

SPRINGER
REFERENCE

Roger Ghanem
David Higdon
Houman Owhadi
Editors

Handbook of Uncertainty Quantification

Handbook of Uncertainty Quantification

Roger Ghanem · David Higdon ·
Houman Owhadi
Editors

Handbook of Uncertainty Quantification

With 508 Figures and 92 Tables



Springer

Editors

Roger Ghanem
Department of Civil and Environmental
Engineering
University of Southern California
Los Angeles, CA, USA

David Higdon
Social and Decision Analytics Laboratory
Virginia Bioinformatics Institute
Virginia Tech University
Arlington, VA, USA

Houman Owhadi
Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA, USA

ISBN 978-3-319-12384-4 ISBN 978-3-319-12385-1 (eBook)
ISBN 978-3-319-12386-8 (print and electronic bundle)
DOI 10.1007/978-3-319-12385-1

Library of Congress Control Number: 2017934651

© Springer International Publishing Switzerland 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This UQ Handbook comes at a time when predictive science is emerging at the confluence of computational, physical, and mathematical sciences. Although UQ appears to be an umbrella term, perhaps due in part to the breadth, novelty, and interdisciplinary nature of the field, it is driven by real challenges of practical importance involving the interplay between physics, modeling, computational hardware, algorithmic complexity, and decisions.

This Handbook provides a glimpse into these new frontiers of uncertainty quantification.

This project was a collaborative effort. The section editors, Drs. Wei Cen, Habib Najm, Bertrand Iooss, Tony Cox, Jim Stewart, and Michael McKerns, provided their own perspective on the Handbook's content and are responsible for both its breadth and cohesion. The editorial staff at Springer provided professional and insightful assistance with diligence and patience. The Handbook would not have been possible without the authors who did all the heavy lifting. We are grateful for their hard work and their confidence.

University of Southern California
Virginia Tech
Caltech
December 2016

Roger Ghanem
David Higdon
Houman Owhadi

Contents

Volume 1

Part I Introduction to Uncertainty Quantification	1
1 Introduction to Uncertainty Quantification.....	3
Roger Ghanem, David Higdon, and Houman Owhadi	
Part II Methodology	7
2 Bayes Linear Emulation, History Matching, and Forecasting for Complex Computer Simulators	9
Michael Goldstein and Nathan Huntley	
3 Inference Given Summary Statistics	33
Habib N. Najm and Kenny Chowdhary	
4 Multi-response Approach to Improving Identifiability in Model Calibration	69
Zhen Jiang, Paul D. Arendt, Daniel W. Apley, and Wei Chen	
5 Validation of Physical Models in the Presence of Uncertainty	129
Robert D. Moser and Todd A. Oliver	
6 Toward Machine Wald	157
Houman Owhadi and Clint Scovel	
7 Hierarchical Models for Uncertainty Quantification: An Overview .	193
Christopher K. Wikle	
8 Random Matrix Models and Nonparametric Method for Uncertainty Quantification.....	219
Christian Soize	
9 Maximin Sliced Latin Hypercube Designs with Application to Cross Validating Prediction Error	289
Yan Chen, David M. Steinberg, and Peter Qian	

10	The Bayesian Approach to Inverse Problems	311
	Masoumeh Dashti and Andrew M. Stuart	
11	Multilevel Uncertainty Integration	429
	Sankaran Mahadevan, Shankar Sankararaman, and Chenzhao Li	
12	Bayesian Cubic Spline in Computer Experiments	477
	Yijie Dylan Wang and C. F. Jeff Wu	
13	Propagation of Stochasticity in Heterogeneous Media and Applications to Uncertainty Quantification	497
	Guillaume Bal	
14	Polynomial Chaos: Modeling, Estimation, and Approximation	521
	Roger Ghanem and John Red-Horse	

Volume 2

Part III	Forward Problems	553
15	Bayesian Uncertainty Propagation Using Gaussian Processes	555
	Ilias Bilionis and Nicholas Zabaras	
16	Solution Algorithms for Stochastic Galerkin Discretizations of Differential Equations with Random Data	601
	Howard Elman	
17	Intrusive Polynomial Chaos Methods for Forward Uncertainty Propagation	617
	Bert Debusschere	
18	Multiresolution Analysis for Uncertainty Quantification	637
	Olivier P. Le Maître and Omar M. Knio	
19	Surrogate Models for Uncertainty Propagation and Sensitivity Analysis	673
	Khachik Sargsyan	
20	Stochastic Collocation Methods: A Survey	699
	Dongbin Xiu	
21	Sparse Collocation Methods for Stochastic Interpolation and Quadrature	717
	Max Gunzburger, Clayton G. Webster, and Guannan Zhang	
22	Method of Distributions for Uncertainty Quantification	763
	Daniel M. Tartakovsky and Pierre A. Gremaud	
23	Sampling via Measure Transport: An Introduction	785
	Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini	

24	Compressive Sampling Methods for Sparse Polynomial Chaos Expansions	827
	Jerrad Hampton and Alireza Doostan	
25	Low-Rank Tensor Methods for Model Order Reduction	857
	Anthony Nouy	
26	Random Vectors and Random Fields in High Dimension: Parametric Model-Based Representation, Identification from Data, and Inverse Problems	883
	Christian Soize	
27	Model Order Reduction Methods in Computational Uncertainty Quantification	937
	Peng Chen and Christoph Schwab	
28	Multifidelity Uncertainty Quantification Using Spectral Stochastic Discrepancy Models	991
	Michael S. Eldred, Leo W. T. Ng, Matthew F. Barone, and Stefan P. Domino	
29	Mori-Zwanzig Approach to Uncertainty Quantification	1037
	Daniele Venturi, Heyrim Cho, and George Em Karniadakis	
30	Rare-Event Simulation	1075
	James L. Beck and Konstantin M. Zuev	
Part IV	Introduction to Sensitivity Analysis	1101
31	Introduction to Sensitivity Analysis	1103
	Bertrand Iooss and Andrea Saltelli	
32	Variational Methods	1123
	Maelle Nodet and Arthur Vidard	
33	Design of Experiments for Screening	1143
	David C. Woods and Susan M. Lewis	
34	Weights and Importance in Composite Indicators: Mind the Gap	1187
	William Becker, Paolo Paruolo, Michaela Saisana, and Andrea Saltelli	
35	Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms	1217
	Clémentine Prieur and Stefano Tarantola	
36	Derivative-Based Global Sensitivity Measures	1241
	Sergey Kucherenko and Bertrand Iooss	
37	Moment-Independent and Reliability-Based Importance Measures	1265
	Emanuele Borgonovo and Bertrand Iooss	

38 Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes	1289
Loïc Le Gratiet, Stefano Marelli, and Bruno Sudret	
39 Sensitivity Analysis of Spatial and/or Temporal Phenomena	1327
Amandine Marrel, Nathalie Saint-Geours, and Matthias De Lozzo	

Volume 3

Part V Risk	1359
40 Decision Analytic and Bayesian Uncertainty Quantification for Decision Support	1361
D. Warner North	
41 Validation, Verification, and Uncertainty Quantification for Models with Intelligent Adversaries	1401
Jing Zhang and Jun Zhuang	
42 Robust Design and Uncertainty Quantification for Managing Risks in Engineering	1421
Ron Bates	
43 Quantifying and Reducing Uncertainty About Causality in Improving Public Health and Safety	1437
Louis Anthony Cox, Jr.	
Part VI Codes of Practice and Factors of Safety	1501
44 Conceptual Structure of Performance Assessments for Geologic Disposal of Radioactive Waste	1503
Jon C. Helton, Clifford W. Hansen, and Cédric J. Salaberry	
45 Redundancy of Structures and Fatigue of Bridges and Ships Under Uncertainty	1541
Dan M. Frangopol, Benjin Zhu, and Mohamed Soliman	
46 Uncertainty Approaches in Ship Structural Performance	1567
Matthew Collette	
47 Uncertainty Quantification's Role in Modeling and Simulation Planning, and Credibility Assessment Through the Predictive Capability Maturity Model	1589
W. J. Rider, W. R. Witkowski, and Vincent A. Mousseau	
48 Uncertainty Quantification in a Regulatory Environment	1613
Vincent A. Mousseau and Brian J. Williams	

Part VII Introduction to Software for Uncertainty Quantification 1649	
49 Dakota: Bridging Advanced Scalable Uncertainty Quantification Algorithms with Production Deployment	1651
Laura P. Swiler, Michael S. Eldred, and Brian M. Adams	
50 Problem Solving Environment for Uncertainty Analysis and Design Exploration	1695
Charles Tong	
51 Probabilistic Analysis Using NESSUS (Numerical Evaluation of Stochastic Structures Under Stress)	1733
John M. McFarland and David S. Riha	
52 Embedded Uncertainty Quantification Methods via Stokhos	1765
Eric T. Phipps and Andrew G. Salinger	
53 Uncertainty Quantification Toolkit (UQTK)	1807
Bert Debusschere, Khachik Sargsyan, Cosmin Safta, and Kenny Chowdhary	
54 The Parallel C++ Statistical Library for Bayesian Inference: QUESO	1829
Damon McDougall, Nicholas Malaya, and Robert D. Moser	
55 Gaussian Process-Based Sensitivity Analysis and Bayesian Model Calibration with GPMSA	1867
James Gattiker, Kary Myers, Brian J. Williams, Dave Higdon, Marcos Carzolio, and Andrew Hoegh	
56 COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management	1909
Edoardo Patelli	
57 SIMLAB Software for Uncertainty and Sensitivity Analysis	1979
Stefano Tarantola and William Becker	
58 OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation	2001
Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin	
Index	2039

About the Editors



Roger Ghanem is the Gordon S. Marshall Professor of Engineering Technology at the University of Southern California where he holds joint appointments in the Departments of Civil and Environmental Engineering and Aerospace and Mechanical Engineering. He has a Ph.D. in Civil Engineering from Rice University. Prior to joining USC, he had served as member of the faculty of civil engineering at both Johns Hopkins University and SUNY Buffalo. Ghanem's expertise is in probabilistic modeling and computational stochastics with focus on high-dimensional formulations and polynomial chaos methods.

His recent work has addressed issues of hierarchical and concurrent coupling in stochastic problems through manifold learning and adaptation. Ghanem has served as president of the Engineering Mechanics Institute (EMI) of ASCE and currently serves on the Executive Council of the US Association for Computational Mechanics (USACM) and as Chair of the SIAM Activity Group on Uncertainty Quantification. He is elected fellow of EMI, USACM, and AAAS.



David Higdon is a Professor in the Social and Decision Analytics Laboratory with the Biocomplexity Institute at Virginia Tech. Previously, he spent 14 years as a scientist or group leader of the Statistical Sciences Group at Los Alamos National Laboratory. He is an expert in Bayesian statistical modeling of environmental and physical systems, combining physical observations with computer simulation models for prediction and inference. His research interests include space-time modeling; inverse problems in a variety of physical applications; statistical modeling in ecology, environmental science, and biology; multiscale statistical models; distributed methods for Markov chain Monte Carlo and posterior exploration; statistical computing; and Monte Carlo and simulation-based methods. Dr. Higdon has served on several advisory groups concerned with statistical modeling and uncertainty quantification and co-chaired

Monte Carlo and posterior exploration; statistical computing; and Monte Carlo and simulation-based methods. Dr. Higdon has served on several advisory groups concerned with statistical modeling and uncertainty quantification and co-chaired

the NRC Committee on Mathematical Foundations of Validation, Verification, and Uncertainty Quantification. He is currently the Editor in Chief for the SIAM-ASA *Journal on Uncertainty Quantification*. Dr. Higdon holds a B.A. and M.A. in Mathematics from the University of California, San Diego, and a Ph.D. in Statistics from the University of Washington.



Houman Owhadi is Professor of Applied and Computational Mathematics and Control and Dynamical Systems in the Computing and Mathematical Sciences Department at the California Institute of Technology. Houman Owhadi's work lies at the interface between applied mathematics, probability, and statistics. At the center of his work are fundamental problems such as the optimal quantification of uncertainties in the presence of limited information, statistical inference/game theoretic approaches to numerical analysis and algorithm design, multiscale analysis with non separated scales, and the geometric integration of structured stochastic systems.

Section Editors

Introduction to Uncertainty Quantification



Roger Ghanem

Department of Civil and Environmental
Engineering
University of Southern California
Los Angeles, CA, USA
Email: ghanem@usc.edu



David Higdon

Social and Decision Analytics
Laboratory
Virginia Bioinformatics Institute
Virginia Tech University
Arlington, VA, USA
Email: dhigdon@vbi.vt.edu



Houman Owhadi

Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA, USA
Email: owhadi@caltech.edu

Methodology



Wei Chen
Department of Mechanical
Engineering
Northwestern University
Evanston, IL, USA
Email: weichen@northwestern.edu

Forward Problems



Habib N. Najm
Sandia National Laboratories
P.O.Box 969, MS 9051
Livermore, CA 94551, USA
Email: hnnajm@sandia.gov

Introduction to Sensitivity Analysis



Bertrand Iooss
Industrial Risk Management Department
EDF R&D
EDF Lab Chatou
6 Quai Watier
78401 Chatou, France
and
Institut de Mathématiques de Toulouse
Université Paul Sabatier
31062 Toulouse
France
Email: bertrand.iooss@edf.fr

Risk



Louis Anthony Cox
Cox Associates
503 Franklin Street
Denver, CO 80218, USA
Email: tcoxdenver@aol.com

Codes of Practice and Factors of Safety



James R. Stewart
Optimization and UQ Department 1441
Sandia National Laboratories
P.O. Box 5800, MS 1318
Albuquerque, NM 87185
USA
Email: jrstewa@sandia.gov

Introduction to Software for Uncertainty Quantification



Mike McKerns
California Institute of Technology
Computing and Mathematical Sciences
102 Steele House MC 9-94
Pasadena CA, 91125
USA
Email: mmckerns@caltech.edu

Contributors

Brian M. Adams Optimization and Uncertainty Quantification Department, Sandia National Laboratories, Albuquerque, NM, USA

Daniel W. Apley Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

Paul D. Arendt CNA Financial Corporation, Chicago, IL, USA

Guillaume Bal Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY, USA

Matthew F. Barone Sandia National Laboratories, Albuquerque, NM, USA

Ron Bates Design Sciences, Engineering Capability, Rolls-Royce plc., Derby, UK

Michaël Baudin Industrial Risk Management Department, EDF R&D France, Chatou, France

James L. Beck Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

William Becker European Commission Joint Research Centre, Ispra (VA), Italy

Ilias Bilionis School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA

Emanuele Borgonovo Department of Decision Sciences, Bocconi University, Milan, Italy

Marcos Carzolio Department of Statistics, Virginia Tech, Blacksburg, VA, USA

Peng Chen Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, USA

Wei Chen Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA

Yan Chen University of Wisconsin-Madison, Madison, WI, USA

Heyrim Cho Department of Mathematics, University of Maryland, College Park, MD, USA

Kenny Chowdhary Quantitative Modeling and Analysis, Sandia National Laboratories, Livermore, CA, USA

Matthew Collette Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, USA

Louis Anthony Cox, Jr. Cox Associates and University of Colorado, Denver, CO, USA

Masoumeh Dashti Department of Mathematics, University of Sussex, Brighton, UK

Bert Debusschere Mechanical Engineering, Sandia National Laboratories, Livermore, CA, USA

Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA

Matthias De Lozzo CEA, DEN, DER, Saint-Paul-lez-Durance, France

Stefan P. Domino Sandia National Laboratories, Albuquerque, NM, USA

Alireza Doostan Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA

Anne Dutfoy Industrial Risk Management Department, EDF R&D France, Saclay, France

Michael S. Eldred Optimization and Uncertainty Quantification Department, Sandia National Laboratories, Albuquerque, NM, USA

Howard Elman Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA

Dan M. Frangopol Department of Civil and Environmental Engineering, Lehigh University, Engineering Research Center for Advanced Technology for Large Structural Systems (ATLSS Center), Bethlehem, PA, USA

James Gattiker Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM, USA

Roger Ghanem Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA, USA

Michael Goldstein Science Laboratories, Department of Mathematical Sciences, Durham University, Durham, UK

Loïc Le Gratiet EDF R&D, Chatou, France

Pierre A. Gremaud Department of Mathematics, North Carolina State University, Raleigh, NC, USA

Max Gunzburger Department of Scientific Computing, The Florida State University, Tallahassee, FL, USA

Jerrad Hampton Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA

Clifford W. Hansen Photovoltaics and Distributed Systems Department, Sandia National Laboratories, Albuquerque, NM, USA

Jon C. Helton Thermal Sciences and Engineering Department, Sandia National Laboratories, Albuquerque, NM, USA

David Higdon Social Decision Analytics Laboratory, Virginia Bioinformatics Institute, Virginia Tech University, Arlington, VA, USA

Andrew Hoegh Department of Statistics, Virginia Tech, Blacksburg, VA, USA

Nathan Huntley Science Laboratories, Department of Mathematical Sciences, Durham University, Durham, UK

Bertrand Iooss Industrial Risk Management Department, EDF R&D, Chatou, France

Institut de Mathématiques de Toulouse, Université Paul Sabatier, Toulouse, France

Zhen Jiang Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA

George Em Karniadakis Division of Applied Mathematics, Brown University, Providence, RI, USA

Omar M. Knio Pratt School of Engineering, Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA

Sergey Kucherenko Department of Chemical Engineering, Imperial College London, London, UK

Susan M. Lewis Southampton Statistical Sciences Research Institute, University of Southampton, Southampton, SO17 1BJ, UK

Chenzhao Li Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN, USA

Sankaran Mahadevan Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN, USA

Olivier P. Le Maître LIMSI-CNRS, Orsay, France

Nicholas Malaya Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Stefano Marelli Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Zürich, Switzerland

Amandine Marrel CEA, DEN, DER, Saint-Paul-lez-Durance, France

Youssef Marzouk Massachusetts Institute of Technology, Cambridge, MA, USA

Damon McDougall Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

John M. McFarland Mechanical Engineering Division, Southwest Research Institute, San Antonio, TX, USA

Tarek Moselhy D. E. Shaw Group, New York, NY, USA

Robert D. Moser Department of Mechanical Engineering, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Vincent A. Mousseau Sandia National Laboratories, Albuquerque, NM, USA

Kary Myers Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM, USA

Habib N. Najm Combustion Research Facility, Reacting Flow Research, Sandia National Laboratories, Livermore, CA, USA

Leo W. T. Ng Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Maelle Nodet Laboratoire Jean Kuntzmann (LJK), University Grenoble Alpes, Grenoble, France

INRIA, Rocquencourt, France

D. Warner North NorthWorks, San Francisco, CA, USA

Anthony Nouy Department of Computer Science and Mathematics, GeM, Ecole Centrale Nantes, Nantes, France

Todd A. Oliver Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Houman Owhadi Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Matthew Parno Massachusetts Institute of Technology, Cambridge, MA, USA

U. S. Army Cold Regions Research and Engineering Laboratory, Hanover, NH, USA

Paolo Paruolo European Commission Joint Research Centre, Ispra (VA), Italy

Edoardo Patelli Institute for Risk and Uncertainty, University of Liverpool, Liverpool, UK

Eric T. Phipps Sandia National Laboratories, Center for Computing Research, Albuquerque, NM, USA

Anne-Laure Popelin Industrial Risk Management Department, EDF R&D France, Chatou, France

Clémentine Prieur Laboratoire Jean Kuntzmann (LJK), University of Grenoble Alpes, INRIA, Grenoble, France

Peter Qian University of Wisconsin-Madison, Madison, WI, USA

John Red-Horse Engineering Sciences Center, Sandia National Laboratories, Albuquerque, NM, USA

W. J. Rider Sandia National Laboratories, Albuquerque, NM, USA

David S. Riha Mechanical Engineering Division, Southwest Research Institute, San Antonio, TX, USA

Cosmin Safta Quantitative Modeling and Analysis, Sandia National Laboratories, Livermore, CA, USA

Nathalie Saint-Geours ITK - Predict and Decide, Clapiers, France

Michaela Saisana European Commission Joint Research Centre, Ispra (VA), Italy

Andrew G. Salinger Sandia National Laboratories, Center for Computing Research, Albuquerque, NM, USA

Cédric J. Salaberry Applied Systems Analysis and Research Department, Sandia National Laboratories, Albuquerque, NM, USA

Andrea Saltelli Centre for the Study of the Sciences and the Humanities (SVT), University of Bergen (UIB), Bergen, Norway

Institut de Ciència i Tecnologia Ambientals (ICTA), Universitat Autònoma de Barcelona (UAB), Barcelona, Spain

Shankar Sankararaman NASA Ames Research Center, SGT Inc., Moffett Field, CA, USA

Khachik Sargsyan Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA

Christoph Schwab Departement Mathematik, Seminar for Applied Mathematics, ETH Zurich, Switzerland

Clint Scovel Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Christian Soize Laboratoire Modélisation et Simulation Multi Echelle (MSME), Université Paris-Est, Marne-la-Vallée, France

Mohamed Soliman School of Civil and Environmental Engineering, Oklahoma State University, Stillwater, OK, USA

Alessio Spantini Massachusetts Institute of Technology, Cambridge, MA, USA

David M. Steinberg Tel Aviv University, Tel Aviv, Israel

Andrew M. Stuart Mathematics Institute, University of Warwick, Coventry, UK

Bruno Sudret Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Zürich, Switzerland

Laura P. Swiler Optimization and Uncertainty Quantification Department, Sandia National Laboratories, Albuquerque, NM, USA

Stefano Tarantola Statistical Indicators for Policy Assessment, Joint Research Centre of the European Commission, Ispra (VA), Italy

Institute for Energy and Transport, European Commission, Joint Research Centre, Ispra (VA), Italy

Daniel M. Tartakovsky Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Charles Tong Computation Directorate, Lawrence Livermore National Laboratory, Livermore, CA, USA

Daniele Venturi Department of Applied Mathematics and Statistics, University of California Santa Cruz, Santa Cruz, CA, USA

Arthur Vidard Laboratoire Jean Kuntzmann (LJK), University Grenoble Alpes, Grenoble, France

INRIA, Rocquencourt, France

Yijie D. Wang Blizzard Entertainment, Irvine, CA, USA

Clayton G. Webster Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN, USA

Christopher K. Wikle Department of Statistics, University of Missouri, Columbia, MO, USA

Brian J. Williams Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM, USA

W. R. Witkowski Sandia National Laboratories, Albuquerque, NM, USA

David C. Woods Southampton Statistical Sciences Research Institute, University of Southampton, Southampton, SO17 1BJ, UK

C. F. Jeff Wu H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Dongbin Xiu Department of Mathematics and Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA

Nicholas Zabaras Warwick Centre for Predictive Modelling, University of Warwick, Coventry, UK

Guannan Zhang Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN, USA

Jing Zhang Department of Industrial and Systems Engineering, New York State University at Buffalo, Buffalo, NY, USA

Benjin Zhu Rizzo Associates, Inc., Pittsburgh, PA, USA

Jun Zhuang Department of Industrial and Systems Engineering, New York State University at Buffalo, Buffalo, NY, USA

Konstantin M. Zuev Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Part I

Introduction to Uncertainty Quantification

Roger Ghanem, David Higdon, and Houman Owhadi

Contents

1 Introduction	3
----------------------	---

1 Introduction

Technology, in common with many other activities, tends toward avoidance of risks by investors. Uncertainty is ruled out if possible. People generally prefer the predictable. Few recognize how destructive this can be, how it imposes severe limits on variability and thus makes whole populations fatally vulnerable to the shocking ways our universe can throw the dice.

Frank Herbert (*Heretics of Dune*)

This handbook of Uncertainty Quantification (UQ) consists of six chapters, each with its own chapter editor. The choice of these chapters reflects a convergence

R. Ghanem (✉)

Department of Civil and Environmental Engineering, University of Southern California,
Los Angeles, CA, USA

e-mail: rghanem@usc.edu

D. Higdon

Social Decision Analytics Laboratory, Biocomplexity Institute, Virginia Polytechnic Institute and
State University, Arlington, VA, USA

e-mail: dhigdon@vbi.vt.edu

H. Owhadi

Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA
e-mail: owhadi@caltech.edu

of opinions on part of the editors in chief and organizes the handbook around methodological developments, algorithms for the statistical exploration of the forward model, sensitivity analysis, risk assessment, codes of practice, and software. Most inference problems of current significance to the UQ community can be assembled using building blocks from these six components. The contributions consist of overview articles of interest both to newcomers and veterans of UQ.

Scientific progress proceeds in increments, and its transformative jumps invariably entail falsifying prevalent theories. This involves comparing predictions from theory with experimental evidence. While this recipe for advancing knowledge remains as effective today as it has been throughout history, its two key ingredients carry within them a signature of their own time and are thus continually evolving. Indeed, both predicting and observing the physical world, the two main ingredients of the scientific process, reflect our perspective on the physical world and are delineated by technology.

The pace of innovation across the whole scientific spectrum, coupled with previously unimaginable capabilities to both observe and analyze the physical world, has heightened the expectations that the scientific machinery can anticipate the state of the world and can thus serve to improve comfort and health and to mitigate disasters.

Uncertainty quantification is the rational process by which proximity between predictions and observations is characterized. It can be thought of as the task of determining appropriate uncertainties associated with model-based predictions. More broadly, it is a field that combines concepts from applied mathematics, engineering, computational science, and statistics, producing methodology, tools, and research to connect computational models to the actual physical systems they simulate. In this broader interpretation, UQ is relevant to a wide span of investigations. These range from seeking detailed quantitative predictions for a well-understood and accurately modeled engineering systems to exploratory investigations focused on understanding trade-offs in a new or even hypothetical physical system.

Uncertainty in model-based predictions arises from a variety of sources including (1) uncertainty in model inputs (e.g. parameters, initial conditions, boundary conditions, forcings), (2) model discrepancy or inadequacy due to the difference between the model and the true system, (3) computational costs, limiting the number of model runs and supporting analysis computations that can be carried out, and (4) solution and coding errors. Verification can help eliminate solution and coding errors. Speeding up a model by replacing it with a reduced order model is a strategy for trading off error/uncertainty between (2) and (3) above. Similarly, obtaining additional data, or higher quality data, is often helpful in reducing uncertainty due to (1) but will do little to reduce uncertainty from other sources. The multidisciplinary nature of UQ makes it ripe for exploiting synergies at the intersection of a number of disciplines that comprise this new field. More specifically, for instance,

- Novel application of principles and approaches from different fields can be combined to produce effective, synergistic solutions for UQ problems,
- The features and nuances of a particular application typically call for specific methodological advances and approaches.
- Novel solutions and approaches often appear from adapting concepts and algorithms from one field of research to another in order to solve a particular UQ problem.
- The concept of “codesign” – building and representing computational models, and analysis approaches and algorithms with HPC architecture in mind – is natural in UQ research, leading to novel solutions in UQ problems.
- Every effort to quantify uncertainty can be leveraged to make new studies – modeling efforts, data collections, computational approaches, etc. – more accurate and/or more efficient.

Managing these trade-offs to the best effect, considering computational costs, personnel costs, cost of data acquisition, etc., depends on the goals of the investigation, as well as the characteristics of the models involved. Unlike more data-driven fields, such as data mining, machine learning, and signal processing, UQ is more commonly focused on leveraging information from detailed models of complex physical systems. Because of this, UQ brings forward a unique set of issues regarding the combination of detailed computational models with experimental or observational data. Quite often the availability of this data is limited, tilting the balance toward leveraging the computational models. Key considerations in UQ investigations include

- The amount and relevance of the available system observations,
- The accuracy and uncertainty accompanying the system observations,
- The complexity of the system being modeled,
- The degree of extrapolation required for the prediction relative to the available observations and the level of empiricism encoded in the model,
- The computational demands (run time, computing infrastructure) of the computational model,
- The accuracy of the computational model’s solution relative to that of the mathematical model (numerical error),
- The accuracy of the computational model’s solution relative to that of the true system (model discrepancy),
- The existence of model parameters that require calibration using the available system observations,
- The availability of alternative computational models to assess the impact of different modeling schemes or implementations on the prediction.

The concept of a well-posed UQ problem is nucleating in response to the flurry of activities in this field. In particular, whether UQ can be framed as a problem

in approximation theory on product spaces, or as an optimization problem that relates evidence to decisions, or as a Bayesian inference problem with likelihoods constrained by hierarchical evidence, points to a convergence of mathematical rigor, engineering pragmatism, and statistical reasoning, all powered by developments in computational science.

Our initial intent for this introductory chapter was to present a common notation that ties a common thread throughout the handbook. This proved premature, and the diversity of these contributions points to a still nascent field of UQ. Although, at present, UQ lacks a coherent general presentation, much like the state of probability theory before its rigorous formulation by Kolmogorov in the 1930s, the potential for such a development is clear, and we hope that this handbook on UQ will contribute to its development by presenting an overview of fundamental challenges, applications, and emerging results.

Part II

Methodology

Bayes Linear Emulation, History Matching, and Forecasting for Complex Computer Simulators

2

Michael Goldstein and Nathan Huntley

Abstract

Computer simulators are a useful tool for understanding complicated systems. However, any inferences made from them should recognize the inherent limitations and approximations in the simulator's predictions for reality, the data used to run and calibrate the simulator, and the lack of knowledge about the best inputs to use for the simulator. This article describes the methods of emulation and history matching, where fast statistical approximations to the computer simulator (emulators) are constructed and used to reject implausible choices of input (history matching). Also described is a simple and tractable approach to estimating the discrepancy between simulator and reality induced by certain intrinsic limitations and uncertainties in the simulator and input data. Finally, a method for forecasting based on this approach is presented. The analysis is based on the Bayes linear approach to uncertainty quantification, which is similar in spirit to the standard Bayesian approach but takes expectation, rather than probability, as the primitive for the theory, with consequent simplifications in the prior uncertainty specification and analysis.

Keywords

Computer simulators • Bayes linear • Emulation • Model discrepancy • History matching • Calibration • Internal discrepancy • Forecasting

Contents

1	Introduction	10
2	Example: Rainfall Runoff Simulator	11
3	The Bayesian Analysis of Computer Simulators for Physical Systems	12

M. Goldstein and N. Huntley (✉)

Science Laboratories, Department of Mathematical Sciences, Durham University, Durham, UK
e-mail: michael.goldstein@durham.ac.uk; nathan.huntley@durham.ac.uk

4	Bayes Linear Analysis	14
5	Emulation	14
6	Example: Emulating FUSE	17
7	Model Discrepancy	20
8	Example: Model Discrepancy for FUSE	21
9	History Matching	22
10	Example: History Matching FUSE	24
10.1	Introducing f_2	26
10.2	Impact of External and Internal Discrepancy	28
11	Forecasting	28
12	Example: Forecasting for FUSE	29
13	Conclusion	30
	Appendix: Internal Discrepancy Perturbations	30
	References	32

1 Introduction

One of the main tools for studying complex real-world phenomena is the creation of mathematical models for such phenomena, typically implemented as computer simulators. There is a growing field of study which is concerned with the uncertainties arising when computer simulators are used to make inferences about real-world behavior. Two characteristic features of this field are, firstly, the need to analyze uncertainties for simulators which are slow to evaluate and, secondly, the need to recognize and assess the difference between the simulator and the physical system which the simulator purports to represent. In this article, the Bayes linear approach to the assessment of uncertainty for such problems is described. This Bayes linear approach has been successfully applied in a variety of areas, including oil reservoir management [3], climate modeling [17], and simulators of galaxy formation [16]. The aim of this paper is to present a survey of the common general methodology followed in each such application.

The structure of the article is as follows. First, the Bayesian analysis of computer simulators for physical systems is discussed, and the role of Bayes linear analysis in this context is described and motivated. This is followed by a discussion of the issues arising when the simulator is slow to evaluate and the role of emulation in such problems. Next, ways to assess the structural discrepancy arising from the mismatch between the simulator and the physical system are described. These methods are then used in the context of history matching, namely, finding collections of simulator evaluations which are consistent with historical observations, within the levels of uncertainty associated with the problem. Finally, the role of the simulator in forecasting within the Bayes linear framework is described. The running example used to illustrate the development is based around flood modeling. This example has the merit that the code is freely available as an R package, so that the interested reader may try out the type of analysis described and compare it to alternative approaches to the same problems. This work was supported by NERC under the PURE research program.

2 Example: Rainfall Runoff Simulator

The running example is of a rainfall runoff simulator, that is, a simulator that predicts stream flow in a river. The particular simulator considered in this article is FUSE (The authors thank their colleagues, from the NERC PURE program, Wouter Buytaert, Nataliya Bulygina, and in particular Claudia Vitolo, for their help in running and interpreting the FUSE simulator.) (Framework for Understanding Structural Errors), described in [2], which can be downloaded for R from R-Forge at https://r-forge.r-project.org/R/?group_id=411. FUSE was designed as a toolbox of different modeling choices (there are 1248 different simulators available in FUSE), but in this article, only one (model 17) is considered. For brevity, this specific simulator within FUSE is referred to throughout simply as “FUSE.”

FUSE takes as its input a time series of average rainfall across the catchment in question, on whatever time scale one desires, and a time series of “potential evapotranspiration” on the same time scale. Its output is a time series of predicted stream flow, on the same scale. For this example, the simulator was run using data from the Pontbren catchment in Wales; the dataset is freely available from the Centre for Ecology & Hydrology (<https://eip.ceh.ac.uk/>). After some processing, the data consist of hourly readings over approximately 15 months, where the rainfall data is derived from six rainfall gauges giving readings every 10 min, and the evapotranspiration is calculated hourly using the Penman-Monteith equation [12]. The output is compared to hourly readings of stream flow at a gauge near the end of the catchment.

FUSE models water storage as consisting of three compartments, the size of which is governed by three corresponding parameters. Flow into, out of, and between these compartments are governed by simple equations that rely on five further parameters. Finally, a time delay, governed by one parameter, is applied to the predicted stream flow. Full details of these equations can be found in [2].

In summary, FUSE is run by providing a time series of rainfall, a time series of evapotranspiration, nine parameters, and finally an initial condition that specifies how much water is in the compartments at the beginning. Figure 2.1 shows an

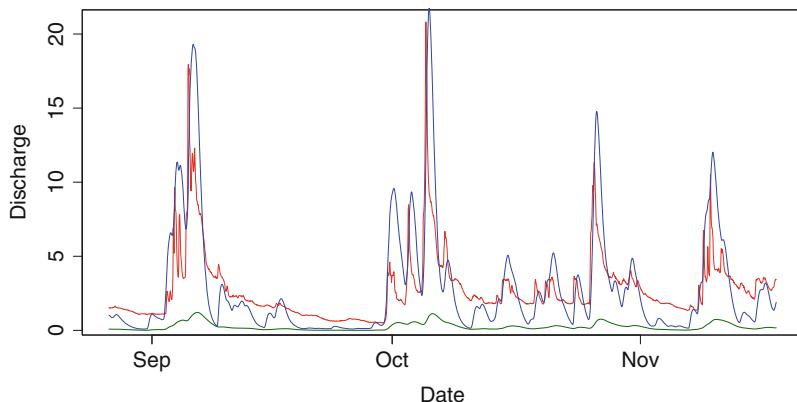


Fig. 2.1 Observed discharge (red) compared with FUSE runs for two choices of parameters

example of FUSE output for a decent choice of parameters (blue line) and a very poor choice of parameters (green line) compared with observed stream flow (red line) for a small section of the data.

3 The Bayesian Analysis of Computer Simulators for Physical Systems

This article is concerned with problems in which computer simulators are used to represent physical systems. Each simulator can be conceived as a function $f(x)$, where x is an input vector representing properties of the physical system and $f(x)$ is an output vector representing system behavior. Typically, some of the elements of x represent unknown physical properties of the system, some are tuning parameters to compensate for approximations in the simulator, and some are control parameters which correspond to ways in which system behavior may be influenced by external actions.

Analysis of simulator behavior provides qualitative insights into the behavior of the physical system. Usually, it is of interest to identify “appropriate” (in some sense) choices, x^* , for the system properties x and to assess how informative $f(x^*)$ is for actual system behavior, y . Often, historical observations, z , observed with error, are available, corresponding to a historical subvector, y_h , of y , which may be used both to test and to constrain the simulator, by comparison with the corresponding subvector $f_h(x)$ of $f(x)$. Typically, there is ensemble of simulator evaluations $F_{[n]} = (f(x_1), \dots, f(x_n))$ made at an evaluation set of input choices $x_{[n]} = (x_1, \dots, x_n)$ which is used to help address these questions.

There are many uncertainties associated with this analysis. The parameter values are unknown, and there are uncertainties in the forcing functions, boundary conditions, and initial conditions required to evaluate the simulator. Many simulators have random outputs. The complexity of the underlying mathematics forces the solutions of the system equations to be approximated. Typically, computer simulators are slow to evaluate, and so the value of the function $f(x)$ must be treated as unknown for all x except for the design values $x_{[n]}$. The data to which the simulator is matched is only observed with error. Even if all of these considerations could be addressed precisely, the computer simulator would still be an imperfect representation of the physical system.

A common representation for simulator discrepancy in practice is to suppose that there is an appropriate choice of system properties x^* (currently unknown), so that $f(x^*)$ contains all the information about the system, expressed as the relation

$$y = f(x^*) + \epsilon \quad (2.1)$$

where ϵ is a random vector expressing the uncertainty about the physical system that would remain were the simulator to be evaluated at x^* . Typically, ϵ is taken to be independent of f and of x^* , though modifications to this form are discussed in later sections. The variance of an individual component ϵ_i expresses one’s confidence in

the ability of the simulator to reproduce the corresponding physical process, while the correlation between components of ϵ expresses the similarity between different types of structural error, for example, whether underpredictions for the simulator in the past suggest that the simulator will continue to underpredict in the future. For many problems, whether this formulation is appropriate is, itself, the question of interest, that is, the goal is to determine whether there are any choices of input parameters for which the simulator outputs are in rough agreement with the system observations in the sense of consistency with (2.1).

The specification is completed by a relationship between observations z and historical system values y_h , which often is taken to be of form

$$z = y_h + e \quad (2.2)$$

where e is the vector of observational errors, taken to be independent of all other quantities in the problem.

A Bayesian analysis of such a problem therefore requires, as a minimum, prior probability distributions over the input space x , over the function values $f(x)$, for each x , and over the simulator discrepancy ϵ , and a likelihood function for e . In principle, given all of these specifications, a full probabilistic synthesis of all the sources of information in the problem can be made, to identify appropriate inputs, make effective forecasts of future system behavior, and select control parameters to optimize the ability to meet the targets of system control. For systems of moderate size and complexity, this approach is tractable, powerful, and successful.

For complex, high-dimensional problems, however, the approach encounters various difficulties. In such cases, the computations for learning from data tend to be technically difficult and time consuming, as the likelihood surface is extremely complicated, and any full Bayes calculation may be extremely non-robust and highly dependent on the initial prior specification. It is difficult to specify meaningful prior distributions over high-dimensional spaces, and the nature of the calculations makes it difficult to carry out a full sensitivity analysis on the various features of the probabilistic specification which may contribute strongly to the final inferences. Such difficulties are particularly acute when repeat evaluations of the inferential process must be made, for example, when using the prior specification to generate informative choices for the simulator design $x_{[n]}$ or using the inferential construction to facilitate real-time control of the physical process being modeled. Such complexities lead, in practice, to the use of conventional conjugate prior forms which sacrifice fidelity to one's best scientific judgements in order to simplify the technicalities of the calculations. In cases where one wishes to avoid the introduction of somewhat arbitrary simplifying assumptions, such an approach may be deemed unsatisfactory.

Much of the complexity of the standard Bayesian approach derives from the extreme level of detail which is required for a full probabilistic specification of all of the uncertainties in the problem. Therefore, it is important to recognize that there is a choice as to the primitive chosen as the basis of the stochastic analysis and that this choice determines the complexity of the calculations that are required for the analysis, as is now described.

4 Bayes Linear Analysis

The conventional Bayesian approach takes probability as the primitive quantity around which all analyses are constructed. However, it is possible to construct an alternative approach to Bayesian inference in which expectation, rather than probability, acts as the primitive. This approach is termed Bayes linear analysis, because of the linearity properties of expectation. For a careful and detailed exposition of the notion of expectation as the appropriate primitive for the subjectivist theory, see [6]. In this work, de Finetti chooses expectation over probability, as if probability is primitive, then all of the probability statements must be made before any of the expectation statements can be, whereas if expectation is primitive, then as many or as few expectation statements as one chooses can be made, so that there is the option of restricting attention to whatever subcollection of specifications one is both willing and able to specify in a meaningful fashion.

Full Bayes analysis can be very informative if conducted carefully, both in terms of the prior specification and the analysis. Bayes linear analysis is partial but easier, faster, and more robust. The approaches may be considered as complementary and use made of either, or both, depending on the requirements of the problem at hand. In this article, attention is restricted to the Bayes linear formulation.

The Bayes linear approach is (relatively) simple in terms of belief specification and analysis, as it is based only on the mean, variance, and covariance specification which is taken as primitive. Just as Bayes analysis is based around a single updating equation, the Bayes linear approach is based around the notion of Bayes linear adjustment. $E_z(y)$, $\text{Var}_z(y)$ are the expectation and variance for the vector y adjusted by the vector z . These quantities are given by

$$E_z(y) = E(y) + \text{Cov}(y, z)\text{Var}(z)^{-1}(z - E(z)), \quad (2.3)$$

$$\text{Var}_z(y) = \text{Var}(y) - \text{Cov}(y, z)\text{Var}(z)^{-1}\text{Cov}(z, y). \quad (2.4)$$

One may view Bayes linear adjustment in two ways. Firstly, this may be viewed as a fast approximation to a full Bayes analysis based on linear fitting. Secondly, one may view the Bayes linear approach as giving the appropriate analysis under a direct partial specification of means, variances, and covariances, where expectation is treated as primitive. This view is based on an axiomatic treatment of temporal coherence [7]. In this view, full Bayes analysis is simply a special case where expectations of families of indicator functions are specified, and probabilistic conditioning is simply the corresponding special case of (2.3). For a full account of the Bayes linear approach, see [11].

5 Emulation

Uncertainty analysis based on computer simulators is particularly challenging in the common situation where the simulator takes a long time to evaluate for a single choice of input parameters. In such problems, the value of the simulator $f(x)$ must

be treated as unknown unless x is a member of the evaluation set, $x_{[n]}$. In such problems, one's uncertainty as to the value of $f(x)$ must be assessed for each x which is not a member of $x_{[n]}$. This uncertainty specification is often referred to as an emulator for the simulator.

Within the Bayes linear formulation, the emulator is usually described in terms of (i) an expectation function, $E(f(x))$, which acts as a fast approximation to the simulator; (ii) a variance function, $\text{Var}(f(x))$, which acts as an assessment of the level of approximation introduced by replacing the simulator by this fast alternative; and (iii) a covariance function, $\text{Cov}(f(x), f(x'))$, which describes the similarity between different simulator evaluations and therefore determines the amount of information about general simulator values $f(x)$ that is available from the evaluation ensemble $F_{[n]}$.

A common choice for the emulator, for an individual component $f_i(x)$ of $f(x)$, is to express the function as

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x) + u_i(x). \quad (2.5)$$

In (2.5), $B = \{\beta_{ij}\}$ are unknown scalars and $g_{ij}(x)$ are known deterministic functions of x (e.g., polynomials). $Bg(x)$ expresses global variation in $f_i(x)$, namely, those features of the surface which can be assessed by making simulator evaluations anywhere in the input space. Therefore, the functional forms g_{ij} must be chosen, and also a second-order specification for the elements of B is required.

As $f_i(x)$ is continuous in x , the residual function $u_i(x)$ is also a continuous function. $u_i(x)$ expresses local variation, namely, those aspects of the surface which can only be learned about by making evaluations of the simulator within a subregion near x . Typically, $u_i(x)$ is represented as a second-order stationary stochastic process, with a correlation function which expresses the notion that the correlation between the value of u_i for any two values x, x' is an increasing function of the distance between the two input values. A common choice is the squared exponential, which is used for the illustrations in this article, and is of form

$$\text{Corr}(u_i(x), u_i(x')) = \exp\left(-\left(\frac{\|x - x'\|}{\theta_i}\right)^2\right). \quad (2.6)$$

This form corresponds to a judgement that the function is very smooth. There are many choices for the form of the correlation function and a wide literature on the merits of different choices [15]. How important this choice is depends largely on the proportion of the uncertainty that can be attributed to the global portion of the emulator. In this article, an approach in which a lot of effort is expended on the choice of global fit is favored. Firstly, this is because the choice of appropriate residual correlation function is a difficult problem, largely because, in practice, most functional outputs display different degrees of smoothness in different regions of the input space. Therefore, reducing dependence on this choice

is often prudent. Secondly, as shall be described, much of the analysis is greatly simplified by having a substantial component of the variation described by the global surface.

An emulator is fit based on an analysis of the evaluation ensemble $F_{[n]}$ in combination with expert judgements as to the form of the response surface. There are as many ways to build the emulator as there are methodologies for statistical modeling of complex functions; a good introduction is [13] and see, for example, [15]. A simple approach, which often is reasonably successful, is to choose the functional forms g_{ij} by least squares fitting and then to fit the correlation function for each $u_i(x)$ to the residuals from the least squares fit, either by some method such as maximum likelihood or by some trial-and-error method based on cross-validation or other forms of assessment of quality of fit. Whichever approach taken, one must apply careful diagnostic testing to ensure reliability of the emulator, see, for example, [1].

Often, the simulator will include parameters that have been judged, by some combination of expert judgement and statistical analysis, to have minor influence on component $f_i(x)$. In such cases, it is often profitable to remove these parameters from the mean function and from u_i . That is, if the parameter set is partitioned into active parameters x_A and inactive parameters x_I for a particular component f_i , then the emulator becomes

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x_A) + u_i(x_A) + \delta_i(x_I). \quad (2.7)$$

The variance of δ_i can be estimated by running the simulator at different values of x_I for fixed x_A . Typically, this variance will be small relative to that of u_i ; if not, it is questionable whether x_I is truly inactive.

Such methods require sufficient simulator evaluations to support the assessment of the emulator. If the simulator is slow to evaluate, then it is often possible to develop a joint uncertainty model, where $f(x)$ is combined with a fast approximate version, $\tilde{f}(x)$ say, based on reducing the level of detail in the representation of the input space or approximating the simulator solver, for example, by increasing the time step or reducing the number of iterations of the solution algorithm. Many evaluations of $\tilde{f}(x)$ can then be made, to fit an emulator of the form

$$\tilde{f}_i(x) = \sum_j \tilde{\beta}_{ij} g_{ij}(x) + \tilde{u}_i(x) \quad (2.8)$$

The form (2.8) acts as a prior specification for $f_i(x)$. Then, a relatively small evaluation set $F_{[n]}$ can be chosen, which, in combination with a representation of the relationship between the forms of the global surface in the two simulators, for example,

$$\beta_{ij} = \alpha_i \tilde{\beta}_{ij} + \gamma_{ij}, \quad u_i(x) = \alpha \tilde{u}_i(x) + r_i(x) \quad (2.9)$$

enables adjustment of the prior emulator to an appropriate posterior emulator for $f_i(x)$. This approach is effective because, in general, it takes far more function evaluations to identify the appropriate forms $g_{ij}(x)$ than to quantify the various coefficients of the model, particularly if it is possible to fit regression components which account for a large component of global variation. See [5] for a description and illustration of this approach and, in particular, of the role of the prior construction over the fast simulator in constructing an efficient small sample evaluation set $F_{[n]}$ on the slow simulator.

6 Example: Emulating FUSE

Instead of attempting to emulate the entire FUSE output, it is more instructive as an example to consider instead the emulation of a few interesting summaries. As well as being simpler, this focuses on quantities that are both physically meaningful and can be emulated well. The quantities of interest will be the maximum stream flow (denoted f_1) and the total stream flow (denoted f_2) in the period from hours 7800 to 8800. This corresponds roughly to September to October in Fig. 2.1.

The evaluation set $x_{[n]}$ is drawn from the hypercuboid of plausible parameter values suggested in [2, Table 3], using a maximin Latin hypercube, rescaling all parameters so they take values in $[0, 1]$. The value of n used was 1000; in practice, this is rather more than is usually necessary to build an emulator, but the speed of the model allows such a large set to be used conveniently. The simulator is then run at each parameter choice in $x_{[n]}$ to give a set of simulated quantities $f_1(x_{[n]})$, $f_2(x_{[n]})$.

Least squares fitting was used to find good choices of $g_{ij}(x)$ in (2.5). The first observation was that it was difficult to find any good global fit for f_2 , with the best attempt giving an adjusted R^2 of only 0.65. This is unlikely to be sufficient to build a good emulator. So initially, an emulator was built for f_1 only – f_2 is revisited later in the account.

For f_1 , the maximum discharge, the best model found was a cubic fit, giving adjusted R^2 of around 0.9. Looking more closely at this model suggested that a transformation of parameter $x_{(8)}$ to its logarithm might be profitable. Doing this not only allowed a simplification to a quadratic fit but also increased the R^2 to 0.958. So, the form of the emulator is now given by

$$f_1(x) = \beta_1 + \sum_{j=1}^9 \beta_{1jj} x_{(j)}^2 + \sum_{j=1}^9 \sum_{k < j} \beta_{1jk} x_{(j)} x_{(k)} + \sum_{j=1}^9 \beta_{1j} x_{(j)} + u_1(x). \quad (2.10)$$

The next step is to seek inactive variables. These can be suggested through stepwise selection, removing at each step the variable that has the smallest influence. Following such a strategy for this example suggested removing parameters $x_{(2)}$, $x_{(4)}$, $x_{(6)}$, and $x_{(7)}$, with the R^2 falling only to 0.95 after all are removed. Removing a fifth variable (either $x_{(3)}$ or $x_{(5)}$) would reduce the R^2 to 0.94; this is still a small change

but it was decided not to remove any further variables. Thus, the final form of the emulator is, following (2.7),

$$f_1(x) = \beta_1 + \sum_{j \in A} \beta_{1jj} x_{(j)}^2 + \sum_{\substack{k, j \in A \\ k < j}} \beta_{1jk} x_{(j)} x_{(k)} + \sum_{j \in A} \beta_{1j} x_{(j)} + u_1(x_A) + \delta_1(x_I) \quad (2.11)$$

where $A = \{1, 3, 5, 8, 9\}$ and recalling that $x_{(8)}$ is still the logarithm of the original parameter.

This is almost everything needed to perform the Bayes linear adjustment in (2.3) and (2.4). The goal is to predict the value f_1 of the simulator at some new parameter choice x , using the values $f_1(x_{[n]})$ calculated earlier. Thus, in the Bayes linear adjustment equations, $f_1(x)$ plays the role of y , and $f_1(x_{[n]})$ plays the role of z .

Applying the Bayes linear adjustment in (2.3) and (2.4) requires prior specifications for the relevant expectations and variances. It is assumed a priori that $E(u_1(x)) = 0$, $E(\delta_1(x)) = 0$ and that both $\text{Var}(u_1(x))$ and $\text{Var}(\delta_1(x))$ do not depend on x . Correlation between $u_1(x)$ and $u_1(x')$ is given by (2.6), and it is assumed that there is no correlation between $\delta_1(x)$ and $\delta_1(x')$. If there was suitable expert knowledge and little data, the remaining values would have to be carefully elicited from the experts. In this example, however, there is little expert knowledge, but there are lots of simulator runs. It is therefore reasonable to use some of the estimates from the least squares fit: for $E(\beta_{1j})$ and $\text{Var}(\beta_{1j})$, the corresponding estimates from the fit can be used (where with 1000 runs, the variances will be very low), and for $\text{Var}(u_1)$, the residual variance can be used. This leaves only $\text{Var}(\delta_1)$: this is estimated by fixing values of x_A and running the simulator for a range of x_I . These runs gave an estimate for $\text{Var}(\delta_1)$ of 3×10^{-5} , which is very small compared with f_1 (usually around 3). This therefore gives some confidence that this estimation method should be sufficient.

Only one quantity remains to be assessed: θ_1 . Following the argument from [16] that, for an emulator whose mean function is a polynomial of order p , a plausible choice for θ_1 is $1/(p+1)$, an initial choice of $1/3$ was tried, which was then refined by cross-validation. For a given value of θ_1 , for each x_i in $x_{[n]}$, one can calculate $E_{f_1(x_{[-i]})}(f_1(x_i))$ and $\text{Var}_{f_1(x_{[-i]})}(f_1(x_i))$ using the other elements $x_{[-i]}$ of $x_{[n]}$ to perform the adjustment. Here, $E_{f_1(x_{[-i]})}(f_1(x_i))$ is the adjusted expectation (see (2.3)) for $f_1(x_i)$ having observed $f_1(x_j)$ for each other x_j in $x_{[n]}$, and $\text{Var}_{f_1(x_{[-i]})}(f_1(x_i))$ is the corresponding adjusted variance (see (2.4)). Then, one can check whether $E_{f_1(x_{[-i]})}(f_1(x_i))$ is close to $f_1(x_i)$, relative to the size of $\text{Var}_{f_1(x_{[-i]})}(f_1(x_i))$. If the prediction is many standard deviations away from $f_1(x_i)$ for many x_i , this suggests a problem with the model: either it is not fit for purpose, or the correlations implied by θ_1 are too strong. On the other hand, if all the predictions are very close to $f_1(x)$ relative to the size of the standard deviations, this suggests that the adjusted variance at many points is too low. This could happen because the correlation between nearby points is being underestimated, and hence a higher θ_1 could be tried to give accurate predictions with a lower variance.

The choice $\theta_1 = 1/3$ leads to the latter case: the emulator successfully predicts all 1000 points to within 3 standard deviations, but almost all points are predicted to within 1 standard deviation and indeed over 80 % of the points are predicted to within 0.5 standard deviations. That is, the emulator variance is too conservative, which would lead to poor performance in the history-matching process later. This could be because θ_1 is too small. Indeed, increasing θ_1 to 0.45 gave more sensible variances and predicted 98 % of the simulations within three standard deviations.

There are several reasons why one might be comfortable with 98 % accuracy. In a sample of 1000, one would expect some predictions to be so inaccurate anyway. Further, more investigation of the parameters that led to poor predictions revealed that they were mostly simulations that gave very high or very low values for f_1 , and the emulator was correctly predicting that these parameters would give extreme values. The only failure of the emulator was in underestimating quite how extreme these were. As shall soon be seen, the two main dangers of an inaccurate emulator are predicting f_1 to be close to reality when it is in fact far away or predicting f_1 to be far from reality when it is in fact close. For this emulator, these cases occur only very rarely.

As a summary of this validation, Fig. 2.2 shows an example of the predictions (black line), three standard deviation bounds (blue lines), and observed values (red line) for predicting 100 parameter choices from the other 900. Observe that large differences between predictions and observations do indeed occur when the adjusted variance is high.

To conclude this section, note that, once the fit has been performed, the updates in (2.3) and (2.4) are performed very quickly. Inverting $\text{Var}(f_1(x_{[n]}))$ needs only be done once, so for a particular candidate point x , the only computations are

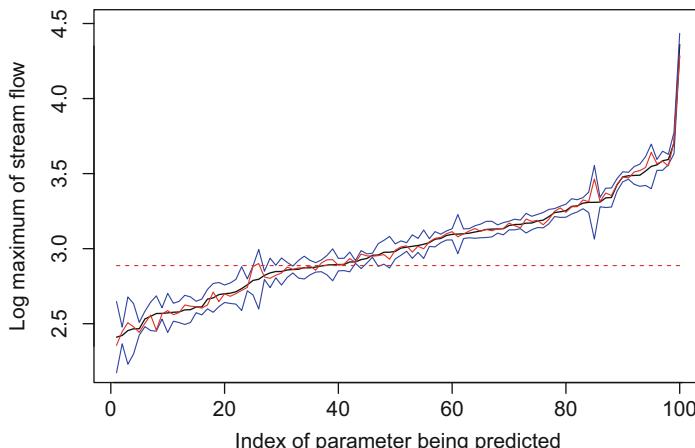


Fig. 2.2 Emulator of $f_1(x)$ (black line) with three standard deviation bounds (blue lines), together with observed value of $f_1(x)$, for 100 parameter choices, predicted from 900 other simulator runs. The horizontal dashed line represents the observed log maximum of stream flow

calculating distances between x and $x_{[n]}$ and then performing a matrix multiplication. Thus, the emulator can be evaluated at very many locations. Even for a simulator as fast as FUSE this is useful; for slower simulators it is invaluable.

7 Model Discrepancy

A physical model is a description of the way in which system properties (the inputs to the model) affect system behavior (the outputs of the model). This description involves two basic types of simplification.

Firstly, the properties of the system are approximated, as these properties are too complicated to describe fully, and anyway they are not completely understood. Secondly, the rules for finding system behavior, given system properties, are approximated, because of necessary mathematical simplifications, simplifications for numerical tractability, and because the physical laws which govern the process are not fully understood.

Neither of these approximations invalidates the modeling process. Problems only arise when these simplifications are forgotten, and the uncertainty analysis of the simulator is confused with the corresponding uncertainty analysis for the physical system itself. As a basic principle, it is always better to recognize than to ignore uncertainty, even if modeling and analysis of such uncertainty is difficult and partial. Thus, it is important to recognize that models do not produce statements about reality, however carefully they are analyzed. Such statements require structural uncertainty assessment, taking account the mismatch between the simulator and the physical system.

One may distinguish two types of model discrepancy, internal and external. The term internal discrepancy is used to refer to any aspect of structural discrepancy whose magnitude can be assessed by experiments on the computer simulator. Internal discrepancy analysis gives a lower bound on the structural uncertainty that must be introduced into the simulator analyses.

How internal discrepancy is assessed depends on the access that one has to the simulator code and one's ability and willingness to carry out the relevant experiments. For example, many simulators assess, at each time point, a state vector from which the system outputs are evaluated, and then propagate the state vector to the next time point by various system rules. If it is straightforward to access the state vector, then it may be possible to introduce an additional stochastic term into the state vector propagation to express uncertainty as to the principles underlying this process. In the flood example, the forcing functions and initial conditions are varied. Also assessed is the effect of allowing various simulator parameters, considered constant within the simulator, to vary slowly over time.

The term external discrepancy relates to all of those aspects of structural discrepancy that have not been addressed by one's assessments for internal discrepancy. In particular, it relates to the inherent limitations of the modeling process embodied in the simulator, based on missing or misconceived aspects of the physical simulator. There are no experiments on the simulator which may reveal the magnitude of this

discrepancy. It is determined by a combination of expert judgements and statistical estimation.

A common representation for simulator discrepancy in practice is to use the form (2.1). Typically, ϵ is taken to be independent of f and of x^* , though there are many possible modifications to this form. For example, (2.1) combines the internal and external discrepancy into a single quantity. However, if the two aspects of discrepancy have been carefully assessed separately, then discrepancy term may be separated into two components, each with their own second-order structure. For example, one might treat the variance of the internal portion of ϵ as a scale parameter dependent on x^* , which is assessed by computer experiments.

More generally, the external discrepancy may be quantified by careful assessment of the reasons for the failure of the simulator to match with reality. One formalism that allows expression of this difference is through what is termed reified modeling (from reify – to treat an abstract concept as if it was real). In this approach, the reason that the simulator is informative for the physical system is considered to be that it is an informative, but imperfect, representation for the actual relationships between system properties and system behavior. The reified model f^* is one's best expression of such relationships. It is not possible to build and evaluate f^* , but it can be emulated.

Therefore, a more careful construction of external discrepancy proceeds by separating the discrepancy between the simulator and the physical system into two parts. The first is the difference between f and f^* and the second is the difference between $f^*(x^*)$ and the system behavior y . There are many ways to construct an emulator for f^* . The simplest is to build it on top of the emulator for f . Suppose that the emulator for f is given by (2.5). Then an emulator for f^* might take the form

$$f^*(x, w) = B^* g(x) + u^*(x) + u^*(x, w)$$

where one might model one's judgements relating B and B^* in a similar manner to (2.9) and specify a correlation function relating $u(x)$ and $u^*(x)$, while treating $u^*(x, w)$, with additional parameters, w , as uncorrelated with the remaining terms in the emulator. For discussion and illustration of reification, including a treatment of structured reification which incorporates systematic modeling for all aspects of simulator deficiency whose effects one wishes to represent explicitly, see [9].

8 Example: Model Discrepancy for FUSE

This article uses a simple method for estimating internal discrepancy, following a similar approach to [10]. The strengths of this method are that it is easy to apply, does not involve any complicated theory or calculations, can incorporate expert knowledge where available but can be performed without any, is trivially parallelizable, and does not require a huge number of simulator runs (although

would still not be suitable for a very slow simulator). The main disadvantage is that the estimate derived is likely to be crude, as shall be seen in later sections.

The core idea is to identify aspects of internal discrepancy, make suitable perturbations of these aspects, run the simulator for each perturbation, and assess the variability of this output. The aspects perturbed in this example were the rainfall and evapotranspiration time series, the initial condition, and the parameter vector. The details of the perturbations themselves can be found in the Appendix. Once the perturbation method is determined, the procedure used is as follows.

1. Select an evaluation parameter set $x_{[n]}$.
2. Generate m perturbations of each type.
3. Combine these to form m overall perturbations.
4. Run the simulator at each x_j in $x_{[n]}$ and each combined perturbation (giving $n \times m$ total simulator runs).
5. For each run, calculate the quantities of interest f_i (in this example, log maximum and total stream flow in the relevant region).
6. For each x_j and each quantity of interest, calculate the standard deviation of the set of m simulations.
7. Take the maximum of these as the estimate of internal discrepancy for each quantity of interest.
8. Check whether the perturbations tend to bias the results in one direction or the other. This does not occur in the example. Were it to occur, then the later assumption that the expected value of internal discrepancy is zero would not be valid.

The justification for taking the maximum rather than, say, the mean, is that the estimate for internal discrepancy should be conservative, because it could be that x^* is in the region of high internal discrepancy. Indeed, it could be argued that even the maximum is not conservative enough, since only a small number of parameters have been sampled. In this example, the authors' experiences with the model suggested that the region of high internal discrepancy was unlikely to include x^* , so the maximum was considered to be acceptable.

Running such perturbations for $n = 50$ and $m = 300$, the log maximum of discharge yielded a maximum internal discrepancy variance of 0.124^2 . Observe that this is very high relative to the emulator variances from earlier. In practice, this will often be the case; later, one way of refining this estimate is considered.

9 History Matching

A typical Bayesian approach to the analysis of relations (2.1) and (2.2) treats the quantity x^* as a true but unknown parameter, specifies a prior distribution for x^* , and carries out a simulator calibration exercise to derive the posterior distribution of x^* given z . However, in many problems, for example, the rainfall runoff simulator being used as an illustration, it may not be believed that there is a unique true but

unknown choice for x^* . Indeed, it is often possible that there are no good choices for x^* , due to deficiencies in the simulator.

A simple alternative to calibration is “history matching.” This method describes the identification of all input choices x for which the match of the simulator to the data is judged to be consistent with the uncertainty expressed in relations (2.1) and (2.2). Sometimes, this analysis will be sufficient for the problem at hand. At other times, it may serve as a precursor to a more traditional simulator calibration. As full Bayes calibration analysis is technically difficult and non-robust for large and complex simulators, it is usually helpful to identify the region of the input space which offers acceptable fits to the data, firstly, to act as a check on the simulator, by testing whether there are any input values for which the simulator gives an acceptable match to the data, and, secondly, to (massively) reduce the search space for the Bayesian algorithm.

The history match is carried out by assessing, for each x , the number of standard deviations between z and $f(x)$. If x is one of the members x_i of the evaluation set, $x_{[n]} = (x_1, \dots, x_n)$, then the value of $f(x)$ is known so that the variance of $(z - f(x))$ is the sum of the observation variance and the discrepancy variance. For each other value of x , instead z must be compared to the emulator mean for $f(x)$, and the difference, $(z - E(f(x)))$, has variance $V(x)$ given by the sum of the observation, discrepancy, and emulator variance.

For each value x , the number of standard errors between z and $E(f(x))$ can be assessed. This distance is expressed as an “implausibility” measure. For example, taking a single component $f_i(x)$ corresponding to data output z_i , a potential implausibility measure is

$$I_{(i)}(x) = \frac{|z_i - E(F_i(x))|^2}{V_i(x)}. \quad (2.12)$$

Large values of $I_{(i)}(x)$ suggest that it is “implausible” that this choice of x will give an acceptable representation of the observed data. Small values of $I_{(i)}(x)$ are consistent with the possibility that x will offer an acceptable representation, but could simply correspond to very large values of the emulator variance, so these values may not necessarily be viewed as plausible. Because the emulator is fast to evaluate, the implausibility measure is correspondingly fast to evaluate as x is varied.

The implausibility calculation can be performed on individual outputs or by using the corresponding multivariate calculation over subvectors. The implausibilities are then combined. For example, one might assess $I_M(x) = \max_i I_{(i)}(x)$ and consider that an input x would not be an acceptable choice if it gave a poor fit to any of the outputs of interest. It is a matter of judgement as to the magnitude of $I(x)$ that is considered unacceptable. There are various probabilistic heuristics that may be used as guidance in this judgement, for example, the “3 σ rule,” which states that for any continuous, unimodal distribution, at least 95 % of the probability lies within 3 standard deviations of the mean (see [14]).

Removing all input points, judged implausible leaves a “non-implausible” subregion of the input space. Efforts may now be refocused on this subregion, by making more simulator runs, refitting the emulator, and repeating the analysis over this subregion, to reduce further the non-implausible region. This cycle of resampling, re-emulating, and reducing the non-implausible space by use of implausibility measures can be repeated until the implausible region cannot be reduced any further.

A key feature that makes history matching so much more tractable than the standard Bayesian analysis is that there is no need to match to all of the data at once. At the beginning, some of the outputs will be relatively straightforward to emulate, while others will be much more difficult, as they demonstrate many different modes of behavior over the input space. Therefore, the initial waves of history matches are based around subvectors whose elements are both straightforward to emulate and physically meaningful. Reducing the size of the input space leads to focussing on parameter values which are constrained to correspond, in at least some aspects, to the actual behavior of the physical system. This means that then various of the other outputs will become straightforward to emulate. Thus, the size of the non-implausible region is progressively reduced by gradually expanding the collection of outputs to which the simulator can be matched. This process may be viewed as an iterative global search for the collection of all choices of x^* which would give acceptable fits to historical data. For examples and discussion of history matching, see, for example, [3] and [16].

10 Example: History Matching FUSE

Most of the components needed in (2.12) are now in place. The emulator gives $E_{f_1(x_{[n]})}(f_1(x))$ for any x . From the data, $z_1 = 2.89$. $V_1(x)$ is the sum of the emulator variance (calculated earlier), the measurement error (assessed as $(1/30)^2$ for the gauges in question), and the model discrepancy. This final component is a sum of the internal discrepancy (estimated as 0.124^2 earlier) and the external discrepancy, assessed informally as the same as the measurement error. Observe that the internal discrepancy is much higher than the external discrepancy and the measurement error, so the exact choices for these two quantities should be expected to have relatively minor influence on the history match.

Visualizing history matching is difficult in many dimensions, but useful two-dimensional plots of implausibility can be drawn. Consider a grid on the five-dimensional active parameter space, with say 50 grid points in each dimension, giving 50^5 points in total. For each x in this grid, $I_1(x)$ can be calculated and compared to 3^2 . Now consider the grid for only two of these five parameters. Each point in this grid corresponds to 50^3 parameters in the full grid. For each such point, the proportion of non-rejected parameters can be computed and these proportions can be plotted. The plots in Fig. 2.3 show such an approach, using parameter pairs $x_{(1)}$ and $x_{(22)}$, and $x_{(19)}$ and $x_{(22)}$. A location is colored red if most parameter choices at this pair are rejected and colored white if most are accepted. This allows

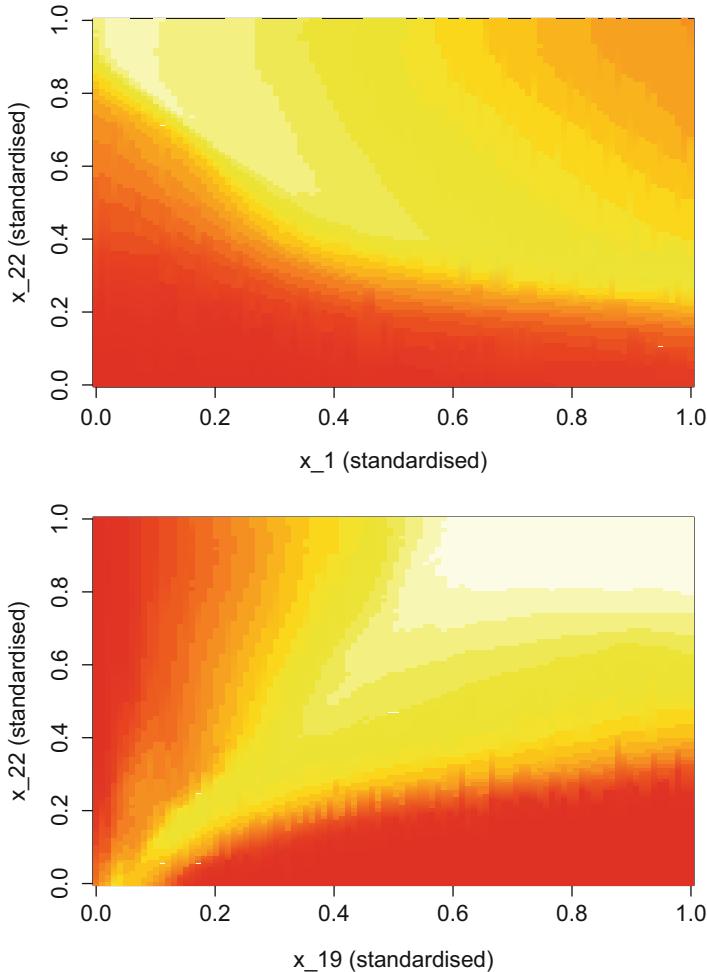


Fig. 2.3 Proportions of parameter space rejected at each pair of $(x_{(1)}, x_{(22)})$ and $(x_{(19)}, x_{(22)})$. Red represents almost all points rejected, and white represents almost all points accepted

visualization of the shape of the non-implausible region or regions. Note that this procedure involves very many calculations and requires the emulator: running the simulator at all these grid points would not be feasible even for FUSE.

The next step is to apply the standard method of refining the non-implausible region through successive “waves” of emulation and history matching. A new hypercube of $n_2 = 2000$ parameter choices $x_{[n_2]}$ was generated, and for each x in this hypercube, $I_1(x)$ from (2.12) was calculated. Any x with $I_1(x) > 9$ was removed. This leaves 1087 non-implausible parameter choices, at which the simulator is run. From this, a new emulator is built for the reduced parameter space (or, if the non-implausible space were several regions, an emulator would

be built for each region – this does not occur in this example). This new emulator takes into account only the behavior of the simulator in the reduced space and so will usually be superior fit to the global model. Given a new set of parameters, the original emulator can be used to discard some implausible parameter choices and then the new emulator used to discard even more. In this particular example, however, the second emulator has relatively little impact: from a third hypercube of 2000 parameters, 915 were discarded using the first emulator, and 44 more were discarded using the second emulator. This suggests that the possibilities of f_1 have been exhausted for the moment.

10.1 Introducing f_2

As expected, very few extra points were removed in the previous step. The next step is to bring back the quantity f_2 , the total stream flow over the period of interest; recall that this was initially difficult to emulate. Now, in the reduced space, a cubic fit is possible, giving R^2 of 0.86, with inactive parameters $x_{(4)}$, $x_{(6)}$, and $x_{(9)}$. This is sufficient to perform a history match. Assessing the nugget effect of these inactive parameters, however, suggested a high nugget variance. That is, one or more of the supposedly inactive parameters has a noticeable influence. Reactivating parameter $x_{(6)}$ improved this, although the nugget variance in this case is still not much below the measurement error, which is a concern. Reactivating other parameters, however, resulted in either a very high emulator variance or a very poor cross-validation result, depending on the choice of δ_2 . It was therefore decided to accept the nugget with parameters $x_{(4)}$ and $x_{(9)}$ inactive.

Some new quantities are required: a new measurement error (estimated as around 42.6^2 , using the assessment of measurement error at a single observation, and a similar argument for correlation as was used for the rain gauge measurement error), internal discrepancy (estimated as 251^2 in the same manner as that for f_1), and a new external discrepancy (assessed informally as $(200/3)^2$). The observed quantity z_2 is 2321. Observe that the measurement error is smaller relative to the observation than for f_1 ; this is a common feature of using total discharge rather than maximum. With an emulator, observed data, and all relevant discrepancy estimates, $I_2(x)$ from (2.12) can again be calculated for any new x .

A new hypercube of $n = 2500$ parameters was generated, and history matching was run sequentially on f_1 and then on f_2 . This yielded 1001 non-implausible parameter choices, at which the simulator was run. Using these 1001, another emulator was built for f_2 , giving a total of four emulators. In this new emulator, parameter $x_{(9)}$ became active, and now the nugget effect dropped to a near-insignificant value. Finally, a history match on 2800 new points with each emulator was performed, yielding 1086 non-implausible points. Leaving out the fourth history match would have yielded 1109 non-implausible points, so it appears that the limit of history matching on f_2 has been reached as well.

As an example, the plots from Fig. 2.3 are extended to cover each wave of history matching; this is shown in Figs. 2.4 and 2.5.

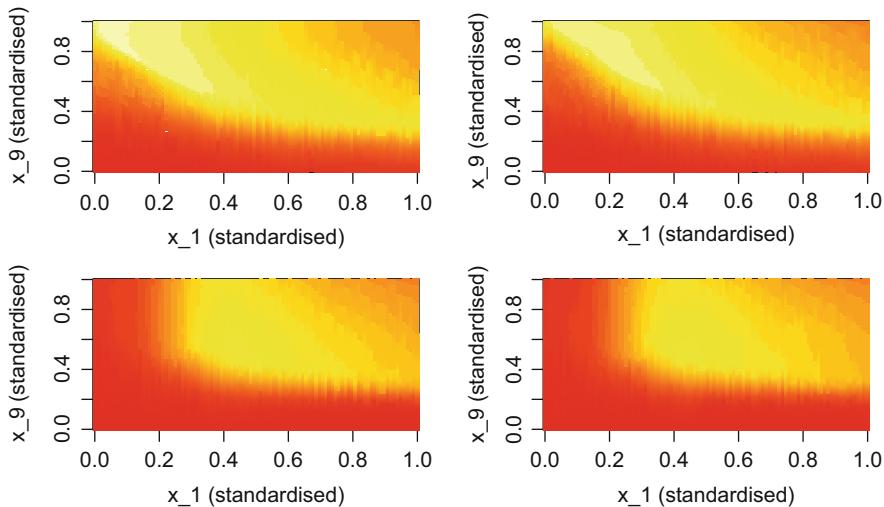


Fig. 2.4 Proportions of parameter space rejected at each pair of $(x_{(1)}, x_{(22)})$, over all four waves of history matching. Red represents almost all points rejected, and white represents almost all points accepted

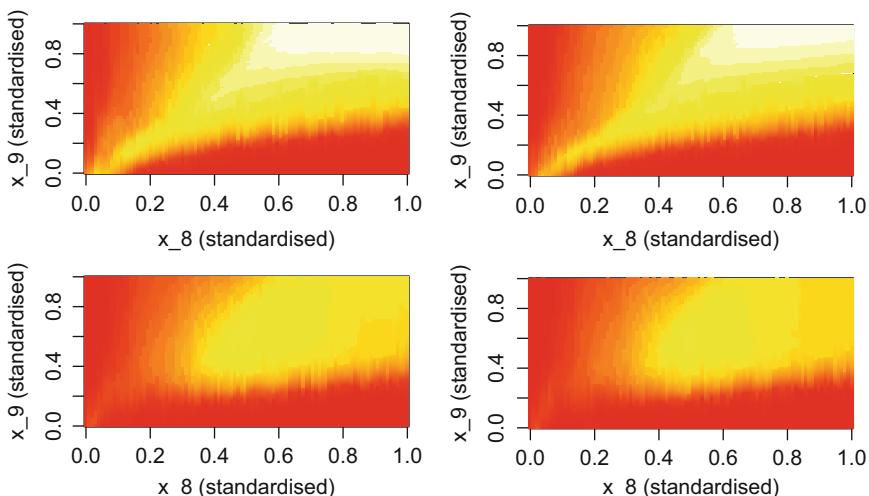


Fig. 2.5 Proportions of parameter space rejected at each pair of $(x_{(19)}, x_{(22)})$, over all four waves of history matching. Red represents almost all points rejected, and white represents almost all points accepted

Having again reached the limit of f_2 , but still with a fairly large non-imausible parameter space, the likely next step in practice would be to seek additional summaries to emulate or perhaps now attempt emulation of the whole time series at once. This would follow essentially the same pattern, so this example is not pursued further in this way.

10.2 Impact of External and Internal Discrepancy

This approach can be used to investigate the influence of the various discrepancies. One question is whether the external discrepancy is too low or conversely whether the estimated internal discrepancy represents most of the model discrepancy. If the history-matching procedure was returning no or very few non-imausible parameters, this might be a sign that the external discrepancy was too low. In the example, the non-imausible region is still large, so there is no particular reason to increase the external discrepancy. Note that if a third summary f_3 was introduced, and this rejected almost all points, then there may be reason to increase the external discrepancy at earlier stages.

On the other hand, it seems likely given its relative size that the internal discrepancy is the driving factor. To investigate this, the external discrepancy was set to zero. A history match under these conditions yielded 1048 non-imausible points, compared with 1086 with external discrepancy included. As expected, these numbers are close. Given that the estimate of internal discrepancy was quite crude, but very influential, this quantity would seem to need more attention.

There is noticeable variation in the 20 estimates for internal discrepancy calculated earlier. Using the maximum of the 20 estimates at all points could therefore be significantly overestimating the internal discrepancy at many points (and underestimating at some). This could be overcome by another use of emulation. The results of the internal discrepancy experiments can be viewed as a slow computer simulator, and so the emulation framework could be used to predict the result of those experiments at any other parameter choice. This would provide an expected value and a variance for the internal discrepancy variance at any parameter choice x , which would be used in the calculation of $I_i(x)$. Pursuing this idea is outside the scope of this article.

One final investigation is to suppose that the simulator is perfect, that is, that the measurement error, internal discrepancy, and external discrepancy are all zero. Running the history-matching procedure rejects all but 3 % of the parameter space. This is very small compared with the reductions already seen. The conclusion is that the emulators are powerful, and it is the intrinsic uncertainties in the simulator and observations that are restricting the ability to remove parameter choices.

11 Forecasting

Consider the prediction of future system outcomes, y_p , from data on past system outcomes, z . In order to make a Bayes linear forecast for y_p given z , the second-order specification relating these two vectors must be constructed. z is linked to y_p through relations (2.1) and (2.2). This specification divides into two parts. The first part arises from uncertainty as to the simulator values and the appropriate choice for x^* . The mean and variance of $f(x)$, for any x , are obtained from the mean and variance functions of the emulator for f . Using these values, the mean and variance of $f(x^*)$ can be computed by first conditioning on x^* and then integrating out x^* ,

over the input region identified by history matching. These values are combined with the second-order specification for the simulator discrepancy ϵ , derived, typically, as a combination of internal and external discrepancies. Adding the observational variance for z gives the joint second-order structure required, from (2.3), (2.4), the Bayes linear adjusted mean and variance for y_p given z ; for details, see [4].

This analysis is fast and straightforward, even for high-dimensional systems. It is sufficiently tractable to serve as a basis for a general forecasting methodology including suggesting informative designs for evaluating the simulator in order to minimize forecast variance, careful sensitivity analyses to identify the impact of the various modeling assessments that have been made on the accuracy of the forecast, and real-time control of the process.

This analysis is largely based on exploiting the global variation in the emulator and is most effective when the local component of emulator variation is small. A more detailed forecast, exploiting the local component of variation, can be made by using the method of Bayes linear calibrated forecasting. In this approach, one must construct a Bayes linear assessment \hat{x} for x^* given z , evaluate the simulator at \hat{x} , to give value $\hat{f} = f(\hat{x})$, compute the covariance between \hat{f} and (z, y_p) , and construct the Bayes linear adjustment of y_p based on both z and \hat{f} ; for details, see [8].

12 Example: Forecasting for FUSE

The quantity chosen to forecast was log maximum in the next rainfall period (the period beginning in October in Fig. 2.1). Only the simplest approach to forecasting is applied in this example. Note that across the non-imausible region, the simulator typically predicts log maximum in the range 2.5–3.5 for this period (the observed value is 3.03), so this gives an idea of the best accuracy of prediction that can be hoped for.

The first step is to build an emulator for this new quantity f_3 in the non-imausible region. This proceeded as usual, using a cubic fit and $\theta_3 = 0.35$. With this emulator, along with estimates of the discrepancy for f_3 and a given x^* , a prediction for $f_3(x^*)$, and hence z_3 , could be made. The same measurement error and external discrepancy as for f_1 were used, and internal discrepancy was estimated as usual from the 20 internal discrepancy experiments.

So, given x^* , the expected value and the variance of z_3 can be calculated. Since x^* is unknown, a uniform prior over the non-imausible region is assumed (since there is no particular information to suggest a nonuniform distribution). Therefore, x^* can be integrated out to find the mean and variance of z_3 , no longer conditional on x^* .

In the example, this gives $E(z_3) = 2.93$ with a standard deviation of 0.31. As expected, the uncertainty on this estimate is high. Although the high internal discrepancy contributes to this, the main problem is the size of and variability within the non-imausible region – this contributes around 75% of the variance. In practice, this would be a sign to either assess discrepancies more carefully and history match again or to find new quantities to match.

13 Conclusion

Emulation and history matching give a powerful and tractable framework for learning about the output of a computer simulator (through the simple statistical approximation of the emulator) and good parameter choices at which to run the simulator (through the iterative procedure of history matching). Bayes linear analysis provides a theoretical basis for performing these history matches without needing a complex and computationally challenging full Bayes analysis, but still allowing incorporation of prior judgements. This framework brings together judgements about parameters and about the simulator and its shortcomings, training runs of the simulator and of faster approximate simulators, and discrepancy estimates on the simulator. As shown in the example, this can involve straightforward and computationally inexpensive calculations to quickly reduce the non-imausible space of parameters. Once this is completed, forecasting for the future follows naturally.

The example contained simple estimations of both external and internal model discrepancy. These calculations can be extended in several ways if a more careful assessment of model discrepancy is required. As discussed earlier, the external discrepancy can be explored more thoroughly by the introduction of the reified model f^* . The internal discrepancy calculations can be improved by emulating the result of the internal discrepancy experiments, that is, by building a model for the standard deviation (and possible expected value) of the internal discrepancy for each parameter choice x . This would give low internal discrepancy in region where there is reason to believe it should be low, avoiding the problem encountered in the example where the high internal discrepancies occurred in implausible regions of the parameter space.

Appendix: Internal Discrepancy Perturbations

In this appendix, a description of the internal discrepancy experiments is provided. The first step is to identify potential quantities to perturb. For FUSE, the obvious quantities to choose are the two input time series and the initial condition. Also, the parameters could be perturbed at every time step. Similarly, the state vectors could be perturbed at every step, but this was not feasible in FUSE. Other possibilities not considered here include the time scale of the simulator and the accuracy of the numerical solver.

The next step is to informally assess the potential influence of each quantity. For example, increasing all rainfall by 10% makes a large difference to the output, whereas increasing all evapotranspiration by 10% makes a smaller but noticeable difference. Meanwhile, making large changes to the initial condition leads to extremely small changes away from the start of the simulation (recall that the quantities of interest are near the end of the simulation). From these initial explorations, each quantity can be categorized: if it has very little influence, it may

not be worth perturbing; if it has a small influence, it may be worth including but not expending much effort on; if it has a large influence, it is worth carefully modeling. The outcome of this exploration for FUSE suggested that the initial condition was hardly relevant, the evapotranspiration was worth including, and the rainfall and parameter perturbations deserved more attention.

The final step is to consider how to generate perturbations of each quantity. The initial condition is ignored. For evapotranspiration, good estimates of observation uncertainty are lacking, but given the low influence of this quantity this is not too worrying: any plausible perturbation should be sufficient. Each evapotranspiration observation was multiplied by a perturbation drawn from a log-normal distribution, such that most observations were perturbed by no more than 10 %. Correlation between observations within 24 h was also included, so if a particular observation has a high perturbation, nearby ones will also. This is motivated by the daily period of the evapotranspiration time series.

Parameter perturbations are performed by multiplying each initial parameter by some random perturbation, with nearby multipliers being correlated. The size of the perturbations were chosen such that the parameters rarely changed by much more than 10 % over the course of a simulation. This creates collections of perturbations that cause the parameters to evolve slowly without sudden large changes and without a large change overall. The parameter perturbations have a significant effect on the output, but expert opinion about how these are likely to change over time and by how much is lacking. In principle, in such a situation one should make the correlation and the magnitude of the perturbations configurable parameters, so as to understand their influence. For this example, however, this complication is avoided. An example of the evolution of a particular choice of $x_{(1)}$ for a particular perturbation can be seen in Fig. 2.6.

Perturbing the rainfall also has a significant effect on the output. In this case, however, there is some more guidance on the perturbations required. Sources of

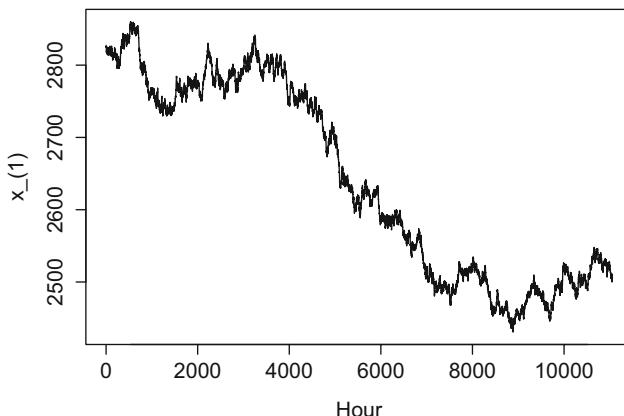


Fig. 2.6 The evolution of parameter $x_{(1)}$ for a particular parameter perturbation

uncertainty in the rainfall was attributed to three significant causes: the “local” gauge measurement error, the process of aggregating readings to the nearest hour, and the process of averaging over the catchment by kriging. Suitable perturbations from these errors were generated and combined.

The overall rainfall perturbations generated for this process typically display occasional noticeable differences but mostly small differences. This suggests that the rainfall error could contribute significantly to discrepancy for maximum stream flow, but not so much for discrepancy for average stream flow.

References

1. Bastos, L.S., O'Hagan, A.: Diagnostics for Gaussian process emulators. *Technometrics* **51**, 425–438 (2008)
2. Clark, M.P., Slater, A.G., Rupp, D.E., Woods, R.A., Vrugt, J.A., Gupta, H.V., Wagener, T., Hay, L.E.: Framework for Understanding Structural Errors (FUSE): a modular framework to diagnose differences between hydrological models. *Water Resour. Res.* **44**, W00B02 (2008)
3. Craig, P.S., Goldstein, M., Seheult, A.H., Smith, J.A.: Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments (with discussion). In: Gastonis, C., et al. (eds.) *Case Studies in Bayesian Statistics*, vol. III, pp. 37–93. Springer, New York (1997)
4. Craig, P.S., Goldstein, M., Rougier, J.C., Seheult, A.H.: Bayesian forecasting using large computer models. *JASA* **96**, 717–729 (2001)
5. Cumming, J., Goldstein, M.: Small sample Bayesian designs for complex high-dimensional models based on information gained using fast approximations. *Technometrics* **51**, 377–388 (2009)
6. de Finetti, B.: *Theory of Probability*, vols. 1 & 2. Wiley, New York (1974, 1975)
7. Goldstein, M.: Subjective Bayesian analysis: principles and practice. *Bayesian Anal.* **1**, 403–420 (2006)
8. Goldstein, M., Rougier, J.C.: Bayes linear calibrated prediction for complex systems. *JASA* **101**, 1132–1143 (2006)
9. Goldstein, M., Rougier, J.C.: Reified Bayesian modelling and inference for physical systems (with discussion). *JSPI* **139**, 1221–1239 (2008)
10. Goldstein, M., Seheult, A., Vernon, I.: Assessing model adequacy. In: Wainwright, J., Mulligan, M. (eds) *Environmental Modelling: Finding Simplicity in Complexity*, 2nd edn., pp. 435–449, Wiley, Chichester (2010)
11. Goldstein, M., Wooff, D.A.: *Bayes Linear Statistics: Theory and Methods*. Wiley, Chichester/Hoboken (2007)
12. Monteith, J.L.: Evaporation and environment. *Symp. Soc. Exp. Biol.* **19**, 205–224 (1965)
13. O'Hagan, A.: Bayesian analysis of computer code outputs: a tutorial. *Reliab. Eng. Syst. Saf.* **91**, 1290–1300 (2006)
14. Pukelsheim, F.: The three sigma rule. *Am. Stat.* **48**, 88–91 (1994)
15. Santner, T., Williams, B., Notz, W.: *The Design and Analysis of Computer Experiments*. Springer, New York (2003)
16. Vernon I., Goldstein M., and Bower, R.: Galaxy Formation: a Bayesian Uncertainty Analysis (with discussion). *Bayesian Anal.* **5**, 619–670 (2010)
17. Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., Yamazaki, K.: History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Clim. Dyn.* **41**(7–8), 1703–1729 (2013)

Habib N. Najm and Kenny Chowdhary

Abstract

In many practical situations, where one is interested in employing Bayesian inference methods to infer parameters of interest, a significant challenge is that actual data is not available. Rather, what is most commonly available in the literature are summary statistics on the data, on parameters of interest, or on functions thereof. In this chapter, we present a general framework relying on the maximum entropy principle, and employing approximate Bayesian computation methods, to infer a joint posterior density on parameters of interest given summary statistics, as well as other known details about the experiment or observational system behind the published statistics. By essentially redoing the experimental fitting using proposed data sets, the method ensures that the inferred joint posterior density on model parameters is consistent with the given statistics *and* with the model.

Keywords

Approximate bayesian computation • Bayesian inference • Maximum entropy • Missing data • Sufficient statistic

Contents

1	Introduction	34
2	Mathematical Setup	36
2.1	Bayesian Inference	36
2.2	Entropic Inference	39
2.3	Sufficient Statistics	42

H.N. Najm (✉)

Combustion Research Facility, Reacting Flow Research, Sandia National Laboratories,
Livermore, CA, USA

e-mail: hnnajm@sandia.gov

K. Chowdhary

Quantitative Modeling and Analysis, Sandia National Laboratories, Livermore, CA, USA
e-mail: kchowdh@sandia.gov

2.4	Consistent Data Sets and Approximate Bayesian Computation	43
3	Algorithmic Structure	44
3.1	Remarks	48
4	Applications	49
4.1	Algebraic System	49
4.2	Chemical System	56
5	Conclusion	65
	References	66

1 Introduction

Statistical inference methods are useful for estimation of uncertain model inputs and/or parameters based on experimental measurements of observable model outputs. Parameter estimation based on fitting to data can, of course, be done using least-squares fitting methods [4, 15, 20]. Further, this fitting can, under certain conditions such as Gaussian noise and model linearity in parameters, provide measures of uncertainty in fitted parameters. However, more generally, when these assumptions are relaxed, the most natural framework for uncertain parameter estimation based on data is a statistical framework [5].

Further, the Bayesian viewpoint on probability theory [2, 18], and associated statistical inference methods [6, 28], provide numerous advantages over alternate frequentist methods. Bayesian inference methods involve an in-built requirement for an explicit statement on prior knowledge. This requisite encapsulation of prior information in the fitting process is often necessary due to the need for regularization because of data sparsity, as is self-evident from the wide-scale use of regularization in deterministic optimization studies. The Bayesian framework incorporates regularization naturally through the prior [25, 31, 32]. Further, priors provide a basis for sequential learning and for learning from multiple experiments and heterogeneous data sources, among other uses. Bayesian methods also provide well-founded means for dealing with nuisance parameters, hypothesis testing, as well as model comparison/selection and averaging [9]. In the following, we place ourselves squarely within the Bayesian probabilistic framework and more particularly in the context of Bayesian inference for parameter estimation.

Given availability of measurement data on observable model outputs, there is a wealth of literature on the practical use of Bayesian inference methods for estimation of model inputs/parameters with quantified uncertainty [6, 28]. However, there are many practical situations where estimation of uncertain model parameters is necessary, but no data is actually available. Rather, what is typically available to the analyst is information in the form of data summaries. Our goal is to discuss the estimation of a joint probability density function (PDF) on the parameters in the absence of actual measurement data, but given such data summaries.

This is in fact a very typical scenario. A probabilistic characterization of uncertain model parameters is not typically available in the literature for a wide range of relevant models. Rather, one might find published information on nominal values and error bars on model parameters. Alternately, one might find similar

information, in terms of nominals and error bars, on specific functions of the parameters, e.g., fitted model output observables.

For example, consider a computational model $F(u, \lambda) = 0$, where u is a vector of observables and λ is a parameter vector. Estimation of uncertain predictions, namely, the PDF $p(u)$, requires knowledge of $p(\lambda)$. However, published literature, where λ is measured, does not report $p(\lambda)$. Rather, what is available is published information on nominals and error bars such as $\lambda = \mu_\lambda \pm \sigma_\lambda$, based on experimental measurements of λ . Alternately, what may be available, e.g., in an experiment where an observable $y = f(\lambda)$ was measured to estimate λ , are nominals and error bars on y , (μ_y, σ_y) . One can interpret $(\mu_\lambda, \sigma_\lambda)$ as specific moments of conditional or marginal densities on λ resulting from experimental fitting. Similarly, one might interpret (μ_y, σ_y) as similar moments of the pushed forward posterior density $p(y|D)$ or as statistics of the data distribution. The appropriate interpretation depends on the specific situation at hand. In general, such information can be viewed as providing summary statistics on the missing data D for the observable y . The core question one is faced with, then, is whether this information may be used in a statistical inference context to arrive at a meaningful density on λ .

This is, in fact, the typical situation in the context of computational modeling of chemical kinetic systems. Consider, for example, computations of a reacting hydrocarbon-air mixture. The governing equation system relies on a chemical source term for each species which involves contributions due to a number of elementary reaction steps. Considering a simple setting, the rate of progress of each reaction is determined by its stoichiometry, the participating species concentrations, and its reaction rate k , whose temperature dependence is commonly parameterized with the Arrhenius expression $k(T) = AT^n e^{-E/RT}$. The parameter vector $\lambda_i = (A_i, n_i, E_i)$ for each reaction i is typically measured using shock tube experiments and reported in different ways using nominals and error bars. One often finds reported best-fit values of (A, n, E) , with error bars only on $\ln A$, not on either of n or E . Alternately, in some cases, one finds published nominals and error bars on $\ln k(T)$ at a number of T -values. In neither case is the actual shock tube ignition data published along with the fitted parameters. Without the data, there is a need for an alternate procedure, aside from the direct application of Bayes' rule, to arrive at a meaningful joint density on the full set of parameters for all reactions. The limited amount of information available is not sufficient to arrive at a joint density exhibiting internal correlations among the Arrhenius rate parameters of each reaction, let alone correlations among parameters of different reactions.¹ Yet, a meaningful joint PDF on the input space is in principle necessary for reliable estimates of uncertainty in predictions with a chemical kinetic model. Again, the challenge is how to construct

¹It is worth noting that, presuming reported error bars in $\ln k(T)$ over a sufficient number of temperature points and considering a Gaussian parameter PDF, the joint correlation structure of $p(A, n, E)$ has been inferred through other means [23]. However, neither of these assumptions is required in the present approach.

a meaningful PDF on model parameters in the absence of actual data, but given some set of summary statistics based on a missing data set.

This problem is in fact solvable with recourse to the maximum entropy (MaxEnt) principle [7, 18]. In the absence of data, but given constraints on a distribution, the MaxEnt density is that density which maximizes relative entropy while satisfying the constraints. Maximizing entropy translates to maximizing uncertainty up to available information. The application of the MaxEnt principle relies on an explicit statement on the constraints in a partition function. This is not easily done given the above scenarios of interest because of the complex nature of the constraints. For example, we are not after a density that simply satisfies the stated nominals and bounds, but rather one that satisfies these *and* is consistent with the fitted model and the fitting that was done on the missing data to arrive at the given statistics. Rather, what is needed is a general computational procedure that allows the estimation of a joint density on parameters of interest given arbitrary and complex constraints ultimately stated as summary statistics. This is the procedure we outline in this chapter. The construction of this “data-free inference” (DFI) procedure is based on the MaxEnt principle, coupled with Approximate Bayesian Computation (ABC) methods. In the following, we outline the mathematical formulation of the method, followed by its general algorithmic structure. We then present illustrations of its use in simple problems as well as more complex problems of practical relevance.

2 Mathematical Setup

2.1 Bayesian Inference

In this section, we will provide the mathematical background for understanding the DFI procedure. We will start with a brief introduction to Bayesian inference and how it relates to entropic inference, i.e., MaxEnt.

For the examples in this chapter, it will be sufficient to only consider finite-dimensional random variables that live in a subset of Euclidean space. In general, however, let $(\Omega, \mathcal{A}, \mathbb{P})$ be the canonical probability triplet, where Ω is the sample space, \mathcal{A} the associated σ -algebra, and \mathbb{P} the probability measure on the σ -algebra \mathcal{A} . Let $X : \Omega \mapsto \mathbb{R}^d$ be a d -dimensional random variable which maps the event space into a subset of finite-dimensional Euclidean space. The random variable induces a probability triplet $(\mathbb{R}^d, \mathcal{B}, \mu)$, where \mathcal{B} is the Borel σ -algebra and $\mu \in \mathcal{P}(\mathbb{R}^d)$ is the measure induced by the random variable, where $\mathcal{P}(\mathbb{R}^d)$ is the space of all probability measures on \mathbb{R}^d . In particular, we have $\mu(B) = \mathbb{P}(\omega \in X^{-1}(B) : B \in \mathcal{B})$. When the probability measure μ is absolutely continuous w.r.t. the Lebesgue measure, we can write the density of the random variable as $\mu(x) \in \mathbb{R}$, $x \in \mathbb{R}^d$. From here on, we will assume that all measures introduced in this chapter are absolutely continuous w.r.t. the Lebesgue measure, which means that all probability measures will have an associated probability density function.

In the classical Bayesian inference procedure, consider a true model, f , which depends on an uncertain parameter $x \in \mathbb{R}^d$ and some auxiliary parameter t , which may represent some known quantities, e.g., time or temperature,

$$y_{\text{true}} = f(t; x), \quad (3.1)$$

where y_{true} represents the noiseless observation of the model. For example, $f(t; x)$ could be a linear function of x , e.g., $f(t; x) = t \cdot x$. Note that in many applications, one can make a reasonable assumption on the model form for $f(t; x)$ and the goal of the inference is then to determine the model parameters. To illustrate this point further, $f(t; x)$ may be a polynomial in t , and x represents the coefficients. Inferring x in this model is referred to as regression.

In the context of Bayesian inference, modeling x as a realization of the random variable $X \in \mathbb{R}^d$, the inference of x is the estimation of the posterior density on X given (typically noisy) observations of y_{true} . For the purposes of forward uncertainty quantification, one might also be interested in propagating the uncertain random variable X through the model $f(t; X)$, to obtain a distribution on the output. For the purposes of this work, we will focus on the Bayesian parameter estimation problem where the goal is to characterize the posterior density on X given observations of the model output.

As alluded to earlier, one does not directly observe measurements in the form of y_{true} . Instead, one observes the true model perturbed by some instrument error/noise. The Bayesian procedure now proceeds as follows. First, let $\beta(x) \in \mathcal{P}(\mathbb{R}^d)$ be a prior density on X , which represents our prior belief on the uncertain parameters [18]. Next, the likelihood or probability that the data came from a model y , not to be confused with y_{true} , with uncertain parameter $X = x$, will be denoted by $\gamma(y|x) \in \mathcal{P}(\mathbb{R}|\mathbb{R}^d)$ and is, not surprisingly, referred to as the likelihood density. Now, we have all the ingredients to apply Bayes' rule. The posterior density on the uncertain random variable X is given by the following ratio:

$$\nu(x|y) = \frac{\gamma(y|x)\beta(x)}{\int_{\mathbb{R}^d} \gamma(y|x)\beta(x)dx}. \quad (3.2)$$

The denominator is the “evidence,” or the marginal likelihood, being the integral of the likelihood over the prior-distributed parameters. For purposes of parameter estimation, it is sufficient to treat the evidence simply as a normalization constant. Thus, there is no need to evaluate the marginal likelihood integral.

For convenience, it is useful to consider the following observation model, where the noise is additive Gaussian. Thus, we write

$$y = f(t; x) + \sigma\eta \quad (3.3)$$

where $\eta \sim \mathcal{N}(0, 1)$ and σ is the standard deviation of the Gaussian noise. Generally, σ may be unknown, to be estimated from the data, and thus treated as

a hyperparameter in the Bayesian inference context. For the present discussion, we consider it known. Given this data model, the likelihood takes on the form

$$\gamma(y|x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\|y - f(t; x)\|^2\right). \quad (3.4)$$

Equation (3.2) provides a density for X given a single observation y . We now consider n observations, $\mathbf{y} = (y_1, \dots, y_n)$. Again, for convenience of illustration, we consider the case of *i.i.d* (independent and identically distributed) noise, such that $\gamma(\mathbf{y}|x) = \prod_{i=1}^n \gamma(y_i|x)$. The resulting posterior is then

$$\nu(x|\mathbf{y}) \propto \prod_{i=1}^n \gamma(y_i|x) \beta(x). \quad (3.5)$$

Now that we have a density for X , given by $\nu(x|\mathbf{y})$, let us assume we can obtain samples from this density, which is by no means a trivial task, but will help us illustrate the next point. If we propagate those samples through the data model in Eq. (3.3), we obtain the *posterior predictive* density $\rho_0(f(t, X) + \sigma\eta)$, where $X \sim \nu(x|\mathbf{y})$ and $\eta \sim \mathcal{N}(0, 1)$ [11, 22]. The posterior predictive is useful as a means of predicting the data and is often employed as a diagnostic to evaluate the quality of the inference. On the other hand, for prediction purposes, in order to estimate model predictions with quantified uncertainty, what is required is to push forward $\nu(x|\mathbf{y})$ through the predictive model. This can be done by propagating random samples of X through $f(t, x)$ alone, without the noise model. In this case, one obtains the *pushed forward posterior* density $\rho(f(t; X))$. Note the distinction between the pushed forward posterior and the posterior predictive. The posterior predictive propagates the uncertainty in X as well as the noise (and any associated uncertainty in noise model hyperparameters), through the data model. The pushed forward posterior, most commonly associated with forward uncertainty quantification, propagates the uncertainty in X through the predictive model $f(t, x)$. We will come back to these definitions after we discuss the idea of entropic inference and maximum entropy.

To motivate this transition, let us review the concept of Bayesian inference and how it relates to DFI. In the Bayesian procedure, one has the observation y^* and a likelihood function, $\gamma(y^*|x)$, which relates the data to the observation, measuring the likelihood of observing y^* given that the uncertain random variable is x . In the DFI context, we no longer observe y^* directly, but rather are given some function or summary statistic of y^* , call it $S(y^*)$, which may be a nonlinear, non-invertible mapping, e.g., $S(y^*)$ could be the variance of the observations y^* , or it could be the mean of the posterior distribution on $x \sim \nu(x|y^*)$. In this scenario, how do we calculate the likelihood for a particular realization y^* of the observed data if we only know $S(y^*)$? In other words, how do we calculate $\tilde{\nu}(S(y^*)|x)$, where $\tilde{\nu}$ does not necessarily have the same form as ν ? The hope is that the likelihood can be written explicitly as a function of $S(y)$, but this is rarely the case. Alternatively, for each

x , one could sample $S(y)$, fit a probability density, and measure the likelihood of $S(y^*)$, but this could be very costly. A less costly and more accurate procedure that integrates these two ideas is the following. The idea is to approximate the likelihood with

$$\tilde{v}(S(y)|x) \approx v(y|x)w(y), \quad (3.6)$$

where $w(y)$ is a probability density itself which weighs how close $S(y)$ is to the observed summary statistics. Essentially, this function holds information about the density of $S(y)$ w.r.t. the data space and is the work horse for this procedure. For example, $w(y) = \delta(S(y) - S(y^*))$, which means that $\tilde{v}(S(y)|x)$ is nonzero only when $y = y^*$ (assuming S is one-to-one). In the next section, we discuss how the Bayesian procedure can proceed with this new likelihood given that we only have $S(y^*)$ and not y^* directly.

2.2 Entropic Inference

In this section, we derive the Bayesian procedure from an entropic point of view [7], in which the DFI setting and the likelihood defined in Eq. (3.6) fit naturally. First, let $q(x)$ be a prior density on the parameters and $q(y|x)$ be the likelihood function relating the data and the parameters. Consider the joint density $q(y, x) = q(y|x)q(x)$ on both the data and parameters. Let $p(y, x)$ be some unknown joint density on the data and parameters, such that p is absolutely continuous w.r.t. q . The relative entropy between p and q is given by

$$\mathcal{E}(p, q) = -D_{\text{KL}}(p\|q) = -\int \log \frac{p(y, x)}{q(y, x)} p(y, x) dy dx, \quad (3.7)$$

where $D_{\text{KL}}(p\|q)$ is the Kullback-Leibler (KL) divergence from p to q [10]. D_{KL} is a pseudo-metric which enjoys some useful properties, such as convexity in both p and/or q , lower semi-continuity, and nonnegativity [10]. However, it should be noted that it is not symmetric nor does it satisfy the triangle inequality, which is why it is a *pseudo*-metric instead of complete metric. The idea of relative entropy originally came from Claude Shannon and was popularized by E.T. Jaynes [18, 26].

We now claim that the posterior density, $q(x|y)$, is a member of the class of densities that maximizes $\mathcal{E}(p, q)$ subject to a particular set of constraints. In the simplest case, where we have an observation y^* , the single constraint on the density is

$$p(y) = \delta(y - y^*), \quad (3.8)$$

so that the data is constrained to be the observation y^* , but in general, this need not be the case. In fact, in our construction, our constraint can be written more generally as

$$p(y) = \delta(S(y) - S(y^*)), \quad (3.9)$$

where $S(y)$ is again some statistic on the observation y .

Considering first the simple case with the constraint in Eq. (3.8), the feasible region for the relative entropy maximization problem is given by

$$\mathcal{P}_0 := \left\{ p(y, x) : p(y) = \delta(y - y^*), \int p(y, x) dy dx = 1 \right\}, \quad (3.10)$$

and this convex optimization problem can be written as

$$\max_{p \in \mathcal{P}_0} \mathcal{E}(p, q). \quad (3.11)$$

To summarize, the problem defined in Eq. (3.11) is to find the joint density $p(y, x) \in \mathcal{P}_0$ which maximizes $\mathcal{E}(p, q)$ or (equivalently) is *closest* to the joint prior $q(y, x)$ in terms of $D_{KL}(p \| q)$.

In order to solve this optimization problem, one can use the method of Lagrange multipliers. Note that since the constraint is a function, the Lagrange multiplier must be infinite-dimensional. Denote the Lagrange multiplier by $\lambda(y) \in \mathbb{R}$. The Lagrange function is given by

$$L(p, \alpha, \lambda) = \mathcal{E}(p, q) + \alpha \left[\int p(y, x) dy dx - 1 \right] + \int \lambda(y) [p(y) - \delta(y - y^*)] dy. \quad (3.12)$$

Differentiating and setting the roots to zero yield the solution

$$p(y, x) = q(y, x) \frac{e^{\lambda(y)}}{Z} \quad (3.13)$$

where Z is a normalization constant, to guarantee that $p(z, x)$ integrates to one, and $\lambda(x)$ is determined from the observation constraint (see [7] for a derivation). In order to satisfy the constraint, we must have $p(y) = \delta(y - y^*)$. Using Eq. (3.13), the marginal density on y is given by

$$p(y) = \int p(y, x) dx = \frac{e^{\lambda(y)}}{Z} q(y) \quad (3.14)$$

which must equal the integral constraint

$$\frac{e^{\lambda(y)}}{Z} q(y) = \delta(y - y^*). \quad (3.15)$$

Thus,

$$p(y, x) = q(x|y) q(y) \frac{e^{\lambda(y)}}{Z} = q(x|y) \delta(y - y^*). \quad (3.16)$$

Marginalizing over y finally shows that

$$p(x) = q(x|y^*) \quad (3.17)$$

which is Bayes' rule!

The entropic inference framework allows us to generalize Bayes' rule for the case when exact observations are not known, such as the case when only indirect functions of the observations, i.e., statistics, $S(y)$ are observed. In general, let $w(y)$ be a density describing the given data set, e.g., $w(y) = \delta(S(y) - S(y^*))$, as described in the previous section, and consider the modified class of feasible densities

$$\mathcal{P}_0 := \left\{ p(y, x) : p(y) = w(y), \int p(y, x) dy dx = 1 \right\}. \quad (3.18)$$

Thus, we are looking at joint densities for which $y \sim w(y)$, among the other constraints. Now, it can be shown (see [7]) that the joint density maximizing the entropy $\mathcal{E}(p\|q)$ is

$$p(y, x) = q(x|y)w(y). \quad (3.19)$$

Again, marginalizing over the data space gives

$$p(x) = \int q(x|y)w(y) dy = E_y[q(x|y)]. \quad (3.20)$$

Recall that marginalizing with the case $w(y) = \delta(y - y^*)$ brings us back to the original Bayes' rule, but in this case, since $w(y)$ may in general be any density, the analogous solution is computing the mean posterior density of x over $w(y)$. Now the DFI algorithm starts to take form. The idea is to first compute consistent data sets $y \sim w(y)$, and then, for each consistent data set, compute the posterior $q(x|y)$. Finally, to obtain samples from $p(x)$, we marginalize the joint samples over $y \sim w(y)$. This is also called linear average pooling. One disadvantage of linear pooling is that it is *not* externally Bayesian [3, 12], i.e., Bayesian updating and pooling do not commute, with the result that each consistent posterior has to be computed individually from each consistent data set, and saved, before proceeding to pooling. This can be quite inefficient. A more computationally efficient option is logarithmic pooling, which *is*, as discussed below, externally Bayesian. In this context, we average/pool the posterior densities as follows:

$$p(x) = \exp \left(\int \log(q(x|y)) w(y) dy \right). \quad (3.21)$$

Intuitively, linear pooling of densities is similar to taking the union of the densities, whereas logarithmic pooling of densities is similar to taking their intersection.

Logarithmic pooling can also be derived using an analogous maximum entropy procedure, but with the additional constraint that the joint density $p(x, y)$ can be written as $p(x)p(y)$. That is, the new feasible set is now

$$\mathcal{P}_2 = \left\{ p(y, x) : p(y) = w(y), \int p(y, x) dy dx = 1, p(y, x) = p(y)p(x) \right\}. \quad (3.22)$$

Then, it can be shown that the maximum entropy solution is Eqn. (3.21).

As already indicated, an important characteristic of logarithmic (or log for short) average pooling is that it is externally Bayesian [3, 12]. This means one can log pool the individual posteriors or, equivalently, calculate the posterior based on the log pooled likelihoods. More formally, suppose we have M samples from $w(y)$: y_1, \dots, y_M . Each data sample has a respective posterior density $q(x|y_i)$ for $i = 1, \dots, M$. We can write $q(x|y_i) = q(y_i|x)\pi(x)/q(y_i)$, where $q(y_i) = \int q(y_i, x) dx$ and $\pi(x)$ is a prior on the parameter space. For simplicity, let us assume that the prior for each posterior density is the same. If we let T denote the logarithmic average pooling operator of the M posterior densities, then it can be shown [12] that

$$T(q(x|y_1), \dots, q(x|y_M)) = T\left(\frac{q(y_1|x)\pi(x)}{q(y_1)}, \dots, \frac{q(y_M|x)\pi(x)}{q(y_M)}\right) \quad (3.23)$$

$$= \frac{T(q(y_1|x), \dots, q(y_M|x))\pi(x)}{T(q(y_1), \dots, q(y_M))}. \quad (3.24)$$

Thus, instead of parameterizing and saving M posteriors to be pooled, one can simply redo the inference one more time with the union of all consistent data sets and with the pooled likelihood.

2.3 Sufficient Statistics

Thus far, we have introduced a Bayesian procedure to infer our uncertain model parameters x where the likelihood as $\tilde{v}(S(y)|x)$ is approximated as $v(y|x)w(y)$, where S might be some mapping of some complicated statistic on the data. Ideally, one would like to observe y directly, since we have an analytic form for the data likelihood. So when is $\tilde{v}(S(y)|x)$ a good substitute for $v(y|x)$? The answer is when $S(y)$ is a *sufficient* statistic for the posterior density on x . In this case, any data set y that produces the same sufficient statistic, $S(y^*)$, as the true observed information will yield the same posterior density on x . In general, when using a likelihood-based inference, any two data sets with the same sufficient statistics, for the underlying model parameters, will yield the same inference on the posterior of those model parameters. This is due to the Fisher factorization theorem [21].

Let $p(y|x)$ be any likelihood function and let $S(y)$ be a sufficient statistic for the model parameter x . We will show that any set of data with the same value of $S(y)$ will yield the same inference, e.g., posterior, on x . By the Fisher factorization theorem, there exist nonnegative functions $p(S(y)|x)$ and $q(y)$ such that

$$p(y|x) = p(S(y)|x)q(y). \quad (3.25)$$

Next, with $\pi(x)$ as the prior on x , we have

$$p(x|y) = \frac{p(y|x)\pi(x)}{p(y)} = \frac{p(S(y)|x)q(y)\pi(x)}{p(y)} = \frac{p(S(y)|x)\pi(x)}{\tilde{p}(y)} = \tilde{p}(x|S(y)), \quad (3.26)$$

where the posterior distribution depends on the data only through the sufficient statistics, up to a normalization factor $q(y)$. Thus, any two data sets yielding the same sufficient statistics for x will yield the same posterior density. Further, it is important to be aware that sufficient statistics are not necessarily unique and multiple data sets could yield the same posterior distribution. Finally, we note that one can apply the DFI procedure irrespective of the sufficiency of the available statistics. Ideally one would indeed like to have sufficient statistics that fully capture the relevant information in the missing data. However, whether available statistics are sufficient or not, DFI seeks the posterior that satisfies them.

2.4 Consistent Data Sets and Approximate Bayesian Computation

Before we move on to the algorithm behind DFI, we need to spend some time discussing the work horse of the procedure, i.e., sampling consistent data sets $y \sim w(y)$. Recall that we need to do this in order to approximate the likelihood which relates the data y to the uncertain parameters x . Ideally, given an observed set of statistics $S(y^*)$, one would like to sample from $w(y) = \delta(S(y) - S(y^*))$, but this would be nearly impossible unless we can explicitly invert the operator S . However, in most cases, e.g., if S calculates the mean, this inverse problem is ill-posed and has infinitely many solutions. Instead, if we relax this constraint that $S(y)$ must exactly match $S(y^*)$, then the problem becomes well-posed again, although our likelihood becomes an approximation. This leads to Approximate Bayesian Computation (ABC) methods [1, 27].

With ABC methods, one has to derive a reasonable way to measure the discrepancy between the observed statistics, $S(y^*)$ and the sample statistic $S(y)$. For example, Berry *et al.* introduced a weight function [3] that penalized certain quantiles of the observed statistics, using the KL divergence pseudo-metric. We proceed in a similar manner. First, let us define the following subset of the data space:

$$\Delta_\varepsilon = \{y \in \mathbb{R}^n : d(S(w), S(y^*)) \leq \varepsilon\}, \quad (3.27)$$

where d is some metric or pseudo-metric that measures the discrepancy between the test statistic and the observed statistic. The subset Δ_ε represents the set of approximately consistent data. Note that as $\varepsilon \rightarrow 0$, the set Δ_ε becomes $\delta(S(y) - S(y^*))$. Next, we replace the likelihood function with the following approximate consistency function

$$w(y) \propto \mathbb{1}_{\Delta_\varepsilon}(y), \quad (3.28)$$

where the right-hand side is the indicator function. Alternatively, we can approximate the data likelihood directly using d , i.e.,

$$w(y) \propto d(S(y), S(y^*)) \quad (3.29)$$

where the right-hand side is at its maximum when $y = y^*$. We can also add a simulated annealing parameter $\delta \geq 1$ where

$$w(y) \propto d(S(y), S(y^*))^\delta. \quad (3.30)$$

In this case, larger δ corresponds to a likelihood which samples more frequently about the maximum. In fact, as $\delta \rightarrow \infty$, we obtain the maximum likelihood estimate. In the following sections, we will define the distance measure d and the observed statistics more precisely.

3 Algorithmic Structure

We describe in the following a specific algorithmic structure for the method. Alternate formulations can be explored of course, which can potentially be more computationally efficient; however, the present construction is both flexible and quite general. The construction relies on Markov chain Monte Carlo (MCMC) methods. It employs a nested pair of MCMC chains to explore the data and parameter spaces, arriving at a pooling of consistent data-marginalized posteriors on the parameters.

With unknown data y and parameters X and given some information I in terms of constraints on some functional of X , we are ultimately after the marginalized posterior

$$p(x|I) = \int p(x, y|I) dy = \int p(x|y) p(y|I) dy \quad (3.31)$$

where $p(x, y|I)$ is the MaxEnt joint posterior on data and parameters that satisfies given constraints. The DFI procedure provides means of estimation of this marginal

density without explicit estimation of the MaxEnt joint density. This is done by generating *consistent* data samples $\{y^1, y^2, \dots, y^M\}$ and *pooling* the resulting parametric posteriors $\{p_1, p_2, \dots, p_M\}$, where $p_i \equiv p(x|y_i)$ [3]. In this context, consistency is used to mean consistency with respect to the given information I , i.e., consistent data samples are samples from the data density $p(y|I)$. Further, as indicated above, pooling [3, 13] provides a means of combining information from different sources, in the form of consistent posteriors p_i , in a manner corresponding to the marginalization in Eq.(3.31), with optionally additional constraints. In particular, the above introduced linear average pooling providing the pooled posterior density as an arithmetic average

$$p_a = \frac{1}{M} \sum_i p_i \quad (3.32)$$

corresponds to the marginalization in Eq.(3.31). This linear pooling operation is *externally filtered*, such that it commutes with marginalization, while, as indicated above, it is not *externally Bayesian*, and therefore it does *not* commute with Bayesian updating [3, 12]. On the other hand, and as already stated, enforcing an additional requirement of independence of the data and parameters, via $p(x, y|I) = p(x|I)p(y|I)$, leads to logarithmic pooling, corresponding to a geometric average

$$p_g = \left[\prod_{i=1}^m p_i \right]^{1/M}, \quad (3.33)$$

an operation which is indeed externally Bayesian.

The key challenge in the algorithm is the generation of the consistent data sets. Once these are found, associated consistent Bayesian posteriors are easily found and pooled as desired. In principle, an MCMC procedure can be used to generate samples from $p(y|I)$, except that, in a computational setting, one does not have this density explicitly. Likelihood-free Approximate Bayesian Computation (ABC) methods have been used in this regard, employing a quasi-likelihood kernel density that enforces the requisite constraints by construction. An MCMC chain that uses this likelihood to judge acceptance of proposed data samples provides the means of generation of consistent data.

The overall algorithmic construction then looks as follows. We have an MCMC chain on data space whose objective is to generate consistent data samples $y \sim p(y|I)$. Generally, this chain can also include some hyperparameters θ that are involved in the formulation of the ABC likelihood, such that this “outer” chain is over the parameter vector $\zeta = (y, \theta)$. Bayes formula for the outer chain can be written as

$$p(\zeta|I) \propto p(I|\zeta)\pi(\zeta) \quad (3.34)$$

where $\pi(\zeta)$ is a requisite prior on ζ and $p(I|\zeta)$ is the ABC likelihood, which generally can combine K kernel densities, functions of associated statistics on ζ , in a manner that enforces requisite constraints. For example,

$$p(I|\zeta) = \prod_{k=1}^K F_k(S_k(\zeta)) \quad (3.35)$$

where $S_k(\zeta)$ is the k -th statistic on ζ and $F_k(\cdot)$ is the associated kernel density, e.g., of the form

$$F_k(S_k(\zeta)) = \frac{w_k}{\epsilon_k} F\left(\frac{\rho(S_k, S_k^*)}{\epsilon_k}\right) \quad (3.36)$$

where w_k and ϵ_k are scaling factors, $\rho(S_k, S_k^*)$ is a measure of distance between $S_k(\zeta)$ and the requisite value S_k^* of this statistic, and $F(\cdot)$ is the Gaussian kernel

$$F(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}. \quad (3.37)$$

An outer single-site MCMC chain on the data space can be written simply as shown in Algorithm 1. In this algorithm illustration, being with no loss of generality a single-site MCMC algorithm, s is the MCMC step index, ζ^s is the ζ at step s , \tilde{p}^s is the corresponding un-normalized posterior value, with

$$\tilde{p}(\zeta|I) \equiv p(I|\zeta)\pi(\zeta), \quad (3.38)$$

and v is a vector of variances for the normal proposal distribution used in the algorithm.

Algorithm 1 Outer chain single-site MCMC algorithm [17]

```

Input:  $\zeta^0 \in \mathbb{R}^N$ ,  $\tilde{p}^0 = \tilde{p}(\zeta^0|I)$ ,  $v \in \mathbb{R}_+^N$ 
Output: MCMC chain samples
foreach  $s = 1, \dots, M$  do
     $\zeta^s \leftarrow \zeta^{s-1}$ 
     $\tilde{p}^s \leftarrow \tilde{p}^{s-1}$ 
    foreach  $i = 1, \dots, N$  do
         $\delta\zeta_i \leftarrow \xi \sim N(0, v_i)$ 
         $\zeta^* \leftarrow (\zeta_1^s, \dots, \zeta_{i-1}^s, \zeta_i^s + \delta\zeta_i, \zeta_{i+1}^s, \dots, \zeta_N^s)$ 
         $\tilde{p}^* \leftarrow \tilde{p}(\zeta^*|I)$ 
         $\alpha \leftarrow \min(1, \frac{\tilde{p}^*}{\tilde{p}^s})$ 
        if  $u \sim U(0, 1) < \alpha$  then
             $\zeta^s \leftarrow \zeta^*$ 
             $\tilde{p}^s \leftarrow \tilde{p}^*$ 
        end
    end
end

```

At any step in the outer chain, given $\zeta^s = (y^s, \theta^s)$, we run an inner MCMC chain on the parameter space to evaluate the requisite statistics $S_k(\zeta^s)$, $k = 1, \dots, K$. In general, we define any statistic $S(\zeta)$ as a functional on the posterior density $p(\beta|y)$ of the model parameters β inferred given data y . Presuming some hyperparameters ϕ for the inner chain also, we define the state vector for the inner chain as $\lambda = (\beta, \phi)$ and write the inner chain posterior as

$$p(\lambda|y) \propto p(y|\lambda)\pi(\lambda) \quad (3.39)$$

where $\pi(\lambda)$ is the prior on λ , capturing prior belief on λ , and $p(y|\lambda)$ is the likelihood of data y given the fit model and λ . The formulation of the inner chain likelihood function depends on the presumed data model for the missing data from the experiment, involving both the fit model and a statistical model for the discrepancy between the model and the data. For example, presuming an additive zero-bias *i.i.d.* Gaussian noise model for the discrepancy, we have the data model

$$y = f(\beta) + \epsilon, \quad \epsilon \sim N(0, \sigma^2) \quad (3.40)$$

such that, with $\lambda = (\beta, \sigma)$, the likelihood based on data $y^s = (y_1^s, \dots, y_J^s)$ is given by

$$p(y^s|\lambda) \equiv L^s(\lambda) = \prod_{j=1}^J \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f(\beta) - y_j^s)^2}{2\sigma^2}\right). \quad (3.41)$$

With this likelihood, and the prior $\pi(\lambda)$, any suitable MCMC chain construction can be employed to generate samples from the posterior $p(\lambda|y^s)$. With these samples, requisite statistics $S_k(\zeta_s)$ can be evaluated. For example, statistics such as $E_{\lambda|y^s}[\beta]$, $\text{Var}_{\lambda|y^s}[\beta]$, $E_{\lambda|y^s}[f(\beta)]$, or $\text{Var}_{\lambda|y^s}[f(\beta)]$ can be easily evaluated and employed in the outer/data chain ABC likelihood to check consistency with associated given values.

With this, we have a number of consistent data sets y^s , and associated consistent posteriors $p(\lambda|y^s)$, for $s = 1, \dots, M$. A suitably pooled posterior can be estimated in a straightforward manner as outlined above, arriving at, e.g., an arithmetically or geometrically averaged posterior as in Eqs. (3.32) and (3.33). In particular, considering logarithmic pooling, which is externally Bayesian, one can, in order to avoid laborious parameterization of posterior densities, pool the data rather than explicitly pooling the posteriors. For example, for the above *i.i.d.* Gaussian likelihood in Eq. (3.41), we have

$$p_g = \left[\prod_{s=1}^M p(\lambda|y^s) \right]^{1/M} \propto \left[\prod_{s=1}^M p(y^s|\lambda) \right]^{1/M} \pi(\lambda) \quad (3.42)$$

$$\propto \left[\prod_{s=1}^M \prod_{j=1}^J \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f(\beta) - y_j^s)^2}{2\sigma^2}\right) \right]^{1/M} \pi(\lambda). \quad (3.43)$$

Thus, a single MCMC chain on the parameters, with the full data vector y^{pool} given by $\{y_j^s\}$, $s = 1, \dots, M$, $j = 1, \dots, J$, and with the modified likelihood

$$p(y^{\text{pool}}|\lambda) = L^{\text{pool}}(\lambda) = \left[\prod_{s=1}^M \prod_{j=1}^J \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f(\beta) - y_j^s)^2}{2\sigma^2}\right) \right]^{1/M} \quad (3.44)$$

provides the logarithmically pooled posterior.

Irrespective of the chosen pooling method, in the limit of sufficiently dense sampling of the data space, the resulting pooled posterior is the sought-after marginal MaxEnt density, being marginalized over the data. It is the density which is consistent with all given information on the model, the experiment, and summary statistics.

3.1 Remarks

The above DFI construction has been employed in a range of problems [3, 8, 24]. Before moving forward to a discussion of illustrative applications, we give in the following some general remarks regarding the algorithm.

1. One of the key challenges in the algorithm is the potential high dimensionality of the data space. Leaving aside any hyperparameters, with N data pairs (x_i, y_i) , the data chain is $2N$ -dimensional. It is quite feasible to imagine many practical problems with large N , with an associated high-dimensional MCMC chain on the data. In this context, it becomes crucial that numerous MCMC details be judiciously chosen to ensure accurate results with reasonable computational efficiency. This includes careful choices of the chain starting point, priors on data space, and the proposal distribution. We have found that data chain burn-in, in particular, can be a significant computational burden in high dimensions. Further, it is precisely because of the challenges in proposal distribution selection in high dimensions that our above-illustrated MCMC chain on the data space relies on single-site proposal structure.
2. The outer chain, by construction, is sampling along a manifold in data space, where the manifold is defined by the given constraints. While the chosen ABC kernel width adds a certain small width to the manifold, sampling along the manifold in data space does remain a challenge. While it remains to be shown, it is likely that geometric MCMC methods that take the geometry of the manifold into consideration in adapting the proposal distribution would be useful in this

regard. Further, in principle, any point on the manifold is equally good. As a result, while “good mixing” may be sought-after as regards jumps orthogonal to the manifold, it is in principle expected that the chain may wander aimlessly *along* the manifold. In order to get good statistics from the data chain, the key requirement is that it gets good “coverage” of the manifold. In order to ensure this, it is useful to run many data chains in parallel, with randomly chosen starting points, and to pool all the resulting consistent data sets. This strategy allows assessment of the convergence of the resulting statistics by estimating the fractional change in select metrics with the inclusion of additional chains.

3. The nested structure of the construction, while flexible and generally useful, is nonetheless a computational burden. It is clearly of interest to explore a single-chain version of the algorithm that employs joint sampling on both the data and parameter spaces.
4. The practical implementation of the algorithm does rely on knowing a number of necessary details about the study behind the reported statistics. For example, aside from knowing the details of the physical system, the details of the model that was fitted, and the operating conditions of the experiment, it is also necessary to know some aspects of the measurement process, for example, the noise structure of the instrument, e.g., the distribution of the noise, its correlation structure if any, and its dependence on operating conditions. Further, if the fitting done in the original experiment employed parameters from previous measurements, these need to be accounted for as nuisance parameters, with specified uncertainty. Moreover, the actual number of data points used in the experiment needs to be estimated. This can be difficult, depending on how much information is reported in the literature. In a case where this or other pertinent details are missing, one can assign a density that encapsulates what one *does* know about the relevant detail and to then marginalize the results over this distribution. For example, if N , the number of data points, is unknown, but can be confidently bounded from below and above, then a MaxEnt uniform probability mass function can be assigned to N in this range. Another important experimental detail that is frequently hard to assess is the precise meaning of error bars reported. This is a challenge in general, aside from the present purposes. Nonetheless, it is an issue in the present context, because everything depends on the meaning of the declared statistics. Frequently, a judicious assumption is necessary, declaring, for example, that the error bars refer to a specific multiple of the standard deviation, or specific quantiles, of the reported observable interpreted as a random variable.

4 Applications

4.1 Algebraic System

4.1.1 Setup

Consider a nonlinear model problem that involves the chemical decomposition of an unstable radical, with initial concentration c_0 at time $t = 0$. Synthetic data pairs

of time and concentration, (t_i, y_i) , are generated using a double exponential decay truth model $s(t)$, with a multiplicative data noise model.

$$s_i = 2e^{-2t_i} + \frac{1}{2}e^{-\frac{1}{2}t_i}, \quad (3.45)$$

$$\begin{aligned}\gamma_i &= \frac{1}{4}s_i + \frac{1}{20}, \\ y_i &= s_i + \gamma_i \varepsilon_i,\end{aligned}$$

where ε_i are i.i.d. $\mathcal{N}(0, 1)$ random variables and t_i are the time values in the set

$$\tau = \{i/2 \quad i = 1, \dots, 10\}, \quad (3.46)$$

with $t_i \in [0, 5]$. Also, at each time point, 10 data points are generated for a total of 100 data points, i.e., the experiment was performed ten times. Here, s_i is the true signal, ε_i is Gaussian white noise, and y_i 's are our observations.

A researcher observes these time-concentration data pairs and reports fitting a *single* exponential decay model to this data (not knowing the true underlying double exponential decay model). Furthermore, instead of publishing the data pairs, (t_i, y_i) , the researcher publishes summary statistics, namely, nominal values and error bars on the predicted concentration from the fitted single exponential decay model at a set of time instances [16]. This means that our $S(y)$ function will involve a complicated and potentially computationally expensive procedure, which will be outlined in the next two subsections.

As readers of this published report, we want to use the author's results and produce detailed predictions with associated uncertainty estimates, i.e., we want to know the full posterior distribution for the uncertain model parameters. We could easily do this if the observed data pairs were published, but in this case, we have no such luck. Thus, we need to infer consistent data sets w.r.t. the given summary statistics, then use those data pairs to perform our own model fitting, and pool those results to obtain an averaged posterior. Before we describe the procedure for generating consistent data sets, let us describe the single exponential decay model that the inference is based on. This will help determine the mapping $S(y)$. It should be emphasized that the fitted model and the procedure for obtaining the processed data products or summary statistics are presumably reported in the author's work.

4.1.2 Single Exponential Model Fit

Let us assume that the author fits a single exponential decay model with the observed, but unpublished, (synthetic) data pairs using a Bayesian procedure. The reported single exponential decay model is

$$\tilde{s}_i = ae^{-kt_i} \quad (3.47)$$

$$\tilde{\gamma}_i = \sigma_1 \tilde{s}_i + \sigma_2 \quad (3.48)$$

$$y_i = \tilde{s}_i + \tilde{\gamma}_i \varepsilon_i. \quad (3.49)$$

for $i = 1, \dots, N_t = 10$, where k is the decay rate parameter, a is the initial concentration, and σ_1, σ_2 are nuisance parameters. With N data points at each t_i , let $n = N \cdot N_t$ be the dimensionality of the observed data. Furthermore, let the t_i 's be known, deterministic variables, while the nuisance parameters, σ_1 and σ_2 , are not known. By assumption, the author infers and marginalizes over these parameters to obtain a distribution on the model parameters of interest, a and k .

The fitting is done using a Bayesian approach, which necessitates the need for a likelihood function for the model, as well as a choice of priors. Let $x = (a, k)$, $\sigma = (\sigma_1, \sigma_2)$, and define the data set $y = (y_1, \dots, y_n)$. Then, the likelihood is

$$p(y|x, \sigma) \propto \left(\prod_{i=1}^n (\sigma_1 a e^{-kt_i} + \sigma_2) \right)^{-1} \cdot \exp \left\{ -\frac{1}{2} \sum_{i=1}^n \left(\frac{y_i - ae^{-kt_i}}{\sigma_1 a e^{-kt_i} + \sigma_2} \right)^2 \right\}. \quad (3.50)$$

Using improper uniform priors on σ , the posterior on x and σ is then given by

$$p(x, \sigma|y) \propto p(y|x, \sigma). \quad (3.51)$$

This summarizes the Bayesian fitting procedure presumably reported to us in the author's work. We can now define the procedure that the author used to produce the summary statistics, which we denote by $S(y)$.

4.1.3 Summary Statistics for Processed Data Products

We presume that the author reported error bars on the predicted concentration at a set of time instances, using the fitted model. More specifically, say the author reported summary statistics in the form of means and 2σ quantiles (equivalently, 95% confidence interval) on the pushed forward posterior at particular time instances.² These types of summary statistics will also be referred to as processed data products (PDPs). Figure 3.1 shows the published summary statistics. From this figure, we will attempt to infer the original data used to create it. To do this, we assume that we know how the summary statistics were computed (without actually knowing the data used to create them).

We now assume that the processed data products were computed by the following procedure. First, given the raw data y^* , we presume that the author determined the posterior distribution $X \sim p(x|y^*)$ on the model parameters and computed mean and quantiles (twice the standard deviation) for $f(t; x)$ where $x \sim p(x|y)$ (this represents the pushed forward posterior) for each t_i . This is summarized by the mapping $S : y \in \mathbb{R}^n \mapsto \mathbb{R}^{2N_t}$ defined as

$$S : y \in \mathbb{R}^n \mapsto x \sim p(x|y) \mapsto (\mu_{f_1}(y), \dots, \mu_{f_{N_t}}(y), \sigma_{f_1}(y), \dots, \sigma_{f_{N_t}}(y)), \quad (3.52)$$

²We will assume that the author had an assumption of normality, hence their interpretation as 95% quantiles.

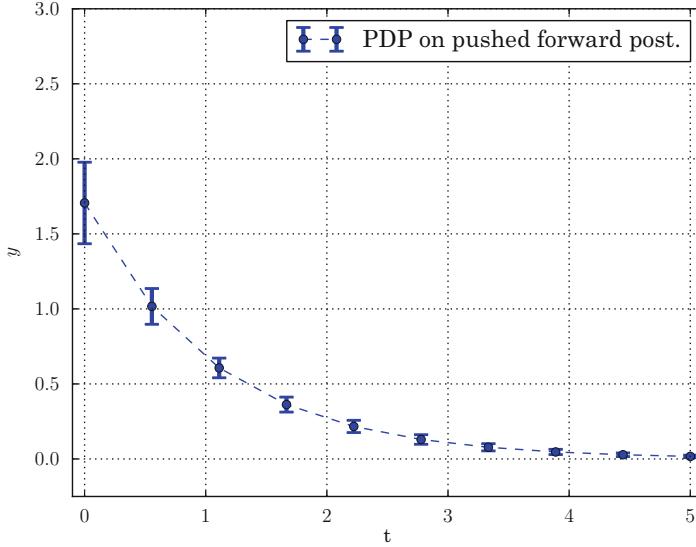


Fig. 3.1 Summary statistics or processed data products provided in the published report. These are error bars on the pushed forward posterior through the single exponential decay model [30]

where

$$\mu_{f_i}(y) = \mathbb{E}_{p(X|y)}[f(t_i; X)] = \int_{\mathbb{R}^d} f(x; t_i) p(x|y) dx, \quad (3.53)$$

$$\sigma_{f_i}^2(y) = \mathbb{E}_{p(X|y)}[(f(t_i; X) - \mu_{f_i})^2] = \int_{\mathbb{R}^d} [f(t_i; x) - \mu_{f_i}]^2 p(x|y) dx. \quad (3.54)$$

It is important to note that the mapping S requires the posterior to be computed, which makes S computationally expensive.

At this point, we, the readers of the published report, only have access to the fitting procedure and the PDPs $S(y^*)$ and *not* the actual observed data y^* . Our goal is to produce a set of data that is consistent with the given summary statistics. In other words, we want to choose the data y s.t. $S(y)$ is *close* to the given $S(y^*)$. The next section details a possible definition for such a discrepancy measure, which we denote as $w(y)$.

4.1.4 PDP Pseudo-metric

Let $y^* \in \mathbb{R}^n$, denote the unpublished, hidden data used to produce the PDP $S(y^*)$. In order to obtain an appropriate pseudo-metric, $w(y)$, for these types of processed data products, it is useful to think of $w(y)$ as a way to penalize the data y for being inconsistent with the given processed data products, i.e., not having $S(y)$ close, in some sense, to $S(y^*)$. We will derive w in two parts. For convenience, let us write

$$w(y) = w^{(1)}(y) \times w^{(2)}(y). \quad (3.55)$$

The first term $w^{(1)}$ will try to force consistency between the data sets and the given pushed forward posterior means (contained in $S(y^*)$). Likewise, the second term $w^{(2)}(y)$ will try to force consistency between the data sets and the pushed forward posterior quantiles (interpreted as $\mu_{f_i}(y) \pm 2\sigma_{f_i}(y)$, also contained in $S(y^*)$).

In order to check consistency between the data sets and the pushed forward posterior mean, we define the function $w^{(1)}(y) : \mathbb{R}^n \mapsto \mathbb{R}$ as follows:

$$w^{(1)}(y) = \prod_{i=1}^{N_t} \exp(-\delta_1(\mu_{f_i}(y) - \mu_{f_i}(y^*))^2) \quad (3.56)$$

where δ_1 is a tunable parameter, i.e., the annealing parameter [14, §12.4]. If we assume that the sampled pushed forward posterior means $\mu_{f_i}(y)$ are distributed normally, i.e., $\mu_{f_i}(y) \sim \mathcal{N}(\mu_{f_i}(y^*), \delta_1^{-2})$, we may interpret $1/\delta_1^2$ as the variance for this sample mean. In practice, because we have to perform sampling to approximate $\mu_{f_i}(y)$, we can justify this approximation from a central limit type argument, where δ_1 is proportional to the square root of the number of sample points. Alternatively, if we substitute the definition of $\mu_{f_i}(y)$ into Eq. (3.56), we get

$$w^{(1)}(y) = \prod_{i=1}^{N_t} \exp \left(-\delta_1 \left(\int_{\mathbb{R}^d} [f(x; t_i) - \mu_{f_i}(y^*)] p(x|y) dx \right)^2 \right). \quad (3.57)$$

It is clear that this weight function is enforcing the mean of the pushed forward posterior.

To enforce the variance or quantiles of the pushed forward posterior on model outputs, one can take a similar approach to (3.56) and simply replace $\mu_{f_i}(y)$ with $\sigma_{f_i}(y)$. However, we choose a more robust approach outlined in [3], which performs a trinomial quantile check. Instead of comparing means and variances directly, which can be highly sensitive to sampling noise, i.e., via Monte Carlo estimation, we sample the pushed forward posterior of a sample data set y and bin the samples according to the 2.5 % – 95 % – 2.5 % quantiles according to the given summary statistics. This binning produces a trinomial density for the sample data set y . Ideally, the trinomial density will have 2.5 % – 95 % – 2.5 % probability masses in each bin. To check consistency, we use relative entropy/KL divergence to compare the sample trinomial density, computed from the pushed forward posterior based on y , with the ideal trinomial density, with probability masses 2.5 % – 95 % – 2.5 %.

In order to properly define this pseudo-metric, we must first make an assumption that the data provided for the standard deviation can be equivalently interpreted as quantiles. That is, if we are given the mean and standard deviation μ and σ , we can equivalently say that 95 % of the data falls within $[\mu - 2\sigma, \mu + 2\sigma]$ and 2.5 % of the data falls below $\mu - 2\sigma$ and above $\mu + 2\sigma$. If the underlying distribution is normal, this assumption is justified. If this assumption cannot be justified, then using (3.56)

with σ_{f_i} instead of μ_{f_i} will suffice. In the case that the processed data products are provided in the form of quantiles, we do not need to make any assumption at all, and the following approach is justified.

That being said, let us assume that the quantiles for the pushed forward posterior, $f(x; t_i)$, are given to us in the form of $[\mu_{f_i}(y^*) - 2\sigma_{f_i}(y^*), \mu_{f_i}(y^*) + 2\sigma_{f_i}(y^*)]$ for $i = 1, \dots, N_t$. We introduce a function $g_i(y)$ which measures the mismatch between the pushed forward samples and the summary statistics. This measure essentially checks that 2.5, 95, and 2.5 percent of the pushed forward posterior samples fall within the bins

$$\begin{aligned} & [-\infty, \mu_{f_i}(y^*) - 2\sigma_{f_i}(y^*)] \\ & [\mu_{f_i}(y^*) - 2\sigma_{f_i}(y^*), \mu_{f_i}(y^*) + 2\sigma_{f_i}(y^*)] \\ & [\mu_{f_i}(y^*) + 2\sigma_{f_i}(y^*), \infty], \end{aligned} \quad (3.58)$$

respectively. In other words, $g_i(y)$ measures the likelihood that the correct binning is achieved. To measure the mismatch, we use the KL density [3, 10]. In order to define $g_i(y)$, let us introduce the following probabilities. Let $p_{f(t_i; x)}(y)$ be the pushed forward posterior density at t_i . Then, define

$$\begin{aligned} p_i(y) &:= \int_{y < \mu_{f_i}(y^*) - 2\sigma_{f_i}(y^*)} p_{f(t_i; x)}(y) dy \\ q_i(y) &:= \int_{y > \mu_{f_i}(y^*) + 2\sigma_{f_i}(y^*)} p_{f(t_i; x)}(y) dy, \\ r_i(y) &:= 1 - p_i(y) - q_i(y). \end{aligned} \quad (3.59)$$

Let $P_i(y) = [p_i, r_i, q_i](y)$ represent a probability density, with $p_i + q_i + r_i = 1$, which represents the percent of samples that fall within the 2.5%–95%–2.5% bins defined in (3.58). Ideally, we would like this density to be $Q = [0.025, 0.95, 0.025]$ for $i = 1, \dots, N_t$. Using the KL density to compare measures, one obtains

$$g_i(y) \propto \exp(-\delta_2 D_{\text{KL}}(P_i(y) \| Q)) \quad (3.60)$$

where increasing $\delta_2 > 0$ acts to improve the match between P_i and Q , by increasing the severity of the application of the constraint. Finally, we want this to hold true for $i = 1, \dots, N_t$ which gives us the corresponding weight

$$w^{(2)}(y) \propto \prod_{i=1}^{N_t} g_i(y) = \prod_{i=1}^{N_t} \exp(-\delta_2 D_{\text{KL}}(P_i(y) \| Q)). \quad (3.61)$$

Combining weights $w_2^{(1)}$ and $w^{(2)}$ gives

$$w(y) \propto \left[\prod_{i=1}^{N_t} \exp(-\delta_1(\mu_{f_i}(y) - \mu_{f_i}(y^*))^2) \right] \times \left[\prod_{i=1}^{N_t} \exp(-\delta_2 D_{\text{KL}}(P_i(y) \| Q)) \right]. \quad (3.62)$$

As we will see in the examples, it may be desirable to only match or penalize a subset of the provided quantiles. That is, let $B \subset \{1, \dots, N_t\}$ and consider the reduced weight function

$$\tilde{w}(y) \propto \left[\prod_{i \in B} \exp(-\delta_1(\mu_{f_i}(y) - \mu_{f_i}(y^*))^2) \right] \times \left[\prod_{i \in B} \exp(-\delta_2 D_{\text{KL}}(P_i(y) \| Q)) \right]. \quad (3.63)$$

Note that \tilde{w} does not enforce a matching between all N_t means and quantiles, but rather a smaller subset B of those same means and quantiles. We now explain in the following how we employ these constraints to generate consistent data sets and arrive at the pooled posterior of interest.

4.1.5 Generating Consistent Data and Pooling the Posterior

With $w(y)$ properly defined, let us utilize Algorithm 1 to generate replica data sets from these processed data products, consistent with the given summary statistics. Note that in Algorithm 1, we do not consider N a random variable. Instead, we fix N at an arbitrarily chosen value.³ Once these data sets have been generated, we then produce a posterior on the model parameters for each consistent data set and pool these distributions. We detail the application of this approach below.

We are given a total of ten error bars at ten different time points, i.e., $N_t = 10$. Using the means and quantiles from the error bars, we generate consistent data using Algorithm 1. For convenience, we will fix N to be 5, which means we have underestimated the dimensionality of the data set (since the original data set contains $N = 10$ observations for each time point). We use Algorithm 1 with $\delta_1 = \delta_2 = 300$. Each run of the algorithm generates 3000 samples. To obtain a sizable number of consistent data sets, we employ MCMC runs for 50 different random seeds to obtain a total of 150,000 data sets. Additionally, as done above, and for the same reasons, instead of matching the full set of 10 quantiles for $t \in \tau$, we ran Algorithm 1 with a smaller subset of error bars, i.e., $B = \{1, 3, 5, 8\}$.

Figure 3.2 shows the quantiles of each data set versus the given quantiles from the PDP. This figure shows how well the algorithm produced consistent data.

After generating consistent data sets, we want to be able to infer the posterior on the model parameters of the original data set. Figures 3.3 and 3.4 show the posterior

³Again, we do this to reduce the computational cost of generating data sets over multiple dimensions. Although we do not marginalize over the dimensionality, we show that for a reasonable choice of the dimension parameter $N = 5$, we can still recover consistent data sets.

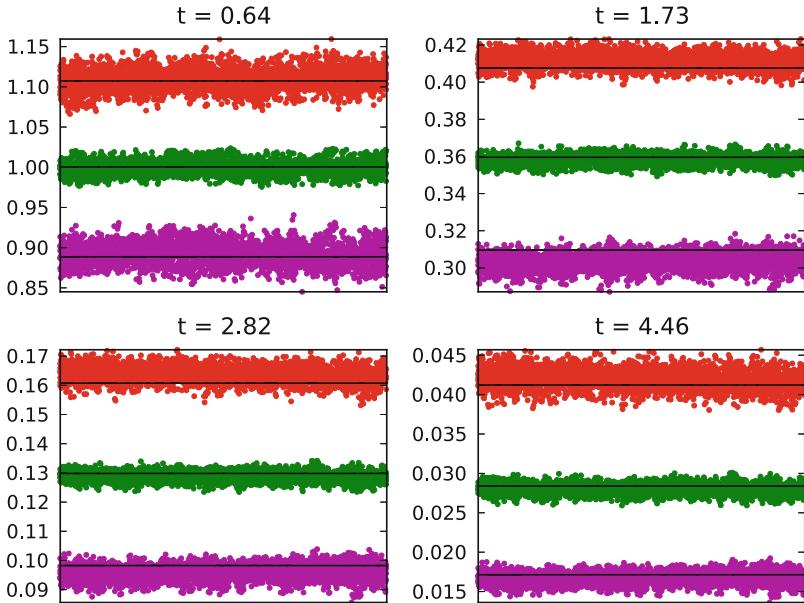


Fig. 3.2 The *black lines* denote the true quantiles (means and 2.5 % and 97.5 % quantiles) based on the original raw data set (hidden from us, but displayed here for comparison purposes) at four different times. The *green dots* represent the means for each consistent data set generated using Algorithm 1 with $\delta_{1,2} = 300$. Likewise, the *red* and *magenta dots* represent the upper and lower quantiles, respectively, for each consistent data [30]

means and covariance of each individual sample versus the original posterior, similar to what is displayed in Figure 3.2. Figure 3.5 displays the log and linear pooled posterior from all 150,000 samples, superposed on the true posterior. In this case, both pooled posteriors are very similar, since the generated data does not exhibit much sample-to-sample variance in the posterior. More importantly, they both agree very well with the true posterior, given (a) the correct interpretation of the error bars in this case, and (b) the evident near sufficiency of the given statistics.

We also show, in Fig. 3.6, quantiles for consistent data when $\delta_2 = 150$ instead of $\delta_2 = 300$. As expected, the quantiles are less tightly consistent, but in fact still within an acceptable amount.

We also repeated the experiment with the dimensionality $N = 10$ to simulate the true dimension of the data set. We found that the results were very similar to $N = 5$ so we omit the results.

4.2 Chemical System

We now present an illustration of the method as applied in the context of parameter estimation in a chemical system. We consider a realistic scenario where we are

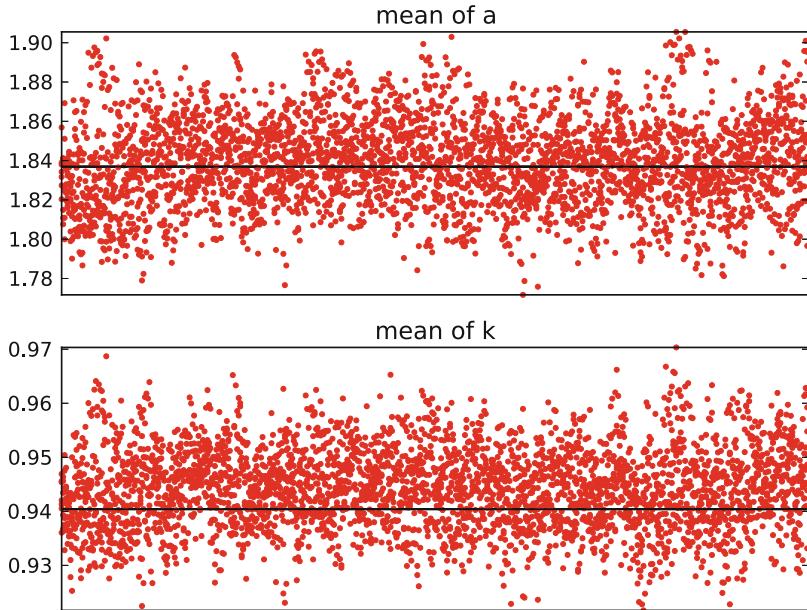


Fig. 3.3 The black lines denote the true posterior means from the raw data set, unbeknown to us. The red dots represent the means for the model fit parameter after each consistent data set, generated using Algorithm 1, is fit to the single exponential decay model via a Bayesian approach [30]

given summary statistics on Arrhenius reaction rate coefficients of a global methane-oxidation reaction based on some missing chemical ignition data set. We consider these statistics to be based on a joint posterior density on the parameters that is not reported and use them to infer a marginal MaxEnt posterior that is consistent with the given information.

More specifically, and in order to enable an examination of the effectiveness of the approach, we consider a synthetic data context, where we first generate methane-air ignition data with a detailed chemical mechanism, namely, GRIMech3.0 [29], over a range of initial temperature, and corrupt it with multiplicative *i.i.d.* Gaussian noise. We then use this data in a Bayesian inference context to calibrate a single-step global methane-oxidation model, arriving at a joint posterior on its two parameters of interest (A, E), being the pre-exponential and activation energy. Evaluating and retaining nominal values and marginal posterior 5 % and 95 % quantiles on each of the two parameters, we next discard the data, use the DFI algorithm to discover the marginal MaxEnt posterior, and compare it to the known “true” posterior based on the original data set.

The data D is a collection of 11 pairs of ignition time and initial temperature [24]. The synthetic data from the detailed model before and after corruption by multiplicative *i.i.d.* Gaussian noise is shown in Fig. 3.7. With this data, and

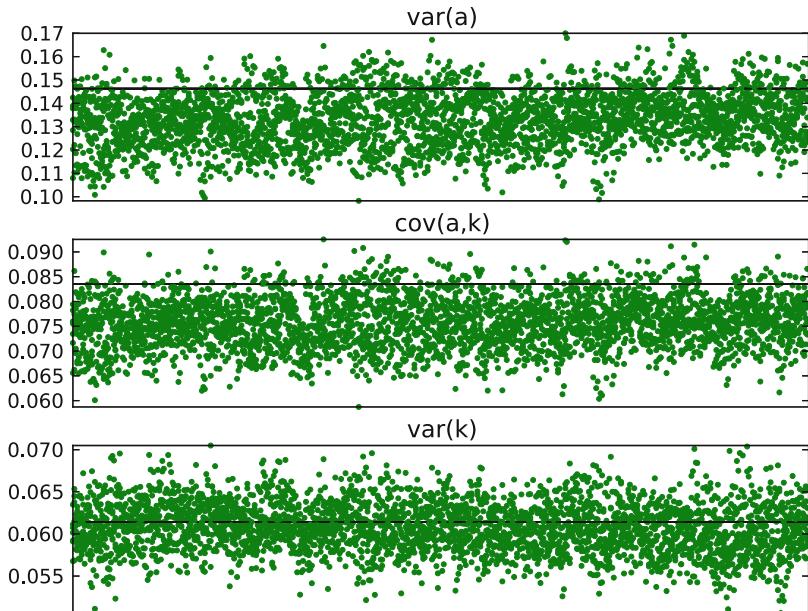


Fig. 3.4 The *black lines* denote the true posterior covariance from the original (synthetic) data set, unbeknown to us. The *green dots* represent the covariance for the model fit parameter after each consistent data set, generated using Algorithm 1, is fit to the single exponential decay model via a Bayesian approach [30]

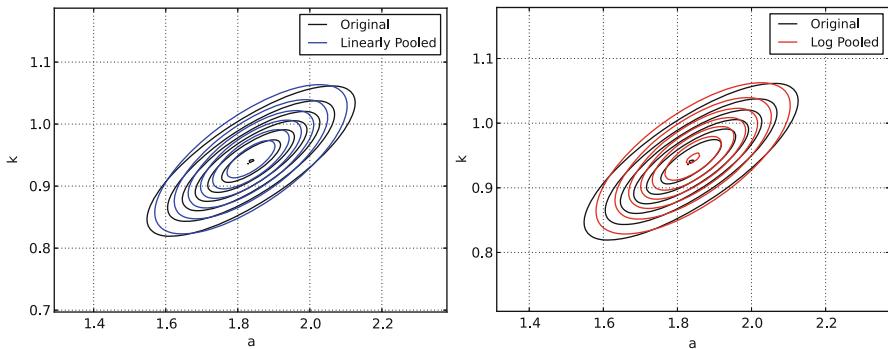


Fig. 3.5 The true posterior, based on the original (synthetic) data set, is shown in *black* contours. The linear and log pooled covariance is shown in the *left* and *right* images, respectively, using Algorithm 1 when we have error bars on the pushed forward posterior [30]

employing a corresponding data model, we use Bayesian inference [24] to arrive at the posterior density on the two parameters of interest $p(\ln A, \ln E | D)$, shown in Fig. 3.8. Extracting the nominal values, as well as the 5 % and 95 % quantiles,

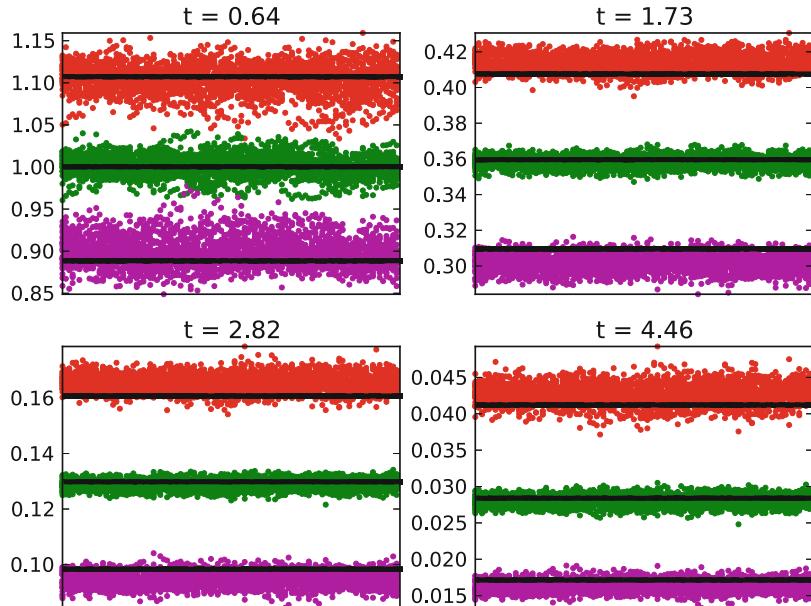


Fig. 3.6 The *black lines* denote the true quantiles (means and 2.5 % and 97.5 % quantiles) based on the original, raw synthetic data set (hidden from us, but displayed here for comparison purposes) at four different times. The *green dots* represent the means for each consistent data set generated using Algorithm 1 with $\delta_{1,2} = 150$. Likewise, the *red* and *magenta dots* represent the upper and lower quantiles, respectively, for each consistent data [30]

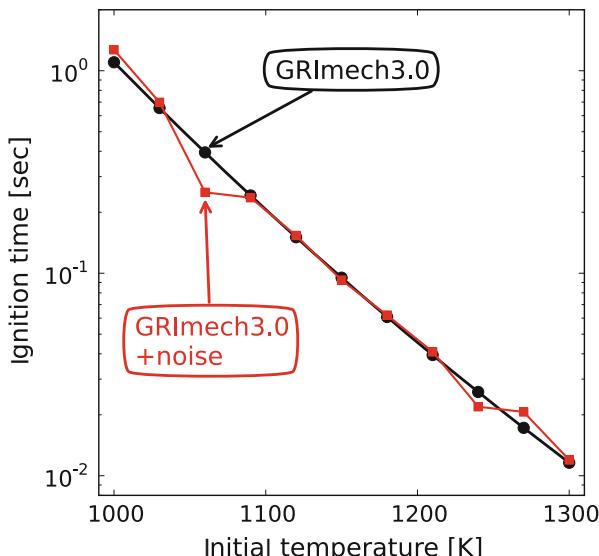
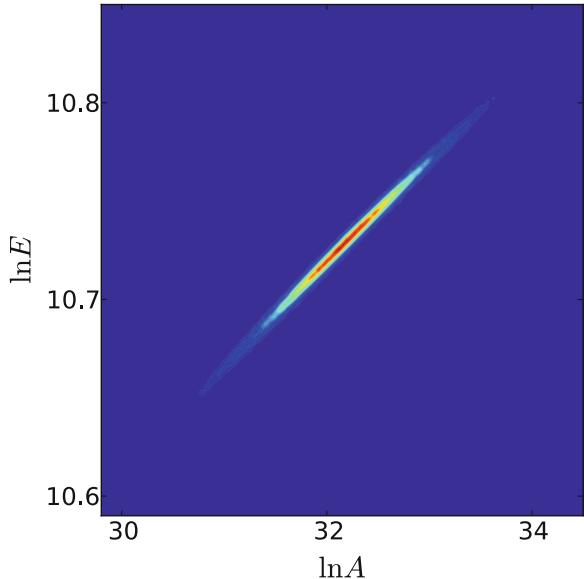


Fig. 3.7 Ignition time computed with GRImech3.0 over a range of variation of initial temperature, including also the corresponding noisy synthetic data points used for inference [17]

Fig. 3.8 Joint 2D marginal posterior on $(\ln A, \ln E)$ [17]



from the marginal parameter posteriors, we then proceed to the application of DFI, forgetting for the moment both the data and the true posterior structure.

For the application of DFI, we presume 8 data pairs, as opposed to the true original number of 11, for illustration. We do this in order to accentuate the point that in an actual application with experimental data that is not reported, the number of missing data points may not necessarily be known. As we indicated earlier, one in principle can propose a density that encapsulates knowledge of the number of points. Here, we simply posit a number, that is different from the truth. An analysis of the impact of different choices for the number of points, not reported here, suggests a small effect on the result of inference in this particular situation.

The construction of the data chain likelihood relies on kernel densities that enforce the nominal values of $(\ln A, \ln E)$ and their 5 %/95 % quantiles. With the data chain state $\zeta = (z, \ln \sigma_d)$, where z is the vector of 8 data pairs and σ_d is the standard deviation of the noise, we define the data chain likelihood as

$$p(I|\zeta) = w_\delta(z, \ln \sigma_d) = F_\delta(z) \frac{p(z, \ln \sigma_d | \beta_0)}{\max_{\beta, \sigma} [p(z, \ln \sigma | \beta)]}. \quad (3.64)$$

where $\beta \equiv (\ln A, \ln E)$, β_0 is the nominal value of β , and $p(z, \ln \sigma_d | \beta_0)$ is evaluated from the likelihood function of the inner chain. Specifically, we have

$$p(z, \ln \sigma | \beta) = p(z | \beta) p(\ln \sigma | \beta) = p(z | \beta, \ln \sigma) \pi(\beta, \ln \sigma) / \pi(\beta) \quad (3.65)$$

where $\pi(\beta, \ln \sigma)$, $\pi(\beta)$ are priors on $(\beta, \ln \sigma)$ and β , respectively, and $p(z | \beta, \ln \sigma)$ is the inner chain likelihood. This likelihood is based on the model prediction of ignition time $\tau^m(T^o, \beta)$ for any initial temperature T^o and parameter vector $\beta = (\ln A, \ln E)$, on the multiplicative *i.i.d.* Gaussian noise data model $\tau = \tau^m(1 + \sigma\epsilon)$, and on the data set $\{(T_i^o, \tau_i^d)\}_{i=1}^N$ for the given data chain step, here with $N = 8$. Specifically, we have

$$p(z | \beta, \ln \sigma) \propto \frac{1}{(2\pi\sigma^2)^{N/2}} \frac{1}{\prod_{i=1}^N \tau^m(T_i^o, \beta)} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N \left[\frac{\tau_i^d}{\tau^m(T_i^o, \beta)} - 1 \right]^2 \right\}. \quad (3.66)$$

Thus, the ratio term on the right-hand side of Eq. (3.64) enforces consistency with the nominal parameter values, as $p(z, \ln \sigma_d | \beta_0)$ measures the likelihood of the proposed outer chain state for the *nominal* value of β . Further, the normalization by the denominator shown in Eq. (3.64) ensures that the ratio is in $[0, 1]$.

The $F_\delta(z)$ term, with $\delta > 0$, is constructed to enforce consistency with the quantiles. Specifically, we use

$$F_\delta(z) \propto \prod_{i=1}^2 f_\delta \left([p_i(z), 1 - p_i(z) - q_i(z), q_i(z)] \mid [0.05, 0.90, 0.05] \right) \quad (3.67)$$

where,

$$p_1(z) = \int_{\ln A < (\ln A)_5} p(\beta | z) d\beta, \quad q_1(z) = \int_{\ln A > (\ln A)_{95}} p(\beta | z) d\beta, \quad (3.68)$$

$$p_2(z) = \int_{\ln E < (\ln E)_5} p(\beta | z) d\beta, \quad q_2(z) = \int_{\ln E > (\ln E)_{95}} p(\beta | z) d\beta, \quad (3.69)$$

where $p(\beta | z)$ is the marginal inner chain posterior on the parameters β , $\{(\ln A)_5, (\ln E)_5\}$ are the specified 5% quantiles and $\{(\ln A)_{95}, (\ln E)_{95}\}$ the specified 95% quantiles on $(\ln A, \ln E)$, and the trinomial density with probabilities $p + r + q = 1$ is denoted $[p, r, q]$. Further, f_δ is defined as [3]

$$f_\delta([p, r, q] | [0.05, 0.90, 0.05]) = \exp \left\{ -\delta \left(p \ln \frac{p}{0.05} + r \ln \frac{r}{0.90} + q \ln \frac{q}{0.05} \right) \right\}, \quad (3.70)$$

being the KL density [3] formed of the KL divergence from the density $[p, r, q]$ to $[0.05, 0.90, 0.05]$, where, for two measures (μ, ν) with ν absolutely continuous w.r.t. μ , we have

$$f_\delta(\mu | \nu) \propto \exp(-\delta \cdot D_{\text{KL}}(\mu \| \nu)) \quad (3.71)$$

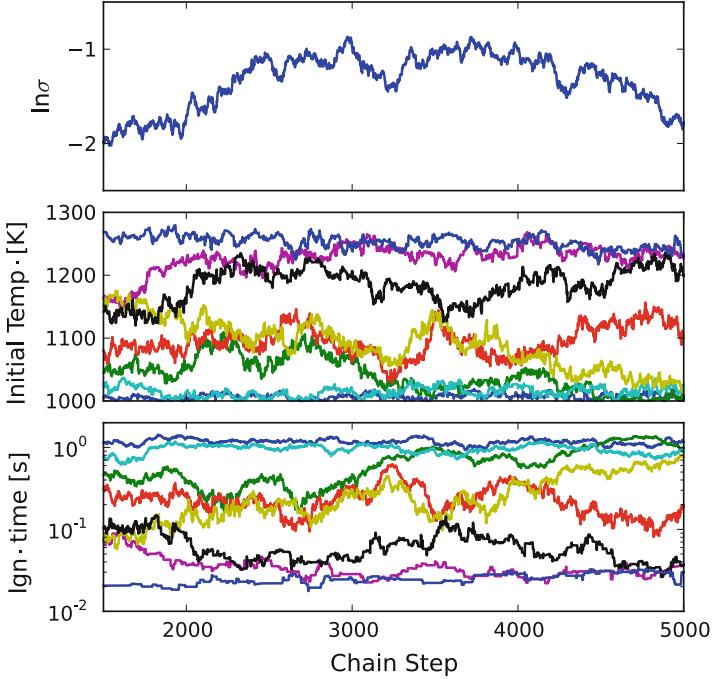


Fig. 3.9 A segment of the DFI data chain, showing the MCMC sampled values of the data vector. In the *bottom* two frames, each line corresponds to the sampled value of each data point, (T_i^o, τ_i) , $i = 1, \dots, 8$ [17]

and $D_{\text{KL}}(\mu \| \nu) \geq 0$ is the KL divergence from μ to ν . The choice of δ provides a control on the sharpness of the density f_δ , with larger values of δ leading to sharper f_δ , being also by construction in $[0, 1]$.

A segment of the data chain is shown in Fig. 3.9. Each state of the chain is composed of ignition time τ_i and initial temperature T_i^o , for $i = 1, \dots, 8$, and the standard deviation of the noise σ . The consistency of data chain samples from 50 independent outer chains is illustrated in Figure 3.10, where the scatter of summary statistics from accepted chain samples is shown on top of the enforced values for both $\ln A$ and $\ln E$. As already indicated, the δ -factor in the outer chain likelihood controls the tightness of the bounds consistency check and therefore the scatter in the quantiles observed in the figure. Further, the consistency of the data samples from the 50 chains, relative to the actual data behind the statistics, is illustrated in Fig. 3.11. Clearly, satisfying the requisite statistics, as fitted using ignition with the global chemistry model, generates data sets that are close to the original data set.

The resulting marginal logarithmically pooled posterior on $(\ln A, \ln E)$, based on the 50 chains, is shown in Fig. 3.12. The figure indicates the near congruence between the DFI posterior and the reference posterior. We note, however, that this agreement between the two posteriors is not necessarily expected. It clearly has to

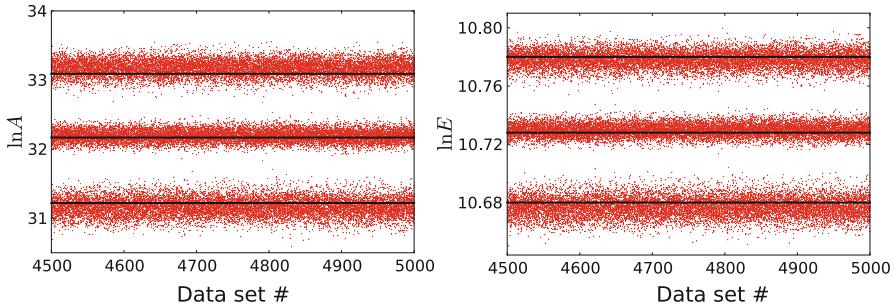


Fig. 3.10 A plot of the means, as well as the 5 % and 95 % quantiles from the parameter posteriors from each data chain step, for all 50 chains, and for the last 500 steps from each chain. The *left frame* shows those for $\ln A$, while the *right frame* shows those for $\ln E$. The *solid lines* indicate the corresponding values from the reference posterior [17]

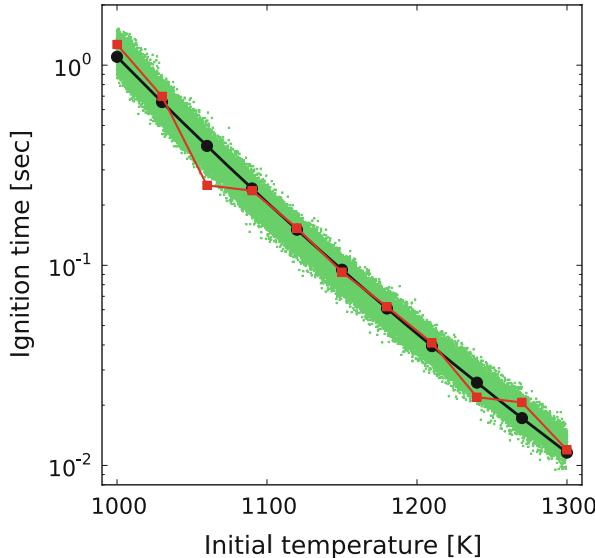


Fig. 3.11 Scatter plot of 50 chains, with the first 1500 burn-in states removed from each chain. Only every 25th chain state is plotted for convenience [17]

do with the amount of given information, the nature of the summary statistics, and the structure of the model. In this case, evidently, posteriors consistent with all this given information are essentially equivalent to the missing posterior. In other words, the available information constitutes a nearly sufficient statistic [6].

Finally, the statistical convergence of the algorithm is highlighted in Fig. 3.13. Here, the convergence of the KL divergence between successive pooled posteriors, with increasing data volume, is illustrated. The abscissa k is the number of

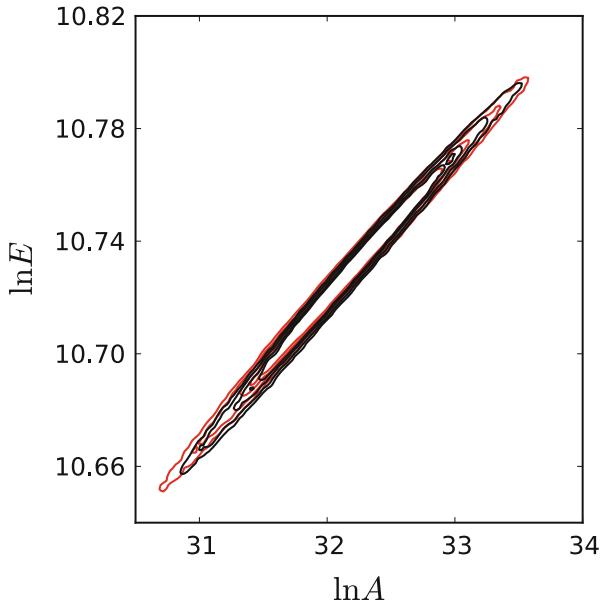


Fig. 3.12 Two-dimensional marginal parameter posteriors on $(\ln A, \ln E)$ resulting from pooling 50 chains of data (red), compared with the reference known posterior (black) [17]

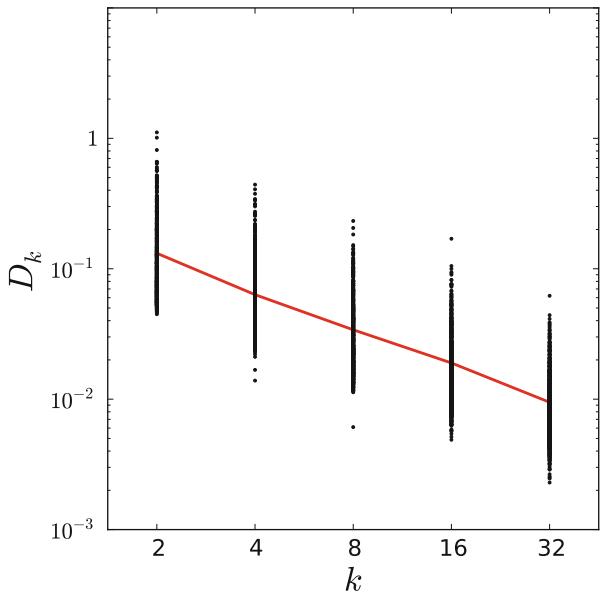


Fig. 3.13 Plot showing the statistical convergence of the algorithm in terms of the KL divergence (KLdiv) between successive pooled posteriors. Also shown is the scatter in each of the KLdiv values. The red line indicates the evolution of the mean with increased number of chains [17]

(equal-length) chains. Thus, if the pooled posterior with k -chains is p_k , we define the KL divergence D_k as

$$D_k = D_{KL}(p_{k/2} || p_k) \equiv \int p_{k/2} \ln \frac{p_{k/2}}{p_k} d\ln A d\ln E d\ln \sigma. \quad (3.72)$$

The results highlight the scatter of D_k , for any given k , resulting from 1024 combinatorial choices of k -chains out of 50. The average divergence, however, exhibits a roughly inverse linear dependence on k , such that $\bar{D}_k \propto 1/k$. Thus, with additional data, we have an associated linear convergence of the pooled posterior.

5 Conclusion

This chapter has focused on the context of Bayesian learning from summary statistics based on missing data. We motivated this focus with practical challenges faced in forward UQ given probabilistically incomplete published information on uncertain model parameters, this being most commonly in the form of summary statistics. We outlined a maximum entropy framework within which the problem can be posed in general and discussed the relationship between the maximum entropy and Bayesian data analysis frameworks. We also outlined a general algorithm, relying on the use of Approximate Bayesian Computation methods for solving the MaxEnt problem, providing sampling from the MaxEnt posterior on the joint data-parametric space, and pooling/marginalizing over the data space to arrive at the marginal posterior density on parameters of interest. Finally, we illustrated the application of this algorithm, with synthetic data, on an algebraic model problem as well as the practical context of estimation of Arrhenius reaction rate coefficients for a global chemical model of methane-air oxidation, based on reported summary statistics on these parameters.

Aside from these applications, the method has been used for estimation of rate constants for an elementary H_2/O_2 reaction, namely, $H + O_2 = OH + O$, based on reported processed data summaries from a shock tube experiment [19]. In this application in particular, the practical utility of having a meaningful joint density on model parameters, rather than the default of ignoring such un-reported correlations, has been illustrated in terms of associated impact on uncertainties in predicted model observables. Similarly, the utility of the method in allowing the handling of other previously measured uncertain model parameters as nuisance parameters, and exploring the correlation among these and the parameters of interest, has been studied.

In closing, it is useful to emphasize that, given the legacy of published literature on physical models in general, where partial information is provided on uncertain model inputs, largely composed of nominal values and error bars on each parameter, with hardly ever any other information on correlations among parameters, let alone the joint PDF on all parameters, there is a significant need for further development and demonstration of statistical methods, such as the above DFI algorithm, that

allow adequate handling of such partial information to build meaningful joint densities on model parameters and enable accurate estimation of uncertainty in computational predictions.

References

1. Beaumont, M.A., Zhang, W., Balding, D.J.: Approximate Bayesian computation in population genetics. *Genetics* **162**(4), 2025–2035 (2002)
2. Bernardo, J., Smith, A.: *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley, Chichester (2000)
3. Berry, R., Najm, H., Debusschere, B., Adalsteinsson, H., Marzouk, Y.: Data-free inference of the joint distribution of uncertain model parameters. *J. Comput. Phys.* **231**, 2180–2198 (2012)
4. Bevington, P., Robinson, D.: *Data Reduction and Error Analysis for the Physical Sciences*, 2nd edn. McGraw-Hill, New York (1992)
5. Box, G.E., Hunter, J.S., Hunter, W.G.: *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd edn. Wiley, New York (2005)
6. Carlin, B.P., Louis, T.A.: *Bayesian Methods for Data Analysis*. Chapman and Hall/CRC, Boca Raton (2011)
7. Caticha, A., Preuss, R.: Maximum entropy and Bayesian data analysis: entropic prior distributions. *Phys. Rev. E* **70**(4), 046127 (2004)
8. Chowdhary, K., Najm, H.: Data free inference with processed data products. *Stat. Comput.* 1–21 (2014). doi:10.1007/s11222-014-9484-y
9. Clyde, M.: Bayesian model averaging and model search strategies (with discussion). In: Bernardo, J., Berger, J., Dawid, A., Smith, A. (eds.) *Bayesian Statistics 6*, pp. 157–185. Oxford University Press, New York (1999)
10. Dupuis, P., Ellis, R.: *A Weak Convergence Approach to the Theory of Large Deviations*. Wiley-Interscience, New York (1997)
11. Gelman, A., Meng, X.L., Stern, H.: Posterior Predictive Assessment of Model Fitness via Realized Discrepancies. *Statistica Sinica* **6**, 733–807 (1996)
12. Genest, C.: A characterization theorem for externally Bayesian groups. *Ann. Stat.* **12**(3), 1100–1105 (1984)
13. Genest, C., Zidek, J.: Combining probability distributions: a critique and an annotated bibliography. *Stat. Sci.* **1**(1), 114–135 (1986)
14. Gregory, P.: *Bayesian Logical Data Analysis for the Physical Sciences*. Cambridge University Press, Cambridge (2010)
15. Hansen, P.C., Pereyra, V., Scherer, G.: *Least Squares Data Fitting with Applications*. The Johns Hopkins University Press, Baltimore (2013)
16. Hebrard, E., Dobrijevic, M.: How measurements of rate coefficients at low temperature increase the predictivity of photochemical models of Titan’s atmosphere. *J. Phys. Chem.* **113**, 11227–11237 (2009)
17. IJUQ: Reprinted from Najm, H.N., Berry, R.D., Safta, C., Sargsyan, K., Debusschere, B.J.: Data-free inference of uncertain parameters in chemical models. *Int. J. Uncertain. Quantif.* **4**, 111–132 (2014); Copyright (2014); with permission from Begell House, Inc
18. Jaynes, E., Bretthorst, G.L. (eds.): *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge (2003)
19. Khalil, M., Najm, H.: Probabilistic inference of reaction rate parameters based on summary statistics. In: Proceedings of the 9th U.S. National Combustion Meeting, Cincinnati (2015)
20. Lakowicz, J.R.: *Principles of Fluorescence Spectroscopy*, 2nd edn. Kluwer Academic, New York (1999). Support Plane Analysis: see pp. 122–123
21. Lehmann, E., Casella, G.: *Theory of Point Estimation*. Springer Texts in Statistics. Springer, New York (2003). <https://books.google.com/books?id=9St7DCbu9AUC>

22. Lynch, S., Western, B.: Bayesian posterior predictive checks for complex models. *Sociol. Methods Res.* **32**(3), 301–335 (2004). doi:10.1177/0049124103257303
23. Nagy, T., Turányi, T.: Determination of the uncertainty domain of the arrhenius parameters needed for the investigation of combustion kinetic models. *Reliab. Eng. Syst. Saf.* **107**, 29–34 (2012)
24. Najm, H., Berry, R., Safta, C., Sargsyan, K., Debusschere, B.: Data free inference of uncertain parameters in chemical models. *Int. J. Uncertain. Quantif.* **4**(2), 111–132 (2014). doi:10.1615/Int.J.UncertaintyQuantification.2013005679
25. Park, T., Casella, G.: The Bayesian Lasso. *J. Am. Stat. Assoc.* **103**(482), 681–686 (2008)
26. Shannon, C.E.: A mathematical theory of communication. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* **5**(1), 3–55 (2001)
27. Sisson, S.A., Fan, Y.: Likelihood-free Markov chain Monte Carlo. In: Brooks, S. (ed.) *Handbook of Markov Chain Monte Carlo*. Chapman & Hall, London (2010)
28. Sivia, D.S., Carlile, C.J.: Molecular-spectroscopy and Bayesian spectral-analysis – how many lines are there. *J. Chem. Phys.* **96**(1), 170 – 178 (1992)
29. Smith, G., Golden, D., Frenklach, M., Moriarty, N., Eiteneer, B., Goldenberg, M., Bowman, C., Hanson, R., Song, S., Gardiner, W., Jr., Lissianski, V., Zhiwei, Q.: GRI mechanism for methane/air, version 3.0 (1999), 30 July 1999. www.me.berkeley.edu/gri_mech
30. STCO: Reprinted from the Chowdhary, K., Najm, H.N.: Data free inference with processed data products. *J. Stat. Comput.* 1–21 (2014); Copyright (2014); with permission from Springer, U.S.
31. Tarantola, A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia (2005)
32. Tibshirani, R.: Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **58**, 267–288 (1996)

Multi-response Approach to Improving Identifiability in Model Calibration

4

Zhen Jiang, Paul D. Arendt, Daniel W. Apley, and Wei Chen

Abstract

In physics-based engineering modeling, two primary sources of model uncertainty that account for the differences between computer models and physical experiments are parameter uncertainty and model discrepancy. One of the main challenges in model updating results from the difficulty in distinguishing between the effects of calibration parameters versus model discrepancy. In this chapter, this identifiability problem is illustrated with several examples that explain the mechanisms behind it and that attempt to shed light on when a system may or may not be identifiable. For situations in which identifiability cannot be achieved using only a single response, an approach is developed to improve identifiability by using multiple responses that share a mutual dependence on the calibration parameters. Furthermore, prior to conducting physical experiments but after conducting computer simulations, in order to address the issue of how to select the most appropriate set of responses to measure experimentally to best enhance identifiability, a preposterior analysis approach is presented to predict the degree of identifiability that will result from using different sets of responses to measure experimentally. To handle the computational challenges of the preposterior analysis, we also present a surrogate preposterior analysis based on the Fisher information of the calibration parameters.

Z. Jiang • W. Chen (✉)

Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA
e-mail: zhenjiang2015@u.northwestern.edu; weichen@northwestern.edu

P.D. Arendt

CNA Financial Corporation, Chicago, IL, USA
e-mail: p.darendt@gmail.com

D.W. Apley

Department of Industrial Engineering and Management Sciences, Northwestern University,
Evanston, IL, USA
e-mail: apley@northwestern.edu

Keywords

Parameter uncertainty • Model discrepancy • Experimental uncertainty • Calibration • Bias correction • (Non)identifiability • Identifiability • Model uncertainty quantification • Calibration parameters • Discrepancy function • Gaussian process • Modular Bayesian approach • Hyperparameters • Simply supported beam • Non-informative prior • Multi-response Gaussian process • Multi-response modular Bayesian approach • Spatial correlation • Non-spatial covariance • Preposterior covariance • Preposterior analysis • Fixed- θ preposterior analysis • Surrogate preposterior analysis • Observed Fisher information

Contents

1	Introduction	70
2	Identifiability Challenge in Model Calibration	73
2.1	Overview of Lack of Identifiability	73
2.2	Case Study: An Illustrative Simply Supported Beam Example	76
2.3	When Is Calibration Identifiability Possible?	81
3	Improving Identifiability in Model Calibration Using Multiple Responses	85
3.1	Multi-response Modular Bayesian Approach	86
3.2	Applying the Multi-response Approach to the Simply Supported Beam	91
3.3	Remarks on the Multi-response Modular Bayesian Approach	95
4	Preposterior Analyses for Predicting Identifiability in Model Calibration	97
4.1	Preposterior Analysis	98
4.2	A Modified Algorithm for Investigating the Behavior of the Preposterior Analysis	104
4.3	Fisher Information-Based Surrogate Preposterior Analysis	105
4.4	Single-Response Case Study: Simply Supported Beam Example	106
4.5	Using the Preposterior Analysis to Select N_e	111
4.6	Multi-response Case Study: Simply Supported Beam Example	112
4.7	Remarks on the Preposterior Analyses Method	117
5	Conclusions	120
	Appendix A: Estimates of the Hyperparameters for the Computer Model MRGP	121
	Appendix B: Posterior Distributions of the Computer Responses	122
	Appendix C: Estimates of the Hyperparameters for the Discrepancy Functions MRGP	123
	Appendix D: Posterior Distribution of the Calibration Parameters	124
	Appendix E: Posterior Distribution of the Experimental Responses	124
	References	125

1 Introduction

Computer simulations have been widely used for design and optimization in many fields of science and engineering, as they are often much less expensive than physical experiments in analyzing complex systems. However, computer simulations never agree completely with experiments, because no computer model is perfect. Several sources of uncertainty accounting for the differences between computer simulations and physical experiments have been reported in the literature [1, 2]. *Parameter uncertainty* and *model discrepancy* are typically the two main

sources; the former is due to the lack of knowledge of physical parameters (e.g., friction coefficient in a finite element analysis) that are naturally fixed but unknown and cannot be directly observed in physical experiments, while the latter is associated with the lack of knowledge of the underlying true physics and is represented by a discrepancy function. Other sources of uncertainty may include *numerical uncertainty* due to numerical errors in implementing computer models, *experimental uncertainty* due to random observational error when taking experimental measurements, and uncertainty due to randomly varying physical parameters. To analyze the differences between computer simulations and physical experiments and to adjust simulation models to better reflect the reality, several model uncertainty quantification methodologies [1–5] have been developed for learning these uncertainties via combining simulation data with physical experimental data. Adjusting predictive models based on identifying the unknown physical parameters and the model discrepancy are referred to as *calibration* [1] and *bias correction* [6–8], respectively.

When there is no discrepancy function, calibration is typically a straightforward, statistically identifiable problem [9–11]. However, it is well known [12–15] that, when a discrepancy function is present, the estimation problem is often poorly identifiable, in the sense that it is difficult to distinguish the effects of calibration parameter uncertainty from the discrepancy function or to estimate these quantities individually. Much of the prior calibration work that has considered a discrepancy function, including work on experimental design for calibrating computer models [1, 2, 4, 16, 17], focused on the more easily attainable objective of good response prediction with the calibrated computer model, even if one is unable to accurately identify the calibration parameters and distinguish their effects from the discrepancy function.

Higdon et al. [2], Loeppky et al. [12], and Han et al. [13] distinguished between two different types of parameters – tuning parameters and calibration parameters – that can serve as input variables in a computer model. Tuning parameters have no meaning in the physical experiment, whereas calibration parameters do have concrete physical meaning but are unknown in reality. Examples of tuning parameters [18] are mesh density in a finite element simulation or some constant in an empirically postulated material flow law (e.g., [19]). Han et al. [13] and the references therein provide a number of examples of calibration parameters that have concrete physical meaning, such as the friction between bone and prosthesis in a prosthetic knee simulation, and they discuss the importance of identifying the true values of the calibration parameters. In this chapter, the situation is considered in which there are unknown calibration parameters that have physical meaning but are unmeasurable, and it is desired to estimate their true values and distinguish their effects from the effects of the discrepancy function.

Recent studies [12, 14, 15] indicate that calibration is usually difficult and that existing methodologies often fail at distinguishing between the effects of parameter uncertainty and model discrepancy and even between the effects of different model parameters. This issue is referred to as (non)*identifiability*. Loosely speaking, identifiability problems occur when different but equally likely combinations of the

calibration parameters and the discrepancy function result in equally good agreement with the observed data. Most of the existing model uncertainty quantification techniques [1, 2, 4, 20] treat the process as a black-box and have the objective of improving the experimental response prediction but with little regard to identifying the true model calibration parameters and model discrepancy. Loepky et al. [12] explored the identifiability issue and concluded that accurate prediction of the experimental response is often possible, even if the individual calibration parameters and discrepancy function are not identified precisely. Furthermore, they concluded that the true values of the calibration parameters are only attainable when the model discrepancy between simulation and reality is neither present nor included in the model updating formulation. Higdon et al. [2] found that in some simple cases the calibration parameters and discrepancy function can be accurately identified, but in other cases their effects cannot be distinguished even though the response predictions may be reasonable.

A broader view should be taken that good identifiability is often critically important for many reasons: (1) Learning the calibration parameters may in itself be a primary goal with broad-reaching implications for product/process improvement (e.g., if these calibration parameters are needed for a system-level simulation or if the calibration parameters themselves reflect the performance of interest but cannot be observed directly). (2) Knowledge of the model discrepancy improves the understanding of the deficiencies of the computer model for improving future generations. (3) It results in more accurate prediction over a broader set of input regions, because the model adjustment from learning the calibration parameters is more global than the adjustment from learning the model discrepancy. In the context of using predictive modeling for engineering design, adjusting the calibration parameters changes the prediction of the experimental response for a wide range of values for the design variables, whereas adjusting the discrepancy function tends to change the prediction of the experimental response predominantly within some neighborhood of the specific design variable settings that were used in the experiments.

In many engineering applications, good identifiability is virtually impossible if considering a single experimental response. In this chapter, we aim to provide a better understanding of the issue of identifiability and offer feasible treatments to predict and to enhance identifiability via a multi-response approach. The remainder of the chapter is organized as follows: Sect. 2 investigates the issue of identifiability using a simply supported beam example. Whereas the example and discussion in Sects. 2.1 and 2.2 convey the conclusion that identifiability is often very difficult, or impossible, in typical implementations, Sect. 2.3 discusses situations in which it is reasonable to expect that good identifiability can be achieved. Section 3 further investigates the identifiability issue and demonstrates that incorporating multiple experimental responses that share a mutual dependence on a common set of calibration parameters can substantially improve identifiability. Section 4 provides a detailed description of a preposterior analysis, in conjunction with the multi-response uncertainty quantification approach, for predicting identifiability prior to conducting physical experiments. Furthermore, to handle computational challenges

in the preposterior analysis, a surrogate preposterior analysis is presented based on the Fisher information of the calibration parameters. The methods are applied to the same simply supported beam example to enhance identifiability. The results of the preposterior and surrogate preposterior analyses are compared to the results of the posterior analysis (after observing the experimental data) to demonstrate the effectiveness of the methods. Conclusions are made in Sect. 5.

2 Identifiability Challenge in Model Calibration

2.1 Overview of Lack of Identifiability

Following the seminal work of Kennedy and O'Hagan [1], the model uncertainty quantification formulation that incorporates the various uncertainties is [1, 2, 4, 21]:

$$y^e(\mathbf{x}) = y^m(\mathbf{x}, \boldsymbol{\theta}^*) + \delta(\mathbf{x}) + \varepsilon, \quad (4.1)$$

where the controllable inputs (aka design variables) are denoted by the vector $\mathbf{x} = [x_1, \dots, x_d]^T$ and the calibration parameters, which are unknown model parameters, are denoted by the vector $\boldsymbol{\theta} = [\theta_1, \dots, \theta_r]^T$. A calibration parameter is defined as any physical parameter that can be specified as an input to the computer model and that is unknown or not measurable when conducting the physical experiments [1]. $\boldsymbol{\theta}^*$ (an $r \times 1$ vector of constants) denotes the true values of the unknown calibration parameters over the course of the physical experiments. $y^e(\mathbf{x})$ is the experimental response, $y^m(\mathbf{x}, \boldsymbol{\theta})$ is the computer model response as a function of \mathbf{x} and $\boldsymbol{\theta}$, $\delta(\mathbf{x})$ is the additive discrepancy function that represents the difference between the model response (using the true $\boldsymbol{\theta}^*$) and the experimental response, and ε accounts for the experimental uncertainty. Experimental uncertainty is typically assumed to follow a normal distribution with mean 0 and variance λ , denoted $\varepsilon \sim \mathcal{N}(0, \lambda)$.

Parameter uncertainty has two forms. A parameter $\boldsymbol{\theta}$ may be constant but unknown and its uncertainty represented via a probability distribution. Alternatively, a parameter \mathbf{x} may vary randomly (from run to run over the physical experiments and/or over any future instances for which the model is used to predict reality) according to some probability distribution. An example of the latter could be the blank thicknesses in a sheet metal stamping operation that vary randomly from blank to blank due to manufacturing variation. Calibration generally refers to the former form of parameter uncertainty, with the goal of identifying the true values of the constant parameters. "Constant" parameters refer to those that do not change over the duration of the physical experiments. Accordingly, in this chapter, only this form of parameter uncertainty is considered. Calibration in the case of parameters that vary randomly throughout the physical experiment would be far more challenging. One might assume some distribution for the randomly varying parameters and define the "calibration" goal as to identify the statistical parameters of the distribution (e.g., the mean and variance of a normal distribution) [20]. However, this would require

far more experimental observations than are needed for the goal in this chapter of identifying constant physical parameters.

The formulation in Eq. (4.1) is comprehensive in that it accounts for parameter uncertainty, model discrepancy, experimental uncertainty, and interpolation uncertainty. To quantify uncertainty in the calibration parameters and discrepancy function, the modular Bayesian approach presented in [1] is used to calculate their posterior distributions. Notice that the discrepancy function in Eq. (4.1) is not directly observable from the collected data, since the true value of θ is unknown. Notation-wise two Gaussian process (GP) models are used to infer the computer model response and the discrepancy function such that:

$$y^m(\cdot, \cdot) \sim \mathcal{GP}(\mathbf{h}^m(\cdot, \cdot)\boldsymbol{\beta}^m, \sigma_m^2 R^m((\cdot, \cdot), (\cdot, \cdot))), \quad (4.2)$$

$$\delta(\cdot) \sim \mathcal{GP}(\mathbf{h}^\delta(\cdot)\boldsymbol{\beta}^\delta, \sigma_\delta^2 R^\delta(\cdot, \cdot)). \quad (4.3)$$

The prior mean function of the computer model response is comprised of the unknown regression coefficients $\boldsymbol{\beta}^m$ and the known regression functions $\mathbf{h}^m(\mathbf{x}, \theta)$. The prior covariance function is the product of an unknown constant σ_m^2 and a correlation function $R^m((\mathbf{x}, \theta), (\mathbf{x}', \theta'))$, where (\mathbf{x}, θ) and (\mathbf{x}', θ') denote two sets of computer model inputs. The notations for the discrepancy function are similarly defined. $R^m((\mathbf{x}, \theta), (\mathbf{x}', \theta'))$ and $R^\delta(\mathbf{x}, \mathbf{x}')$ are parameterized by a $(d+r) \times 1$ vector $\boldsymbol{\omega}^m$ and a $d \times 1$ vector $\boldsymbol{\omega}^\delta$ such that:

$$R^m((\mathbf{x}, \theta), (\mathbf{x}', \theta')) = \exp \left\{ - \sum_{k=1}^d \omega_k^m (x_k - x'_k)^2 - \sum_{k=d+1}^{d+r} \omega_k^m (\theta_k - \theta'_k)^2 \right\}, \quad (4.4)$$

$$R^\delta(\mathbf{x}, \mathbf{x}') = \exp \left\{ - \sum_{k=1}^d \omega_k^\delta (x_k - x'_k)^2 \right\}. \quad (4.5)$$

The modular Bayesian approach incorporates the prior knowledge of calibration parameters θ , estimates the *hyperparameters* of the GP models $\phi = [\boldsymbol{\beta}^m, \sigma_m^2, \boldsymbol{\omega}^m, \boldsymbol{\beta}^\delta, \sigma_\delta^2, \boldsymbol{\omega}^\delta, \lambda]$, and evaluates the posterior distribution of θ . Because Eq. (4.1) accounts for several different forms of uncertainty and also includes \mathbf{x} , it is viewed as providing a comprehensive and widely applicable model updating formulation for design under uncertainty. However, one limitation is that for some systems it is very difficult to distinguish between the effects of parameter uncertainty and model discrepancy. More specifically, the same predicted experimental response can result from many different but equally likely combinations of the calibration parameters and the discrepancy function. This constitutes a lack of identifiability [12, 22] of the calibration parameters and discrepancy function, even though the experimental response still may be accurately predicted.

Figure 4.1 illustrates this with an example having a scalar input $x(d = 1)$ and a scalar calibration parameter $\theta(r = 1)$. $\hat{\theta}^{(1)}$, $\hat{\theta}^{(2)}$, and $\hat{\theta}^{(3)}$ denote three possible estimates of the calibration parameter θ ; $y^m(x, \hat{\theta}^{(1)})$, $y^m(x, \hat{\theta}^{(2)})$, and $y^m(x, \hat{\theta}^{(3)})$ are the computer simulation models corresponding to the three different values of θ .

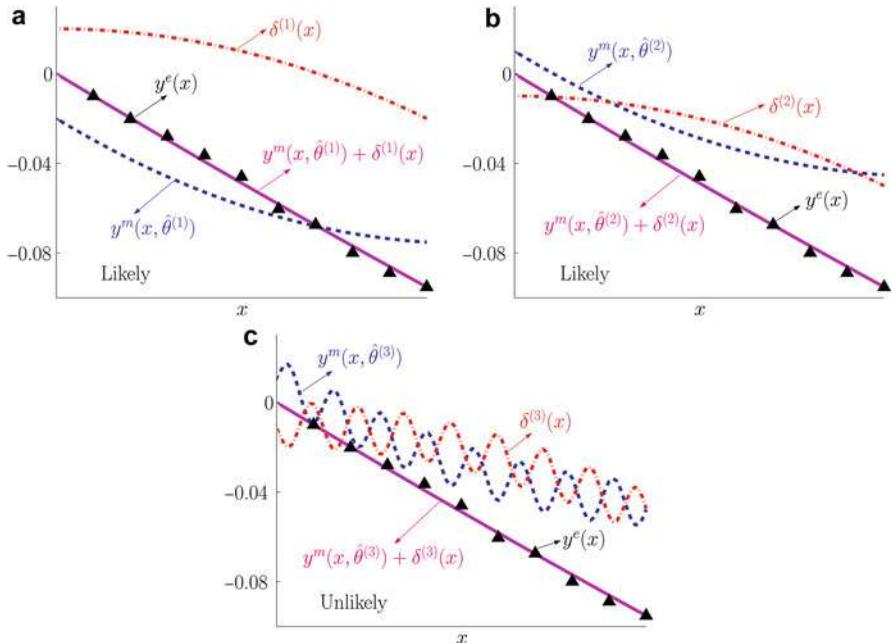
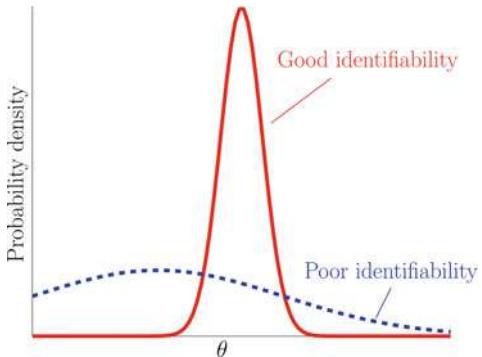


Fig. 4.1 An illustration of identifiability. While it is easy to tell that (c) is the least likely estimate of θ , it may be impossible to identify which of (a) and (b) is better

For each simulation realization $y^m(x, \hat{\theta}^{(i)})$ ($i = 1, \dots, 3$), an estimated discrepancy function $\delta^{(i)}(x)$ ($i = 1, \dots, 3$) can be found so that the resulting experimental response predictions $y^m(x, \hat{\theta}^{(1)}) + \delta^{(1)}(x)$, $y^m(x, \hat{\theta}^{(2)}) + \delta^{(2)}(x)$, and $y^m(x, \hat{\theta}^{(3)}) + \delta^{(3)}(x)$ are quite similar and in equally good agreement with the experimental data within the experimental region. Intuitively from the smoothness of the observed experimental data, the combination of $\hat{\theta}^{(3)}$ and $\delta^{(3)}(x)$ seems less likely than the other two combinations to be the true parameter value and true discrepancy function as the simulation is highly nonlinear. Rigorous calculation can show that its posterior probability density $p(\hat{\theta}^{(3)}|y^m, y^e)$ has a smaller value than $p(\hat{\theta}^{(1)}|y^m, y^e)$ and $p(\hat{\theta}^{(2)}|y^m, y^e)$. However, it may be virtually and computationally impossible to identify which of $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$ is a better estimate of θ , since in both cases (Fig. 4.1a, b) the computer simulations are consistent with the experiments to a similar degree, and the values of $p(\hat{\theta}^{(1)}|y^m, y^e)$ and $p(\hat{\theta}^{(2)}|y^m, y^e)$ are close.

Figure 4.2 depicts how the posterior distribution of θ can be used to assess the level of identifiability for a hypothetical single-parameter example. A tight posterior distribution of θ with a clear mode indicates good identifiability. In sharp contrast, with a widely dispersed posterior distribution, the identifiability is poor. As in much of the prior work that assessed calibration identifiability (e.g., [4, 12, 13]), identifiability can be quantified via the posterior covariance matrix of the set of calibration parameters (posterior to observing the physical experimental

Fig. 4.2 Posterior distribution of calibration parameter as a demonstration of identifiability for the single-parameter case



data, in addition to the simulation data). One might also consider using the posterior covariance function of the discrepancy function as a measure of identifiability; however, this is infinite dimensional and cumbersome. Moreover, the posterior covariance of the discrepancy function is closely related to the posterior covariance of the calibration parameters, in the sense that precisely estimated calibration parameters generally imply a precisely estimated discrepancy function (which is somewhat obvious from Eq. (4.1)). Consequently, the posterior covariance matrix of the calibration parameters is more manageable to work with. Here, the term “identifiability” is used rather informally to refer to whether one can expect to achieve reasonable and useful estimation of the calibration parameters with typical finite sample sizes (which is sometimes possible), as opposed to whether estimators are consistent and guaranteed to asymptotically converge to the true values when sample sizes approach infinity in some sense (which is perhaps never possible under realistic assumptions).

2.2 Case Study: An Illustrative Simply Supported Beam Example

To further illustrate this identifiability problem, an example using a simply supported beam [14] is presented in Fig. 4.3. The beam has length of 2 m and a rectangular cross-section with a height of 52.5 mm and width of 20 mm. One end of the beam is fixed (no vertical or horizontal displacement permitted), while the other end of the beam is supported by a roller (no vertical displacement permitted). A static force is applied to the midpoint of the beam to induce various responses (e.g., stress and displacement). In this example, the magnitude of the applied force is chosen as the design variable x . Refer to Table 4.1 for the notation of this example.

The computer model response $y^m(x, \theta)$ (angle of deflection at the end of the beam in radians) is obtained from a finite element analysis (FEA) model representing a simply supported beam (using Abaqus 6.9) with a simplified (perfectly plastic) material law. For illustrative purposes, the “experimental response” $y^e(x)$ (angle of deflection at the end of the beam in radians) is taken to be the response from a FEA model using a more elaborate material law (a power law for the plastic region with

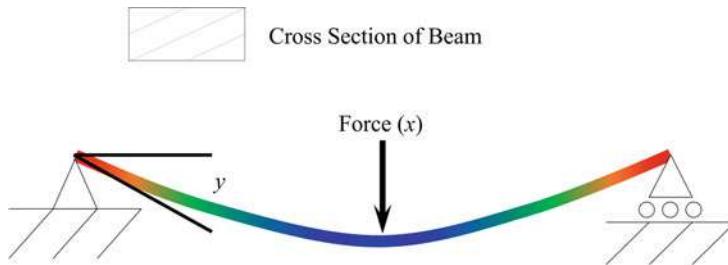


Fig. 4.3 Schematic of the simply supported beam

Table 4.1 Notation for the simply supported beam example

Variable	Description
x	Static force (N) applied to the midpoint of the beam (on the range [1300 2300] N)
θ	Young's modulus (GPa), on which both the simulation model and physical reality depend (on the range [150 300] GPa), $\theta^* = 206.8$ GPa is the true value
y	The angle of deflection (radians) at the end of the beam
$y^m(x, \theta)$	y calculated via a finite element analysis (FEA) with a simplified material model
$y^e(x)$	"Experimentally measured" y (for illustrative purposes, calculated from a FEA model with a more elaborate material model assuming $\theta = \theta^*$)
$\delta(x)$	Discrepancy function
ε	Random error, assumed 0 in this example

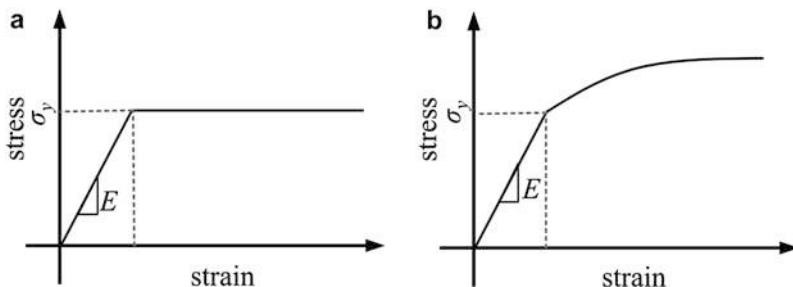


Fig. 4.4 Material stress-strain curves for (a) the computer model and (b) the "physical experiments." E is Young's modulus and the calibration parameter. σ_y is the yield stress (225 MPa)

constant stress $C = 2068$ MPa and strain-hardening exponent $n = 0.5$) [23]. These material laws are in part governed by Young's modulus, E , which is treated as the unknown calibration parameter θ (Fig. 4.4). For the "physical experiments," the true value of the calibration parameter is set to $\theta^* = 206.8$ GPa but is treated as unknown during the analysis and assigned a uniform prior over the range $150 \leq \theta \leq 300$ GPa. The prior distribution was chosen to be relatively wide in order to minimize the effect of the prior on the posterior distributions and to avoid choosing a prior that does not contain any support at the true value θ^* (in which case the posterior will

not contain any support at θ^* either). In practice, it is recommended choosing a conservatively wide prior in order to ensure that there is some support at the true value of the parameters.

The goal is to use the observed simulation and experimental data to infer the calibration parameter and the discrepancy function of the model updating formulation of Eq. (4.1). It should be noted that this example is intended only to illustrate the modular Bayesian approach and the identifiability challenges. There are many standard experimental methods to determine Young's modulus using specimens of the material.

To infer the calibration parameter and the discrepancy function, the modular Bayesian approach discussed in [1] is applied to this example. We begin by fitting a GP model of the computer model using a 4 by 4 grid over the input space ($1300 \leq x \leq 2300$ N and $150 \leq \theta \leq 300$ GPa). The experimental data \mathbf{y}^e is observed at $\mathbf{X}^e = [1300, 1400, 1500, \dots, 2300]^T$ and indicated in Fig. 4.5a by the black dots. The experimental data, together with the uniform prior for θ and the hyperparameter MLEs, are used to estimate the hyperparameters of the GP model for the discrepancy function. Finally, the posterior distributions for the discrepancy function and the calibration parameter are calculated via Legendre-Gauss quadrature. The posterior distributions are shown in Fig. 4.5. From Fig. 4.5a, the posterior distribution of the experimental response is very accurate and precise, because there was a relatively

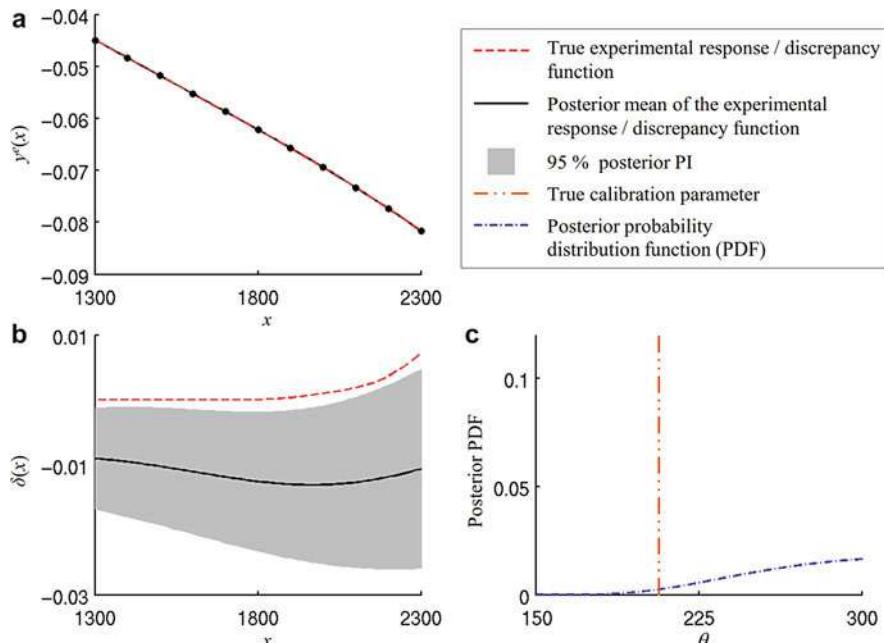


Fig. 4.5 The posterior distributions for (a) the experimental response, (b) the discrepancy function, and (c) the calibration parameter showing a lack of identifiability

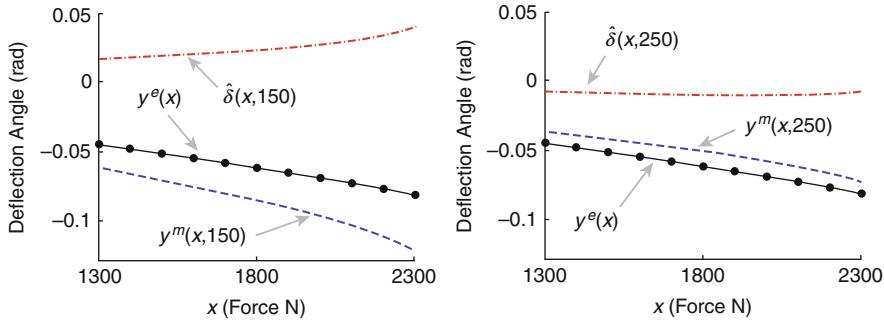


Fig. 4.6 The computer response $y^m(x, \theta)$, the experimental response $y^e(x)$ (bullets indicate experimental data), and the estimated discrepancy function $\hat{\delta}(x, \theta) = y^e(x) - y^m(x, \theta)$ for (a) $\theta = 150$ GPa and (b) $\theta = 250$ GPa

large amount of experimental data and no experimental error ε . In spite of this, the calibration parameter and the discrepancy function are not identifiable as illustrated below.

To assess identifiability, the posterior distributions of the calibration parameter (Fig. 4.5c) and the discrepancy function (Fig. 4.5b) are used. The large posterior variance of the calibration parameter and the large width of the prediction intervals for the discrepancy function indicate the poor identifiability of the calibration parameter and discrepancy function. Indeed, several different combinations of these are approximately equally probable and give approximately the same (quite accurate and precise) predicted response in Fig. 4.6a. It is also worth noting neither posterior reflects the true values of the calibration parameter and the discrepancy function. Because of the high uncertainty reflected by the posterior distributions, the system is not identifiable, even with the relatively dense spacing of the experimental data. Moreover, collecting additional experimental observations would help little.

To understand how the large uncertainty and the inaccuracy of the posterior distributions lead to a lack of identifiability, consider the estimated discrepancy function $\hat{\delta}(x, \theta) = y^e(x) - y^m(x, \theta)$ shown in Fig. 4.6 for two different values of θ . For θ underestimated at 150 GPa in Fig. 4.6a (the true $\theta^* = 206.8$ GPa) and for θ overestimated at 250 GPa in Fig. 4.6b, neither θ nor the resulting estimated discrepancy function are inconsistent with their priors or in disagreement with the data. They both result in very similar, and quite accurate, prediction of the experimental response. Consequently, without additional information, it is virtually impossible to distinguish between the effects of θ and $\delta(x)$ and accurately identify each in this example.

To improve identifiability, previous literature [4, 24] recommended using informative (i.e., low variance or tightly dispersed) prior distributions for the calibration parameters, the discrepancy function, or both. However, specifying informative priors for the discrepancy function (e.g., a linear or quadratic functional form [20, 25]) is often difficult because one rarely has significant prior knowledge.

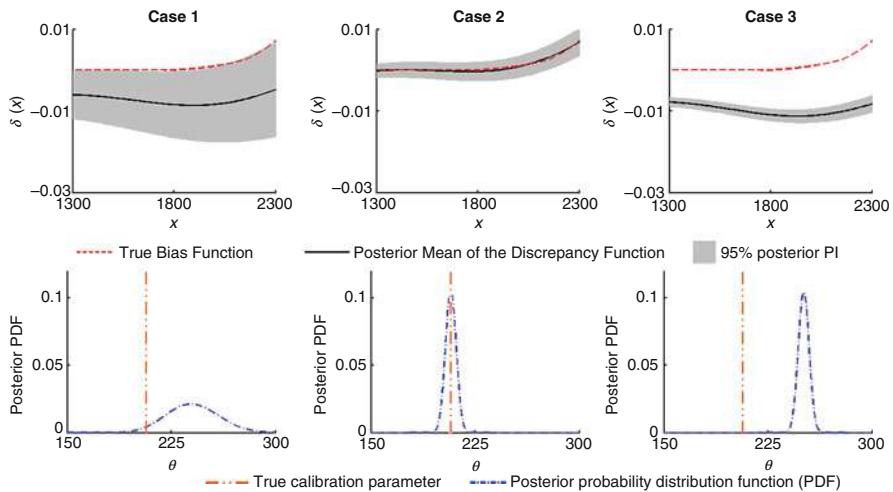


Fig. 4.7 Posterior distributions for the discrepancy function and the calibration parameter for the three prior distributions in Table 4.2

Table 4.2 Prior distribution (normal) and posterior distribution means and standard deviations for the calibration parameter

	Case 1	Case 2	Case 3
Prior mean	225.00	206.80	250.00
Prior std. dev.	22.36	3.87	3.87
Post mean	240.63	207.43	250.68
Post std. dev.	18.68	3.84	3.83

Likewise, one does not have such prior information on the calibration parameters (otherwise, they would be viewed as known parameters and not considered calibration parameters). However, to illustrate how this would improve identifiability, three versions of the preceding beam example are considered using three different normal prior distributions for θ , each with different means and/or variances. The resulting posterior distributions for θ and the discrepancy function are shown in Fig. 4.7 and Table 4.2.

Case 1 assumes a less informative (larger standard deviation) prior distribution centered about the midpoint of the calibration parameter range (mean of 225 GPa). In this case, the posterior distribution for the calibration parameter and the discrepancy function were neither accurate nor precise, which indicates a lack of identifiability. In Case 2, an informative prior (small standard deviation) with a mean equal to the true θ results in an identifiable system, which is evident by the accurate and precise posterior distributions. In contrast, Case 3 shows the inherent danger of using an informative prior that is inaccurate albeit precise. In this case, the posterior for the discrepancy function has a small prediction interval and the posterior of the calibration parameter has a small variance, which would lead one to believe that they are precisely estimated. However, the results of posteriors are inaccurate and do not reflect the true discrepancy function and θ .

In conclusion, since one rarely has such accurate and informative prior knowledge about the sources of uncertainty present in the engineering system, assuming informative priors for the calibration parameters or the discrepancy function is not a satisfying solution to the identifiability problem.

2.3 When Is Calibration Identifiability Possible?

The simply supported beam represents a system for which we cannot accurately and precisely identify both the calibration parameters and the discrepancy function (e.g., via their posterior distributions). This leads one to question whether identifiability is ever possible, given the nature of the model in Eq. (4.1). This section demonstrates that the answer is yes, but under certain assumptions. This is illustrated with a simple example in which the requisite assumptions are that the discrepancy function is reasonably smooth and can be represented by a smooth GP model (i.e., a GP with small roughness parameters, ω). Loosely speaking, “smooth” means that the function does not change abruptly for small changes in the design variables. In the context of our GP-based modeling, smooth means that the function is consistent with a GP model having relatively small roughness parameters (ω), as discussed later at the end of this section.

As evident from the posterior distributions of Fig. 4.5, the simply supported beam is not identifiable. The conceptual explanation (see the discussion surrounding Fig. 4.6) is that, for many different values of θ , the estimated discrepancy function is smooth and consistent with a GP model. In contrast, an example is considered next in which the estimated discrepancy function has behavior that is inconsistent with a GP model when $\theta \neq \theta^*$, but consistent when $\theta = \theta^*$. It will be shown that this is the essential ingredient for good identifiability [14]. Consider the following step function as the computer model

$$y^m(x, \theta) = \begin{cases} 1 - \theta & x < 2 \\ 1 + \theta & x \geq 2 \end{cases}, \quad (4.6)$$

with $1 \leq x \leq 3$ and $0.5 \leq \theta \leq 3$. Notice that the magnitude of the step is dictated by the calibration parameter θ . Plots of $y^m(x, \theta)$ for $\theta = 1.0$ and 3.0 , respectively, are shown in Fig. 4.8a, c. The physical experimental responses were generated according to Eq. (4.1) with the true calibration parameter $\theta^* = 2.5$, experimental error $\varepsilon = 0$, and the true discrepancy function $\delta(x)$ generated as one random realization of a smooth GP with a prior mean of 0 and a Gaussian covariance function with $\omega^\delta = \sigma_\delta = 1$.

To quantitatively assess identifiability, again, the modular Bayesian approach is used. A GP model for the computer model (the step function of Eq. (4.6)) was created using a total of 168 simulation runs, as shown in Fig. 4.9. 144 runs were observed on a 12 by 12 grid. Additionally, to accurately capture the step behavior of the computer model, 24 additional runs were conducted at $x = 1.99$ and 2.01 , for 12 evenly spaced values of θ . The GP model of the computer model

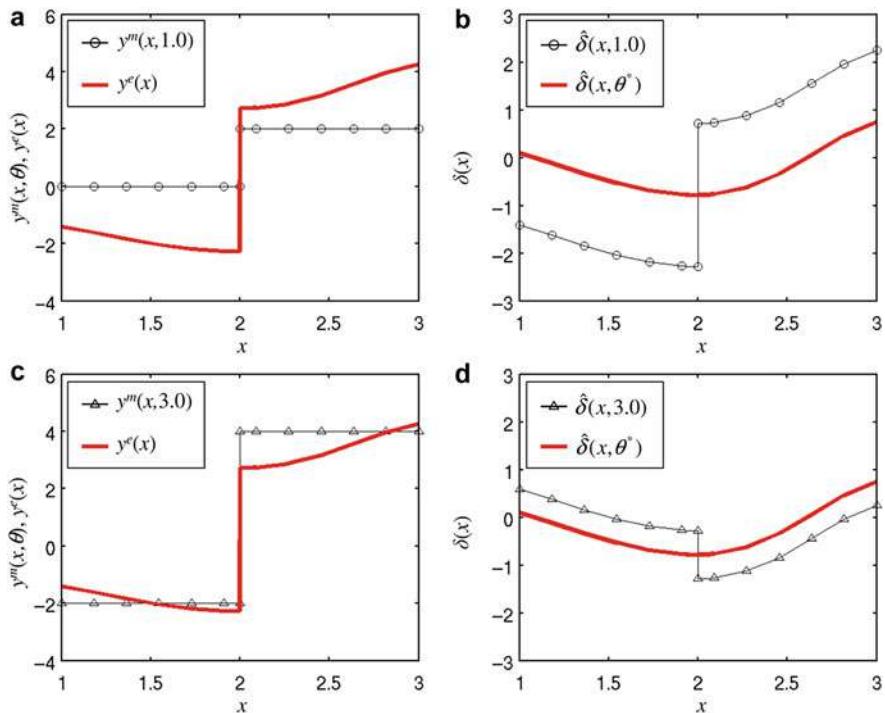
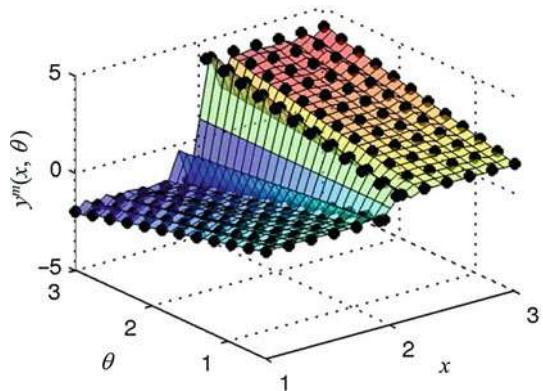


Fig. 4.8 Computer model and experimental response for Eqs. (4.6) and (??) with (a) $\theta = 1.0$ and (c) $\theta = 3.0$. Corresponding estimated discrepancy function $\hat{\delta}(x, \theta)$ for (b) $\theta = 1.0$ and (d) $\theta = 3.0$

Fig. 4.9 Fitted GP model of the computer model of Eq. (4.6) based on 168 simulation runs, y^m (black dots)



was difficult to fit using the 168 simulations, due to the discontinuity at $x = 2$. Typically, a GP model with a Gaussian correlation function is not a good choice to model a step function because of its inherent smoothness (see Ref. [26, p. 123] and subsequent discussion). However, Fig. 4.9 shows that, with the large amount of

data and allowing large roughness parameters (the MLEs of the hyperparameters are $\omega^m = [\omega^x, \omega^\theta]^T$, where $\omega^x = 377.52$ and $\omega^\theta = 50$, and ω^θ fell at the upper boundary of what is allowed in the MLE algorithm for numerical purposes), a fairly accurate GP metamodel results. Although the fitted GP metamodel is somewhat rough and interpolates less accurately near the discontinuity, it serves to illustrate when identifiability is achievable.

After creating the GP model for the computer model, the modular Bayesian approach estimates the hyperparameters for the GP model representing the discrepancy function. 14 experimental data points were collected at $\mathbf{X}^e = [1, 1.167, 1.333, \dots, 3, 1.99, 2.01]^T$ (the black dots in Fig. 4.10a). The same x locations were chosen for the simulation and experimental runs in order to focus on identifiability rather than interpolation performance. Finally, the posterior distributions for the calibration parameters, the experimental response, and the discrepancy function are calculated. 10 shows the resulting posterior distributions.

The accurate and precise (low posterior standard deviation) posterior distributions for the discrepancy function in Fig. 4.10b and the calibration parameter in Fig. 4.10c demonstrates that good identifiability is achievable for this example.

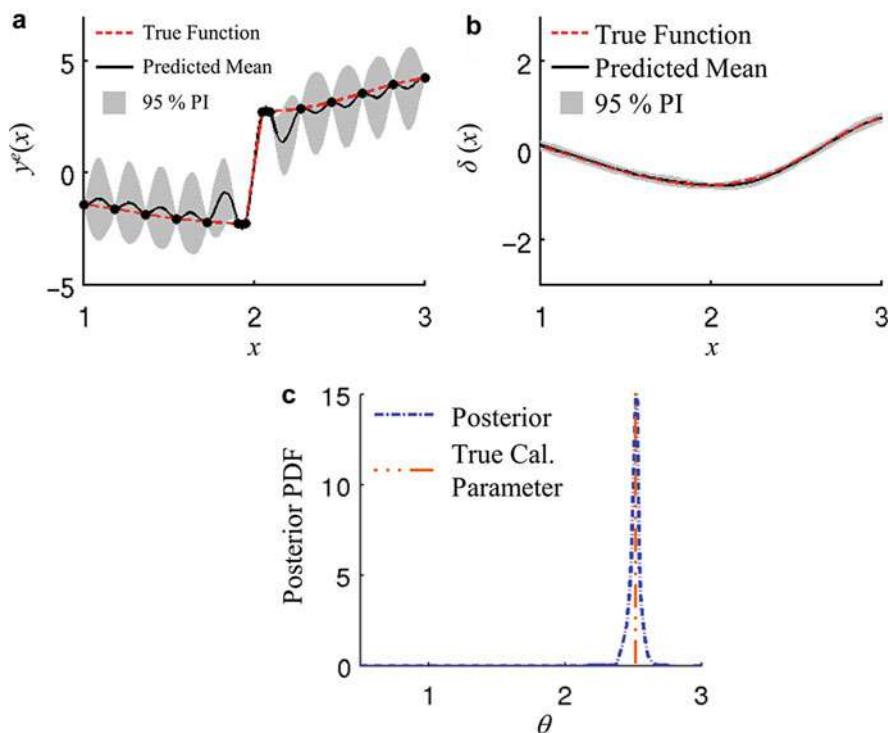


Fig. 4.10 Posterior distributions for (a) the experimental response (black dots indicate experimentally observed response values), (b) the discrepancy function, and (c) the calibration parameter θ

Notice that the posterior mean of the experimental response in Fig. 4.10a is a less accurate and precise predictor near the step discontinuity. In spite of this, reasonable identifiability of the calibration parameter was achieved, which was the objective of this analysis.

To understand why the system is identifiable in this example, consider the estimated discrepancy functions $\hat{\delta}(x, \theta)$ for two incorrect values of θ shown in Fig. 4.8b ($\theta = 1.0$) and Fig. 4.8d ($\theta = 3.0$). For reference, the estimated discrepancy function $\hat{\delta}(x, \theta^*)$ for the true $\theta^* = 2.5$ is shown in both figures. The salient feature is that $\hat{\delta}(x, \theta)$ for the incorrect values of θ have discontinuities that are inconsistent with a smooth GP model for the discrepancy function. The only value of θ that results in a $\hat{\delta}(x, \theta)$ consistent with model is the true value $\theta^* = 2.5$. By a $\hat{\delta}(x, \theta)$ “inconsistent” with the model, it means that, when that value of θ and its corresponding $\hat{\delta}(x, \theta)$ are plugged into the likelihood function, the resulting likelihood is exceptionally low. Because the estimated discrepancy function entails a high likelihood only when θ is close to θ^* , and the likelihood drops off abruptly as θ deviates from θ^* , the second derivative of the likelihood function (with respect to θ) is quite large. This translates to a large observed Fisher information, which results in a tight posterior distribution for θ . This is, in essence, the definition of identifiability. The preceding example illustrates that identifiability is possible when small changes in the calibration parameters about their true values result in an estimated discrepancy function (which reflects the differences between the observed experimental data and the simulation model) that is inconsistent with the GP model of the discrepancy function.

The identifiability issues in the preceding discussion can also be viewed in a slightly different light. In essence, the approach evaluates identifiability by considering the simulation data, considering the experimental data, and then assessing the likelihood that the observed differences between the simulation and experimental data are explained by the discrepancy function, by an adjustment of the calibration parameters, or by various combinations thereof. Hypothetically, if there is roughly equal likelihood that the differences between the simulation and experimental data are explained by two (or more) different combinations of a discrepancy function and an adjustment in the calibration parameters, then identifiability would be poor in this case. On the other hand, if the likelihood is high that one particular combination of discrepancy function and adjustment in the calibration parameters accounts for the differences between the simulation and experimental data, then identifiability would be good in this case.

In either case, the likelihood must be evaluated with respect to the assumed prior distribution for the discrepancy function. And in this sense, identifiability is possible only by making certain assumptions on the prior distribution model for the discrepancy function. Although identifiability is achieved in the step example via certain assumptions on the prior distributions, this is quite different than assuming a tight, informative prior distribution for the calibration parameters. The latter would also generally result in identifiability (i.e., a tight posterior), but it would be an artificial identifiability in that one would have to begin with precise knowledge of

the parameters via a tight prior. In all of the examples of this section, only relatively non-informative prior distributions have been used for the calibration parameters. Note that, in the preceding, the term “likelihood” is used, but it is technically a Bayesian posterior probability, which considers both the likelihood and the prior distributions for all parameters and hyperparameters.

3 Improving Identifiability in Model Calibration Using Multiple Responses

In engineering systems where there is a lack of identifiability with a single response, this section introduces the use of multiple responses as additional information to improve identifiability. Multiple responses are considered to be a collection of single responses of different quantities (e.g., load force, displacement, stress, etc.), each of which can also be measured over time and/or space. When the multiple responses are mutually dependent on the same set of calibration parameters, the information obtained from them can be combined to better infer the true value of the calibration parameters. Note that multiple responses usually can be “measured” for free in computer simulations, as they are automatically calculated in the simulation. Hence, the added cost of observing multiple responses is that associated with their measurement in the physical experiments.

To quantify the uncertainty that results from having to interpolate the responses between discrete input sites at which they are observed in expensive computer simulations or physical experiments, multi-response surrogate models are necessary. One existing body of work uses a multi-response Gaussian process (MRGP) model, which is defined by its mean and covariance functions, as a surrogate model representing only the computer model. Most existing work uses the same mathematical structure for the mean function. However, for the covariance function, two different mathematical structures are common. References [27–30] assign an indexing variable to each response, which is treated as an additional input over which the covariance function is defined. This is only applicable if the multiple responses are inherently ordered in a manner that permits such an indexing.

More recently, Conti et al. [31, 32] defined a covariance function that is the combination of a spatial correlation function and a discrete covariance matrix, the latter representing the covariance between responses. This covariance parameterization is relatively simple yet flexible, and the resulting MRGP model is a straightforward extension of the well-established single-response Gaussian process model [32]. Therefore, this covariance function is adopted here to construct the MRGP model for both the computer model and the discrepancy functions.

Although using multiple responses to quantify computer model uncertainty based on experimental data has been considered in previous literature, the “multiple responses” were restricted to a single response measured at multiple spatial and/or temporal locations [33–36]. Furthermore, the prior work focuses solely on predicting the experimental response and not on identifiability of the calibration parameters

and the discrepancy functions. As demonstrated in Sect. 2.2, different combinations of the calibration parameters and the discrepancy functions can often result in the same experimental response prediction, which is the root of the identifiability problem.

The previous section shed light on the challenging problem of identifying the calibration parameters and the discrepancy function when combining single-response data from a computer model and physical experiments. The main objective of this section is to show that identifiability can be improved by including multiple responses in the model updating formulation. A multi-response modular Bayesian approach is developed in [15] to extend the single-response modular Bayesian approach [1], which is used to calculate the posterior distributions of the calibration parameters and the discrepancy functions. Subsequently, the multi-response modular Bayesian approach is applied to the same simply supported beam example introduced in Sect. 2.2. Whereas identifiability was not achievable using only a single response in Sect. 2.2, it will be shown here that identifiability can be substantially enhanced using certain combinations of multiple responses.

3.1 Multi-response Modular Bayesian Approach

The single-response modular Bayesian approach [1] calculates the posterior distributions, which quantify the posterior uncertainty of the calibration parameters and the discrepancy function, by integrating data from both simulations and experiments. In this section, the modular Bayesian approach is extended to incorporate multiple responses [15]. To do this, the model updating formulation of Kennedy and O'Hagan [1] is reformulated as [15, 34]:

$$y_i^e(\mathbf{x}) = y_i^m(\mathbf{x}, \boldsymbol{\theta}^*) + \delta_i(\mathbf{x}) + \varepsilon_i, \quad (4.7)$$

for $i = 1, \dots, q$, where q denotes the number of response variables. The vector $\mathbf{x} = [x_1, \dots, x_d]^T$ denotes the design variables (aka controllable inputs) and $\boldsymbol{\theta} = [\theta_1, \dots, \theta_r]^T$ denotes a vector of calibration parameters with $\boldsymbol{\theta}^*$ denoting their true physical values, which are unknown to the user. $y_i^m(\mathbf{x}, \boldsymbol{\theta})$ ($i = 1, \dots, q$) denotes the i th response from the computer model as a function of \mathbf{x} and $\boldsymbol{\theta}$, and $y_i^e(\mathbf{x})$ denotes the corresponding experimentally observed response.

Note that all of the responses from the computer model mutually depend on the same set of calibration parameters $\boldsymbol{\theta}$, and the experimental responses depend on the same true calibration parameter value $\boldsymbol{\theta}^*$. $\delta_i(\mathbf{x})$ denotes the i th discrepancy function, which represents the difference between the computer model (with input $\boldsymbol{\theta}^*$) and the physical experiments. Finally, ε_i is independent random experimental error (e.g., observation error), which is assumed normal with mean 0 and variance λ_i .

Similar to the procedure for implementing the single-response modular Bayesian approach as briefly mentioned in Sect. 2.1, the multi-response modular Bayesian approach fits a MRGP model to the computer model and the discrepancy functions

in two separate modules (Modules 1 and 2). After fitting each MRGP model, one calculates the posterior distributions of the calibration parameters, the discrepancy functions, and the experimental responses (Modules 3 and 4). Details of each module in the multi-response modular Bayesian approach are provided in the following.

3.1.1 Multi-response Gaussian Process Model for the Computer Model (Module 1)

In Module 1, a MRGP model is fit to the computer responses and used to infer their values at input sites other than those that were simulated. Following Conti et al. [31, 32], the prior for the MRGP model is:

$$\mathbf{y}^m(\cdot, \cdot) \sim \mathcal{GP}(\mathbf{h}^m(\cdot, \cdot)\mathbf{B}^m, \boldsymbol{\Sigma}^m R^m((\cdot, \cdot), (\cdot, \cdot))), \quad (4.8)$$

where $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta}) = [y_1^m(\mathbf{x}, \boldsymbol{\theta}), \dots, y_q^m(\mathbf{x}, \boldsymbol{\theta})]$ denotes the multiple responses from the computer model. The prior mean function is comprised of an arbitrary vector of specified regression functions $\mathbf{h}^m(\mathbf{x}, \boldsymbol{\theta}) = [h_1^m(\mathbf{x}, \boldsymbol{\theta}), \dots, h_p^m(\mathbf{x}, \boldsymbol{\theta})]$ and a matrix of unknown regression coefficients $\mathbf{B}^m = [\beta_1^m, \dots, \beta_q^m]$, where $\beta_i^m = [\beta_{1,i}^m, \dots, \beta_{p,i}^m]^T$. A frequently used regression function is $\mathbf{h}^m(\mathbf{x}, \boldsymbol{\theta}) = 1$, which corresponds to a constant prior mean. The prior covariance function is the product of an unknown nonspatial $q \times q$ covariance matrix $\boldsymbol{\Sigma}^m$ and a spatial correlation function $R^m((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}'))$, where $(\mathbf{x}, \boldsymbol{\theta})$ and $(\mathbf{x}', \boldsymbol{\theta}')$ denote two sets of computer model inputs. This prior covariance function can be rewritten as $\text{Cov}[y_i^m(\mathbf{x}, \boldsymbol{\theta}), y_j^m(\mathbf{x}', \boldsymbol{\theta}')] = \boldsymbol{\Sigma}_{ij}^m R^m((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}'))$ where $\boldsymbol{\Sigma}_{ij}^m$ is the covariance between the i th and j th computer model responses at the same input values. Further, this covariance structure reduces to the covariance function for a single-response Gaussian process model when $q = 1$. A Gaussian correlation function is used:

$$R^m((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')) = \exp \left\{ -\sum_{k=1}^d \omega_k^m (x_k - x'_k)^2 \right\} \exp \left\{ -\sum_{k=1}^r \omega_{d+k}^m (\theta_k - \theta'_k)^2 \right\}, \quad (4.9)$$

parameterized by the $(d+r) \times 1$ vector of roughness parameters $\boldsymbol{\omega}^m$, which represent the rate at which the correlation between $y_i^m(\mathbf{x}, \boldsymbol{\theta})$ and $y_i^m(\mathbf{x}', \boldsymbol{\theta}')$ decays to zero as $(\mathbf{x}, \boldsymbol{\theta})$ and $(\mathbf{x}', \boldsymbol{\theta}')$ diverge. Lower values of $\boldsymbol{\omega}^m$ indicate a smoother MRGP model for the response $y_i^m(\mathbf{x}, \boldsymbol{\theta})$.

Reference [31] provides some justification, beyond the relatively simple parameterization that results in tractable computations, for using the covariance structure in Eq. (4.9). In particular, if multiple responses mutually depend on the same set of inputs (both design variables and calibration parameters), then one might expect the responses to be correlated and to share a similar spatial correlation structure or smoothness. The formulation of Eq. (4.9) assumes such a shared spatial correlation structure. On the other hand, if the responses are weakly correlated with different

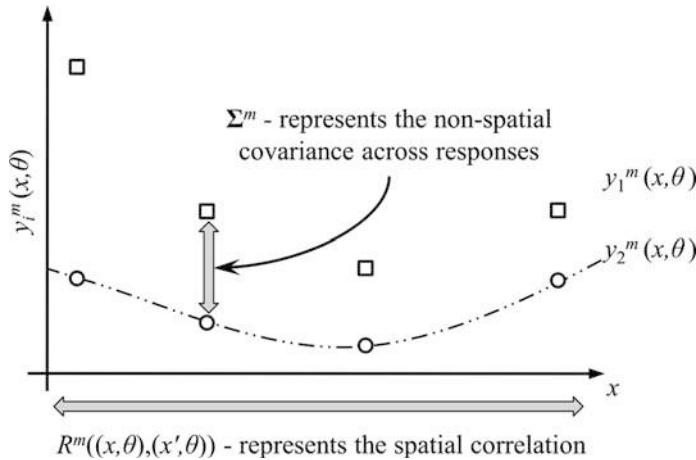


Fig. 4.11 Schematic showing the MRGP model's covariance function represented by spatial correlation and nonspatial covariance for a fixed θ

spatial correlations, then one can model each response independently, allowing each to have a different spatial correlation function [31].

Figure 4.11 illustrates the separation of the covariance function into spatial and nonspatial portions for the case of two responses. The gray arrow below the x -axis of Fig. 4.11 signifies the spatial correlation portion of the covariance function, which is represented by $R^m((x, \theta), (x', \theta))$ for computer model inputs x and x' with fixed θ . Alternatively, the nonspatial covariance matrix portion of the covariance function, Σ^m , represents the covariance across response functions, as signified in Fig. 4.11 by the gray arrow between the responses. Whereas the spatial correlation function dictates the smoothness of both responses, the nonspatial covariance matrix dictates the similarity between any trends or fluctuations (not represented by the prior mean function) in the two responses.

To obtain maximum likelihood estimates (MLEs) of the hyperparameters $\phi^m = \{\mathbf{B}^m, \Sigma^m, \omega^m\}$ for the computer model MRGP, the multivariate normal log-likelihood function based on the simulation data $\mathbf{Y}^m = [\mathbf{y}_1^m, \dots, \mathbf{y}_q^m]$ is maximized, where $\mathbf{y}_i^m = [y_i^m(\mathbf{x}_1^m, \theta_1^m), \dots, y_i^m(\mathbf{x}_{N_m}^m, \theta_{N_m}^m)]^T$, collected at N_m input sites $\mathbf{X}^m = [\mathbf{x}_1^m, \dots, \mathbf{x}_{N_m}^m]^T$ and $\boldsymbol{\Theta}^m = [\boldsymbol{\theta}_1^m, \dots, \boldsymbol{\theta}_{N_m}^m]^T$. Previous literature [37, 38] suggest a space-filling experimental design for the input settings of \mathbf{X}^m and $\boldsymbol{\Theta}^m$. Furthermore, to improve the numerical stability of the MLE algorithm, the inputs \mathbf{X}^m and $\boldsymbol{\Theta}^m$ can be transformed to the range of 0 to 1, and the simulation data can be standardized to have a sample mean of 0 and a sample standard deviation of 1 [39]. Details of the MLE algorithm can be found in Appendix A.

Inference of the computer model response $y_i^m(\mathbf{x}, \theta)$ at any \mathbf{x} and θ , can be determined by inserting the MLEs of the hyperparameters into the MRGP model posterior mean and covariance equations, Eqs. (4.22) and (4.23) of Appendix B. After obtaining the MLEs of the hyperparameters, the next step is to estimate the

hyperparameters of the MRGP model representing the discrepancy functions, as follows.

3.1.2 Multi-response Gaussian Process Model for the Discrepancy Functions (Module 2)

The MRGP model of the experimental response is the sum of two MRGP models representing both the computer model $y_i^m(\mathbf{x}, \boldsymbol{\theta})$ and the discrepancy functions $\delta_i(\mathbf{x})$. This MRGP model will then infer the experimental response $y_i^e(\mathbf{x})$ at any point \mathbf{x} .

To define the MRGP model for the experimental response, the MRGP model for the discrepancy functions needs to be defined first, which is the focus of Module 2. The prior of the MRGP model for the discrepancy functions is

$$\boldsymbol{\delta}(\cdot) \sim \mathcal{GP}\left(\mathbf{h}^\delta(\cdot)\mathbf{B}^\delta, \boldsymbol{\Sigma}^\delta R^\delta(\cdot, \cdot)\right), \quad (4.10)$$

where $\boldsymbol{\delta}(\mathbf{x}) = [\delta_1(\mathbf{x}), \dots, \delta_q(\mathbf{x})]$ denotes the multiple discrepancy functions. The mean function is comprised of a vector of specified regression functions $\mathbf{h}^\delta(\mathbf{x}) = [h_1^\delta(\mathbf{x}), \dots, h_s^\delta(\mathbf{x})]^T$ and a matrix of unknown regression coefficients $\mathbf{B}^\delta = [\boldsymbol{\beta}_1^\delta, \dots, \boldsymbol{\beta}_q^\delta]$, where $\boldsymbol{\beta}_i^\delta = [\beta_{1,i}^\delta, \dots, \beta_{s,i}^\delta]^T$. The prior covariance function of this MRGP model is the product of an unknown nonspatial covariance matrix $\boldsymbol{\Sigma}^\delta$ and a spatial correlation function $R^\delta(\mathbf{x}, \mathbf{x}')$, which is a Gaussian correlation function (similar to Eq. (4.9) but without the $\boldsymbol{\theta}$ inputs) parameterized by the $d \times 1$ vector of roughness parameters $\boldsymbol{\omega}^\delta$. This prior covariance function can be rewritten as $\text{Cov}[\delta_i(\mathbf{x}), \delta_j(\mathbf{x}')] = \boldsymbol{\Sigma}_{ij}^\delta R^\delta(\mathbf{x}, \mathbf{x}')$, where $\boldsymbol{\Sigma}_{ij}^\delta$ is the covariance between the i th and j th discrepancy functions at the same input values. Following the work of Kennedy and O'Hagan [1], the computer model, the discrepancy functions, and the experimental uncertainty are assumed a priori statistically independent (i.e., $\text{Cov}[y_i^m(\mathbf{x}, \boldsymbol{\theta}), \delta_j(\mathbf{x}')] = 0$, $\text{Cov}[y_i^m(\mathbf{x}, \boldsymbol{\theta}), \varepsilon_j] = 0$, and $\text{Cov}[\delta_i(\mathbf{x}), \varepsilon_j] = 0$ for $i, j \in \{1, \dots, q\}$ and all \mathbf{x} , \mathbf{x}' , and $\boldsymbol{\theta}$). These assumptions of a priori independence simplify many of the calculations required in the multi-response modular Bayesian approach. It should be noted that the posterior distributions for these quantities are generally highly correlated, even though their prior distributions are independent.

The priors for the computer model MRGP model [Eq. (4.8)] and the discrepancy functions MRGP model [Eq. (4.10)] are combined via Eq. (4.7) to form the prior for the experimental responses MRGP model:

$$\begin{aligned} \mathbf{y}^e(\cdot) | (\boldsymbol{\theta} = \boldsymbol{\theta}^*) &\sim \mathcal{GP}(m^e(\cdot, \boldsymbol{\theta}^*), V^e((\cdot, \boldsymbol{\theta}^*), (\cdot, \boldsymbol{\theta}^*))), \\ m^e(\mathbf{x}, \boldsymbol{\theta}) &= \mathbf{h}^m(\mathbf{x}, \boldsymbol{\theta})\mathbf{B}^m + \mathbf{h}^\delta(\mathbf{x})\mathbf{B}^\delta, \\ V^e((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta})) &= \boldsymbol{\Sigma}^m R^m((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta})) + \boldsymbol{\Sigma}^\delta R^\delta(\mathbf{x}, \mathbf{x}') + \boldsymbol{\lambda}, \end{aligned} \quad (4.11)$$

where $\mathbf{y}^e(\mathbf{x}) = [y_1^e(\mathbf{x}), \dots, y_q^e(\mathbf{x})]$ denote the multiple experimental responses. The $q \times q$ diagonal covariance matrix $\boldsymbol{\lambda}$ with diagonal entries $\lambda_1, \dots, \lambda_q$ represents the experimental uncertainty, i.e., $\varepsilon_i \sim \mathcal{N}(0, \lambda_i)$. Since Eq. (4.11) depends on the true calibration parameters $\boldsymbol{\theta}^*$, the objective is now to estimate the hyperparameters of

the MRGP model for the discrepancy functions without knowing the true value of θ^* .

Kennedy and O'Hagan [1] developed a procedure to obtain estimates of the hyperparameters of a single discrepancy function using the prior distribution of the calibration parameters. Specifically, this procedure constructs a likelihood function, marginalized with respect to the prior distribution of the calibration parameters, from the simulation and experimental data and using the MLEs of the computer model hyperparameters from Module 1. The experimental data consists of experimental response observations at N_e input sites, $\mathbf{X}^e = [\mathbf{x}_1^e, \dots, \mathbf{x}_{N_e}^e]^T$ (transformed to the range [0, 1]) with responses $\mathbf{Y}^e = [\mathbf{y}_1^e, \dots, \mathbf{y}_q^e]$, where $\mathbf{y}_i^e = [y_i^e(\mathbf{x}_1^e), \dots, y_i^e(\mathbf{x}_{N_e}^e)]^T$ (standardized). Then the MLE estimates of the hyperparameters $\hat{\phi}^\delta = \{\mathbf{B}^\delta, \Sigma^\delta, \omega^\delta, \lambda\}$ occur at the location of the maximum of the likelihood function (marginalized with respect to the priors of the calibration parameters). Appendix C contains a detailed description of this procedure. After estimating the hyperparameters of the two MRGP models in Modules 1 and 2, Bayes theorem is used to calculate the posterior distribution of θ , as follows.

3.1.3 Posterior of the Calibration Parameters (Module 3)

Module 3 calculates the posterior of the calibration parameters using the MLEs of the hyperparameters from Modules 1 and 2, the simulation data, and the experimental data. The posterior of the calibration parameters is:

$$p(\theta|\mathbf{d}, \hat{\phi}) = \frac{p(\mathbf{d}|\theta, \hat{\phi})p(\theta)}{p(\mathbf{d}|\hat{\phi})}, \quad (4.12)$$

where $\hat{\phi}$ are the MLEs of $\phi = \{\phi^m, \phi^\delta\}$, $\mathbf{d} = [vec(\mathbf{Y}^m)^T \ vec(\mathbf{Y}^e)^T]^T$ is the complete response data, and $p(\theta)$ is the prior of the calibration parameters (e.g., a uniform distribution defined a priori). The likelihood function $p(\mathbf{d}|\theta, \hat{\phi})$ is a multivariate normal distribution whose mean and covariance are determined by the two MRGP models for the computer model [Eq. (4.8)] and the experimental responses [Eq. (4.11)]. The denominator $p(\mathbf{d}|\hat{\phi})$ of Eq. (4.12) is the marginal distribution of the data [40], which does not depend upon θ (i.e., the denominator is a normalizing constant). Appendix D details the equations for calculating the posterior distribution in Eq. (4.12). The posterior distribution of the calibration parameters and the MLEs of the hyperparameters from Modules 1 and 2 influence the prediction of the experimental responses in the next and final module.

3.1.4 Prediction of the Experimental Responses and Discrepancy Function (Module 4)

After estimating the hyperparameters in Modules 1 and 2 and collecting the simulation and experimental data, the conditional (given a specific value of θ) posterior distribution for the experimental responses can be calculated at any point \mathbf{x} . The unconditional posterior distribution is then obtained by marginalizing the conditional posterior distribution with respect to the posterior of the calibration

parameters calculated in Module 3, as discussed in Appendix E. Therefore, the marginalized posterior distribution of the multiple experimental responses accounts for parameter uncertainty, model discrepancy, interpolation uncertainty, and experimental uncertainty. Since the discrepancy functions are also represented by a MRGP model, their posterior distributions can be calculated in a similar manner. One should note that the MRGP model prediction of the discrepancy functions will depend on the value of the calibration parameters. Therefore, similar to the prediction of the physical experiments, the prediction of the discrepancy functions is marginalized with respect to the posterior distribution of the calibration parameters.

3.2 Applying the Multi-response Approach to the Simply Supported Beam

This section revisits the simply supported beam example, introduced in Sect. 2.2, to show how using multiple responses can improve the identifiability of the calibration parameters and the discrepancy functions. Briefly (refer to Sect. 2.2 for details), the simply supported beam is fixed at one end and supported by a roller on the other end. A static force is applied to the midpoint of the beam to induce various responses, e.g., deflection and stress. The magnitude of this force was chosen as the design variable x , while Young's modulus was treated as the calibration parameter θ . For generating the “physical experiment” (which is taken to be the same computer simulations but with a more sophisticated material law) data, the true value of the calibration parameter is set to $\theta^* = 206.8 \text{ GPa}$. However, θ^* is treated as unknown during the analysis and we use a uniform prior distribution over the range $150 \leq \theta \leq 300 \text{ GPa}$. The experimental uncertainty was set to zero (i.e., $\lambda_i = 0$).

3.2.1 Identifiability with Single Responses

Before using multiple responses for the simply supported beam, the resulting identifiability from using only single responses is first examined. Table 4.3 shows the six responses measured from the beam (for both the computer model y_i^m and the physical experiments y_i^e). The simulation data for each response \mathbf{y}_i^m was

Table 4.3 Responses measured in the simply supported beam example

Response	Description
y_1	Strain at the midpoint of the beam (mm)
y_2	Plastic strain at the midpoint of the beam (mm)
y_3	Angle of deflection at the end of the beam (radians) (y of Sect. 2.2)
y_4	Internal energy of the beam (Joules)
y_5	Displacement at the middle of the beam (m)
y_6	Plastic dissipation energy (Joules)

collected using a 4 by 4 grid over the input space ($1300 \leq x \leq 2300$ N and $150 \leq \theta \leq 300$ GPa). The experimental data \mathbf{y}_e^e was collected at $\mathbf{X}^e = [1300, 1400, 1500, \dots, 2300]^T$ ($N_e = 11$). For each single response, the modular Bayesian approach was used to calculate the posterior distributions of the calibration parameter, the discrepancy function, and the experimental response.

Because of the relatively large amount of experimental data and no experimental uncertainty, the posterior distribution of the experimental response for each single response was accurate and precise (see Fig. 4.5a in Sect. 2.2 for y_3 of Table 4.3). Even so, the calibration parameter and the discrepancy function are not identifiable as evident from the high uncertainty reflected by their posterior distributions, as shown in Fig. 4.12 for responses 1 through 4 (similar results were obtained for y_5 and y_6 but are not shown due to space). Table 4.4 shows the resulting posterior mean and standard deviation of the calibration parameter for each single response. The large posterior standard deviations in Table 4.4 again indicate a lack of identifiability. Since the experimental data are already relatively dense, additional

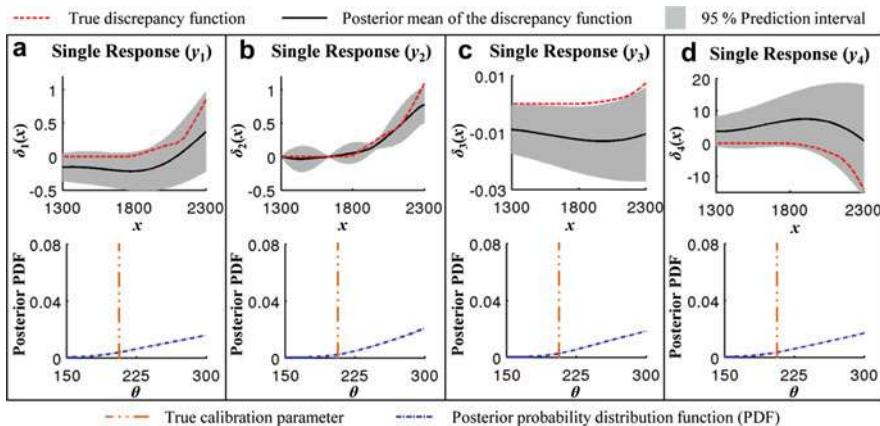


Fig. 4.12 Posterior distributions of the discrepancy function and the calibration parameter using a single response for (a) y_1 , (b) y_2 , (c) y_3 , and (d) y_4

Table 4.4 Single-response (SR) posterior mean ($\mu_{i,post}^{SR}$ in GPa) and standard deviation ($\sigma_{i,post}^{SR}$ in GPa) of the calibration parameter θ

Response	SR posterior distribution of θ	
y_i	$\mu_{i,post}^{SR}$	$\sigma_{i,post}^{SR}$
y_1	256.74	31.68
y_2	264.70	27.15
y_3	262.18	26.67
y_4	261.79	26.92
y_5	261.15	27.84
y_6	254.48	33.30

data will not improve identifiability when using only a single response. Instead, in the remainder of this section, the use of multiple responses is explored to enhance identifiability.

3.2.2 Identifiability with Multiple Responses

With the goal of enhancing identifiability, the procedure described in Sect. 2 combines information from multiple responses. In general, there will often be many responses that are automatically calculated in the computer model and that could potentially be measured experimentally. Ideally, one might like to include all of the responses in the model updating formulation, but this is not practical for a number of reasons. First, it can become more computationally expensive to implement the modular Bayesian approach with additional responses and including too many responses may result in numerical instabilities. Furthermore, some responses may be largely redundant, containing nearly the same information as other responses, which will not improve identifiability. As seen below, different combinations of responses result in drastically different identifiability. Third, although multiple responses are available for free in the computer simulation, their experimental measurement may involve prohibitive costs.

For ease of illustration and computational reasons, the multiple responses considered are various pairs of responses. To explore how different pairs of responses affect identifiability in the simply supported beam example, the relevant posterior distributions for the 15 pairs of responses from Table 4.3 are calculated. Specifically, for each pair of responses, the multi-response modular Bayesian approach discussed in Sect. 3.1 is used to calculate the posterior distributions of the calibration parameter, the discrepancy functions, and the experimental responses. The modular Bayesian approach begins with Module 1, which creates a MRGP model of the computer model using the simulation data \mathbf{y}_i^m (observed at the same input settings \mathbf{X}^m as in Sect. 3.2.1). In Module 2, the hyperparameters of the MRGP model for the discrepancy functions are estimated from the experimental data \mathbf{y}_i^e (observed at the same input settings \mathbf{X}^e as in Sect. 3.2.1), the prior distribution of the calibration parameter, and the hyperparameter MLEs from Module 1. Finally, Modules 3 and 4 calculate the posterior distributions for the calibration parameter, the discrepancy functions, and the experimental responses. As with the single responses in Sect. 3.2.1, the posterior of the experimental responses was extremely accurate and precise, although the identifiability of the calibration parameters varied widely from pair to pair of responses.

Table 4.5 shows the resulting posterior mean and standard deviation of the calibration parameter for each set of responses. In cases where the posterior distribution is not normal, as in Fig. 4.12, the standard deviation can be an oversimplification of the dispersion of the posterior distribution, but it is still perhaps the most relevant single measure of identifiability. To quantify identifiability improvement, the multi-response posterior standard deviation $\sigma_{ij,post}^{MR}$ is compared to the best single-response posterior standard deviation of the pair (i.e., the smaller of the two single-response posterior standard deviations $\sigma_{ij,min}^{SR} = \min(\sigma_{i,post}^{SR}, \sigma_{j,post}^{SR})$). The

Table 4.5 Multi-response posterior mean $\mu_{ij,post}^{MR}$ (in GPa) and standard deviation $\sigma_{ij,post}^{MR}$ (in GPa) of the calibration parameter θ and identifiability improvement compared to the single response

Responses			MR posterior distribution of θ		$\sigma_{ij,min}^{SR}$	Std. dev. improvement
y_i	y_j		$\mu_{ij,post}^{MR}$	$\sigma_{ij,post}^{MR}$		
y_1	y_2	(1)	259.81	27.37	31.68	14%
y_1	y_3		213.70	10.13	26.67	62%
y_1	y_4		215.00	9.29	26.92	65%
y_1	y_5		219.83	15.57	27.84	44%
y_1	y_6		247.33	24.08	31.68	24%
y_2	y_3		218.82	10.96	26.67	59%
y_2	y_4		228.28	18.35	26.92	32%
y_2	y_5		224.50	15.42	27.84	45%
y_2	y_6		265.96	26.46	33.30	21%
y_3	y_4	(3)	209.23	5.49	26.67	79%
y_3	y_5		210.39	6.27	26.67	76%
y_3	y_6		210.30	4.86	26.67	82%
y_4	y_5		213.23	3.63	26.92	87%
y_4	y_6		211.91	3.90	26.92	86%
y_5	y_6		210.35	4.67	27.84	83%

posterior standard deviation improvement, “std. dev. improvement” in Table 4.5, is defined as $(\sigma_{ij,min}^{SR} - \sigma_{ij,post}^{MR}) / \sigma_{ij,min}^{SR}$.

To better visualize improved identifiability, Fig. 4.13 shows the posterior distributions of the calibration parameter and the discrepancy functions for three sets of multiple responses with different degrees of identifiability improvement. The sets of responses shown in Fig. 4.13a, b, and c will from hereon be referred to as Case 1, 2, and 3, respectively, and are labeled in Table 4.5.

As can be seen in Table 4.5 and Fig. 4.13, the extent of the identifiability improvement varies widely from pair to pair. In Case 1, the combination of the strain y_1 and the plastic strain y_2 enhances identifiability little beyond that of a single response, and the resulting posterior distributions in Fig. 4.12a, b for the single responses are similar to those in Fig. 4.13a for the pair of responses (a 14% improvement, from Table 4.5). In contrast, in Case 2, the combination of the plastic strain y_2 and the angle of deflection y_3 resulted in a much more substantial improvement in identifiability. Specifically, the posterior distributions shown in Fig. 4.13b represent a 59% improvement in posterior standard deviation (see Table 4.5).

In Case 3, the combination of the angle of deflection y_3 and the internal energy y_4 resulted in even more substantially enhanced identifiability. The corresponding posterior distribution in Fig. 4.13c is much more narrowly dispersed (a 79% improvement, from Table 4.5) than for the single-response cases. Additionally, the posterior mean of the calibration parameter is quite close to the true value of

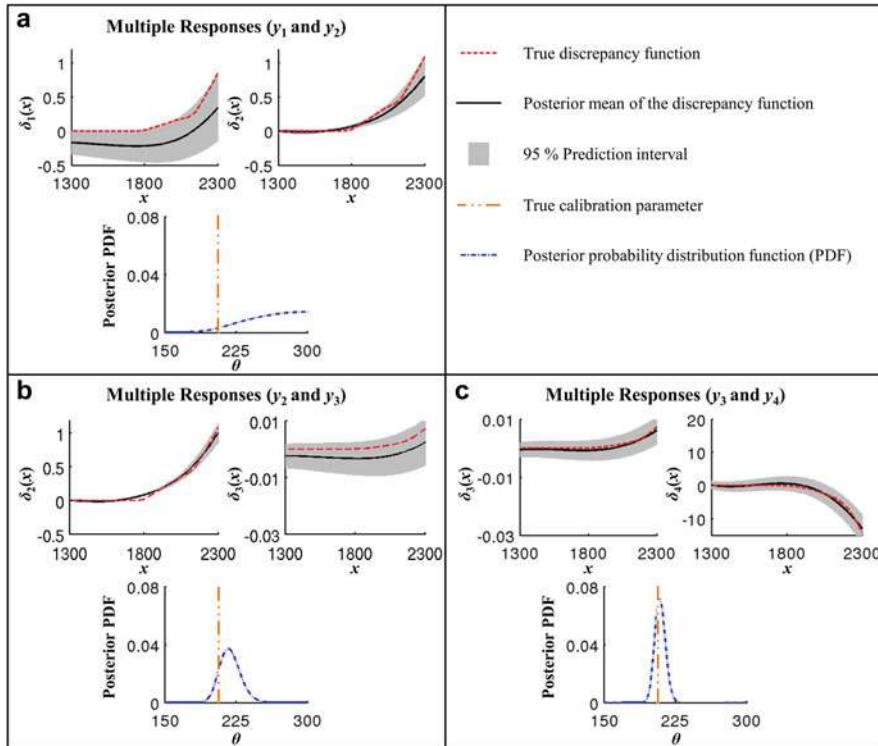


Fig. 4.13 Posterior distributions of the discrepancy functions and the calibration parameter using multiple responses for (a) y_1 and y_2 , (b) y_2 and y_3 , and (c) y_3 and y_4

$\theta^* = 206.8$ (marked by the vertical line in Fig. 4.13c). Moreover, the posteriors of the discrepancy functions $\delta_3(x)$ and $\delta_4(x)$ capture the true functions with relatively small uncertainty. These three cases are representative of the varied improvements in identifiability for all pairs of responses listed in Table 4.5.

In the simply supported beam example, 12 out of the 15 pairs of responses resulted in an identifiability improvement of 25% or more. The degree of improvement varied from little (e.g., 14%), to moderate (e.g., 24% and 32%), to substantial (e.g., 87%). This example illustrates that, if identifiability cannot be obtained using a single response, multiple responses may be used to achieve improved identifiability.

3.3 Remarks on the Multi-response Modular Bayesian Approach

In this section, it has been shown that identifiability can be enhanced by using multiple responses in model updating. The approach is most beneficial when data from a single response results in a lack of identifiability. The existing

single-response modular Bayesian approach was extended to multiple responses, and this extension required the use of multi-response Gaussian process (MRGP) models to represent the multiple computer simulation responses and the multiple discrepancy functions. The multi-response modular Bayesian approach allows one to calculate the posterior distributions of the calibration parameters upon which the multiple responses share a mutual dependence, the posterior distributions of the discrepancy functions of each response, and the predictions of multiple experimental responses.

By applying the multi-response modular Bayesian approach to the simply supported beam example, it was shown that including multiple responses can improve identifiability. In this example, the majority of paired responses resulted in an identifiability improvement of over 50%, and some pairs resulted in an improvement of over 85%. Across all pairs, the resulting improvement in identifiability varied from 14% to 87%. Overall, the simply supported beam example illustrates the effectiveness of using additional information from multiple responses to improve identifiability, when identifiability cannot be achieved using a single response.

As with the single-response modular Bayesian approach, one limitation of the proposed multi-response modular Bayesian approach is the computational cost, which is predominantly affected by the number of observations. Using additional responses also increases the computational cost, but not as significantly as adding observed data. For example, when using two responses, 17 s (on a single Intel 2.66 GHz processor) of computation time was required to calculate the posterior distributions of the simply supported beam example; whereas, when using a single response, only 10 s of computation time was required. The increase in computation time was a result of calculating the posterior distributions for two experimental responses and two discrepancy functions in Module 4. Also, one should notice that the increase in computational cost was due to the Bayesian computations, and not the computer simulations, because no additional computer simulations were run.

Another limitation of the multi-response approach is that considering additional responses can result in numerical instabilities. Theoretically, using additional responses should potentially improve, and never worsen, identifiability. However, additional responses can also introduce numerical conditioning problems. For example, when using more than two responses in the simply supported beam example, estimation of Σ^δ for the discrepancy functions in Module 2 became problematic. Due to the high correlations between the responses, the estimate of Σ^δ was close to singular, which caused numerical instability in the subsequent calculations.

Although in the examples the number of calibration parameters was less than or equal to the number of responses, this is not a requirement for identifiability (much like how regression modeling with a single-response variable can be used to fit a model with multiple parameters). Of course, using more responses will generally improve identifiability, but this improvement will depend on the specifics of the system (including the form of the computer model, the physical experiments, and the discrepancy functions), the multiple responses being measured (as seen in Figs. 4.12 and 4.13), and the location of the measurements in \mathbf{x} . In light of this, it is difficult to

generalize how many responses are necessary to achieve identifiability based on the number of calibration parameters.

Together, this section and the previous section (Sect. 2) shed light on the challenging problem of identifying calibration parameters and discrepancy functions when combining data from a computer model and physical experiments. They demonstrate that identifiability is often possible and can be reasonably achieved with proper analyses in certain engineering systems.

4 Preposterior Analyses for Predicting Identifiability in Model Calibration

It was shown in Sect. 2.3 that the degree of identifiability depends strongly on the nature of the computer model response as a function of the input variables and the calibration parameters, as well as on the prior assumptions regarding the discrepancy function (e.g., the smoothness of the discrepancy function). Computer simulations are inherently multi-response, as there are many intermediate and final response variables at many different spatial and temporal locations that are automatically calculated during the course of the simulation. Section 3.2 showed that, for systems with poor identifiability based on a single measured physical experimental response, identifiability may be improved by measuring multiple physical responses. In all situations, identifiability obviously depends on the design of the physical experiment, i.e., the number of experimental runs and the input settings for each run.

The same can be said when estimating the parameters of any parametric model based on physical experimental data, and standard criteria for designing such physical experiments are often based on measures related to parameter identifiability. However, calibration of computer simulation models involves a fundamental distinction: Prior to designing the physical experiment, one can learn via simulation a great deal about the nature of the functional dependence of the response(s) on the input variables and on the calibration parameters, and this knowledge can be exploited when designing the physical experiment to provide better identifiability. Indeed, because of the cost and difficulty in developing and placing apparatus for measuring certain responses, it is generally not feasible to measure experimentally all of the great many responses that are automatically calculated in the simulations. Moreover, it is generally not necessary, because measuring only a subset of the responses may result in acceptable identifiability. It is preferable to select a relatively small but most appropriate subset of responses to measure experimentally, to best enhance identifiability.

In order to accomplish this, one needs to predict or approximate the degree of identifiability prior to conducting the physical experiments, but after conducting the computer simulations. The primary purpose of this section is to introduce a preposterior analysis developed in [41–43] and investigate the use of the preposterior covariance matrix [44, 45] of the calibration parameters for this purpose and to demonstrate that it provides a reasonable prediction of the actual posterior

covariance, at least for the examples that are considered. Consequently, the preposterior covariance matrix can serve as a reasonable criterion for designing physical experiments in order to achieve good identifiability when calibrating computer simulation models.

It is worth noting that a preposterior analysis cannot improve one's knowledge of the calibration parameters, because it is conducted prior to observing any physical experimental data. However, it can provide a reasonable quantification of the uncertainty of the calibration parameters that one expects will result after observing the experimental data. This is analogous to what occurs in standard optimal design of experiments [46,47] for fitting a response surface via linear regression. Given the design matrix and the standard deviation of the random observation error, one can calculate the covariance matrix of the estimated parameters prior to conducting the experiment and use this as the experimental design criterion. Of course, one cannot estimate the parameters until the experimental data are collected.

The format of the remainder of the section is as follows. Section 4.1 presents the approach to calculate the preposterior covariance matrix of the calibration parameters. Section 4.2 discusses a modification of the preposterior analysis that provides insight into the behavior of the preposterior covariance as a means of predicting the posterior covariance. Section 4.3 provides a detailed description of the surrogate preposterior analysis, in conjunction with the multi-response modular Bayesian approach, for substantially reducing the computational cost in the preposterior analysis. Sections 4.4, 4.5, and 4.6 illustrates the effectiveness of the preposterior analysis in predicting identifiability prior to conducting physical experiments by using a number of examples and discusses the issue of identifiability of the calibration parameters. Remarks are made in Sect. 4.7.

4.1 Preposterior Analysis

The preposterior analysis framework [41–43] for selecting the experimentally measured responses is shown in Fig. 4.14. After conducting the simulations but before conducting the experiments, the user first defines a number of candidate subsets of responses for preposterior analysis. This could be accomplished based on some heuristics (e.g., one that will be discussed in Sect. 4.3) and/or on which responses are deemed inexpensive to measure. If there are N_c candidate subsets of responses, the analysis evaluates the degree of identifiability for each via the preposterior covariance.

We next provide an overview of the algorithm for calculating the preposterior covariance matrix. Following this, detailed descriptions of each step are given in the algorithm. The algorithm begins by fitting the Gaussian process model for $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$ based on the computer simulation data (Step 1a). The user then specifies the physical experimental input settings and assigns priors for the GP model of $\delta(\mathbf{x})$, for the calibration parameters, and for $\boldsymbol{\epsilon}$ (Steps 1b and 1c). Because the preposterior analysis is conducted prior to observing actual experimental data, the main body of the algorithm is a Monte Carlo (MC) simulation (Step 2) in which, on each

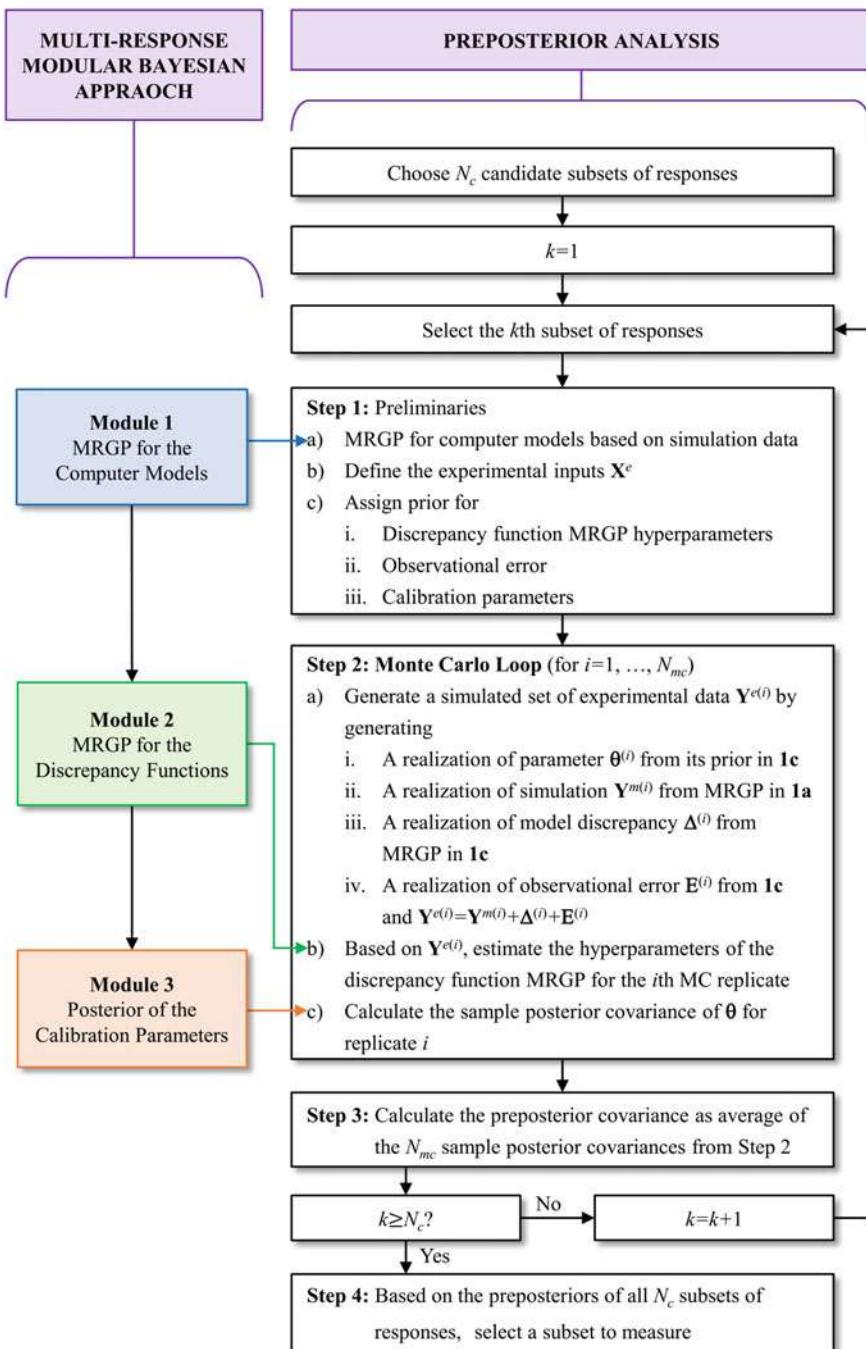


Fig. 4.14 Flowchart of the preposterior analysis for selecting experiment responses

MC replicate, a hypothetical set of physical experimental response observations is generated (Step 2a) at the specified input settings based on the knowledge of $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$ from the computer simulation data and the relevant prior distributions for $\delta(\mathbf{x}), \boldsymbol{\theta}$, and $\boldsymbol{\varepsilon}$. Then Step 2b and 2c calculate a posterior covariance matrix for $\boldsymbol{\theta}$ for the experimental response observations generated in that MC replicate. The preposterior covariance matrix for $\boldsymbol{\theta}$ is taken to be (Step 3) the average of the posterior covariance matrices over all MC replicates. Within Step 2, Modules 2 and 3 of the standard modular Bayesian approach are used. The subset of responses that yields the tightest preposterior distribution (cost/difficulty of measurement can also be taken into consideration) would be deemed the most likely to achieve good identifiability.

Each step of the preposterior analysis is described in detail as follows. The algorithm is presented in a multi-response context; however one should note that it can be easily “degenerated” to a single-response version, by replacing relevant multi-response vectors/matrices (e.g., $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$, $\delta(\mathbf{x})$, Σ^m , Σ^δ) by their single-response scalar counterparts (e.g., $y^m(\mathbf{x}, \boldsymbol{\theta})$, $\delta(\mathbf{x})$, σ_m^2 , σ_δ^2).

4.1.1 Step 1: Preliminaries

In this step, an MRGP model is fit to the simulation data, and several quantities needed in the subsequent steps are defined. First, Module 1 of the multi-response modular Bayesian approach described in Sect. 3.1.1 is implemented to construct the MRGP model for the computer simulations based on the simulation data $\mathbf{Y}^m = [\mathbf{y}_1^m, \dots, \mathbf{y}_q^m]$, where $\mathbf{y}_i^m = [y_i^m(\mathbf{x}_1^m, \boldsymbol{\theta}_1^m), \dots, y_i^m(\mathbf{x}_{N_m}^m, \boldsymbol{\theta}_{N_m}^m)]^T$, collected at N_m input sites $\mathbf{X}^m = [\mathbf{x}_1^m, \dots, \mathbf{x}_{N_m}^m]^T$ and $\boldsymbol{\Theta}^m = [\boldsymbol{\theta}_1^m, \dots, \boldsymbol{\theta}_{N_m}^m]^T$. The maximum likelihood estimation (MLE) method estimates the hyperparameters $\boldsymbol{\phi}^m = \{\mathbf{B}^m, \Sigma^m, \boldsymbol{\omega}^m\}$ of the computer simulations, from which expressions for the posterior distribution of $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$ are obtained. In the MRGP model, \mathbf{Y}^m has mean vector $\mathbf{H}^m \mathbf{B}^m$, where $\mathbf{H}^m = [\mathbf{h}^m(\mathbf{x}_1^m)^T, \dots, \mathbf{h}^m(\mathbf{x}_{N_m}^m)^T]^T$ with each $\mathbf{h}^m(\mathbf{x})$ a specified regression basis function and \mathbf{B}^m a matrix of coefficients; and \mathbf{Y}^m has covariance matrix $\Sigma^m \otimes \mathbf{R}^m$, where \mathbf{R}^m is the $N_m \times N_m$ matrix with k th-row, j th-column entry equal to the prior correlation $R^m((\mathbf{x}_j^m, \boldsymbol{\theta}_j^m), (\mathbf{x}_k^m, \boldsymbol{\theta}_k^m))$ between $\mathbf{y}^m(\mathbf{x}_j^m, \boldsymbol{\theta}_j^m)$ and $\mathbf{y}^m(\mathbf{x}_k^m, \boldsymbol{\theta}_k^m)$. For all examples in this chapter, a constant prior mean function of unknown magnitude (i.e., $\mathbf{h}^m(\mathbf{x}) = 1$ and \mathbf{B}^m a $1 \times q$ vector hyperparameter to be estimated) and a Gaussian correlation function R^m were used. For numerical stability of the MLE algorithm, \mathbf{X}^m and $\boldsymbol{\Theta}^m$ are transformed to the range $[0, 1]$, and each \mathbf{y}_i^m is standardized to have a sample mean of 0 and a sample standard deviation of 1 [39].

In subsequent steps, the MRGP model is used to infer the simulation response at input settings within the design domain where no simulations are conducted. A multivariate normal random number generator will be used with mean and covariance taken to be the posterior mean and covariance of the MRGP model of $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$ from Step 1a. The posterior mean and covariance are calculated using results from APPENDIX B. Here plug-in MLEs for Σ^m and $\boldsymbol{\omega}^m$ with a

non-informative prior for \mathbf{B}^m are used. Over the remainder of the preposterior analysis, the MRGP model of $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$ is not updated from that of Step 1a.

Next, a total of N_e experimental input settings $\mathbf{X}^e = [\mathbf{x}_1^e, \dots, \mathbf{x}_{N_e}^e]^T$ are defined, representing the input settings, i.e., the experimental design that one intends to use for the physical experiment (in the MC simulations of Step 2, simulated experimental response values will be generated at these input settings). Step 1c then assigns the prior distribution parameters for the priors of MRGP hyperparameters in the discrepancy function. The prior for discrepancy function MRGP hyperparameters $\boldsymbol{\phi}^\delta \setminus \lambda = \{\mathbf{B}^\delta, \boldsymbol{\Sigma}^\delta, \boldsymbol{\omega}^\delta\}$ [analogous to the hyperparameters $\boldsymbol{\phi}^m$ for the MRGP model of $\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta})$] captures the prior uncertainty and nonlinearity of the discrepancy functions; $\boldsymbol{\Sigma}^\delta$ in Eq. (4.10) controls the magnitude of uncertainty of the MRGP model, and $\boldsymbol{\omega}^\delta$ controls its roughness (rougher surfaces correspond to larger values of $\boldsymbol{\omega}^\delta$). This prior for $\boldsymbol{\phi}^\delta$ is used only in Step 2a-iii of the MC loop, in which values of $\boldsymbol{\phi}^\delta$ are sampled from the specified prior before generating a hypothetical discrepancy function. Because the modular Bayesian approach calculates MLEs of $\boldsymbol{\phi}^\delta$ in Step 2b, the prior for $\boldsymbol{\phi}^\delta$ is not incorporated via maximum a posteriori estimators, although one could modify the approach to do this if desired. For computational convenience, $\{\mathbf{B}^\delta, \boldsymbol{\Sigma}^\delta, \boldsymbol{\omega}^\delta\}$ can be assigned with point mass priors.

Additionally in Step 1c, the user specifies the prior distribution for the experimental error, which is assumed to be *i.i.d.* normal with mean 0 and a prior variance chosen based on prior knowledge of the level of measurement error. λ is treated in the same manner as $\boldsymbol{\phi}^\delta$. Its prior is used in Step 2a-iv when generating a random value for $\boldsymbol{\epsilon}$, but this prior is not used when calculating the MLE of λ in Step 2b.

Lastly in Step 1c, the user assigns a prior distribution $p(\boldsymbol{\theta})$ to the calibration parameters $\boldsymbol{\theta}$. The prior for calibration parameters $\boldsymbol{\theta}$ should have support spanning the entire range of possible values for $\boldsymbol{\theta}$; for a less informative prior, one can use a uniform distribution over a broad range, while a normal distribution with specified mean and variance can be used for a more informative prior. Since the uncertainty in the calibration parameters is of direct interest, Module 3 of the modular Bayesian approach in Step 2c calculates a full posterior distribution for $\boldsymbol{\theta}$ based on the specified prior, in contrast to the MLE-only estimation for $\boldsymbol{\phi}^m$ and $\boldsymbol{\phi}^\delta$ in Modules 1 and 2. Based on the posterior of $\boldsymbol{\theta}$, Module 3 also calculates the posterior covariance of $\boldsymbol{\theta}$ for each MC replicate.

4.1.2 Step 2: Monte Carlo (MC) Loop (for $i = 1, \dots, N_{mc}$)

In order to calculate the preposterior covariance, a Monte Carlo (MC) sampling strategy is applied to generate N_{mc} replicates of hypothetical experimental response data based on the information calculated or specified in Step 1. For each replicate, after generating the hypothetical experimental data, the multi-response modular Bayesian approach is applied to estimate the hyperparameters of the discrepancy function MRGP models and to calculate the hypothetical posterior covariance for this replicate. For the i th MC replicate, the following steps are involved. The superscript (i) added to a quantity indicates that it is for the i th MC replicate.

(a) Generate a simulated set of experimental data $\mathbf{Y}^{e(i)}$.

A realization $\boldsymbol{\theta}^{(i)}$ of the calibration parameters is first generated from its prior distribution specified in Step 1. The generated $\boldsymbol{\theta}^{(i)}$ will be treated as a “true” value for $\boldsymbol{\theta}$ when generating the hypothetical physical experimental response observations on the i th MC replicate.

Next, using the MRGP model for the computer simulation obtained in Step 1, a realization of the computer simulation response values is generated at parameter values $\boldsymbol{\theta}^{(i)}$ and input settings \mathbf{X}^e , denoted by $\mathbf{Y}^{m(i)} = \hat{\mathbf{y}}^m(\mathbf{X}^e, \boldsymbol{\theta}^{(i)}) = [\hat{y}_1^{m(i)}, \dots, \hat{y}_q^{m(i)}]$ where $\hat{y}_j^{m(i)} = [\hat{y}_j^{m(i)}(\mathbf{x}_1^e, \boldsymbol{\theta}^{(i)}), \dots, \hat{y}_j^{m(i)}(\mathbf{x}_{N_e}^e, \boldsymbol{\theta}^{(i)})]^T$ ($j = 1, \dots, q$). Here, $\mathbf{Y}^{m(i)}$ is generated from a multivariate normal distribution whose mean vector and covariance matrix are determined by the MRGP model. The hat notation “ $\hat{\cdot}$ ” over $\mathbf{y}^m(\mathbf{X}^e, \boldsymbol{\theta}^{(i)})$ denotes that we are drawing interpolated data from the MRGP model (instead of the original model) of \mathbf{y}^m .

Subsequently, using the priors for $\boldsymbol{\phi}^\delta \setminus \boldsymbol{\lambda} = \{\mathbf{B}^\delta, \boldsymbol{\Sigma}^\delta, \boldsymbol{\omega}^\delta\}$ specified in Step 1c and the MRGP model of Eq. (4.10), a realization for the discrepancy functions is generated at the design settings \mathbf{X}^e , denoted by $\boldsymbol{\Delta}^{(i)} = [\boldsymbol{\delta}_1^{(i)}, \dots, \boldsymbol{\delta}_q^{(i)}]$ where $\boldsymbol{\delta}_j^{(i)}$ ($j = 1, \dots, q$) represents the realization of discrepancy function for the i th response at \mathbf{X}^e . Similar to generating $\mathbf{Y}^{m(i)}$, a multivariate normal distribution is used to generate $\boldsymbol{\Delta}^{(i)}$, but with mean vector and covariance matrix determined by the MRGP model for the discrepancy functions. Specifically, the mean is $\mathbf{H}^\delta \mathbf{B}^{\delta(i)}$, where $\mathbf{H}^\delta = [\mathbf{h}^\delta(\mathbf{x}_1^e)^T, \dots, \mathbf{h}^\delta(\mathbf{x}_{N_e}^e)^T]^T$ with $\mathbf{h}^\delta(\mathbf{x})$ some specified regression basis functions (in the examples, constant $\mathbf{h}^\delta(\mathbf{x}) = 1$ will be used), and the covariance is the $\boldsymbol{\Sigma}^{\delta(i)} \otimes \mathbf{R}^\delta$, where \mathbf{R}^δ is the $N_e \times N_e$ matrix with k th-row, j th-column entry equal to the prior correlation between $\delta(\mathbf{x}_k^e)$ and $\delta(\mathbf{x}_j^e)$ using parameters $\boldsymbol{\omega}^{\delta(i)}$.

Finally, a realization $\mathbf{E}^{(i)}$ of the observation errors is generated at the design settings \mathbf{X}^e , assuming that the experimental error ε_j ($j = 1, \dots, q$) follows *i.i.d.* normal distributions with mean 0 and variance λ_j specified in Step 1c. $\mathbf{E}^{(i)}$ is an $N_e \times q$ matrix whose uth -row, vth -column entry is the realization of observation error for the v th response at the uth design setting \mathbf{x}_u^e . Based on Eq. (4.7), the realization $\mathbf{Y}^{e(i)}$ of the simulated experimental responses at the design settings \mathbf{X}^e is calculated via:

$$\mathbf{Y}^{e(i)} = \mathbf{Y}^{m(i)} + \boldsymbol{\Delta}^{(i)} + \mathbf{E}^{(i)}. \quad (4.13)$$

(b) Based on $\mathbf{Y}^{e(i)}$, estimate the hyperparameters of the discrepancy function MRGP $\boldsymbol{\delta}^{(i)}(\mathbf{x})$ for the i th MC replicate.

This is a direct application of Module 2 of the multi-response modular Bayesian approach described in Sect. 3.1.2, but with an actual set of experimental data \mathbf{Y}^e replaced by the hypothetical set $\mathbf{Y}^{e(i)}$ generated on the i th MC replicate. Specifically, Module 2 estimates the hyperparameters of the MRGP model for the discrepancy functions for the i th MC replicate using MLE methods based

on combining the simulation data \mathbf{Y}^m obtained in Step 1 and the hypothetical experimental data $\mathbf{Y}^{e(i)}$ generated in Step 2a for the i th MC replicate.

(c) Calculate the sample posterior covariance of $\boldsymbol{\theta}$ for replicate i .

This is also a direct application of Module 3 of the multi-response modular Bayesian approach described in Sect. 3.1.3, but with an actual set of experimental data \mathbf{Y}^e replaced by the hypothetical set $\mathbf{Y}^{e(i)}$ generated on the i th MC replicate. Specifically, Module 3 calculates the sample posterior covariance $\text{Cov}^{(i)}[\boldsymbol{\theta} | \mathbf{Y}^m, \mathbf{Y}^{e(i)}]$ of the posterior distribution of $\boldsymbol{\theta}$ in Eq. (4.12). For low-dimensional $\boldsymbol{\theta}$, Legendre-Gauss quadrature [48, 49] can be used to calculate the posterior covariance. For higher-dimensional $\boldsymbol{\theta}$, other numerical approaches such as Markov chain Monte Carlo (MCMC) [50] or sampling-resampling [51] methods can be used.

4.1.3 Step 3: Calculate the Preposterior Covariance as Average of the N_{mc} Posterior Covariances from Step 2

Iterating through the N_{mc} MC replicates in Step 2 produces $\{\text{Cov}^{(i)}[\boldsymbol{\theta} | \mathbf{Y}^m, \mathbf{Y}^{e(i)}] : i = 1, 2, \dots, N_{mc}\}$. The preposterior covariance matrix of the calibration parameters is defined as $\mathbb{E}[\text{Cov}^{(i)}[\boldsymbol{\theta} | \mathbf{Y}^m, \mathbf{Y}^e] | \mathbf{Y}^m]$, where the outer expectation is with respect to the actual, yet-to-be-observed experimental response data \mathbf{Y}^e , conditioned on the already-observed simulation data \mathbf{Y}^m . As an estimate of the preposterior covariance matrix, we use:

$$\tilde{\Sigma}_{\boldsymbol{\theta}} \text{ (or } \tilde{\sigma}_{\theta}^2 \text{ for a scalar } \theta\text{)} = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \{\text{Cov}^{(i)}[\boldsymbol{\theta} | \mathbf{Y}^m, \mathbf{Y}^{e(i)}]\}. \quad (4.14)$$

4.1.4 Step 4: Based on the Preposterior Covariances of All Subsets of Responses, Select a Subset to Measure Experimentally

In Bayesian analyses, the posterior covariance constitutes a standard quantification of parameter identifiability [14, 15]. The preposterior covariance serves as a prediction of the posterior covariance that is expected to result after conducting the physical experiment based on the knowledge obtained from the observed simulation response surface. Consequently, the preposterior covariance is used to guide the selection of the subset of responses to measure experimentally that has the most potential to enhance identifiability. For a simple case with a single calibration parameter, the subset of responses that leads to the smallest preposterior variance (tightest preposterior distribution) would be deemed as the most likely to achieve good identifiability. For cases with multiple calibration parameters, scalar metrics of the preposterior covariance matrix, such as its trace, determinant, maximum eigenvalue, etc., can be used to determine which subset yields the tightest preposterior distribution and, subsequently, which subset of responses to measure experimentally.

4.2 A Modified Algorithm for Investigating the Behavior of the Preposterior Analysis

To investigate the behavior of the preposterior analysis, consider the following modification to the algorithm of Fig. 4.14 [41]: In Step 2a-i, instead of generating a different $\theta^{(i)}$ randomly from its prior distribution on each MC replicate, the same fixed value (denoted by θ^t , a value specified by the user outside the MC loop) is used for all replicates. This algorithm is referred to as the *fixed- θ preposterior analysis*, and the resulting estimate of the preposterior covariance matrix is denoted by $\tilde{\Sigma}_{\theta}(\theta^t)$. Recall that $\theta^{(i)}$ in the Fig. 4.14 algorithm is only used when generating the hypothetical experimental observations $\mathbf{Y}^{e(i)}$, for which $\theta^{(i)}$ is treated as the true values of the calibration parameters. In reality there is only a single true value θ^* that the laws of nature will dictate when generating the actual experimental data \mathbf{Y}^e , based on which the actual posterior covariance of θ will be calculated. In light of this, one might wonder how well the preposterior analysis of Fig. 4.14 [which has no knowledge of the true θ^* when generating $\mathbf{y}^{e(i)}$ and must, therefore, average the results over values of $\theta^{(i)}$ drawn from the prior $p(\theta)$] can predict the actual posterior covariance matrix. In the ideal situation that $\tilde{\Sigma}_{\theta}(\theta^t)$ does not depend on θ^t , we might expect the preposterior covariance to provide a good prediction of the posterior covariance. In Sects. 4.4, the fixed- θ preposterior analysis is used to investigate the extent to which $\tilde{\Sigma}_{\theta}(\theta^t)$ depends on θ^t in the examples considered.

Although it may seem counterintuitive that $\tilde{\Sigma}_{\theta}(\theta^t)$ might not depend (strongly) on θ^t , this is precisely the situation in the standard linear regression formulation $\mathbf{y} = \mathbf{X}\theta + \text{error}$, where \mathbf{X} is the design matrix, and \mathbf{y} is the observation vector. In the linear regression formulation, under mild assumptions, the covariance matrix of the regression estimate of θ is $[\mathbf{X}^T \mathbf{X}]^{-1}$ multiplied by the error variance, which is independent of the true θ . This characteristic allows one to design experiments, prior to observing \mathbf{y} , that are “optimal” regardless of the true θ . The situation is more complex for the calibration problem, in part because of the black-box, potentially nonlinear dependence of $y^m(\mathbf{x}, \theta)$ on θ . However, in the next section, it is demonstrated that the relative (when comparing different experimental designs) dependence of $\tilde{\Sigma}_{\theta}(\theta^t)$ on θ^t in the examples is mild enough that the preposterior covariance still allows one to distinguish good experimental designs from poor ones.

Notice that the fixed- θ preposterior covariance $\tilde{\Sigma}_{\theta}(\theta^t)$ is related to the preposterior covariance $\tilde{\Sigma}_{\theta}$ via:

$$\tilde{\Sigma}_{\theta} = \int \tilde{\Sigma}_{\theta}(\theta^t) p(\theta^t) d\theta^t, \quad (4.15)$$

where $p(\cdot)$ is the prior distribution of θ specified in Step 1c-iii. In other words, the preposterior covariance is the expected value of the fixed- θ preposterior covariance with respect to the prior of θ .

4.3 Fisher Information-Based Surrogate Preposterior Analysis

Clearly, the proposed multi-response preposterior analysis is very computationally intensive. For a system with N responses, there are $N(N - 1)/2$ combinations of two responses and 2^N total combinations of any subset of the N responses. Even for a single subset of responses, the computational cost can be substantial. The MC strategy requires a large number N_{mc} of replicates, and for each MC replicate, Modules 2 and 3 (which themselves involve a MC simulation or numerical integration) of the modular Bayesian approach must be implemented. Therefore, to make the preposterior analysis feasible for engineering applications with many system responses, a more computationally efficient surrogate preposterior analysis is developed in [42, 43] that can be used to eliminate the responses that are unlikely to lead to good identifiability, thereby substantially reducing the number of response combinations that must be included in the preposterior analysis.

For the r -dimensional calibration parameter vector θ , the *observed Fisher information* $\mathcal{I}(\theta)$ is a matrix whose uth -row, vth -column entry is the negative second-order derivative of the log-likelihood function:

$$[\mathcal{I}(\theta)]_{u,v} = -\frac{\partial^2}{\partial \theta_u \partial \theta_v} \log p(\mathbf{Y}^m, \mathbf{Y}^e | \theta, \hat{\phi}) \quad (u, v = 1, 2, \dots, r), \quad (4.16)$$

where $p(\mathbf{Y}^m, \mathbf{Y}^e | \theta, \hat{\phi})$ is the likelihood function for the simulation and experimental data together, as in Eq. (4.12), and θ represents the true parameters. It measures the amount of information that yet-to-be-collected experimental data \mathbf{Y}^e , together with the already-observed \mathbf{Y}^m , provides about calibration parameter θ . It should be noted that θ in Eq. (4.16) is meant to be the true values of the calibration parameters. Hence, in Eq. (4.16) the distribution of the computer response \mathbf{Y}^m (which is known by itself does not depend on θ). Only the distribution of \mathbf{Y}^e depends on θ .

The Fisher-like surrogate preposterior criterion that will be introduced shortly in this section, which is used only for reducing the number of response combination that must be considered, is a modified version of Eq. (4.16). To handle the complication that the yet-to-be-collected experimental data \mathbf{Y}^e are unknown in Eq. (4.16), the simple substitution $\hat{\mathbf{Y}}^m(\mathbf{X}^e, \theta)$ for \mathbf{Y}^e is made. Here, $\hat{\mathbf{y}}^m(\mathbf{X}^e, \theta)$ represents the predicted value of \mathbf{Y}^e via interpolating the data from the MRGP model of \mathbf{y}^m . After generating this fictitious realization of \mathbf{Y}^e , the modular Bayesian approach (Step 2b of the flowchart in Fig. 4.14) is used to estimate the hyperparameters $\hat{\phi}$ for substitution into Eq. (4.16). Because \mathbf{Y}^e is replaced by its prediction, this will tend to result in a small estimated variance parameters for the discrepancy function, which will naturally result in some level of underestimation of the identifiability in θ . However, this surrogate procedure is only used for *relative* ranking of the identifiability that results from the various combinations of responses, and the results (See Sect. 4.6) indicate that the surrogate analysis does a reasonable job of

preserving the relative ranking. The advantage of this approach is computational – instead of generating multiple realizations of \mathbf{Y}^e , only a single realization $\hat{\mathbf{y}}^m(\mathbf{X}^e, \boldsymbol{\theta})$ is generated.

There is an additional complication. The $\boldsymbol{\theta}$ in Eq. (4.16) represents the true parameters, and these are unknown. To handle this, Eq. (4.16) is replaced by the “averaged” observed Fisher information matrix, averaging (4.16) with respect to the prior distribution $p(\boldsymbol{\theta})$ of $\boldsymbol{\theta}$. To calculate this, Monte Carlo simulation is used, as outlined in Steps 2 and 3 of the flowchart in Fig. 4.15. Specifically, N'_{mc} realizations $\boldsymbol{\theta}^{(i)} (i = 1, 2, \dots, N'_{mc})$ are drawn from $p(\boldsymbol{\theta})$ and take the u th-row, v th-column entry ($u, v = 1, 2, \dots, r$) of our averaged observed Fisher information matrix to be:

$$[\hat{\mathcal{I}}]_{u,v} = \frac{1}{N'_{mc}} \sum_{i=1}^{N'_{mc}} \left\{ -\frac{\partial^2}{\partial \theta_u \partial \theta_v} \log p \left(\mathbf{Y}^m, \hat{\mathbf{y}}^m(\mathbf{X}^e, \boldsymbol{\theta}^{(i)}) | \boldsymbol{\theta}, \hat{\boldsymbol{\phi}}^{(i)} \right) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}} \right\}, \quad (4.17)$$

where $\hat{\boldsymbol{\phi}}^{(i)}$ denotes the values of $\boldsymbol{\phi}$ estimated in the modular Bayesian algorithm on the i th MC replicate. Various scalar metrics of $\hat{\mathcal{I}}$ (e.g., trace, determinant, maximum eigenvalue, etc.) can be used as the surrogate measure of identifiability. For a simple case with a single calibration parameter $\boldsymbol{\theta}$, $\hat{\mathcal{I}}$ is a scalar. The larger $\hat{\mathcal{I}}$ is, the more information there is about $\boldsymbol{\theta}$ from \mathbf{Y}^m and \mathbf{Y}^e , and the more likely it is to achieve good identifiability.

After calculating $\hat{\mathcal{I}}$ from Eq. (4.17), the top N_c subsets of responses (based on one of the scalar measures of identifiability extracted from $\hat{\mathcal{I}}$) are chosen to be further analyzed in the full preposterior analysis. Figure 4.15 is a flowchart of the entire Fisher information-based surrogate preposterior procedure.

The surrogate analysis is clearly much more efficient than the full preposterior analysis. The number of MC replicates N'_{mc} can be significantly smaller than what is required in the preposterior analysis (N_{mc}), since the MC sampling now is only with respect to $\boldsymbol{\theta}$. In addition, and more importantly, the calculations in each MC replicate are much simpler. Within each MC replicate, neither an inner level MC simulation nor numerical integration is needed. The main computation within each MC replicate is to calculate the MLEs of the hyperparameters $\hat{\boldsymbol{\phi}}^{(i)}$ and then to evaluate the likelihood at a few discrete values in the neighborhood of $\boldsymbol{\theta}^{(i)}$ to calculate the second-order derivative in Eq. (4.17) numerically.

4.4 Single-Response Case Study: Simply Supported Beam Example

In this section, the same beam example (see Sects. 2.2 and 3.2) illustrates how the preposterior analysis can be used, after conducting the computer simulations but before conducting the physical experiments, to predict the identifiability that will

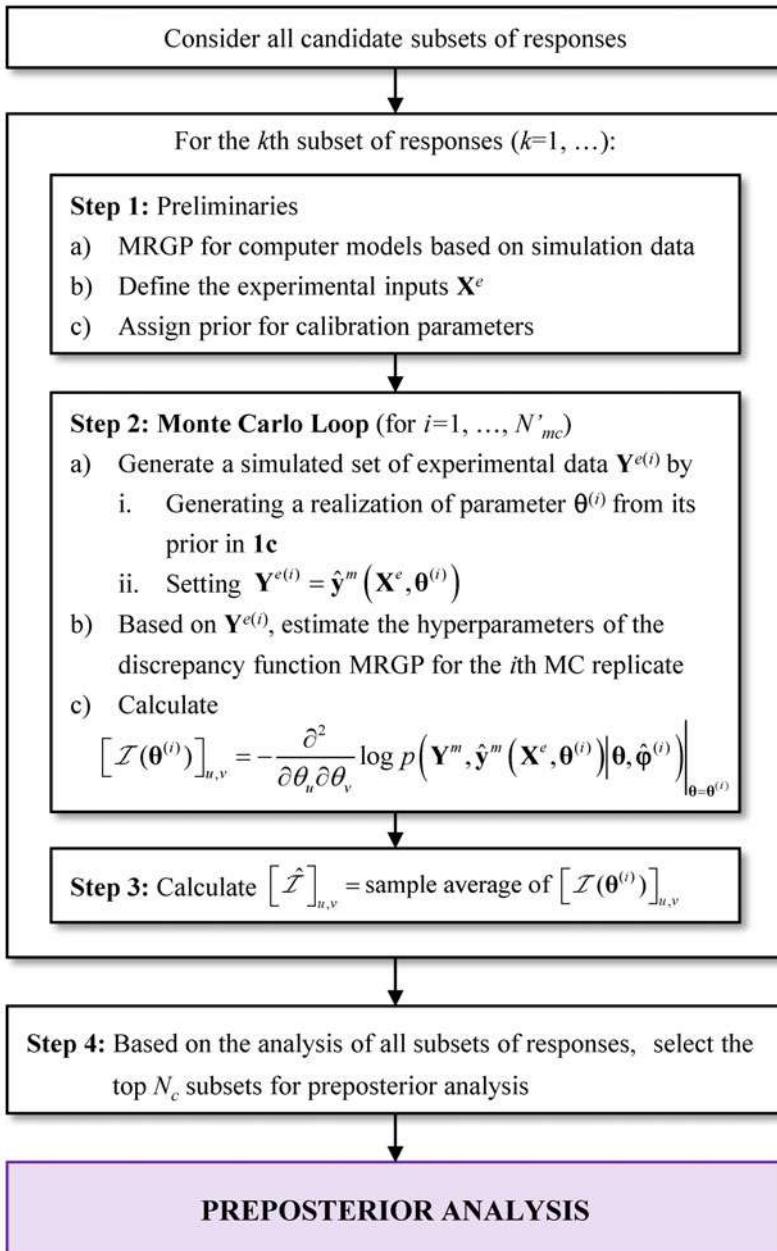


Fig. 4.15 Flowchart of the surrogate preposterior analysis

result from a proposed experimental design. The system in this example is inherently difficult to identify with a single response even with a large amount of observed physical experiments.

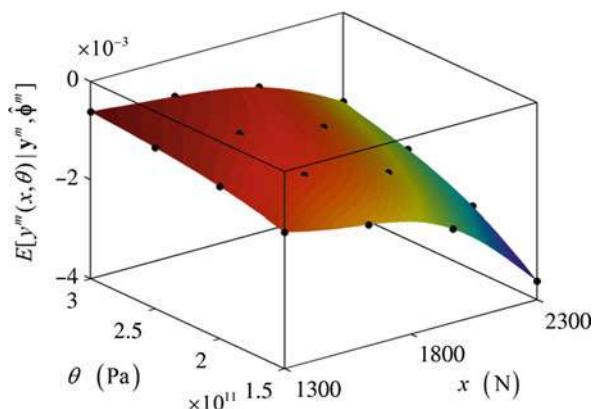
4.4.1 Computer Model, Physical Experiments, and Preliminaries

The design variable x is the magnitude of a static force applied to the midpoint of the beam, and the response y is the strain at the midpoint of the beam (at the top of the cross-section). The calibration parameter θ is the Young's modulus. For Step 1a, the simulations were observed on a 4×4 evenly spaced grid ($N_m = 16$) over the input space ($1300 \leq x \leq 2300$ N and $150 \leq \theta = 300$ GPa). The simulation response observations \mathbf{y}^m and the posterior mean of $y^m(x, \theta)$ from Step 1a are shown in Fig. 4.16. Because of the smooth nature of $y^m(x, \theta)$, there was very small posterior uncertainty in $y^m(x, \theta)$ over the entire input space.

Regarding Step 1b, different analyses will be conducted for different values of N_e . For each N_e , \mathbf{X}^e is evenly spaced over the input range $1300 \leq x \leq 2300$. For purposes of comparing the preposterior covariance to the posterior covariance, actual "physical" experimental data were generated via the same FEA model but using a more elaborate material law that results in a discrepancy between the computer model and the experimental data. Observation error with variance $\lambda = 6.63 \times 10^{-12}$ m² was added to the experimental data, and the true value of the Young's modulus calibration parameter was taken to be $\theta^* = 206.8$ GPa. These physical experimental data are not used in the preposterior algorithm of Fig. 4.14 and are used only to verify the results of the preposterior algorithm.

For Step 1c, $\phi^\delta \setminus \lambda$ is assigned a point mass prior with mass at $\{\boldsymbol{\beta}^\delta, \sigma_\delta^2, \boldsymbol{\omega}^\delta\} = \{0.00, 1.11 \times 10^{-7}, 0.002\}$. The value 1.11×10^{-7} m² for σ_δ^2 was chosen so that the bounds of the 99.7% prediction interval of the discrepancy function were approximately $[-1, 1] \times 10^{-3}$ m. The correlation parameters $\boldsymbol{\omega}^\delta$ were chosen to represent a relatively smooth GP model for the discrepancy function. λ is also assigned a point mass prior with mass at 6.63×10^{-12} m². Recall that the prior distributions for $\phi^\delta \setminus \lambda$ and λ are used only in Step 2a to generate random realizations of $\mathbf{y}^{e(i)}$. In general, the MLEs of $\phi^\delta \setminus \lambda$ and λ are calculated in Step 2b. For this example, however, λ is treated as a known value and is not estimated in Step 2b. This is reasonable in systems for which the random error variance can be estimated externally, simply by taking replicate experimental measurements at the same input

Fig. 4.16 Posterior mean of the computer model GP for the beam example. The bullets indicate \mathbf{y}^m



settings. Lastly, $p(\theta)$ was chosen to be a non-informative uniform distribution over the full range of θ .

4.4.2 Comparing the Preposterior and Posterior Covariances

We begin by comparing the fixed- θ preposterior covariance $\tilde{\Sigma}_\theta(\theta^t)$ for various θ^t to the actual posterior covariance based on the experimental data from the more elaborate FEA model. As described in Sect. 4.2, for each θ^t , $\tilde{\Sigma}_\theta(\theta^t)$ was calculated using the algorithm of Fig. 4.14 with $\theta^{(i)}$ from Step 2a-i replaced by the same value θ^t for all N_{mc} replicates. Figure 4.17 plots the fixed- θ preposterior standard deviation (STD) [i.e., the square root of $\tilde{\Sigma}_\theta(\theta^t)$] versus θ^t for two different experimental designs with $N_e = 3$ and $N_e = 8$. The individual points plotted at $\theta = 206.8 and represented by the shaded circle (for the design with $N_e = 3$) and shaded square (for the design with $N_e = 8$) are the actual posterior STDs of θ that result by applying Modules 1–3 of the standard modular Bayesian approach to a set of experimental data generated using the true value $\theta^* = 206.8. These individual points are included in Fig. 4.17 for purposes of comparing to the preposterior STDs that our algorithm calculates, which are represented by the series of connected points. In Fig. 4.17 both the fixed- θ preposterior STD and the posterior STD have been normalized (divided) by the STD of the prior $p(\theta)$.$$

Although the fixed- θ preposterior STDs differ from the posterior STD, their relative change as N_e increases is comparable. In particular, the relative difference in the fixed- θ preposterior covariance for the design with $N_e = 3$ and the design with $N_e = 8$ is reasonably independent of the fixed θ^t , and this relative difference is quite consistent with the relative difference in the actual posterior covariance for the

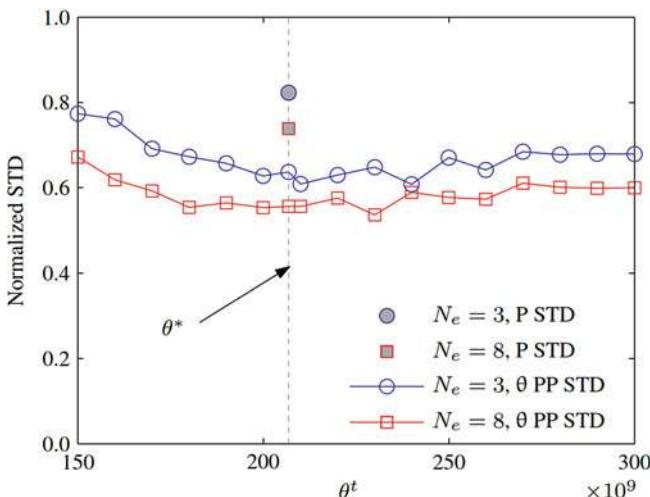
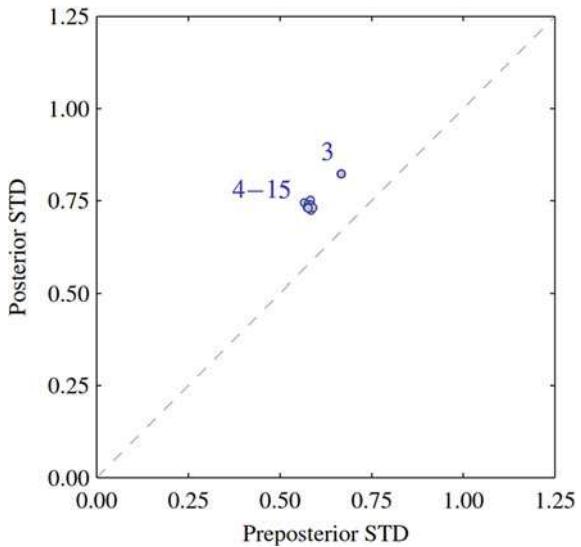


Fig. 4.17 Plot of the fixed- θ preposterior STD and the posterior STD (both normalized by the prior STD of θ) versus θ^t for the beam example. “ θ PP STD” is the fixed- θ preposterior STD and “P STD” is the posterior STD

Fig. 4.18 Plot of the preposterior STD versus the posterior STD (both normalized by the prior STD of θ) for the beam example. The numbers indicate N_e for each case



true $\theta^* = 206.8$ GPa. Consequently, the relative difference in the preposterior STD [which was calculated as the integral of the fixed- θ preposterior STD in Fig. 4.17, with respect to $p(\theta)$] accurately reflects the relative difference in the actual posterior STD in this example. Specifically, when N_e is increased from 3 to 8 in Fig. 4.17, the posterior STD decreases by 10.3% (from 0.8231 to 0.7383), and the preposterior STD decreases by 12.5% (from 0.6676 to 0.5840). This correctly indicates that the experimental design with $N_e = 8$ will result in somewhat better identifiability than the experimental design with $N_e = 3$, as measured by the posterior STD that would result after the experiments are conducted.

The preposterior STD was then calculated from the fixed- θ preposterior STDs in Fig. 4.17, as described in Sect. 4.2. The analyses were repeated for various N_e , and Fig. 4.18 shows a scatter plot of the preposterior STD versus the posterior STD (each normalized by prior STD θ) for $N_e = 3, 4, \dots, 15$. The scatter plot conveys a number of interesting characteristics of the beam example. Although the preposterior STD slightly underestimates the posterior STD for the different designs, the trend is the same: When N_e increases from 3 to 4, the preposterior STD correctly predicts that the posterior STD will decrease. Then, as N_e increases from 4 through 15, the preposterior STD again correctly predicts that there will be negligible further decrease in the posterior STD. The overall conclusion that a user might draw from this preposterior analysis is that the system inherently suffers from a lack of identifiability, because an increase in N_e beyond 4 results in almost no further improvement in the preposterior STD. Sections 2 and 3 discussed the reasons behind the lack of identifiability in detail and also demonstrated that identifiability is greatly improved when multiple responses are measured experimentally. In the next subsection, a second example will be in which the identifiability of the system continues to improve as experimental data are added.

4.5 Using the Preposterior Analysis to Select N_e

One way in which the preposterior analysis can be used is to help the user choose the number N_e of experimental runs [41]. Figure 4.19, which plots the normalized preposterior STD versus N_e for the beam example, illustrates how this might be accomplished. One might conclude that increasing N_e beyond 4 will provide little further knowledge of θ^* and that the beam system is inherently difficult to identify with only the single experimentally measured response (which was strain at the beam midpoint).

The beam example had only a single input variable x and used uniformly spaced grid designs. We now consider an example that uses Latin hypercube designs with two input variable and a calibration parameter. Suppose the computer model is:

$$y^m(x_1, x_2, \theta) = \sin(\theta x_1) + x_1 x_2, \\ x_1 \in [0, 1], \quad x_2 \in [0, 1], \quad \theta \in [0, 4],$$

and the physical experimental data are generated from the model:

$$y^e(x_1, x_2) = y^m(x_1, x_2, \theta^*) + \delta(x_1, x_2) + \varepsilon, \\ \varepsilon \sim \mathcal{N}(0, 0.01^2).$$

We began with a Latin hypercube design for the computer experiment with 40 points in three dimensions $\{x_1, x_2, \theta\}$. An optimal Latin hypercube design based on the maximin criterion was used. A Gaussian process model was fit to the computer experiment data via MLE in Step 1a of Fig. 4.14. For the physical experimental

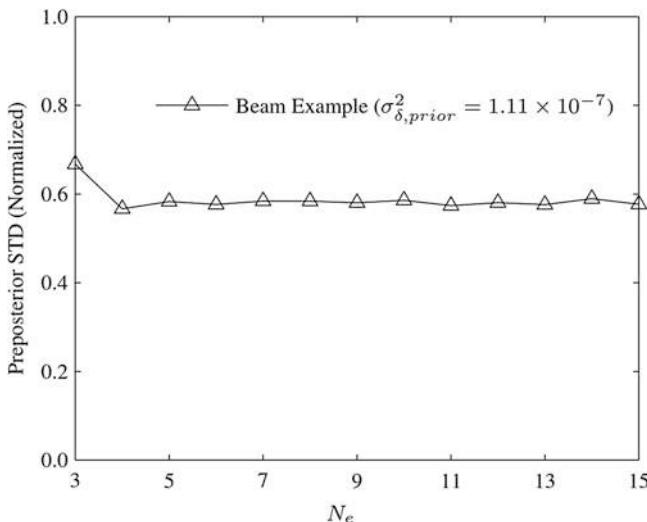


Fig. 4.19 Preposterior STD (normalized) versus N_e for the beam example

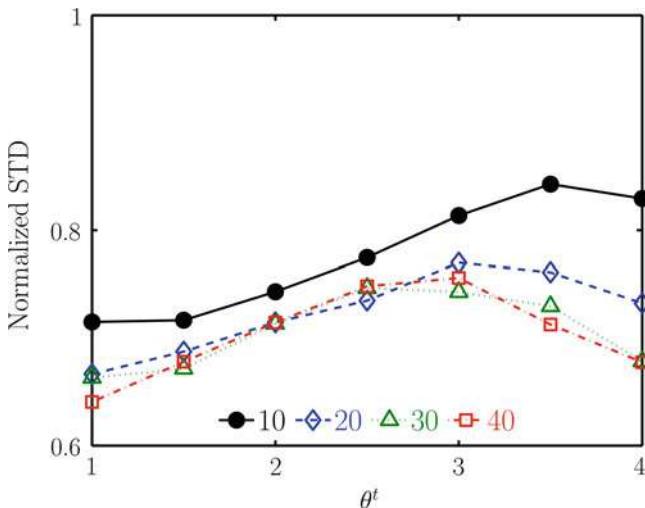


Fig. 4.20 Fixed- θ preposterior STD (normalized by the prior STD of θ) versus θ^t for Latin hypercube designs of various size N_e (N_e is specified in the legend)

design specified in Step 1b, maximin Latin hypercube designs of various size $N_e = \{10, 20, 30, 40, 50\}$ were used over two dimensions $\{x_1, x_2\}$. Because of the random nature of the Latin hypercube designs, none of the physical experimental designs had \mathbf{x} settings that coincided with the computer experiment \mathbf{x} settings. Next, in Step 1c-i, the GP hyperparameters of the discrepancy function were set to be $\beta^\delta = 0$, $\omega^\delta = 3$, and $\sigma_\delta = 0.75$, which corresponds to assigning them point mass priors. The relatively large value of σ_δ results in a variety of relatively large discrepancy functions being generated within Step 2 of the preposterior algorithm. In Step 1c-ii, $\lambda = 0.01^2$ was specified. Lastly, in Step 1c-iii, the prior for θ was specified to be uniform over the entire range $[0, 4]$.

From the preposterior analysis, Fig. 4.20 shows the resulting fixed- θ preposterior standard deviation versus θ^t for the designs with different N_e , and Fig. 4.21 shows the (normalized) preposterior standard deviation as a function of N_e . From Fig. 4.21, the preposterior standard deviation does not substantially decrease beyond $N_e = 30$ roughly.

4.6 Multi-response Case Study: Simply Supported Beam Example

In this section, the simply supported beam example is once again employed, to demonstrate the effectiveness of the preposterior and surrogate preposterior analyses approach under the multi-response scenario. The physical meanings of the six output

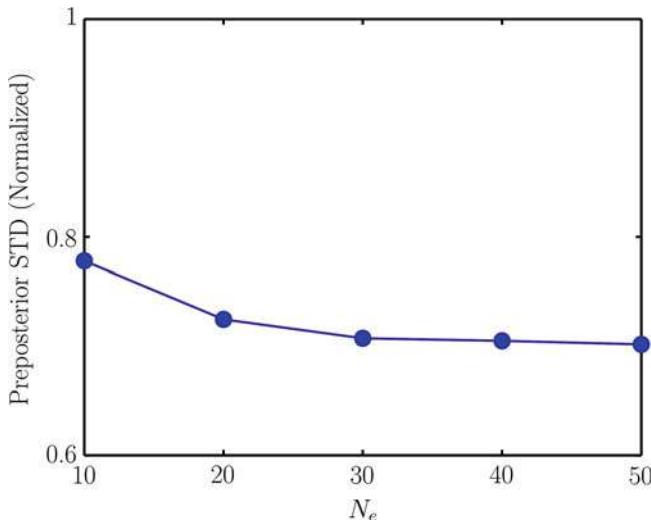


Fig. 4.21 Preposterior STD (normalized by the prior STD of θ) versus N_e for the Latin hypercube designs

variables y_i ($i = 1, \dots, 6$) that are calculated in the computer simulation are listed in Table 4.3. They are considered as candidates to measure experimentally.

As in Sects. 2.2 and 3.2, it turns out that the identifiability of this system is rather poor using any one of the six responses. On the other hand, using two responses together may potentially enhance identifiability. Section 3.2 provided detailed results of posterior analysis on each pair of the six responses and demonstrates the enhancement of identifiability. However, it did not provide a strategy for predicting which pair of responses will best enhance identifiability prior to conducting the physical experiment.

In this section, identifiability is predicted prior to collecting the physical experimental data, although the simulation data for all six responses are available on a 4×4 grid over the (x, θ) space with a range of $1300 \leq x \leq 2300$ N and $150 \leq \theta \leq 300$ GPa. The objective is to select which two out of six responses to measure experimentally, in order to best enhance identifiability. The preposterior and surrogate preposterior analyses using the simulation data are used to predict the degree of identifiability. The candidate subsets of responses that are considered are all 15 $\{y_i, y_j : i, j = 1, \dots, 6, i < j\}$. In the following discussion, the candidate subset $\{y_4, y_5\}$ is used to demonstrate the main procedures of the two analyses.

In the preliminary step of both preposterior and surrogate preposterior analyses, a MRGP model is built for $\{y_4, y_5\}$ from the simulation data. Figure 4.22 plots the predicted response surfaces from the MRGP model, which are quite close to the true simulation response surface, because the latter is relatively smooth in this example. The experimental settings \mathbf{X}^e for the input variable are defined as 11 points uniformly spaced over the design region, i.e., $\mathbf{X}^e = [1300, 1400, \dots, 2200, 2300]$ N.

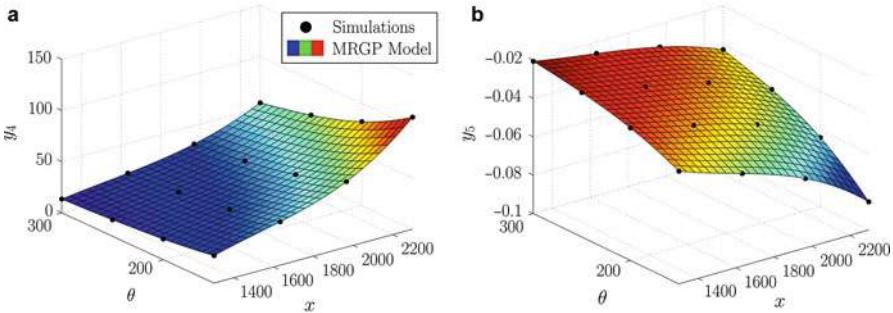


Fig. 4.22 The MRGP model for (a) y_4 and (b) y_5

The prior for the calibration parameter θ is a uniform distribution over $[150, 300]$ GPa. Additionally for the preposterior analysis, point mass priors are assigned to the hyperparameters of the discrepancy functions with masses at $\mathbf{B}^\delta = \mathbf{0}$, $\Sigma^\delta = \text{diag}(1.11 \times 10^{-7}, 1.11 \times 10^{-7})$, and $\omega^\delta = 2$. The experimental errors are assigned independent normal distributions: $\varepsilon_4 \sim \mathcal{N}(0, 0.0070)$ and $\varepsilon_5 \sim \mathcal{N}(0, 2.943 \times 10^{-9})$.

In the MC loop of the preposterior analysis, there are a total of $N_{mc} = 1600$ MC replicates for this specific subset of responses. Within the i th MC replicate, a realization of the simulation response $\mathbf{Y}^{m(i)}$ at the input settings \mathbf{X}^e is generated from the MRGP model fitted in Step 1a (the mean of which is shown in Fig. 4.22), a realization of the model discrepancy $\Delta^{(i)}$ is generated based on the specified prior for the hyperparameters, and a realization of the observational error $\mathbf{E}^{(i)}$ is generated based on the distributions of ε_4 and ε_5 . Based on this, a realization of experimental data $\mathbf{Y}^{e(i)}$ is calculated via Step 2a-iv and can be used for the multi-response modular Bayesian approach to calculate the sample posterior variance for the i th MC replicate. Assuming a uniform prior for θ over the range $[150, 300]$ GPa, the preposterior variance is estimated as the average of all 1600 sample posterior variances, which is $\tilde{\sigma}_\theta^2 = 2.0054 \text{ GPa}^2$.

In the MC loop of the surrogate preposterior analysis, N'_{mc} equally spaced values of θ are considered. Within the i th MC replicate, a realization of the simulation $\mathbf{Y}^{m(i)}$ is generated, as in the preposterior analysis. However, without generating $\Delta^{(i)}$ and $\mathbf{E}^{(i)}$, we directly take $\mathbf{Y}^{e(i)} = \mathbf{Y}^{m(i)}$ and apply the multi-response modular Bayesian approach to calculate the negative second-order derivative of log-likelihood function (as the term inside the brackets in Eq. (4.17)). The results for all 16 values of θ are plotted in Fig. 4.23b. The Fisher information-based scalar predictor \hat{I} , taken to be the average of the 16 values, is 351.8. Because θ is scalar with a uniform prior for this example, taking the average over the 16 evenly spaced values of θ is more computationally efficient than generating random draws of θ from its prior and using Eq. (4.17).

To better illustrate the relationship between the preposterior and the surrogate preposterior analyses, the preposterior analysis is conducted in the same “fixed- θ ” manner described in the preceding paragraph for the surrogate preposterior analysis.

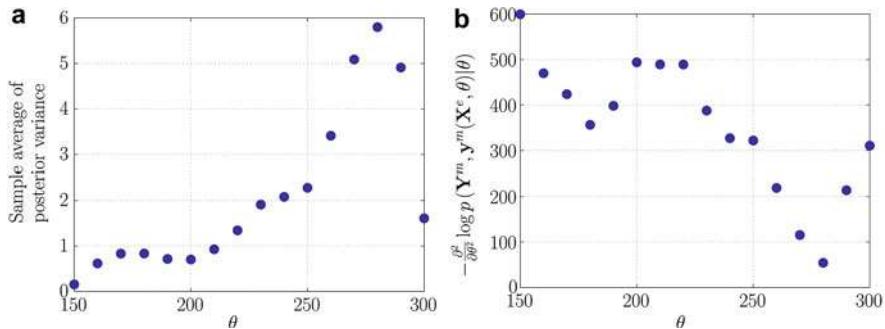


Fig. 4.23 16 equally spaced values of θ , and the corresponding (a) sample average of posterior variance (unit: GPa^2) using hypothetical data over 100 MC simulations per θ , and (b) negative second-order derivative of the log-likelihood function

That is, the 16 different values of θ equally spaced within $[150, 300]$ GPa are considered, and for each specific θ value, 100 MC replicates are used to calculate the sample posterior variance. A procedure identical to that described in Sect. 4.1, but with θ fixed over the 100 MC replicates, was used to estimate the preposterior variance (as the average of the 100 sample posterior variances over the 16 values of θ). The results are shown in Fig. 4.23a. Comparing Fig. 4.23a, b, a clear negative correlation can be seen between the preposterior variance and the Fisher information criterion. This is expected, considering that the preposterior variance is an estimate of the actual posterior variance, which is closely related to the inverse of the Fisher information; a larger value of the former generally corresponds to a larger value of the latter.

The same procedure was repeated for every other pair of responses. Table 4.6 shows the results of the preposterior and surrogate preposterior analyses. The ranks in columns 2 and 3 are based on the predicted identifiability; 1 corresponds to the smallest preposterior variance/largest $\hat{\mathcal{I}}$ and 15 to the largest preposterior variance/smallest $\hat{\mathcal{I}}$. The rankings provide us with predictions of which subsets are most likely to enhance identifiability (lower ranks) and which less likely (higher ranks). The “posterior” columns in Table 4.6, which are taken from Table 4.5, show the actual posterior covariances that resulted from each combination of measured responses after the experiments were conducted. They serve as a basis for comparison, and ideally we would like the rankings from the posterior analyses to coincide with the predicted rankings from the preposterior and surrogate preposterior analyses. It can be seen that the preposterior analysis is in good relative agreement with the posterior. Although the values of preposterior and posterior standard deviations are off by roughly a factor of 2.5, both analyses indicate that $\{y_4, y_5\}$ together lead to the best identifiability, while $\{y_1, y_2\}$ together lead to the worst identifiability. Overall, the rankings are in very close agreement. The improvement on identifiability relative to the worst case (i.e., $\{y_1, y_2\}$ for both analyses) is also calculated and provided in the table. The relative improvements are also in very close agreement. The

Table 4.6 Comparisons between posterior, preposterior, and surrogate preposterior analyses

Responses		Posterior			Preposterior			Surrogate preposterior	
y_i	y_j	σ_θ	Rank	Improvement	$\tilde{\sigma}_\theta$	Rank	Improvement	$\hat{\sigma}$	Rank
y_4	y_5	3.63	1	86.7%	1.4161	1	87.0%	351.8	1
y_4	y_6	3.90	2	85.8%	1.5224	3	86.0%	186.9	6
y_5	y_6	4.67	3	83.0%	1.6528	4	84.8%	218.2	3
y_3	y_6	4.86	4	82.2%	1.6618	5	84.7%	161.6	7
y_3	y_4	5.49	5	80.0%	1.4801	2	86.4%	273.0	2
y_3	y_5	6.27	6	77.1%	2.1117	6	80.6%	202.9	5
y_1	y_4	9.29	7	66.1%	3.0933	7	71.6%	209.6	4
y_1	y_3	10.13	8	63.0%	3.2583	8	70.1%	113.3	9
y_2	y_3	10.96	9	60.0%	3.6595	9	66.4%	93.34	10
y_2	y_5	15.42	10	43.4%	6.0765	11	44.2%	114.3	8
y_1	y_5	15.57	11	43.1%	5.5379	10	49.2%	90.92	11
y_2	y_4	18.35	12	33.9%	8.7816	12	19.4%	65.94	14
y_1	y_6	24.08	13	12.0%	9.1008	13	16.5%	81.19	12
y_2	y_6	26.46	14	3.3%	10.1850	14	6.5%	80.99	13
y_1	y_2	27.37	15	-	10.8931	15	-	47.09	15

results provided by the surrogate preposterior analysis are also in close agreement with the posterior standard deviation results, although slightly less so than the preposterior standard deviations. The top 7 pairs of responses coincide with the top 7 pairs from the actual posterior analysis. Hence, the surrogate preposterior analysis would have effectively narrowed down the candidate pairs to consider in the preposterior analysis. Considering its extremely low computational cost, the surrogate preposterior analysis is a useful enhancement to the preposterior analysis for reducing the number of response pairs to consider.

The results from three analyses in Table 4.6 are in good accordance with the underlying physics of the system. For example, the strain y_1 and the plastic strain y_2 are perfectly correlated with each other; their values are off by a constant (equal to the value of elastic strain) after plastic deformation. Therefore, their combination adds no more information about θ and enhances identifiability little beyond using either single response. In contrast, the internal energy y_4 and the midpoint displacement y_5 follow a nonlinear relationship, and thus the degree of improvement in identifiability is substantial.

It is not surprising to observe the absolute differences between the posterior variance and preposterior variances. In the MC simulations of the preposterior analysis, the hypothetical experimental data are generated based on discrepancy functions generated from their assigned prior distribution. In contrast, the posterior variance calculation is based on the single realization that is the actual discrepancy function. Consequently, the Bayesian analysis modules inside the MC loops are hypothetical and are not expected to obtain the same values of preposterior variance as the actual posterior variance. The surrogate preposterior analysis involves further

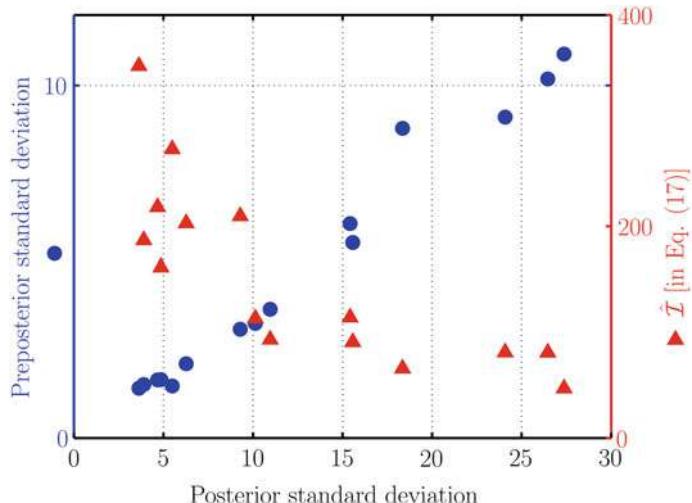


Fig. 4.24 Preposterior standard deviation and Fisher information-based identifiability predictor, versus posterior standard deviation, demonstrating very high correlation

approximations. However, it appears to accomplish its intended purpose, as the preposterior and surrogate preposterior analyses do a reasonable job of predicting the relative degree of identifiability and guide users in selecting the best set of responses to measure experimentally. Figure 4.24 illustrates this by plotting the preposterior standard deviation and the Fisher information criterion $\hat{\mathcal{I}}$ versus the actual posterior standard deviation. The preposterior standard deviation is roughly in proportion to the actual posterior standard deviation, and $\hat{\mathcal{I}}$ is negatively correlated with the posterior standard deviation, which indicates that for this case study preposterior and surrogate preposterior analyses are sufficient in predicting identifiability.

The improvement on identifiability is further illustrated in Figs. 4.25 and 4.26. While neither analyzing y_4 nor analyzing y_5 alone can provide an informative posterior distribution of θ (Fig. 4.25), uncertainty quantification considering both y_4 and y_5 provides a much tighter posterior distribution of θ , and the mean of the posterior is close to the true value of θ (Fig. 4.26a). In contrast, the subset of $\{y_1, y_2\}$ provides a dispersed posterior distribution of θ (Fig. 4.26b). The level of uncertainty is well predicted by both the preposterior and surrogate preposterior analyses.

4.7 Remarks on the Preposterior Analyses Method

Because of identifiability issues, it is often difficult to distinguish parameter uncertainty from uncertainty in the discrepancy function when using the GP-based method for calibrating computer models. In this section, it has been shown that the preposterior covariance of θ calculated via the algorithm of Fig. 4.14, and the

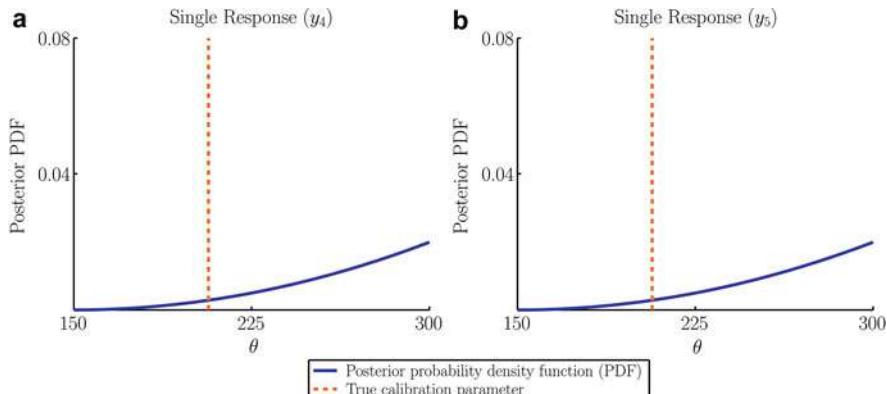


Fig. 4.25 Posterior distribution of the calibration parameter using a single response

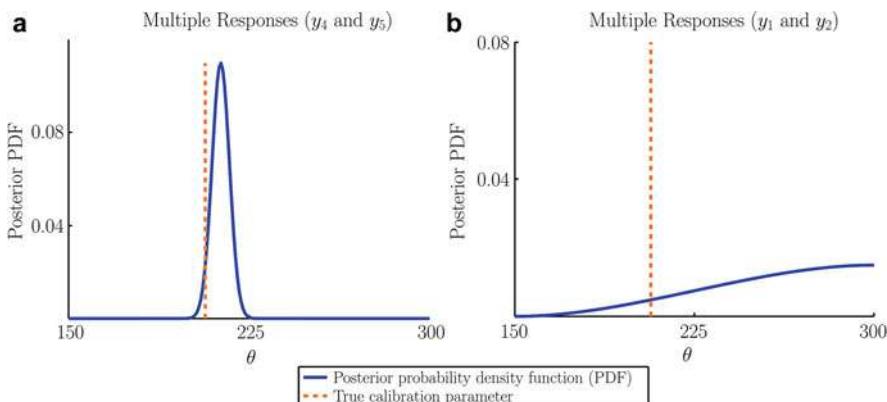


Fig. 4.26 Posterior distribution of the calibration parameter using multiple responses

Fisher information criterion $\hat{\mathcal{I}}$ calculated via the algorithm of Fig. 4.15, constitute reasonable predictions of the posterior covariance that will result when the physical experimental data are collected, at least for the examples considered. As such, the preposterior and the surrogate preposterior analyses can provide insight into the identifiability of a system and serve as criteria for helping to design an effective physical experimental design, based on the results of a previously conducted set of computer simulations.

Regarding the conditions on $\delta(\mathbf{x})$ required for reasonable identifiability, clearly there must be some, and $\delta(\mathbf{x})$ is not allowed to be any function. The assumption that $\delta(\mathbf{x})$ can be represented by a Gaussian process model is not a particularly strong one, because, for various choices of correlation function, a Gaussian process prior can produce an extremely wide variety of stochastic realizations of $\delta(\mathbf{x})$. This flexibility of the Gaussian process model is one of the reasons for its popularity in the calibration literature. As discussed in Sect. 2.3, good identifiability is more difficult

to achieve if $\delta(\mathbf{x})$ is not relatively smooth, which is reflected by the correlation parameters ω^δ (smaller ω^δ corresponds to a smoother discrepancy function). After collecting the physical experimental data, one can always estimate ω^δ to gauge the smoothness of $\delta(\mathbf{x})$. But perhaps the best and most direct measure of whether $\delta(\mathbf{x})$ is sufficiently smooth to allow identifiability is to simply look at the posterior standard deviation of θ , which directly reflects the impact of the smoothness of $\delta(\mathbf{x})$ on identifiability. However, prior to collecting the physical experimental data, it may be difficult to accurately gauge the smoothness of $\delta(\mathbf{x})$. In this case, if one specifies a specific ω^δ (or a range of ω^δ with some prior distribution assigned to it) to use in the preposterior algorithm, then the estimated preposterior covariance of θ should be interpreted as what would result if the discrepancy function truly had the assumed degree of smoothness. One should keep in mind that this complication (not knowing the actual smoothness of $\delta(\mathbf{x})$ until after the experimental data are collected, and hence not knowing how good the design will be until after the experiment is conducted) is certainly not unique to the calibration problem. In any experimental design problem, one never knows with certainty whether the design is good until after the experiment is conducted. For example, an experiment designed for estimating a linear model will end up being a very poorly designed experiment if, after conducting the experiment, it is discovered that the true response surface is really quadratic. The prevailing viewpoint in experimental design is to accept the fact that one may need to conduct a follow-up experiment based on what was observed in the initial experiment, and this same viewpoint applies to the calibration problem: If one designs a calibration experiment based on an assumed ω^δ , and, after conducting the experiment, it is discovered that the actual ω^δ is much different, then one could design and conduct a better follow-up experiment based on the improved knowledge of $\delta(\mathbf{x})$ learned from the initial experiment.

In the examples, the preposterior standard deviation has been used to compare and choose from among a small set of experiments that were designed using other methods. The examples considered evenly spaced grid designs and Latin hypercube designs of various sizes and focused on selecting the appropriate size. More generally, the algorithm in Figs. 4.14 and 4.15 could be used directly to compare any set of given designs (one would specify the input settings for each design in Step 1b). The approach could in principle also be used as a formal design optimization criterion to customize the exact values for the experimental \mathbf{x} settings, choose which of the simulation response variables to measure experimentally, etc. However, in its current form, the computational expense of the preposterior algorithm (Fig. 4.14) is prohibitive for these purposes. For the beam example (Sect. 4.4), each MC replicate (i.e., each iteration of Step 2 in Fig. 4.14) took roughly 0.018 min on average on a single-processor, 3 GHz, i7 machine. The computational expense is largely independent of the size N_e of the physical experiment if N_e is less than N_m , as will often be the case. Because Step 2 accounts for most of the computational expense of the entire algorithm, the overall expense is roughly proportional to N_{mc} . For example, using 1700 MC replicates for the beam example, the entire algorithm took roughly 31 min to calculate the preposterior standard deviation for each experimental design. Using 1700 MC replicates was more than

sufficient for this example, and no appreciable difference was found in the results when MC replicates were increased to 8500. The computational expense will also increase with the number of variables. For the example depicted in Figs. 4.20 and 4.21, each MC replicate took roughly 0.028 min on average, and this was also largely independent of N_e . In order to use the approach in a formal design optimization algorithm, one would have to calculate the preposterior standard deviation for a great many different candidate designs as the exact locations of the \mathbf{x} settings are varied. Hence, the surrogate preposterior analysis (Fig. 4.15) is needed to reduce computational expenses and to adapt the algorithm for these purposes.

5 Conclusions

To use a computer model for simulation-based design, designers must build confidence in using the model for predicting physical systems, which is accomplished by quantifying uncertainty via model updating. The model updating formulation proposed by Kennedy and O'Hagan is the most comprehensive and applicable to design under uncertainty. However, a limitation of this model updating formulation is that it can be difficult to distinguish between the effects of the calibration parameters (parameter uncertainty) and the discrepancy function (model discrepancy), which results in a lack of identifiability. It is important to identify, or differentiate, between these two sources of uncertainty in order to better understand the underlying sources of model uncertainty and ultimately further improve the model to better represent reality.

The degree of identifiability can be measured by the posterior covariance of the calibration parameters in a typical model uncertainty quantification framework. A better understanding of the identifiability problem is provided in this chapter in relation to model updating by using several illustrative examples. It was first shown how the calibration parameters and the discrepancy function were not identifiable in the simply supported beam example, even when significant amounts of simulation and experimental data are available. There are several different combinations of the computer model (with different values for the calibration parameter) and the estimated discrepancy function that combine to accurately predict the experimental response, which translates to poor identifiability. To improve identifiability, one can use informative prior distributions for the calibration parameters and/or the discrepancy function; however, this is a crude solution. Furthermore, in practice, informative priors typically do not exist. In the step function example in Sect. 2.3, it was shown that identifiability is possible under the relatively mild assumption of a smooth discrepancy function. In this example, the estimated discrepancy function is only smooth and consistent with the GP model when the calibration parameter equals the true value, which resulted in good identifiability. In conclusion, identifiability is a highly nuanced and difficult problem, but not impossible.

In spite of the identifiability challenges, good identifiability may often be achieved in model uncertainty quantification by measuring multiple experimental

responses that are automatically calculated in the simulation and that share a mutual dependence on a common set of calibration parameters. An intriguing phenomenon was observed that some combinations of responses may result in drastically different identifiability than others: By revisiting the beam example, it was shown that measuring certain responses will achieve substantial improvement in identifiability, while measuring other combinations of responses provides little improvement in identifiability beyond measuring only a single response. To take advantage of this, a method is needed for predicting multi-response identifiability prior to conducting the physical experiments, to allow users to choose the most appropriate set of responses to measure experimentally.

As it is generally not feasible, nor necessary, to measure experimentally all of the great many responses that are automatically calculated in the simulations, a preposterior analysis was introduced to address the issue of how to select the most appropriate subset of responses to measure experimentally, to best enhance identifiability. Prior to conducting the physical experiments but after conducting the computer simulations, the preposterior analysis can predict the relative improvement in identifiability that will result using different subsets of responses. It is based on Monte Carlo simulations within a modular Bayesian multi-response Gaussian process (MRGP) framework. The case studies showed that the approach is effective in predicting which subset of responses will provide the largest improvement in identifiability. Even though there are absolute differences between the preposterior and actual posterior covariances, the relative differences and the rankings derived from them are quite consistent, indicating that the method can be used effectively to choose the best combination of responses to measure experimentally.

Furthermore, to render the approach computationally feasible in engineering applications with a large number of responses, a simpler, surrogate preposterior analysis based on the expected Fisher information of the calibration parameters was introduced. The expected Fisher information matrix is the frequentist counterpart to the Bayesian preposterior covariance matrix of the parameters. It was demonstrated that, while being much faster to calculate than the preposterior covariance, it still provides a reasonable indication of the resulting identifiability. For real engineering applications with many system responses (and hence many different combinations of responses), it is recommended using the surrogate preposterior analysis to eliminate the responses that are unlikely to lead to good identifiability and reduce the number of response combinations that are considered in the preposterior analysis.

Appendix A: Estimates of the Hyperparameters for the Computer Model MRGP

To obtain the MLEs of the hyperparameters for the computer model MRGP model, the multivariate normal likelihood function is first constructed as:

$$\begin{aligned}
p(vec(\mathbf{Y}^m) | \mathbf{B}^m, \Sigma^m, \omega^m) &= (2\pi)^{-qN_m/2} |\Sigma^m|^{-N_m/2} |\mathbf{R}^m|^{-q/2} \\
&\times \exp \left\{ -\frac{1}{2} vec(\mathbf{Y}^m - \mathbf{H}^m \mathbf{B}^m)^T (\Sigma^m \otimes \mathbf{R}^m)^{-1} vec(\mathbf{Y}^m - \mathbf{H}^m \mathbf{B}^m) \right\},
\end{aligned} \tag{4.18}$$

where $vec(\cdot)$ is the vectorization of the matrix (stacking of the columns), \otimes denotes the Kronecker product, \mathbf{R}^m is a $N_m \times N_m$ correlation matrix whose i th-row, j th-column entry is $R^m((\mathbf{x}_i^m, \theta_i^m), (\mathbf{x}_j^m, \theta_j^m))$, and $\mathbf{H}^m = [\mathbf{h}^m(\mathbf{x}_1^m, \theta_1^m)^T, \dots, \mathbf{h}^m(\mathbf{x}_{N_m}^m, \theta_{N_m}^m)^T]^T$. Taking the log of Eq. (4.18) yields:

$$\begin{aligned}
\ln(p(vec(\mathbf{Y}^m) | \mathbf{B}^m, \Sigma^m, \omega^m)) &= -\frac{qN_m}{2} \ln(2\pi) - \frac{N_m}{2} \ln(|\Sigma^m|) - \frac{q}{2} \ln(|\mathbf{R}^m|) \\
&- \frac{1}{2} vec(\mathbf{Y}^m - \mathbf{H}^m \mathbf{B}^m)^T (\Sigma^m \otimes \mathbf{R}^m)^{-1} vec(\mathbf{Y}^m - \mathbf{H}^m \mathbf{B}^m).
\end{aligned} \tag{4.19}$$

The MLE of \mathbf{B}^m is found by setting the derivative of Eq. (4.19) with respect to \mathbf{B}^m equal to zero, which gives:

$$\hat{\mathbf{B}}^m = [(\mathbf{H}^m)^T (\mathbf{R}^m)^{-1} \mathbf{H}^m]^{-1} (\mathbf{H}^m)^T (\mathbf{R}^m)^{-1} \mathbf{Y}^m. \tag{4.20}$$

The MLE of Σ^m is found using result 4.10 of Ref. [52], which yields:

$$\hat{\Sigma}^m = \frac{1}{N_m} (\mathbf{Y}^m - \mathbf{H}^m \hat{\mathbf{B}}^m)^T (\mathbf{R}^m)^{-1} (\mathbf{Y}^m - \mathbf{H}^m \hat{\mathbf{B}}^m). \tag{4.21}$$

Finally, the MLE of ω^m , denoted by $\hat{\omega}^m$, is found by numerically maximizing Eq. (4.19) after plugging in the MLEs of \mathbf{B}^m and Σ^m .

Appendix B: Posterior Distributions of the Computer Responses

After observing \mathbf{Y}^m , the posterior of the computer response $y_i^m(\mathbf{x}, \theta)$ given \mathbf{Y}^m (and given ω^m and Σ^m and assuming a non-informative prior for \mathbf{B}^m) is Gaussian with mean and covariance:

$$\mathbb{E}[y^m(\mathbf{x}, \theta) | \mathbf{Y}^m, \phi^m] = \mathbf{h}^m(\mathbf{x}, \theta) \hat{\mathbf{B}}^m + \mathbf{r}^m(\mathbf{x}, \theta)^T (\mathbf{R}^m)^{-1} (\mathbf{Y}^m - \mathbf{H}^m \hat{\mathbf{B}}^m) \tag{4.22}$$

$$\begin{aligned}
\text{Cov}[y^m(\mathbf{x}, \theta), y^m(\mathbf{x}', \theta') | \mathbf{Y}^m, \phi^m] &= \Sigma^m \{ R^m((\mathbf{x}, \theta), (\mathbf{x}', \theta')) \\
&- \mathbf{r}^m(\mathbf{x}, \theta)^T (\mathbf{R}^m)^{-1} \mathbf{r}^m(\mathbf{x}', \theta') + [\mathbf{h}^m(\mathbf{x}, \theta)^T - (\mathbf{H}^m)^T (\mathbf{R}^m)^{-1} \mathbf{r}^m(\mathbf{x}, \theta)]^T \\
&\times [(\mathbf{H}^m)^T (\mathbf{R}^m)^{-1} \mathbf{H}^m]^{-1} [\mathbf{h}^m(\mathbf{x}, \theta)^T - (\mathbf{H}^m)^T (\mathbf{R}^m)^{-1} \mathbf{r}^m(\mathbf{x}, \theta)] \}
\end{aligned} \tag{4.23}$$

where $\mathbf{r}^m(\mathbf{x}, \boldsymbol{\theta})$ is a $N_m \times 1$ vector whose i th element is $R^m((\mathbf{x}_i^m, \boldsymbol{\theta}_i^m), (\mathbf{x}, \boldsymbol{\theta}))$. Using an empirical Bayes approach, the MLEs of the hyperparameters from Appendix A are plugged into Eqs. (4.22) and (4.23) to calculate the posterior distribution of the computer responses. Notice that Eqs. (4.22) and (4.23) are analogous to the single-response GP model results.

Appendix C: Estimates of the Hyperparameters for the Discrepancy Functions MRGP

To estimate the hyperparameters $\boldsymbol{\phi}^\delta = \{\mathbf{B}^\delta, \boldsymbol{\Sigma}^\delta, \boldsymbol{\omega}^\delta, \boldsymbol{\lambda}\}$ of the MRGP model representing the discrepancy functions, the procedure outlined by Kennedy and O'Hagan [1] is used and modified to handle multiple responses. This procedure begins by obtaining a posterior of the experimental responses given the simulation data and the hyperparameters from Module 1, which has prior mean and covariance:

$$\mathbb{E}[\mathbf{y}^e(\mathbf{x})|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m, \boldsymbol{\theta} = \boldsymbol{\theta}^*] = \mathbb{E}[\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta}^*)|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m] + \mathbf{h}^\delta(\mathbf{x})\mathbf{B}^\delta, \quad (4.24)$$

$$\begin{aligned} \text{Cov}[\mathbf{y}^e(\mathbf{x}), \mathbf{y}^e(\mathbf{x}')|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m, \boldsymbol{\theta} = \boldsymbol{\theta}^*] &= \boldsymbol{\Sigma}^\delta R^\delta(\mathbf{x}, \mathbf{x}) + \boldsymbol{\lambda} \\ &\quad + \text{Cov}[\mathbf{y}^m(\mathbf{x}, \boldsymbol{\theta}^*), \mathbf{y}^m(\mathbf{x}', \boldsymbol{\theta}^*)|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m], \end{aligned} \quad (4.25)$$

where $\hat{\boldsymbol{\phi}}^m$ are the MLEs of the hyperparameters for the computer model MRGP model. Since Eqs. (4.24) and (4.25) depend on the unknown true value of $\boldsymbol{\theta}^*$, these two equations are integrated with respect to the prior distribution of $\boldsymbol{\theta}(p(\boldsymbol{\theta}))$ via:

$$\begin{aligned} \mathbb{E}[\mathbf{y}^e(\mathbf{x})|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m] &= \int \mathbb{E}[\mathbf{y}^e(\mathbf{x})|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m, \boldsymbol{\theta}] p(\boldsymbol{\theta}) d\boldsymbol{\theta}, \\ \text{Cov}[\mathbf{y}^e(\mathbf{x}), \mathbf{y}^e(\mathbf{x}')|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m] &= \int \text{Cov}[\mathbf{y}^e(\mathbf{x}), \mathbf{y}^e(\mathbf{x}')|\mathbf{Y}^m, \hat{\boldsymbol{\phi}}^m, \boldsymbol{\theta}] p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \end{aligned} \quad (4.26)$$

Kennedy and O'Hagan [53] provide closed form solutions for Eq. (4.26) under the conditions of Gaussian correlation functions, constant regression functions, and normal prior distributions for $\boldsymbol{\theta}$ (for details, refer to Section 3 of [53] and Section 4.5 of [1]). In this chapter, similar closed form solutions are used except that a uniform prior distribution is assumed for $\boldsymbol{\theta}$.

After observing the experimental data, \mathbf{Y}^e , one can construct a multivariate normal likelihood function with mean and variance from Eq. (4.26). The MLEs of $\boldsymbol{\phi}^\delta$ maximize this likelihood function. The MLE of \mathbf{B}^δ can be found by setting the analytical derivative of this likelihood function with respect to \mathbf{B}^δ equal to zero (see Section 2 of Ref. [53]). However, there are no analytical derivatives with respect to the hyperparameters $\boldsymbol{\Sigma}^\delta, \boldsymbol{\omega}^\delta$, and $\boldsymbol{\lambda}$. Therefore, numerical optimization techniques are needed to find these MLEs.

Appendix D: Posterior Distribution of the Calibration Parameters

The posterior for the calibration parameters in Eq.(4.12) involves the likelihood function $p(\mathbf{d}|\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})$ and the marginal posterior distribution for the data $p(\mathbf{d}|\hat{\boldsymbol{\phi}})$. The likelihood function is multivariate normal with mean vector and covariance matrix defined as:

$$\begin{aligned}\mathbf{m}(\boldsymbol{\theta}) &= \mathbf{H}(\boldsymbol{\theta})\hat{\mathbf{B}} \\ &= \begin{bmatrix} \mathbf{I}_q \otimes \mathbf{H}^m & \mathbf{0} \\ \mathbf{I}_q \otimes \mathbf{H}^m(\mathbf{X}^e, \boldsymbol{\theta}) & \mathbf{I}_q \otimes \mathbf{H}^\delta \end{bmatrix} \begin{bmatrix} \text{vec}(\hat{\mathbf{B}}^m) \\ \text{vec}(\hat{\mathbf{B}}^\delta) \end{bmatrix},\end{aligned}\quad (4.27)$$

$$\mathbf{V}(\boldsymbol{\theta}) = \begin{bmatrix} \hat{\Sigma}^m \otimes \mathbf{R}^m & \hat{\Sigma}^m \otimes \mathbf{C}^m \\ \hat{\Sigma}^m \otimes \mathbf{C}^{m T} & \hat{\Sigma}^m \otimes \mathbf{R}^m(\mathbf{X}^e, \boldsymbol{\theta}) + \hat{\Sigma}^\delta \otimes \mathbf{R}^\delta + \hat{\lambda} \otimes \mathbf{I}_{N_e} \end{bmatrix}, \quad (4.28)$$

where $\hat{\mathbf{B}} = (\mathbf{H}(\boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{H}(\boldsymbol{\theta}))^{-1} \mathbf{H}(\boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{d}$, which is calculated based on the entire data set (instead of using the estimates from Modules 1 and 2 for \mathbf{B}^m and \mathbf{B}^δ) as detailed in Section 4 of [53]. $\mathbf{H}^m(\mathbf{X}^e, \boldsymbol{\theta}) = [\mathbf{h}^m(\mathbf{x}_1^e, \boldsymbol{\theta})^T, \dots, \mathbf{h}^m(\mathbf{x}_{N_e}^m, \boldsymbol{\theta})^T]^T$ and $\mathbf{H}^\delta = [\mathbf{h}^\delta(\mathbf{x}_1^e)^T, \dots, \mathbf{h}^\delta(\mathbf{x}_{N_e}^\delta)^T]^T$ denote the specified regression functions for the computer model and the discrepancy functions at the input settings \mathbf{X}^e . \mathbf{C}^m denotes the $N_m \times N_e$ matrix with i th-row, j th-column entries $R^m((\mathbf{x}_i^m, \boldsymbol{\theta}_i^m), (\mathbf{x}_j^e, \boldsymbol{\theta}))$. $\mathbf{R}^m(\mathbf{X}^e, \boldsymbol{\theta})$ denotes the $N_e \times N_e$ matrix with i th-row, j th-column entries $R^m((\mathbf{x}_i^e, \boldsymbol{\theta}), (\mathbf{x}_j^e, \boldsymbol{\theta}))$. \mathbf{R}^δ denotes the $N_e \times N_e$ matrix with i th-row, j th-column entries $R^\delta(\mathbf{x}_i^e, \mathbf{x}_j^e)$. Finally, \mathbf{I}_q and \mathbf{I}_{N_e} denote the $q \times q$ and $N_e \times N_e$ identity matrices.

The marginal posterior distribution for the data $p(\mathbf{d}|\hat{\boldsymbol{\phi}})$ is:

$$p(\mathbf{d}|\hat{\boldsymbol{\phi}}) = \int p(\mathbf{d}|\boldsymbol{\theta}, \hat{\boldsymbol{\phi}}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}, \quad (4.29)$$

which can be calculated using any numerical integration technique. In this chapter, Legendre-Gauss quadrature is used for the low-dimensional examples. Alternatively, Markov chain Monte Carlo (MCMC) could be used to sample complex posterior distributions such as those in Eq.(4.12).

Appendix E: Posterior Distribution of the Experimental Responses

Since a MRGP model represents the experimental responses, the conditional (given $\boldsymbol{\theta}$) posterior distribution at any point \mathbf{x} is Gaussian with mean and covariance defined as (assuming a non-informative prior on \mathbf{B}^m and \mathbf{B}^δ and using the empirical Bayes approach that treats $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$ as fixed):

$$\mathbb{E}[\mathbf{y}^e(\mathbf{x})^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}] = \mathbf{h}(\mathbf{x}, \boldsymbol{\theta})\hat{\mathbf{B}} + \mathbf{t}(\mathbf{x}, \boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1}(\mathbf{d} - \mathbf{H}(\boldsymbol{\theta})\hat{\mathbf{B}}), \quad (4.30)$$

$$\begin{aligned} \text{Cov}[\mathbf{y}^e(\mathbf{x})^T, \mathbf{y}^e(\mathbf{x}')^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}] &= \hat{\Sigma}^m R^m((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta})) + \hat{\Sigma}^\delta R^\delta(\mathbf{x}, \mathbf{x}') \\ &+ \hat{\lambda} - \mathbf{t}(\mathbf{x}, \boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{t}(\mathbf{x}', \boldsymbol{\theta}) + (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta})^T - \mathbf{H}(\boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{t}(\mathbf{x}, \boldsymbol{\theta}))^T \\ &\times (\mathbf{H}(\boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{H}(\boldsymbol{\theta}))^{-1} (\mathbf{h}(\mathbf{x}', \boldsymbol{\theta})^T - \mathbf{H}(\boldsymbol{\theta})^T \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{t}(\mathbf{x}', \boldsymbol{\theta})), \end{aligned} \quad (4.31)$$

where:

$$\mathbf{t}(\mathbf{x}, \boldsymbol{\theta}) = \begin{bmatrix} \hat{\Sigma}^m \otimes \mathbf{R}^m((\mathbf{X}^m, \boldsymbol{\Theta}^m), (\mathbf{x}, \boldsymbol{\theta})) \\ \hat{\Sigma}^m \otimes \mathbf{R}^m((\mathbf{X}^e, \boldsymbol{\theta}), (\mathbf{x}, \boldsymbol{\theta})) + \hat{\Sigma}^\delta \otimes \mathbf{R}^\delta(\mathbf{X}^e, \mathbf{x}) \end{bmatrix}, \quad (4.32)$$

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) = [\mathbf{I}_q \otimes \mathbf{h}^m(\mathbf{x}, \boldsymbol{\theta}) \ \mathbf{I}_q \otimes \mathbf{h}^\delta(\mathbf{x})]. \quad (4.33)$$

$\mathbf{R}^m((\mathbf{X}^m, \boldsymbol{\Theta}^m), (\mathbf{x}, \boldsymbol{\theta}))$ is a $N_m \times 1$ vector whose i th entry is $R^m((\mathbf{x}_i^m, \boldsymbol{\theta}_i^m), (\mathbf{x}, \boldsymbol{\theta}))$, $\mathbf{R}^m((\mathbf{X}^e, \boldsymbol{\theta}), (\mathbf{x}, \boldsymbol{\theta}))$ is a $N_e \times 1$ vector whose i th entry is $R^m((\mathbf{x}_i^e, \boldsymbol{\theta}), (\mathbf{x}, \boldsymbol{\theta}))$, and $\mathbf{R}^\delta(\mathbf{X}^e, \mathbf{x})$ is a $N_e \times 1$ vector whose i th entry is $R^\delta(\mathbf{x}_i^e, \mathbf{x})$.

To calculate the unconditional posterior distributions (marginalized with respect to $\boldsymbol{\theta}$) of the experimental responses, the conditional posterior distributions are marginalized with respect to the posterior distribution of the calibration parameters from Module 3. The mean and covariance of the unconditional posterior distributions can be written as:

$$\mathbb{E}[\mathbf{y}^e(\mathbf{x})^T | \mathbf{d}, \hat{\boldsymbol{\phi}}] = \mathbb{E}[\mathbb{E}[\mathbf{y}^e(\mathbf{x})^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}]], \quad (4.34)$$

$$\begin{aligned} \text{Cov}[\mathbf{y}^e(\mathbf{x})^T, \mathbf{y}^e(\mathbf{x}')^T | \mathbf{d}, \hat{\boldsymbol{\phi}}] &= \mathbb{E}[\text{Cov}[\mathbf{y}^e(\mathbf{x})^T, \mathbf{y}^e(\mathbf{x}')^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}]] \\ &+ \text{Cov}[\mathbb{E}[\mathbf{y}^e(\mathbf{x})^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}], \mathbb{E}[\mathbf{y}^e(\mathbf{x}')^T | \boldsymbol{\theta}, \mathbf{d}, \hat{\boldsymbol{\phi}}]], \end{aligned} \quad (4.35)$$

where the outer expectation and covariance are with respect to the posterior distribution of the calibration parameters. Equations (4.34) and (4.35) are derived using the law of total expectation and the law of total covariance [54]. Due to the complexity of the posterior distribution of the calibration parameters, the marginalization requires numerical integration methods. For the examples in this chapter, Legendre-Gauss quadrature is used.

References

1. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B* **63**(3), 425–464 (2001)
2. Higdon, D., Kennedy, M.C., Cavendish, J., Cafeo, J., Ryne, R.: Combining field data and computer simulations for calibration and prediction. *SIAM J. Sci. Comput.* **26**(2), 448–466 (2004)
3. Reese, C.S., Wilson, A.G., Hamada, M., Martz, H.F.: Integrated analysis of computer and physical experiments. *Technometrics* **46**(2), 153–164 (2004)

4. Bayarri, M.J., Berger, J.O., Paulo, R., Sacks, J., Cafeo, J.A., Cavendish, J., Lin, C.H., Tu, J.: A framework for validation of computer models. *Technometrics* **49**(2), 138–154 (2007)
5. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
6. Chen, W., Xiong, Y., Tsui, K.L., Wang, S.: A design-driven validation approach using bayesian prediction models. *J. Mech. Des.* **130**(2), 021101 (2008)
7. Qian, P.Z.G., Wu, C.F.J.: Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics* **50**(2), 192–204 (2008)
8. Wang, S., Tsui, K.L., Chen, W.: Bayesian validation of computer models. *Technometrics* **51**(4), 439–451 (2009)
9. Drignei, D.: A kriging approach to the analysis of climate model experiments. *J. Agric. Biol. Environ. Stat.* **14**(1), 99–112 (2009)
10. Akkaram, S., Agarwal, H., Kale, A., Wang, L.: Meta modeling techniques and optimal design of experiments for transient inverse modeling applications. Paper presented at the ASME International Design Engineering Technical Conference, Montreal (2010)
11. Huan, X., Marzouk, Y.M.: Simulation-based optimal Bayesian experimental design for nonlinear systems. *J. Comput. Phys.* **232**(1), 288–317 (2013)
12. Loepky, J., Bingham, D., Welch, W.: Computer model calibration or tuning in practice. Technical Report, University of British Columbia, Vancouver, p. 20 (2006)
13. Han, G., Santner, T.J., Rawlinson, J.J.: Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics* **51**(4), 464–474 (2009)
14. Arendt, P.D., Apley, D.W., Chen, W.: Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *J. Mech. Des.* **134**(10) (2012)
15. Arendt, P.D., Apley, D.W., Chen, W., Lamb, D., Gorsich, D.: Improving identifiability in model calibration using multiple responses. *J. Mech. Des.* **134**(10) (2012)
16. Ranjan, P., Lu, W., Bingham, D., Reese, S., Williams, B., Chou, C., Doss, F., Grosskopf, M., Holloway, J.: Follow-up experimental designs for computer models and physical processes. *J. Stat. Theory Pract.* **5**(1), 119–136 (2011)
17. Williams, B.J., Loepky, J.L., Moore, L.M., Macklem, M.S.: Batch sequential design to achieve predictive maturity with calibrated computer models. *Reliab. Eng. Syst. Saf.* **96**(9), 1208–1219 (2011)
18. Tuo, R., Wu, C.F.J., Vu, D.: Surrogate modeling of computer experiments with different mesh densities. *Technometrics* **56**(3), 372–380 (2014)
19. Maheshwari, A.K., Pathak, K.K., Ramakrishnan, N., Narayan, S.P.: Modified Johnson-Cook material flow model for hot deformation processing. *J. Mater. Sci.* **45**(4), 859–864 (2010)
20. Xiong, Y., Chen, W., Tsui, K.L., Apley, D.W.: A better understanding of model updating strategies in validating engineering models. *Comput. Methods Appl. Mech. Eng.* **198**(15–16), 1327–1337 (2009)
21. Liu, F., Bayarri, M.J., Berger, J.O., Paulo, R., Sacks, J.: A Bayesian analysis of the thermal challenge problem. *Comput. Methods Appl. Mech. Eng.* **197**(29–32), 2457–2466 (2008)
22. Arendt, P., Apley, D.W., Chen, W.: Updating predictive models: calibration, bias correction, and identifiability. Paper presented at the ASME 2010 International Design Engineering Technical Conferences, Montreal (2010)
23. Chakrabarty, J.: Theory of Plasticity, 3rd edn. Elsevier/Butterworth-Heinemann, Burlington (2006)
24. Liu, F., Bayarri, M.J., Berger, J.O.: Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Anal.* **4**(1), 119–150 (2009)
25. Joseph, V., Melkote, S.: Statistical adjustments to engineering models. *J. Qual. Technol.* **41**(4), 362–375 (2009)
26. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning. MIT Press, Cambridge (2006)
27. Qian, P.Z.G., Wu, H.Q., Wu, C.F.J.: Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**(3), 383–396 (2008)

28. McMillan, N.J., Sacks, J., Welch, W.J., Gao, F.: Analysis of protein activity data by gaussian stochastic process models. *J. Biopharm. Stat.* **9**(1), 145–160 (1999)
29. Cressie, N.: Statistics for Spatial Data. Wiley Series in Probability and Statistics. Wiley, New York (1993)
30. Ver Hoef, J., Cressie, N.: Multivariable spatial prediction. *Math. Geol.* **25**(2), 219–240 (1993)
31. Conti, S., Gosling, J.P., Oakley, J.E., O'Hagan, A.: Gaussian process emulation of dynamic computer codes. *Biometrika* **96**(3), 663–676 (2009)
32. Conti, S., O'Hagan, A.: Bayesian emulation of complex multi-output and dynamic computer models. *J. Stat. Plan. Inference* **140**(3), 640–651 (2010)
33. Williams, B., Higdon, D., Gattiker, J., Moore, L.M., McKay, M.D., Keller-McNulty, S.: Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis* **1**(4), 765–792 (2006)
34. Bayarri, M.J., Berger, J.O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R.J., Paulo, R., Sacks, J., Walsh, D.: Computer model validation with functional output. *Ann. Stat.* **35**(5), 1874–1906 (2007)
35. McFarland, J., Mahadevan, S., Romero, V., Swiler, L.: Calibration and uncertainty analysis for computer simulations with multivariate output. *AIAA J.* **46**(5), 1253–1265 (2008)
36. Drignei, D.: A kriging approach to the analysis of climate model experiments. *J. Agric. Biol. Environ. Stat.* **14**(1), 99–114 (2009)
37. Kennedy, M.C., Anderson, C.W., Conti, S., O'Hagan, A.: Case studies in gaussian process modelling of computer codes. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1301–1309 (2006)
38. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
39. Rasmussen, C.E.: Evaluation of Gaussian Processes and Other Methods for Non-linear Regression. University of Toronto (1996)
40. Lancaster, T.: An Introduction to Modern Bayesian Econometrics. Blackwell, Malden (2004)
41. Arendt, P.D., Apley, D.W., Chen, W.: A preposterior analysis to predict identifiability in experimental calibration of computer models. *IIE Trans.* **48**(1), 75–88 (2016)
42. Jiang, Z., Chen, W., Apley, D.W.: Preposterior analysis to select experimental responses for improving identifiability in model uncertainty quantification. Paper presented at the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Portland (2013)
43. Jiang, Z., Apley, D.W., Chen, W.: Surrogate preposterior analyses for predicting and enhancing identifiability in model calibration. *Int. J. Uncertain. Quantif.* **5**(4), 341–359 (2015)
44. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer Series in Statistics. Springer, New York (1985)
45. Carlin, B.P., Louis, T.A.: Empirical bayes: Past, present and future. *J. Am. Stat. Assoc.* **95**(452), 1286–1289 (2000)
46. Wu, C., Hamada, M.: Experiments: Planning, Analysis, and Optimization. Wiley, New York (2009)
47. Montgomery, D.C.: Design and Analysis of Experiments, 7th edn. Wiley, Hoboken (2008)
48. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions. Dover, New York (1972)
49. Beyer, W.H.: CRC Standard Mathematical Tables, 28 edn. CRC, Boca Raton (1987)
50. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2004)
51. Smith, A., Gelfand, A.: Bayesian statistics without tears: a sampling-resampling perspective. *Am. Stat.* **46**(2), 84–88 (1992)
52. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis, 6th edn. Prentice Hall, Upper Saddle River (2007)
53. Kennedy, M.C., O'Hagan, A.: Supplementary Details on Bayesian Calibration of Computer Models, pp. 1–13. University of Sheffield, Sheffield (2000)
54. Billingsley, P.: Probability and Measure, Anniversary Edition. John Wiley & Sons, Inc., Hoboken (2011)

Validation of Physical Models in the Presence of Uncertainty

5

Robert D. Moser and Todd A. Oliver

Abstract

As the field of computational modeling continues to mature and simulation results are used to inform more critical decisions, validation of the physical models that form the basis of these simulations takes on increasing importance. While model validation is not a new concept, traditional techniques such as visual comparison of model outputs and experimental observations without accounting for uncertainties are insufficient for assessing model validity, particularly for the case where the intended purpose of the model is to make extrapolative predictions. This work provides an overview of validation of physical models in the presence of uncertainty. In particular, two issues are discussed: comparison of model outputs and observational data when both the model and observations are uncertain, and the process of building confidence in extrapolative predictions. For comparing uncertain model outputs and data, a Bayesian probabilistic perspective is adopted in which the problem of assessing the consistency of the model and the observations becomes one of Bayesian model checking. A broadly applicable approach to Bayesian model checking for physical models is described. For validating extrapolative predictions, a recently developed process termed predictive validation is discussed. This process relies on the ideas of Bayesian model checking but goes beyond comparison of model and data to assess the conditions necessary for reliable extrapolation using physics-based models.

R.D. Moser (✉)

Department of Mechanical Engineering, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA
e-mail: rmoser@ices.utexas.edu

T.A. Oliver

Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA
e-mail: oliver@ices.utexas.edu

Keywords

Extrapolative predictions • Posterior predictive assessment • Validation under uncertainty

Contents

1	Introduction	130
1.1	Measuring Agreement Under Uncertainty	131
1.2	Different Uses of Models	132
2	Comparing Model Outputs and Data in the Presence of Uncertainty	134
2.1	Sources of Uncertainty in Validation Tests	134
2.2	Assessing Consistency of Models and Data	136
2.3	Data for Validation	143
3	Validating Models for Prediction	145
3.1	Mathematical Structure for Prediction	145
3.2	Validation for Prediction	148
4	Conclusions and Challenges	154
	References	155

1 Introduction

Over the last century, the field of computational modeling has grown tremendously, from virtually nonexistent to pervasive. During this time, simultaneous advances in simulation algorithms and computer hardware have enabled the development and application of increasingly complicated and detailed models to represent evermore complex physical phenomena. These advances are revolutionizing the ways in which models are used in the design and analysis of complex systems, enabling simulation results to be used in support of critical design and operational decisions [13, 26]. With continued advances in models, algorithms, and hardware, numerical simulations will only become more critical in modern science and engineering.

Given the importance of computational modeling, it is increasingly important to assess the reliability, in light of the purpose of a given simulation, of the models that form the basis of computational simulations. This reliability assessment is the domain of validation. While the concept of model validation is not new, it has recently received renewed attention due to the rapid growth in the use of models as a basis for making decisions [2, 4, 29]. This article provides an overview of the state of the art in validation of physical models in the presence of uncertainty.

In science and engineering, the word validation is often used to refer to simple comparisons between model outputs and experimental data such as plotting the model results and data on the same axes to allow visual assessment of agreement or lack thereof. While comparisons between model and data are at the core of any validation procedure, there are a number of problems with such naive comparisons. First, these comparisons tend to lead to qualitative rather than quantitative assessments of agreement. While such qualitative assessments are often

instructive and important, they are clearly incomplete, particularly as a basis for making decisions regarding model validity. Second, in naive comparisons, it is common to ignore or only partially account for uncertainty – e.g., uncertainty in the experimental observations or the model input parameters. Without accounting for these uncertainties, it is not possible to appropriately determine whether the model and data agree. Third, by focusing entirely on the agreement in the observable quantities, such comparisons neglect the intended uses of the model and, in general, cannot on their own determine whether the model is sufficient for the intended purposes.

These drawbacks of straightforward but naive comparisons highlight the two primary difficulties in model validation. First, one must quantitatively measure the agreement between model outputs and experimental observations while accounting for uncertainties in both. This fact is widely recognized, particularly in the statistics community, and there are a number of possible approaches. Second, depending on the intended use of the model, an assessment of the agreement between model outputs and available data is not sufficient for validation. Recognizing the purpose of the model is crucial to designing an appropriate validation approach.

1.1 Measuring Agreement Under Uncertainty

While the intended uses of a model will be important in a complete assessment of the validity of the model for those uses, all validation methods rely in some way on an assessment of whether the model is consistent with some set of observational data. In general, both the observations and the model – either through input parameters, the model structure, or both – are subject to uncertainties that must be accounted for in this comparison. Indeed, if both the model and the experiments are free from any uncertainty, then they can only be consistent if the model perfectly reproduces all the data. To define consistency in the far more common situation where at least some aspect of the model and/or data is uncertain, one must supply mathematical representations of all relevant uncertainties, a quantitative method for comparing uncertain quantities using the chosen representations of uncertainty, and a tolerance defining how closely model and data must “agree” to be declared consistent.

A wide range of formalisms have been proposed to represent uncertainty [8, 10, 12, 21, 28, 34], and there is still considerable controversy in the literature regarding the most appropriate approach, especially for so-called “epistemic” uncertainty (see below). This work focuses on the Bayesian interpretation of probability, where probability provides a representation of the degree of plausibility of a proposition, to represent all uncertainties [21, 35]. This choice is popular and has many advantages, including a well-defined method for updating probabilistic models to incorporate new data (Bayes’ theorem) and an extensive and rapidly growing set of available algorithms for both forward and inverse UQ [1, 9, 23, 24, 27, 31]. While a full discussion of the controversy over uncertainty representations is beyond the scope of this article, for the purposes of model validation, most uncertainty representations that have been proposed are either overly simplistic (e.g., using

only intervals to represent uncertainty) or reduce to probability in special cases (e.g., mixed interval/probability methods [11] or Dempster-Shafer theory [33, 34]). Thus, independent of the noted controversy regarding uncertainty representations, a method for assessing consistency between model outputs and data, where both are represented using probability, is required.

One subtlety that arises in a validation assessment is that there are two types of uncertainty that may occur in a complex physical system. First is uncontrolled variabilities in the system, which, in the context of a validation test, result in observations that differ with repetition of the test. Such uncertainties are called aleatoric (from Latin *alea* for dice game). Probabilistic representations of aleatoric uncertainties describe frequencies of occurrence. The second form of uncertainty arises from incomplete knowledge of the system, which in the context of a validation test results in no variability in the observation with repetition of the test. Such uncertainties are called epistemic and can be represented using the Bayesian interpretation of probability. In this case, probability describes the plausibility of outcomes [8, 21, 35]. Because the interpretation of probability is different for aleatoric and epistemic uncertainties, they will need to be distinguished when formulating validation criteria (see Sect. 2.2.2).

Note that what is considered epistemic or aleatoric depends on the details of the problem. In some validation scenarios, a parameter or input could be constrained to be the same on repeated observations, while in another scenario, it is uncontrolled. A simple example is a mechanical part which has uncertainties in geometry due to manufacturing variability. In a validation scenario in which the same part is used in repeated observations, this uncertainty is epistemic. But, in a scenario in which repeated observations are made each with a different part, the uncertainty is aleatoric.

Given the choice of probability to represent uncertainty, it is natural to define consistency in terms of the plausibility of the observations arising from the probabilistic model of the experiment, which represents uncertainties in both physical model and the observation process. Of course, there are still many ways to define a “plausible outcome.” Here, the plausibility of the data as an outcome of the model is defined using highest posterior density credibility sets and tail probabilities of the observable or relevant test quantities. These ideas are described in more detail in Sect. 2.

1.2 Different Uses of Models

Computational models are used for many different purposes. In science and engineering, these different purposes can be split into three broad categories: (1) investigation of the consequences of theories, (2) analysis of experimental data, and (3) prediction.

Scientific theories often lead to models that are sufficiently complex that computations are required to evaluate whether the theory is consistent with reality. When computation is used in this way, the computational model will be an expression of

the theory in a scenario in which experimental data will be available. In addition to the theory being tested, the computational model may include representations (models) of, for example, the experimental facility and the diagnostic instruments. These auxiliary models should be endowed with uncertainties as appropriate. The validation question is then a simple one: given the uncertainties in the auxiliary models and the experimental data, is the data consistent with the computational model? This can be assessed using the techniques discussed in Sect. 2. If an inconsistency is detected, then either the theory being tested or one or more of the auxiliary models is invalid. Assuming the auxiliary models have been sufficiently tested so that their reliability is not in doubt, the theory being tested must be rejected. Alternatively, a lack of detectable inconsistency implies only that the analysis has failed to invalidate the theory.

Models are also used to analyze data obtained from experiments. In particular, it is often the case that the quantity one wishes to measure, e.g., the flow velocity at a point in a wind tunnel experiment, is not directly observable. Instead, one measures a different quantity, e.g., a voltage in a hot wire anemometer circuit, which is related to the quantity of interest (QoI) through a model. As when investigating the consequences of a theory, any detectable inconsistency between the model output and a reliable reference for the quantity being inferred – to be clear, this reliable reference must be from an independent source, such as a different instrument in a calibration experiment – cause the model to be invalid for data analysis purposes. However, a lack of detectable inconsistency does not imply that the model is valid for data analysis. One must also ensure that the intended data analysis does not require the model to extrapolate beyond the range of independent reference data. This extra step is necessary because once the model is used in an extrapolatory mode, it is being used to make predictions, which requires substantially more validation effort, as discussed below.

The most difficult validation situation is when one wishes to use the model to make predictions. To understand this difficulty, it is necessary to be precise about what it means to make a prediction. A prediction is a model-based computation of a specific QoI for which there is no observational data, for instance, because the quantity cannot be measured, because the scenario of interest cannot be produced in the laboratory, or because the system being modeled has not yet been built. Indeed, the prediction is necessary precisely because the QoI is not experimentally observable at the time the information is required, e.g., to inform a decision-making process. Thus, prediction implies extrapolation.

It is well known that a model may be adequate for computing one quantity while not another or in one region of the scenario space and not another. Thus, when extrapolation is involved, it is insufficient to simply compare the model against data to determine consistency. This consistency is necessary but not sufficient because it does not account for the fact that the prediction quantity and scenario are different from the observed quantities and scenarios. Thus, a key challenge in model validation for prediction is in determining the implications of the agreement or disagreement between the model output and the validation data on the accuracy and reliability of the desired prediction. For example, one important question is,

given some observed discrepancy between the model and data, is the model likely to produce predictions of the QoI with unacceptably large error?

While this type of question has generally been left to expert judgment [2, 4], a recently proposed predictive validation process aims to systematically and quantitatively address such issues [30]. The process involves developing stochastic models to represent uncertainty in both the physical model and validation data, allowing rigorous assessment of the agreement between the model and data under uncertainty, as discussed in Sect. 2. Further, these stochastic models, coupled with the common structure of physics-based models, allow one to pose a much richer set of validation questions and assessments to determine whether extrapolation with the model to the prediction is supported by the available data and other knowledge. This predictive validation process will be discussed further in Sect. 3.

2 Comparing Model Outputs and Data in the Presence of Uncertainty

Appropriate validation processes for mathematical models of physical systems depend on the purpose of the model, as discussed in Sect. 1. But, regardless of this purpose, the process will rely on the comparison of outputs of the mathematical models with observations of physical systems to which the model can be applied. Such comparisons are complicated by the presence of uncertainties in both the mathematical model and the observations. In the presence of uncertainty, the relatively straightforward question of whether a model and observations “agree” becomes a more subtle question of whether a model with all its uncertainties is consistent with the observations and all their uncertainties. This section discusses sources of uncertainty in validation tests (Sect. 2.1) and techniques for making comparisons in the presence of uncertainty (Sect. 2.2). Section 2.3 gives some general guidance on selecting validation data.

2.1 Sources of Uncertainty in Validation Tests

To analyze the sources of uncertainty in validation tests, it is helpful to introduce an abstract structure for such a test. Consider a mathematical model \mathcal{U} of some physical phenomenon, which is a mapping from some set of input quantities x to output quantities u (in general, a model for a quantity will be indicated by a calligraphic upper case symbol). The model will in general involve a set of model parameters α_u , which had to be calibrated using data from observations of the phenomenon. The α_u are generally uncertain. In addition, in some situations, the model may be known to be imperfect, so that there is an error ε_u . Therefore,

$$u = \mathcal{U}(x; \alpha_u) + \varepsilon_u, \quad (5.1)$$

where the error is represented here as additive, though other choices are possible. The error ε_u is imperfectly known and may be represented by an “inadequacy model” $\mathcal{E}_u(x; \beta_u)$ [30], with inadequacy model parameters β_u that are also calibrated and uncertain.

Observations of the phenomenon modeled by \mathcal{U} are generally made in the context of some larger system. This larger system has observable quantities v , which will be the basis of the validation test. The validation system must also be modeled with a model \mathcal{V} that is a mapping from a set of inputs y and the modeled quantities u to the observables v . The dependence on u is necessary since the system involves the phenomenon being modeled by \mathcal{U} . The model \mathcal{V} will in general involve model parameters α_v , which are uncertain, and \mathcal{V} may itself be imperfect with error ε_v , which is modeled as \mathcal{E}_v with parameters β_v . Thus, a preliminary representation of the validation system is given by

$$v = \mathcal{V}(u, y; \alpha_v) + \mathcal{E}_v(u, y; \beta_v) = \tilde{\mathcal{V}}(u, y; \alpha_v, \beta_v), \quad (5.2)$$

where $\tilde{\mathcal{V}}$ is the validation system model enriched with the inadequacy model \mathcal{E}_v .

To complete the validation model, u in (5.2) is expressed in terms of the model \mathcal{U} , which means that the inputs x to \mathcal{U} must be determined from the inputs y to \mathcal{V} using a third model \mathcal{X} with parameters α_x , which may also be imperfect, introducing uncertain errors ε_x , modeled as \mathcal{E}_x with parameters β_x , yielding

$$x = \mathcal{X}(y; \alpha_x) + \mathcal{E}_x(y; \beta_x) = \tilde{\mathcal{X}}(y; \alpha_x, \beta_x). \quad (5.3)$$

Because the model \mathcal{U} of the phenomenon is introduced into a larger model of the validation system, it is called an “embedded model” [30].

Finally, errors δ_v are introduced in the physical observations of v themselves, commonly identified as observation or instrument error. The complete model of the validation test is then

$$v = \tilde{\mathcal{V}}[\mathcal{U}(\tilde{\mathcal{X}}(y; \alpha_x, \beta_x), \alpha_u) + \mathcal{E}_u(\tilde{\mathcal{X}}(y; \alpha_x, \beta_x)), y; \alpha_v, \beta_v] + \delta_v. \quad (5.4)$$

Here, the \mathcal{E}_u term is retained explicitly to emphasize that the validation test is directed at the physical model \mathcal{U} and the associated inadequacy model \mathcal{E}_u , if any. In this model of the validation test, there are four types of uncertainties: uncertainties in the model parameters ($\alpha_x, \alpha_u, \alpha_v, \beta_x, \beta_u$, and β_v); uncertainties in the validation inputs y ; uncertainties due to the model errors (\mathcal{E}_u , \mathcal{E}_x , and \mathcal{E}_v); and finally uncertainties due to the observation or instrument errors (δ_v). Note that in some cases, it may be convenient to include the response of the measuring instrument(s) in the validation system model \mathcal{V} . In this case, the instrument errors are included in ε_v .

Clearly, the design of an experimental observation will seek to minimize the uncertainties not directly related to the model being studied (i.e., other than the uncertainties in α_u and \mathcal{E}_u). Furthermore, in the event that the model of the validation (5.4) is found to be inconsistent with observations of v , all that can be said

is that at least one of the models involved ($\mathcal{U} + \mathcal{E}_u$, $\tilde{\mathcal{V}}$, $\tilde{\mathcal{X}}$ and/or the representation of the observation error) or some input to one of these models is inconsistent with reality. For such a validation to meaningfully test the model \mathcal{U} of the phenomenon of interest, the validation problem should, if possible, be designed so that auxiliary models \mathcal{V} and \mathcal{X} are much more reliable than \mathcal{U} . In this way, any inconsistency between model and observation will strongly implicate the model \mathcal{U} that is being tested.

This abstract structure of a validation test might be better understood through reference to a relatively simple example. Let \mathcal{U} be a simple homogeneous linear elastic constitutive model for the stress-strain relationship in some solid part, with u being the stress tensor field and x the strain tensor field. The parameters α_u are the Lamé constants or equivalently the Young's modulus and the Poisson ratio for the material. No inadequacy model is included. A validation test might be conducted by placing the part in a testing machine, which applies a specified overall load force through a fixture in which the part is mounted. The observed quantities v could be the displacement of one or more points on the part, and δ_v would represent the error in determining this displacement experimentally.

The validation system model \mathcal{V} would include an equilibrium continuum equation for the part and possibly the fixture, a model for the connection between the part and the fixture, and a representation of the load characteristics of the testing machine. Parameters α_v might include those describing the material of the fixture, the connection between part and fixture, and the testing machine. The inputs y could include the applied load, the geometry of the part, the load configuration, and other settings of the testing machine. The error ε_v might account for uncontrolled non-idealities in the way the part is mounted in the fixture or in the testing machine. Finally, as a consequence of determining the displacement of points on the part using a continuum representation, the displacement everywhere would be determined. The model \mathcal{X} would then include the mapping from the displacement field in the continuum model used in \mathcal{V} to the strain field, which would not introduce any additional modeling errors ε_x because it is a simple kinematic definition .

The validation system model (5.4) defines the expected relation between the generally uncertain inputs y and observations v . This model includes uncertainties due to the model parameters (the α 's), the modeling errors (the ε 's), and the observation or instrument errors (δ). With a mathematical characterization of these uncertainties, (5.4) makes an uncertain claim as to the values of the observed quantities. The validation test is then to make observations \hat{v} of the physical system and determine whether the \hat{v} are consistent with the uncertain claims regarding v . Assessing this consistency is the subject of the following section.

2.2 Assessing Consistency of Models and Data

From the above discussion, it is clear that a mathematical representation of the many uncertainties in the validation system is needed. A number of such uncertainty representations have been proposed, and as discussed in Sect. 1, many of the issues

surrounding validation under uncertainty are not unique to any particular uncertainty representation. However, in the current discussion of how one actually makes an assessment of the consistency of models and observation in the presence of uncertainty, it will be helpful to consider a particular uncertainty representation: Bayesian probability. This representation is used here for the reasons discussed in Sect. 1.

Since, for the purposes of this work, uncertainty is represented using Bayesian probability, the question of consistency of the model with data falls in the domain of Bayesian model checking. There is a rich literature on this subject, which, for brevity, cannot be discussed extensively here. Instead, a broadly applicable approach to model checking, which is generally consistent with common notions of validation for models of physical systems, is described. The ideas outlined are most closely aligned with those of Andrew Gelman and collaborators [14–18], and the reader is directed to these references for a more detailed statistical perspective. In particular, see [16] for a broad perspective on the meaning and practice of Bayesian model checking.

2.2.1 Single Observation of a Single Observable

The simplest case to consider is that of a validation test with a single observable v , which is taken here to be a real-valued continuous random variable. When uncertainties are represented using Bayesian probability, the validation system model (5.4) yields a probability density $p(v)$ for v . If a single measurement of v is made, yielding a value \hat{v} , the question is then whether the model expressed as the probability density $p(v)$ is consistent with the observation \hat{v} . A straightforward way to assess this consistency is to plot the distribution $p(v)$, as in Fig. 5.1. Indeed graphical representations of model and data are often very informative. It is clear in this figure that if the observation is $\hat{v} = v_i$ for $i = 1, 2$, or 3 , then the observation is consistent with the model because these points fall within the significant probability mass of the distribution. On the other hand, if $\hat{v} = v_i$ for $i = 4$ or 5 , it is clear that the model is inconsistent with the observation. In making these assessments, one asks how likely it is for the observed valued \hat{v} to arise as a sample from a random variable with distribution $p(v)$, and for the values v_i for $i = 1, \dots, 5$, the answer to this question is clear. If, however, $\hat{v} = v_6$, the answer is not obvious, and for these marginal cases, some criterion would be needed to decide whether model and data are consistent. More usefully, a continuum of levels of consistency may be admitted, and one may ask for a measure of the (in)consistency between the model and data.

This is a general issue in statistical analysis. A common approach is to consider the probability of obtaining an observation more extreme than the observation in hand. That is, one can compute the “tail probability” $P_>$ as

$$P_> = P(v > \hat{v}) = \int_{\hat{v}}^{\infty} p(v) dv \quad (5.5)$$

which is the probability, according to the model, of an observation being greater than \hat{v} ($P_<$ is defined analogously). This is the well-known (Bayesian) p -value, and

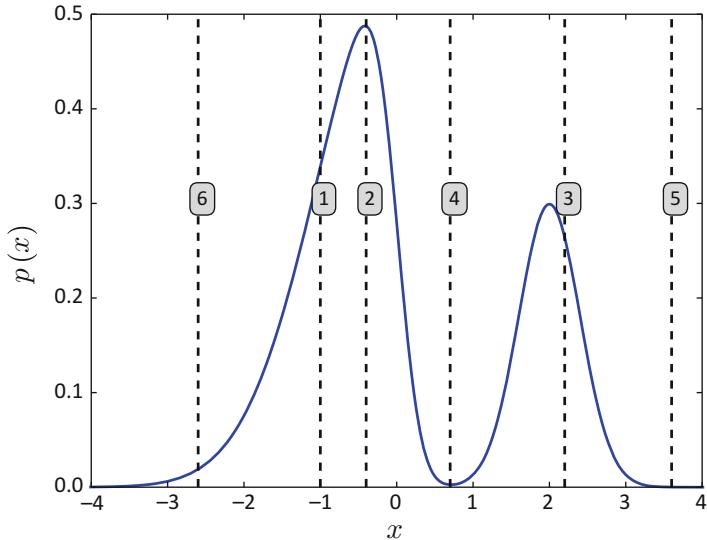


Fig. 5.1 Hypothetical model output distribution and a number of possible observations illustrating observations that are clearly consistent with the output distribution ($i = 1, 2, 3$), observations that are clearly inconsistent ($i = 4, 5$), and observations where the agreement is marginal ($i = 6$)

if it is sufficiently small (e.g., 0.05 or 0.01), one could conclude that \hat{v} is an unlikely outcome of the model, so that the validity of the model is suspect. This leads to the concept of a credibility interval, an interval in which, according to the model, it is highly probable (e.g., probability of 0.95 or 0.99) that an observation will fall. Given a probability distribution $p(v)$, there are many possible credibility intervals, or more generally credibility sets, with a given probability. One could, for example, choose a credibility interval centered on the mean of the distribution or one defined so that the probability of obtaining an observation greater than the upper bound of the interval is equal to that of an observation less than the lower bound. Either of these credibility intervals has the disturbing property of including the point v_4 in Fig. 5.1, which is clearly not a likely draw from the distribution plotted.

A credibility region that is more consistent with an intuitive understanding of credible observations for skewed and/or multimodal distributions such as that shown in Fig. 5.1 is the highest posterior density (HPD) credibility region [7]. The β -HPD ($0 \leq \beta \leq 1$) credible region S is the set for which the probability of belonging to S is β and the probability density for each point in S is greater than that of points outside S . Thus, for a multimodal distribution like that shown in Fig. 5.1, an HPD region may consist of multiple disjoint intervals [20] around the peaks, leaving out the low probability density regions between the peaks.

However, because HPD credibility sets are defined in terms of the probability density, they are not invariant to a change of variables. This is particularly undesirable when formulating a validation metric because it means that one's

conclusions about model validity would depend on the arbitrary choice of variables (e.g., whether one considers the observable to be the frequency or the period of an oscillation). To avoid this problem, a modification of the HPD set is introduced in which the credible set is defined in terms of the probability density relative to a specified distribution q [30]. An appropriate definition of q would be one that represents no information about v [21]. Using this definition of the highest posterior *relative density* (HPRD), a conceptually attractive credibility metric can be defined as $\gamma = 1 - \beta_{\min}$, where β_{\min} is the smallest value of β for which the observation \hat{v} is in the HPRD-credibility set for v according to the model. That is,

$$\gamma = 1 - \int_S p(v) d v, \quad \text{where } S = \left\{ v : \frac{p(v)}{q(v)} \geq \frac{p(\hat{v})}{q(\hat{v})} \right\}. \quad (5.6)$$

When γ is smaller than some tolerance, say less than 0.05 or 0.01, \hat{v} is considered an implausible outcome of the model – i.e., there is an inconsistency between the model and the observation.

2.2.2 Multiple Observations of a Single Observable

Of course, it is common to make multiple measurements of the observable v , especially if the measurement is noisy. Consider the case where the observational uncertainties represented in the model – which lead to the appearance of δ_v in (5.4) – are purely aleatoric and independent for each observation. Further, assume for the purposes of this discussion that any epistemic uncertainties in the model are negligible. In this case, the model implies that each observation is an independent sample of the distribution $p(v)$, and the validation question is whether a set of N observations \hat{v}_i for $i = 1, 2 \dots N$ is consistent with sampling from $p(v)$. It is clearly erroneous to check whether each individual observation is in a given credibility region, as the probability of at least one sample falling outside the credibility region will increase to 1 as N increases, even if the samples are in fact drawn according to the model distribution $p(v)$. A number of correction methods for this effect have been developed in the statistics literature [19, 25].

More generally, one must determine whether a given vector of observational values $\hat{V} = (\hat{v}_1, \dots, \hat{v}_N)$ is unlikely to have arisen as instances of random variables, in this case iid random variables with distribution $p(v)$, which is a common problem in statistical hypothesis testing. An obvious extension to the HPRD regions described above can be defined in terms of the joint distribution $p(V)$ of the vector of N iid random variables $V = (v_1, \dots, v_N)$, which because of independence can be written as:

$$p(V) = \prod_{i=1}^N p(v_i). \quad (5.7)$$

The HPRD-credibility metric can then be written as:

$$\gamma = 1 - \int_S p(V) d V, \quad \text{where } S = \left\{ V : \frac{p(V)}{q(V)} \geq \frac{p(\hat{V})}{q(\hat{V})} \right\}. \quad (5.8)$$

While this directly answers the question of how credible the observations are as samples from the model distribution, it is generally difficult to compute when N is large since it involves evaluating a high-dimensional integral over a complex region. An alternative approach is to consider one or more test quantities [17, 18]. A test quantity $T(V)$ is a mapping from an N -vector to a scalar. When evaluated for a random vector, it is a random scalar, which is designed to summarize some important feature of V . The idea then is to ask whether $T(\hat{V})$ is a plausible sample from the distribution of $T(V)$. One could, for example, compute p -values for this comparison. The HPRD metric could also be used, but $p(T(V))$ which is part of its definition is usually difficult to compute.

In addition to being potentially more tractable, the use of test statistics has another advantage. With rare exceptions, the uncertainty representations leading to the stochastic model being validated are based on crude and/or convenient assumptions about the uncertainties. Indeed, iid Gaussian random variables are often used to model experimental noise, and while this is sometimes justified, it is often assumed purely out of convenience. Thus, one does not necessarily expect, nor is it generally required, for the model distribution $p(v)$ to be representative of the random processes that led to the variability in \hat{v} from observation to observation. In this case, it is necessary only that the uncertainty representations will characterize what is important about the uncertainty for the purposes for which the model is to be used. While the HPRD metric given in (5.8) does not take this into account, validating using test quantities gives one the opportunity to choose T to characterize an important aspect of the uncertainty. For example, if the model is to be used to evaluate extreme deviations from nominal behavior or conditions, then it might make sense to perform validation comparisons based on the test quantity $T(V) = \max_i v_i$. A few example test quantities are discussed in the next subsection.

Finally, the assumption of negligible epistemic uncertainty, while useful in simplifying this discussion, is not generally applicable. This assumption can be removed by marginalizing over the epistemic uncertainties represented in the model, as described in Sect. 2.2.4.

2.2.3 Defining Test Quantities

To select validation test quantities, one should consider what characteristics of the aleatoric uncertainty are important in the context of the model and its planned use. One common consideration is that the mean and variance should be consistent with observations. A straightforward test quantity is simply the sample average $A(V)$, that is,

$$A(V) = \frac{1}{N} \sum_{i=1}^N v_i \quad (5.9)$$

The validation comparison then reduces to asking whether the distribution of $p(A(V))$ implied by the model is consistent with the observation $A(\hat{V})$. Since the v_i determined from the model are iid, if N is sufficiently large, the central limit

theorem implies that $A(V)$ is approximately $\mathcal{N}(\mu, \sigma^2/N)$, where μ and σ^2 are the mean and variance of v . This leads to the Z-test in statistics.

To test whether the variability of v is consistent with the observed variability of \hat{v} , the test quantity

$$X^2(V) = \sum_{i=1}^N \frac{(v_i - \mu)^2}{\sigma^2} \quad (5.10)$$

could be used, which is a χ^2 discrepancy. Note that this test quantity is different because it depends explicitly on characteristics of the model (mean and variance). If in addition to being iid, the v obtained from the model are normally distributed, the model distribution $p(X^2(V))$ will be the χ^2 distribution, with N degrees of freedom.

When the test quantity has a known distribution, as $A(V)$ and $X^2(V)$ discussed above do, it simplifies assessing consistency using, for example, HPRD criteria or p-values. This is so because tail integrals of these distributions are known. However, it is not necessary that exact distributions be known, since the relevant integrals can be approximated using Monte Carlo or other uncertainty propagation algorithms.

For example, in some problems, one is concerned with improbable events with large consequences. In this case, one is interested in the tail of the distribution of v and the extreme values that v may take. A simple test quantity that is sensitive to this aspect of the distribution of v is the maximum attained value. That is,

$$M(V) = \max_i v_i \quad (5.11)$$

The random variable $M(V)$ that is implied by the model can be sampled by simply generating N samples of v and determining the maximum. Thus the p-value relative to the observation $M(\hat{Z})$ can be computed by Monte Carlo simulation.

There is a large literature on statistical test quantities (cf. [22]) for use in a wide variety of applications. Texts on statistics should be consulted for more details. Among commonly used statistical tests are those which test whether a population is consistent with a distribution with known characteristics (e.g., the mean and/or variance) as with the average and χ^2 test quantities discussed above. These are also useful when the model can be used to compute these characteristics with much higher statistical accuracy than they can be estimated from the data (e.g., one can generate many more samples from the model than there are data samples). In other situations, the model may be expensive to validate, so that the number of samples of the posterior distribution of the model is limited. In this case, test quantities that test whether two populations are drawn from distributions with the same characteristics are useful. A simple example is the two sample t-test (Welch's test), which is used to test whether two populations have the same mean.

2.2.4 General Posterior Model Checks

The discussion in Sects. 2.2.1 and 2.2.2 is instructive but does not apply to the usual situation. In particular, it is common to have multiple observations of

multiple different quantities, the predictions of which are affected by both epistemic and aleatoric uncertainties. This section generalizes the validation comparisons discussed previously to this more complicated situation.

Consider the model of the observable expressed in (5.4). This model includes uncertain parameters, the α 's and β 's; representations of modeling errors, ε_u , ε_x , and ε_v ; and the representation of the observation errors δ_v . Generally, the uncertainties in the parameters are considered to be epistemic, that is, there are ideal values of these parameters, which are imperfectly known. Their values do not change from observation to observation in the same system. The observation error δ_v is often considered to be aleatoric, for example, from instrument noise. However, there may also be systematic errors in the measurements, which are imperfectly known and thus epistemically uncertain. Similarly, depending on the nature of the phenomena involved and how they are modeled, the errors ε_u , ε_x , and ε_v in the models \mathcal{U} , \mathcal{X} , and \mathcal{V} may be epistemic, aleatoric, or a mixture of the two. The models for these uncertainties could then include an epistemic part and an aleatoric part, that is,

$$\varepsilon_x \approx \mathcal{E}_x^e + \mathcal{E}_x^a, \quad \delta_v \approx \mathcal{D}_v^e + \mathcal{D}_v^a, \quad (5.12)$$

where superscript e and a are for epistemic and aleatoric, respectively. One challenge is then to compare the model and possibly repeated observations under these mixed uncertainties.

In addition to multiple observations, there may be multiple observables. These observables may be of the same quantity (e.g., the temperature) for different values of the model inputs y (e.g., at different points in space or time) or of different quantities (e.g., the temperature and the pressure). The observable v should thus be considered to be a vector of observables of dimension n . In general, the aleatoric uncertainties in the various observables will be correlated. In the model, these correlations will arise both from the way the aleatoric inadequacy uncertainties impact the observables through the model and from correlations inherent in the dependencies of the aleatoric uncertainties (the \mathcal{E}^a 's and \mathcal{D}_v^a) on model inputs. Due to the presence of aleatoric uncertainties, a validation test may make multiple (N) observations \hat{v} of the observable vector v , resulting in a set of observations $\hat{V} = \{\hat{v}_1, \dots, \hat{v}_N\}$, which are taken to be independent and identically distributed (iid) here.

To facilitate further discussion, let us consolidate the epistemic uncertainties (including those associated with \mathcal{E}_u^e , \mathcal{E}_x^e , \mathcal{E}_v^e and \mathcal{D}_v^e) into a vector θ , which may be infinite dimensional. The validation model for the observables (5.4) can then be considered to be a statistical model \mathcal{S} for the aleatorically uncertain v , which depends on the epistemically uncertain θ , that is, $v = \mathcal{S}(\theta)$. The validation question is then whether the set of observations \hat{V} is consistent with \mathcal{S} , given what is known about θ .

If θ is known precisely (i.e., no epistemic uncertainty), then the situation is similar to that discussed in Sect. 2.2.2. For a given θ , the model \mathcal{S} defines a probability distribution $p(v|\theta)$ in the n -dimensional space of observables. The observables are not independent, so this is not a simple product of one-dimensional distributions, but this introduces no conceptual difficulties. A set of N observations

then defines a probability space of dimension nN . Because the observations are iid, the probability distribution is written:

$$p(V|\mathcal{S}, \theta) = \prod_{i=1}^N p(v_i|\mathcal{S}, \theta). \quad (5.13)$$

Here, the probability density is conditional on the model \mathcal{S} because these are the distributions of outputs v implied by the model. Consistency of \hat{V} with \mathcal{S} can then in principle be determined through something like the HPRD-credibility metric (5.6). However, as discussed in Sect. 2.2.2, this is generally not computationally tractable, nor is it generally desirable. Alternatively, one or more test quantities T may be defined to characterize what is important about the aleatoric variation of v , and as discussed in Sect. 2.2.2, the consistency between the observed $T(\hat{V})$ and the distribution obtained from the model $p(T(V)|\mathcal{S}, \theta)$ can be tested. For example, one could use the p-value $P_>$, which now depends on θ .

When the parameters θ are uncertain, with uncertainty that is entirely epistemic by construction, the validation question is whether the observed value of the test quantity $T(\hat{V})$ is plausible for plausible values of θ . In validation, it is presumed that the parameters in the models (the α 's and β 's in (5.4)) have been calibrated (e.g., via Bayesian inference) and that the epistemic uncertainties are now expressed as probability distributions $p(\theta|\mathcal{S}, \hat{w})$, where \hat{w} represents the data for the observables w used to calibrate the parameters. This calibration data may or may not be included in the validation data \hat{v} . In Bayesian inference, this is the posterior distribution, and so the relevant distribution of $T(V)$ or $P_>$ is that induced by the posterior distribution of θ .

As suggested by Box [6], Rubin [32], and Gelman et al. [17], in this situation, the consistency of the observations \hat{V} with the model can be determined by considering the distribution of $T(V)$ implied by the distribution of θ :

$$p(T(V)|\mathcal{S}, \hat{w}) = \int_{\theta} p(T(V)|\mathcal{S}, \theta) p(\theta|\mathcal{S}, \hat{w}) d\theta. \quad (5.14)$$

This is termed the posterior predictive distribution by Gelman *et al* in [17], though they were referring to the case in which \hat{w} is the same as \hat{v} . It then can be determined whether the observed value $T(\hat{V})$ of the test quantity is consistent with the distribution $p(T(V)|w)$, using, for example, p-values:

$$P_> = \int_{\theta} P(T(V) > T(\hat{V})|\mathcal{S}, \theta) p(\theta|\mathcal{S}, \hat{w}) d\theta. \quad (5.15)$$

2.3 Data for Validation

Of course, to perform a validation comparison, it is necessary to have data to which to compare. The question is, what should this data be? Logically, it is required that for a model to be valid, it must be consistent with all available relevant observational

data. While true, this does not provide useful guidance for designing experimental observations for the purpose of validation. Selecting appropriate validation data requires consideration of several problem-specific issues, so it is difficult to specify generally applicable techniques for designing validation tests. Instead, a list of broad guidelines for designing validation experiments is provided.

1. Often, the first point at which models are confronted with data is when they are calibrated. As part of the calibration process, the calibrated model should also be validated against the calibration data. Because the model has been calibrated to fit the calibration data, consistency with the data will not greatly increase confidence in the model. But if the model is inconsistent with the data with which it has been calibrated, it is a very strong indictment of the model. With parsimonious models, which describe a rich phenomenon with few parameters, failure to reproduce the calibration data is a common mode of validation failure.
2. To increase confidence in the validity of a model, it should be tested against data that was not used in its calibration. Sometimes this is done by holding back some portion of the calibration data set so that it can be used only for validation, which leads to cross-validation techniques [3]. This is generally of limited utility for physics-based models. A much stronger validation test is to use a completely different data set, from experiments with different validation models \mathcal{V} in (5.4). For example, calibration of a model might be done using a set of relatively simple experiments in a laboratory facility, while validation experiments are in more complex scenarios in completely different facilities.
3. The development of computational models often involves various approximations and assumptions. To increase confidence in the models, one should design validation experiments that test these approximations, that is, experiments and measurements should be designed so that the observed quantities are expected to be sensitive to any errors introduced by the approximation. Sensitivity analysis applied to the model can help identify such experiments. Furthermore, test quantities used in validation assessment should also be designed to be sensitive to the approximations being tested.
4. Models of complex physical systems commonly involve sub-models (embedded models) of several physical phenomena that are of questionable reliability. When possible, the individual embedded models should be calibrated and validated separately, using experiments that are designed to probe the modeled phenomena individually. The resulting experiments will generally be much simpler, less expensive, easier to instrument, and easier to simulate than the complete system. This simplicity could allow many more experiments and more measurements to be used for calibration and/or validation. When possible, further experiments simultaneously involving a few of the relevant phenomena should be performed to validate models in combination. Experiments of increasing complexity and involving more phenomena can then be pursued, until measurements in systems similar in complexity to the target system are performed. This structure has been described as a “validation pyramid” [5], with abundant simple inexpensive,

data-rich experiments at the bottom and increasingly expensive and limited experiments as one goes up the pyramid.

The role of the experiments higher in the validation pyramid is generally different from those lower down. The lower level experiments are designed to calibrate models and validate that the models provide a quantitatively accurate representation of the modeled phenomena. Experiments higher in the pyramid are intended to test the modeling of the interactions of different phenomena. Finally, measurements in systems as similar as possible to the target system, at conditions as close as possible to the conditions of interest, are used to detect whether there are unexpected phenomena or interactions that may affect predictions in the target system.

5. As discussed in Sect. 3, when a computational model is used for predictions of unobserved QoIs, the reliability of the predictions depends on the embedded models being used in conditions that have been well tested, with validation observations that are sensitive to errors in the model in the same way as the prediction QoIs. Validation tests should therefore be designed to challenge embedded models as they will be challenged in the prediction scenarios. The conditions that the embedded model experiences during predictions can be evaluated through model simulations of the prediction scenario, and sensitivities of the QoIs to an embedded model can be determined through sensitivity analysis conducted in the system model.

3 Validating Models for Prediction

As mentioned in Sect. 1, when a physical model is being used to make predictions, a detectable inconsistency between the model output and experimental data is not, on its own, sufficient to invalidate the model for use in the prediction. Indeed, it is common in engineering to use models which are known to be inconsistent with some relevant data but which are sufficient for the predictions for which they are to be used. In this situation, the important validation question is not whether the model is scientifically valid – often it is known *a priori* that it is not – but rather whether a prediction made with the model is reliable. This is generally a much more difficult question since it is essentially asking whether an extrapolation from available information will be reliable. Recently, a process for addressing this question in the context of physics-based mathematical models was developed [30]. This section will outline the components of this process.

3.1 Mathematical Structure for Prediction

An abstract structure of a validation test is defined in Sect. 2.1. Here, this mathematical structure is extended to include features common in models of physical systems. To fix the main ideas, this structure is presented first in the simplest possible context (Sect. 3.1.1), with a more general abstraction outlined briefly in Sect. 3.1.2.

3.1.1 Simplest Case

Mathematical models of the response of physical systems are generally based in part, indeed to the greatest extent possible, on physical theories that are known a priori to be reliable in the context of the desired prediction. The mathematical expression of these theories is a reliable, but incomplete, model of the system, which can be written as:

$$\mathcal{R}(u, \tau; r) = 0, \quad (5.16)$$

where \mathcal{R} is an operator expressing the reliable theory, u is the state, r is a set of variables that defines the problem scenario, and τ is an additional quantity that must be known to solve the system of equations. For instance, in fluid mechanics, \mathcal{R} could be a nonlinear differential operator expressing conservation of mass, momentum, and energy with u including the fluid density, velocity, and temperature fields. In this case, τ would include the pressure, viscous stresses, and heat flux, and r would include parameters like the Reynolds and Mach numbers as well as details of the flow geometry and boundary conditions.

This structure in which the model is based at least in part on reliable theory is common in physics-based models. The foundation on theory whose validity is not in question will be important for making predictions. However, the reliable theory on its own rarely forms a closed system of equations, which is represented in (5.16) by τ . If τ could be determined from u and r using a model as reliable as \mathcal{R} itself, then the combination of (5.16) with this high-fidelity model for τ would form a closed system of equations, and solutions of this system would be known to be reliable.

In general, such models – i.e., models that are both closed and known a priori to be highly reliable – are not available in the context of complex prediction problems. Instead the quantity τ must be represented using a lower-fidelity model or a model whose reliability is not known a priori. In this case, the model for τ , denoted \mathcal{T} , is referred to as an embedded model:

$$\tau = \mathcal{T}(u; s, \theta), \quad (5.17)$$

where s is a set of scenario parameters, possibly distinct from r , for the embedded model, and θ is a set of calibration or tuning parameters. Since the embedded model \mathcal{T} is not known a priori to be reliable for the desired prediction, it will be the focus of the validation process.

The system of equations consisting of (5.16) and (5.17) is closed, but for calibration, validation, and prediction, some additional relationships are necessary. In particular, expressions for the experimental observables v and the prediction QoIs q are required. Here, it is assumed that there are maps \mathcal{V} and \mathcal{Q} that determine v and q , respectively, from the model state u , the modeled quantity τ , and the global scenario r :

$$v = \mathcal{V}(u, \tau; r), \quad (5.18)$$

$$q = \mathcal{Q}(u, \tau; r). \quad (5.19)$$

This closed set of Eqs. (5.16), (5.17), (5.18), and (5.19), which allows calculation of both the experimental observables and predictions QoIs, is referred to as a composite model, since it is a composition of high- and low-fidelity components.

Because the foundation of the composite model is a reliable theory whose validity is not questioned, it is possible for such a model to make reliable predictions despite the fact that a less reliable embedded model is also involved. All that is required is that the less reliable embedded model should not be used outside the range where it has been calibrated and tested. This restriction does not necessarily limit the reliability of the composite model in extrapolation since the relevant scenario space for each embedded model is specific to that embedded model, not the composite model in which it is embedded. For example, an elastic constitutive relation for the deformation of a material can only be relied upon provided the strain remains within the bounds in which it has been calibrated and tested. Despite this restriction, a model for a complex structure made from the material, which is based on conservation of momentum, can reliably predict a wide range of structural responses.

3.1.2 Generalizations

The simple problem statement in Sect. 3.1.1 is sufficient to introduce many of the concepts that are critical in validation for prediction, including the distinction between the QoI and available observable quantities and the notion of an embedded model. However, there are several important generalizations that are required to represent the validation and prediction process in complex physical systems. These are outlined below. A more detailed description can be found in [30].

- **Multiple embedded models:** In a complex system, there will generally be multiple physical phenomena for which an embedded model is needed (e.g., thermodynamic models, chemical kinetics models, and molecular transport embedded models in a composite model of a combustion system). Thus, the composite model will generally depend on N_τ quantities τ_i , each with associated models \mathcal{T}_i , calibration parameters θ_i , and scenario parameters s_i .
- **Multiple reliable models:** The experimental systems in which measurements are made for validation and calibration are commonly different from, usually simpler than, the prediction system. For each of N_e experiments, there will in general be a different reliable model \mathcal{R}^j , with associated state variables u^j , observables v^j , scenario parameters r^j , and set of quantities requiring embedded models $\{\tau\}^j$. There are also, therefore, N_e observation models \mathcal{V}^j and sets of embedded models $\{\mathcal{T}\}^j$. For each experiment, the set of modeled quantities $\{\tau\}^j$ must include at least one member of the set of modeled quantities $\{\tau\}^0$ used in the prediction model, but may include other quantities requiring embedded models that are not relevant to the prediction (e.g., to represent an instrument or the laboratory facility).
- **Differing state variables:** In general, the different reliable models for each experiment \mathcal{R}^j have different state variables u^j . The dependence of an

embedded model \mathcal{T}_k^0 on these different states must be represented. To this end, each embedded model \mathcal{T}_k^0 is formulated to be dependent on an argument w_k that is consistent for the prediction and all experiments. There is then a mapping defined by the operator \mathcal{W}_k^j that maps the state variable u^j to the argument w_k .

With these extensions, a generalized version of the problem statement described in Sect. 3.1.1 can be formulated:

$$\begin{aligned} \mathcal{R}(u, \{\tau\}^0, r) &= 0 \\ q &= \mathcal{Q}(u, \{\tau\}^0, r) \\ \mathcal{R}^j(u^j, \{\tau\}^j, r^j) &= 0 \quad \text{for } 1 \leq j \leq N_e \\ v^j &= \mathcal{V}^j(u^j, \{\tau\}^j, r^j) \quad \text{for } 1 \leq j \leq N_e \end{aligned} \quad (5.20)$$

In this formulation, the embedded models required for the prediction ($j = 0$) and the validation scenarios ($1 \leq j \leq N_e$) are given by

$$\tau_k^j = \mathcal{T}_k^j(\mathcal{W}_k^j(u), \theta_k^j, s_k^j) \quad \text{for } 1 \leq k \leq N_\tau^j \quad (5.21)$$

where N_τ^j is the number of embedded models in scenario j .

Like the simple problem statement from Sect. 3.1.1, in this generalized problem the high-fidelity theory forming the basis of the model can enable reliable predictions, despite the need to extrapolate from available data. However, additional complexity arises from confounding introduced by the presence of multiple embedded models in the validation experiments. To avoid this confounding, one would ideally use experiments where there are no extra embedded models beyond those needed for the prediction or where any such extra embedded models introduce small error or uncertainty in the context of the experiment. Of course, the assessment of any extra embedded models would itself form another validation exercise.

To further avoid confounding uncertainties, it is preferable to use experiments in which only one of the embedded models used in the prediction model is exercised. Such experiments are powerful because they provide the most direct assessment possible of the embedded model in question. However, even if experiments that separately exercise all of the embedded models necessary for prediction are available, in general these experiments alone are not sufficient for validation because they cannot exercise couplings and interactions between the modeled phenomena. This fact leads to the idea of a validation pyramid [5], as discussed in Sect. 2.3.

3.2 Validation for Prediction

A fundamental challenge in the validation of predictions is that even if a model is consistent with all available data, as determined by the techniques discussed in

Sect. 2, this does not imply that the model is valid for making predictions. The reason is that the prediction QoI may be sensitive to some error or omission in the model that the observed quantities are not. To preclude this possibility and gain confidence in the prediction, further assessments of the validation process are needed. These are discussed in Sect. 3.2.2 below.

The opposite situation represents a different fundamental challenge in the validation of predictions. Even if a model is found to be inconsistent with available data, this does not imply that the model is invalid for making the desired predictions. The reason is that the prediction QoI may be insensitive to the error or omission in the model that caused the inconsistency with the observations. To determine whether a prediction can be made despite the errors in the model requires that the impact of the modeling errors on the predicted QoI be quantified. This quantification of uncertainty due to model inadequacy is discussed in Sect. 3.2.1.

3.2.1 Accounting for Model Error

If a discrepancy between a model and observations is detected, it may nonetheless be possible to make a reliable prediction, provided the impact of model error responsible for this discrepancy on the predicted QoI can be quantified. This can be difficult because there is no direct mapping from the observables to the QoIs – i.e., given only v , one cannot directly evaluate q . Referring to the problem statement in Sect. 3.1.1, it is clear that any model error must be due to the embedded model \mathcal{T} , since all the other components of the model (\mathcal{R} , \mathcal{V} , and \mathcal{Q}) are presumed to be reliable. In essence, the embedded model must be enriched to include a representation for the uncertainty introduced by model errors. In the simple case from Sect. 3.1.1, one could write

$$\tau \approx \mathcal{T}(u, s; \theta) + \mathcal{E}_\tau(u, s; \alpha), \quad (5.22)$$

where \mathcal{E}_τ is an uncertainty representation of the model error ε_τ , which may depend on additional parameters α . Given the choice to use probability to represent uncertainty, it is natural that \mathcal{E}_τ is a stochastic model, even when the physical phenomenon being modeled is inherently deterministic. Of course, an additive model is not necessary; other choices are possible. More importantly, the form of \mathcal{E}_τ must be specified. The specification of a stochastic model \mathcal{E}_τ is driven by physical knowledge about the nature of the error as well as practical considerations necessary to make computations with the model tractable. For example, when the enriched model (5.22) is introduced into (5.16) so that it can be solved for u , which is now stochastic, the fact that \mathcal{E}_τ depends on u will in general make this solution difficult. To ameliorate this problem, one can attempt to formulate \mathcal{E}_τ to be independent of u or to define \mathcal{E}_τ through an auxiliary equation of the form $f(u, \mathcal{E}_\tau; z) = 0$, where z is an auxiliary random variable that is independent of u . In this latter case, the auxiliary equation can then be solved together with (5.16). Other practical formulations for introducing u dependence in \mathcal{E}_τ may also be possible. Although general principles for developing physics-based uncertainty models need

to be developed, the specification of such a model is clearly problem-dependent and, thus, will not be discussed further here.

For the current purposes, it is sufficient to observe that the model \mathcal{E}_τ is posed at the source of the structural inadequacy – i.e., in the embedded model for τ . The combination of the physical and uncertainty models forms an enriched composite model, which takes the following form in the general case corresponding to (5.20):

$$\begin{aligned}\mathcal{R}(u, \{\mathcal{T}\}^0 + \{\mathcal{E}_\tau\}^0, r) &= 0 \\ q = \mathcal{Q}(u, \{\mathcal{T}\}^0 + \{\mathcal{E}_\tau\}^0, r) \\ \mathcal{R}^i(u^i, \{\mathcal{T}\}^i + \{\mathcal{E}_\tau\}^i, r^i) &= 0 \quad \text{for } 1 \leq i \leq N_e \\ v^i = \mathcal{V}^i(u^i, \{\mathcal{T}\}^i + \{\mathcal{E}_\tau\}^i, r^i) &\quad \text{for } 1 \leq i \leq N_e\end{aligned}\tag{5.23}$$

The inadequacy models, $\{\mathcal{E}_\tau\}^0$ and $\{\mathcal{E}_\tau\}^i$, appear naturally in the calculation of both the observables and the QoIs, both directly through the possible dependence of \mathcal{V}^i and \mathcal{Q} on embedded models and indirectly via the dependence of the state u on the embedded models appearing in \mathcal{R} . The structural uncertainty can therefore be propagated to both the observables and the QoIs without additional modeling assumptions. Furthermore, one can learn about the inadequacies – i.e., calibrate and test the corresponding models – from data on the observables and then transfer that knowledge to the prediction of the QoIs. This ability enables quantification of the impact of modeling inadequacies on the unobserved QoIs.

Enriching the embedded models with representations of the uncertainty due to model inadequacy is done with the goal of explaining all observed discrepancies between the model and observations. Therefore, with these enrichments included, the validation process discussed in Sect. 2 should reveal no inconsistencies with all relevant data. Once this is confirmed, there is no longer a validation failure, and one may proceed to evaluating whether the validation process is sufficient to warrant confidence in predictions of the QoIs.

3.2.2 Predictive Assessment

Since prediction requires extrapolation from available information, a prediction cannot be validated based on agreement between the predictive model (or some part of it) and data. This agreement alone is only sufficient to determine that the model is capable of predicting the observed quantities in the observed scenarios. To go beyond this, additional knowledge about the model and its relationship to both the validation experiments and the prediction are required. In particular, one must assess whether:

1. The calibration and validation of the embedded models are sufficient to give confidence in the prediction
2. The embedded models are being used within their domain of applicability
3. The resulting prediction with its uncertainties is sufficient for the purpose for which the prediction is being made

These predictive assessments are outlined in the following paragraphs and in more detail in [30].

Adequacy of Calibration and Validation

The fundamental issue in assessing the adequacy of the calibration and validation is whether the available data inform and challenge the model in ways that are relevant to the desired prediction. This assessment is necessarily based, at least in part, on knowledge regarding the physics of the problem. For example, in many domains, arguments based on dimensional analysis can help determine the relevance of an experiment on a scale model to the case of interest. Whenever possible, such information must be used. To augment such traditional analyses, one must consider whether the QoIs are sensitive to some characteristic of an embedded model, or the associated inadequacy model, that has not been adequately informed and tested in the preceding calibration and validation processes. In particular, if the QoIs are sensitive to an aspect of the model to which the data are insensitive, then the prediction depends in some important way on things that have not been constrained by the data. In this case, the prediction can only be credible if there is other reliable information that informs this aspect of the embedded models. To assess this then requires a sensitivity analysis to identify what is important about the embedded models for making the predictions. This sensitivity analysis is necessarily concerned with the sensitivities after calibration, because it is the calibrated model that is to be used for prediction. There are several ways in which the calibration and validation processes might be found to be insufficient. The most relevant examples are described briefly below:

1. Suppose that the prediction QoI is highly sensitive to one of the embedded models \mathcal{T} , as measured, for example, by the Fréchet derivative of the QoI with respect to \mathcal{T} at some representative θ . If none of the validation quantities are sensitive to \mathcal{T} , then the validation process has not provided a test of the validity of \mathcal{T} , and a prediction based on \mathcal{T} would be unreliable. More plausibly, it may be that none of the validation quantities for scenarios higher in the validation pyramid are sensitive to \mathcal{T} . The integration of \mathcal{T} into a composite model similar to that used in the predictions would then not have been tested, which would make its use in the prediction suspect [30]. To guard against this and similar possible failures of \mathcal{T} , the predictive assessment process should determine whether validation quantities in scenarios “close enough” to the prediction scenario are sufficiently sensitive to \mathcal{T} to provide a good test of its use in the prediction. The determination of what is “close enough” and what constitutes sufficient sensitivity must be made based on knowledge of the model and the approximations that went into it and of the way the models are embedded into the composite models of the validation and prediction scenarios.
2. Suppose that the prediction QoI is highly sensitive to the value of a particular parameter θ in an embedded model. In this case, it is important to determine whether the value of this parameter is well constrained by reliable information. If, for example, none of the calibration data has informed the value of θ , then

only other available information (prior information in the Bayesian context) has determined its value. Further, if none of the validation quantities are sensitive to the value of θ , then the validation process has not tested whether the information used to determine θ is in fact valid in the current context. The prediction QoI is then being determined to a significant extent by the untested prior information used to determine θ , which leaves little confidence in the reliability of the prediction, unless the prior information is itself highly reliable (e.g., θ is the speed of light). Alternatively, when the available prior information is questionable (e.g., θ is the reaction rate of a poorly understood chemical reaction), the predictions based on θ will not be reliable.

3. Suppose that uncertainty in the prediction QoI is largely due to the uncertainty model \mathcal{E} representing the inadequacy of the embedded model \mathcal{T} . In this case, it is important to ensure that \mathcal{E} is a valid description of the inadequacy of \mathcal{T} . As with the embedded model sensitivities discussed above, validation tests from high on the validation pyramid are most valuable for assessing whether the uncertainty model represents inadequacy in the context of a composite model similar to that for the prediction. If, however, the available validation data are for quantities that are insensitive to \mathcal{E} , then the veracity of \mathcal{E} in representing the uncertainty in the QoI will be suspect. Reliable predictions will then be possible only if there is independent information that the inadequacy representation is trustworthy.

Domain of Applicability of Embedded Models

In general, it is expected that the embedded models making up the composite model to be used in a prediction will involve various approximations and/or will have been informed by a limited set of calibration data. This will limit the range of scenarios for which the model can be considered reliable, either because the approximations will become invalid or because the model will be used outside the range for which it was calibrated. It is therefore clearly necessary to ensure that the embedded models are being used in a scenario regime in which they are expected to be reliable.

As discussed in Sect. 3.1, reliable extrapolative predictions are possible because the scenario parameters relevant to an embedded model need not be the same as those for the global composite model in which it is embedded. For example, when modeling the structural response of a building, the scenario parameters include the structural configuration and the loads. However, the scenario parameters for the linear elasticity embedded model used for the internal stresses would be the local magnitude of the strain, as well as other local variables such as the temperature. For each embedded model then, one needs to identify the scenario parameters that characterize the applicability of the model and the range of those parameters over which the model and its calibration are expected to be reliable. It is then a simple matter of checking the solution of the composite model to see if any of the embedded models are being used “out of range.” For some embedded models, defining the range of applicability in this way is straightforward. However, for some types of embedded models – e.g., an embedded model that involves an additional equation

that has nonlocal dependence on the state – defining the relevant scenario space and, hence, the region of scenario space that defines the domain of applicability is significantly more difficult.

Sufficiency of the Prediction and Uncertainties

The focus of the previous assessments is on ensuring that the calibration and validation processes have been sufficiently rigorous to warrant confidence in an extrapolative prediction and its uncertainty. However, a prediction with an uncertainty that is too large to inform the decision for which the prediction is being performed is not sufficient, even if that uncertainty has been determined to be a good representation of what can be predicted about the QoI. The requirements for prediction uncertainty to inform a decision based on the prediction depend on the nature of the decision, and determination of this requirement is outside the scope of the current discussion. However, once such a requirement is known, the prediction uncertainties can be checked to determine whether these requirements are met and therefore whether the prediction is useful.

Of course, when the prediction uncertainty fails to meet the established tolerance, some action must be taken to reduce this uncertainty. While a full discussion of this process is beyond the scope of the current discussion, the predictive validation activities previously described provide a wealth of information that can provide guidance as to how to proceed. For example, parameters that have large posterior uncertainty and that are influential to the QoIs are good candidates for further calibration based on new experiments. Alternatively, embedded models for which the associated inadequacy model introduces significant uncertainty are good candidates for new model development.

A Major Caveat

The predictive assessment process can determine whether, given what is known about the system, the calibration and validation processes are sufficient to make a reliable prediction. But the well-known problem of “unknown unknowns” remains. If the system being simulated involves an unrecognized phenomenon, then clearly an embedded model to represent it will not be included in the composite model for the system. As with the examples above, the prediction QoI could be particularly sensitive to this phenomenon, while the validation observables are not sensitive. In this situation, one would not be able to detect that anything is missing from the composite model. Further one could not even identify that the validation observables were insufficient, that is, the predictive assessment could not detect the inadequacy of the validation process. This is a special case of a broader issue. The predictive validation process developed here relies explicitly on reliable knowledge about the system and the models used to represent it. This knowledge is considered to not need independent validation and is thus what allows for extrapolative predictions. However, if this externally supplied information is in fact incorrect, then the predictive validation process may not be able to detect it.

4 Conclusions and Challenges

As the importance of computational simulations in science and engineering continues to grow, so does the importance of validating the physical models that form the basis of those simulations. Validation is traditionally defined as a comparison between model outputs and experimental observations intended to reveal any important discrepancies between the model and reality. To make this process rigorous, one must account for uncertainties that affect the experimental observations and the computational results. Thus, in order to draw validation conclusions, it is necessary to define metrics that measure the agreement or lack thereof between uncertain experimental observations and uncertain model outputs. When these uncertainties are represented using probability, a number of such “validation metrics” are available, including highest posterior density credibility intervals and Bayesian p -values, both of which can be used in combination with appropriately chosen test quantities when necessary or desirable.

When the purpose of the computational simulation is prediction, agreement between uncertain model outputs and available uncertain data is in general necessary but not sufficient for validating the prediction because prediction requires extrapolation. In this situation, predictive validation is a process for building confidence in simulation-based predictions by exploiting typical features of physics-based models.

A number of issues remain before systematic validation methodologies like those described here can become standard in computational science and engineering. First, all of the ideas described here depend heavily on the development of probabilistic models to represent uncertainties. In some situations, such as when abundant sample data are available for an aleatorically uncertain variable, these models are straightforward to build. However, in many cases, particularly those involving complex epistemic uncertainties, this process is less clear. For example, general techniques and best practices for representing uncertainty due to model inadequacy, particularly when the modeled quantity is a field, and for representing correlations between experimental measurements when few replications are available must be developed. These difficulties are often related to the more general problem of representing qualitative information such as expert opinion that, while often crucial in accurately characterizing likely values of epistemic parameters or realistic modeling errors, can be challenging to represent quantitatively in a defensible manner.

Second, the methods discussed here require uncertainty propagation through the models being validated. When these models are computationally expensive and/or the space of uncertain variables is high dimensional, it is well known that typical algorithms, such as Monte Carlo sampling or stochastic collocation, often require too many forward model evaluations to be computationally tractable. Better algorithms are necessary to enable routine uncertainty analysis using complex models. The required algorithmic advances are also necessary to enable routine validation of these models.

References

1. Adams, B.M., Ebeida, M.S., Eldred, M.S., Others: Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.2 User's Manual. Sandia National Laboratories, Albuquerque (2014). <https://dakota.sandia.gov/documentation.html>
2. AIAA Computational Fluid Dynamics Committee on Standards: AIAA Guide for Verification and Validation of Computational Fluid Dynamics Simulations. AIAA Paper number G-077-1999 (1998)
3. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. *Stat. Surv.* **4**, 40–79 (2010). doi:10.1214/09-SS054, <http://dx.doi.org/10.1214/09-SS054>
4. ASME Committee V&V 10: Standard for Verification and Validation in Computational Solid Mechanics. ASME (2006)
5. Babuška, I., Nobile, F., Tempone, R.: Reliability of computational science. *Numer. Methods Partial Differ. Equ.* **23**(4), 753–784 (2007). doi:10.1002/num.20263
6. Box, G.E.P.: Sampling and Bayes' inference in scientific modeling and robustness. *R. Stat. Soc. Ser. A* **143**, 383–430 (1980)
7. Box, G., Tiao, G.C.: Bayesian Inference in Statistical Analysis. Wiley Classics, New York (1973)
8. Cox, R.T.: The Algebra of Probable Inference. Johns Hopkins University Press, Baltimore (1961)
9. Cui, T., Martin, J., Marzouk, Y.M., Solonen, A., Spantini, A.: Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Probl.* **30**(11), 114015 (2014)
10. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1988)
11. Ferson, S., Ginzburg, L.R.: Different methods are needed to propagate ignorance and variability. *Reliab. Eng. Syst. Saf.* **54**, 133–144 (1996)
12. Fine, T.L.: Theories of Probability. Academic, New York (1973)
13. Firm uses doe's fastest supercomputer to streamline long-haul trucks. Office of Science, U.S. Department of Energy, Stories of Discovery and Innovation (2011). <http://science.energy.gov/discovery-and-innovation/stories/2011/127008/>
14. Gelman, A.: Comment: ‘Bayesian checking of the second levels of hierarchical models’. *Stat. Sci.* **22**, 349–352 (2007). doi:doi:10.1214/07STS235A
15. Gelman, A., Rubin, D.B.: Avoiding model selection in Bayesian social research. *Sociol. Methodol.* **25**, 165–173 (1995)
16. Gelman, A., Shalizi, C.R.: Philosophy and the practice of Bayesian statistics. *Br. J. Math. Stat. Psychol.* **66**(1), 8–38 (2013)
17. Gelman, A., Meng, X.L., Stern, H.: Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica* **6**, 733–807 (1996)
18. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis, 3rd edn. CRC Press, Boca Raton (2014)
19. Hsu, J.: Multiple Comparisons: Theory and Methods. Chapman and Hall/CRC, London (1996)
20. Hyndman, R.J.: Computing and graphing highest density regions. *Am. Stat.* **50**(2), 120–126 (1996)
21. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge/New York (2003)
22. Kanji, G.K.: 100 Statistical Tests, 3rd edn. Sage Publications, London/Thousand Oaks (2006)
23. Le Maître, O., Knio, O., Najm, H., Ghanem, R.: Uncertainty propagation using wiener-haar expansions. *J. Comput. Phys.* **197**(1), 28–57 (2004)
24. Li, J., Marzouk, Y.M.: Adaptive construction of surrogates for the Bayesian solution of inverse problems. *SIAM J. Sci. Comput.* **36**(3), A1163–A1186 (2014)
25. Miller, R.G.J.: Simultaneous Statistical Inference, 2nd edn. Springer, New York (1981)

26. Miller, L.K.: Simulation-based engineering for industrial competitive advantage. *Comput. Sci. Eng.* **12**(3), 14–21 (2010). doi:10.1109/MCSE.2010.71
27. Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **41**, 35–52 (2009)
28. Oberkampf, W.L., Helton, J.C., SENTZ, K.: Mathematical representation of uncertainty. AIAA 2001-1645
29. Oden, J.T., Belytschko, T., Fish, J., Hughes, T.J.R., Johnson, C., Keyes, D., Laub, A., Petzold, L., Srolovitz, D., Yip, S.: Revolutionizing engineering science through simulation: a report of the National Science Foundation blue ribbon panel on simulation-based engineering science (2006). http://www.nsf.gov/pubs/reports/sbes_final_report.pdf
30. Oliver, T.A., Terejanu, G., Simmons, C.S., Moser, R.D.: Validating predictions of unobserved quantities. *Comput. Methods Appl. Mech. Eng.* **283**, 1310–1335 (2015). doi:<http://dx.doi.org/10.1016/j.cma.2014.08.023>, <http://www.sciencedirect.com/science/article/pii/S004578251400293X>
31. Petra, N., Martin, J., Stadler, G., Ghattas, O.: A computational framework for infinite-dimensional Bayesian inverse problems, Part II: stochastic Newton memc with application to ice sheet flow inverse problems. *SIAM J. Sci. Comput.* **36**(4), A1525–A1555 (2014)
32. Rubin, D.B.: Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Ann. Stat.* **12**, 1151–1172 (1984)
33. SENTZ, K., Ferson, S.: Combination of evidence in Dempster–Shafer theory. Technical report SAND 2002-0835, Sandia National Laboratory (2002)
34. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
35. Van Horn, K.S.: Constructing a logic of plausible inference: a guide to Cox’s theorem. *Int. J. Approx. Reason.* **34**(1), 3–24 (2003)

Houman Owhadi and Clint Scovel

Abstract

The past century has seen a steady increase in the need of estimating and predicting complex systems and making (possibly critical) decisions with limited information. Although computers have made possible the numerical evaluation of sophisticated statistical models, these models are still designed *by humans* because there is currently no known recipe or algorithm for dividing the design of a statistical model into a sequence of arithmetic operations. Indeed enabling computers to *think as humans*, especially when faced with uncertainty, is challenging in several major ways: (1) Finding optimal statistical models remains to be formulated as a well-posed problem when information on the system of interest is incomplete and comes in the form of a complex combination of sample data, partial knowledge of constitutive relations and a limited description of the distribution of input random variables. (2) The space of admissible scenarios along with the space of relevant information, assumptions, and/or beliefs, tends to be infinite dimensional, whereas calculus on a computer is necessarily discrete and finite. With this purpose, this paper explores the foundations of a rigorous framework for the scientific computation of optimal statistical estimators/models and reviews their connections with decision theory, machine learning, Bayesian inference, stochastic optimization, robust optimization, optimal uncertainty quantification, and information-based complexity.

H. Owhadi (✉) • C. Scovel

Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA
e-mail: owhadi@caltech.edu; clintscovel@gmail.com

Keywords

Abraham Wald • Decision theory • Machine learning • Uncertainty quantification • Game theory

Contents

1	Introduction	158
2	The UQ Problem Without Sample Data	160
2.1	Čebyšev, Markov, and Krein	160
2.2	Optimal Uncertainty Quantification	161
2.3	Worst-Case Analysis	163
2.4	Stochastic and Robust Optimization	163
2.5	Čebyšev Inequalities and Optimization Theory	164
3	The UQ Problem with Sample Data	165
3.1	From Game Theory to Decision Theory	165
3.2	The Optimization Approach to Statistics	166
3.3	Abraham Wald	167
3.4	Generalization to Unknown Pairs of Functions and Measures and to Arbitrary Sample Data	170
3.5	Model Error and Optimal Models	170
3.6	Mean Squared Error, Variance, and Bias	171
3.7	Optimal Interval of Confidence	172
3.8	Ordering the Space of Experiments	172
3.9	Mixing Models	172
4	The Complete Class Theorem and Bayesian Inference	173
4.1	The Bayesian Approach	173
4.2	Relation Between Adversarial Model Error and Bayesian Error	174
4.3	Complete Class Theorem	175
5	Incorporating Complexity and Computation	177
5.1	Machine Wald	178
5.2	Reduction Calculus	178
5.3	Stopping Conditions	179
5.4	On the Borel-Kolmogorov Paradox	179
5.5	On Bayesian Robustness/Brittleness	180
5.6	Information-Based Complexity	181
6	Conclusion	182
Appendix		182
Construction of $\pi \odot \mathbb{D}$		182
Proof of Theorem 2		183
Conditional Expectation as an Orthogonal Projection		184
References		185

1 Introduction

During the past century, the need to solve large complex problems in applications such as fluid dynamics, neutron transport, or ballistic prediction drove the parallel development of computers and numerical methods for solving ODEs and PDEs. It is now clear that this development lead to a paradigm shift. Before: each new PDE required the development of new theoretical methods and the employment of large teams of mathematicians and physicists; in most cases, information on solutions was

only qualitative and based on general analytical bounds on fundamental solutions. After: mathematical analysis and computer science worked in synergy to give birth to robust numerical methods (such as finite element methods) capable of solving a large spectrum of PDEs without requiring the level of expertise of an A. L. Cauchy or level of insight of a R. P. Feynman. This transformation can be traced back to sophisticated calculations performed by arrays of *human computers* organized as parallel clusters such as in the pioneering work of Lewis Fry Richardson [1, 90], who in 1922 had a room full of clerks attempt to solve finite-difference equations for the purposes of weather forecasting, and the 1947 paper by John Von Neumann and Herman Goldstine on Numerical Inverting of Matrices of High Order [154]. Although Richardson's predictions failed due to the use of unfiltered data/initial conditions/equations and large time-steps not satisfying the CFL stability condition [90], his vision was shared by Von Neumann [90] in his proposal of the Meteorology Research Project to the US Navy in 1946, qualified by Platzman [120] as "perhaps the most visionary prospectus for numerical weather prediction since the publication of Richardsons book a quarter-century earlier."

The past century has also seen a steady increase in the need of estimating and predicting complex systems and making (possibly critical) decisions with limited information. Although computers have made possible the numerical evaluation of sophisticated statistical models, these models are still designed by *humans* through the employment of multidisciplinary teams of physicists, computer scientists, and statisticians. Contrary to the original *human computers* (such as the ones pioneered by L. F. Richardson or overseen by R. P. Feynman at Los Alamos), these *human teams* do not follow a specific algorithm (such as the one envisioned in Richardson's Forecast Factory where 64,000 human computers would have been working in parallel and at high speed to compute world weather charts [90]) because there is currently no known recipe or algorithm for dividing the design of a statistical model into a sequence of arithmetic operations. Furthermore, while *human computers* were given a specific PDE or ODE to solve, these *human teams* are not given a well-posed problem with a well-defined notion of solution. As a consequence, different *human teams* come up with different *solutions* to the design of the statistical model along with different estimates on uncertainties.

Indeed enabling computers to *think* as *humans*, especially when faced with uncertainty, is challenging in several major ways: (1) There is currently no known recipe or algorithm for dividing the design of a statistical model into a sequence of arithmetic operations. (2) Formulating the search for an optimal statistical estimator/model as a well-posed problem is not obvious when information on the *system of interest* is incomplete and comes in the form of a complex combination of sample data, partial knowledge of constitutive relations, and a limited description of the distribution of input random variables. (3) The space of admissible scenarios along with the space of relevant information, assumptions, and/or beliefs tends to be infinite dimensional, whereas calculus on a computer is necessarily discrete and finite.

The purpose of this paper is to explore the foundations of a rigorous/rational framework for the scientific computation of optimal statistical estimators/models for complex systems and review their connections with decision theory, machine

learning, Bayesian inference, stochastic optimization, robust optimization, optimal uncertainty quantification, and information-based complexity, the most fundamental of these being the simultaneous emphasis on *computation* and *performance* as in machine learning initiated by Valiant [149].

2 The UQ Problem Without Sample Data

2.1 Čebyšev, Markov, and Krein

Let us start with a simple warm-up problem.

Problem 1. Let \mathcal{A} be the set of measures of probability on $[0, 1]$ having mean less than $m \in (0, 1)$. Let μ^\dagger be an unknown element of \mathcal{A} and let $a \in (m, 1)$. What is $\mu^\dagger[X \geq a]$?

Observe that given the limited information on μ^\dagger , $\mu^\dagger[X \geq a]$ could a priori be any number in the interval $[\mathcal{L}(\mathcal{A}), \mathcal{U}(\mathcal{A})]$ obtained by computing the sup (inf) of $\mu[X \geq a]$ with respect to all possible candidates for μ^\dagger , i.e.,

$$\mathcal{U}(\mathcal{A}) := \sup_{\mu \in \mathcal{A}} \mu[X \geq a] \quad (6.1)$$

and

$$\mathcal{L}(\mathcal{A}) := \inf_{\mu \in \mathcal{A}} \mu[X \geq a]$$

where

$$\mathcal{A} := \{\mu \in \mathcal{M}([0, 1]) \mid \mathbb{E}_\mu[X] \leq m\}$$

and $\mathcal{M}([0, 1])$ is the set of Borel probability measures on $[0, 1]$. It is easy to observe that the extremum of (6.1) can be achieved only when μ is the weighted sum of a Dirac mass at 0 and a Dirac mass at a . It follows that, although (6.1) is an infinite dimensional optimization problem, it can be reduced to the simple one-dimensional optimization problem obtained by letting $p \in [0, 1]$ denote the weight of the Dirac mass at 1 and $1 - p$ the weight of the Dirac mass at 0: *Maximize p subject to $ap = m$* , producing the Markov bound $\frac{m}{a}$ as solution.

Problems such as (1) can be traced back to Čebyšev [77, Pg. 4] “Given: length, weight, position of the centroid and moment of inertia of a material rod with a density varying from point to point. It is required to find the most accurate limits for the weight of a certain segment of this rod.” According to Krein [77], although Čebyšev did solve this problem, it was his student Markov who supplied the proof in his thesis. See Krein [77] for an account of the history of this subject along with substantial contributions by Krein.

2.2 Optimal Uncertainty Quantification

The generalization of the process described in Sect. 2.1 to complex systems involving imperfectly known functions and measures is the point of view of optimal uncertainty quantification (OUQ) [3, 69, 72, 96, 114, 142]. Instead of developing sophisticated mathematical solutions, the OUQ approach is to develop optimization problems and reductions, so that their solution may be implemented on a computer, as in Bertsimas and Popescu's [15] convex optimization approach to Čebyšev inequalities, and the Decision Analysis framework of Smith [133].

To present this generalization, for a topological space \mathcal{X} , let $\mathcal{F}(\mathcal{X})$ be the space of real-valued measurable functions and $\mathcal{M}(\mathcal{X})$ be the set of Borel probability measures on \mathcal{X} . Let \mathcal{A} be an arbitrary subset of $\mathcal{F}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$, and let $\Phi: \mathcal{A} \rightarrow \mathbb{R}$ be a function producing a quantity of interest.

Problem 2. Let (f^\dagger, μ^\dagger) be an unknown element of \mathcal{A} . What is $\Phi(f^\dagger, \mu^\dagger)$?

Therefore, in the absence of sample data, in the context of this generalization, one is interested in estimating $\Phi(f^\dagger, \mu^\dagger)$, where $(f^\dagger, \mu^\dagger) \in \mathcal{F}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$ corresponds to an *unknown reality*: the function f^\dagger represents a *response function* of interest, and μ^\dagger represents the probability distribution of the inputs of f^\dagger . If \mathcal{A} represents all that is known about (f^\dagger, μ^\dagger) (in the sense that $(f^\dagger, \mu^\dagger) \in \mathcal{A}$ and that any $(f, \mu) \in \mathcal{A}$ could, a priori, be (f^\dagger, μ^\dagger) given the available information), then [114] shows that the quantities

$$\mathcal{U}(\mathcal{A}) := \sup_{(f, \mu) \in \mathcal{A}} \Phi(f, \mu) \quad (6.2)$$

$$\mathcal{L}(\mathcal{A}) := \inf_{(f, \mu) \in \mathcal{A}} \Phi(f, \mu) \quad (6.3)$$

determine the inequality

$$\mathcal{L}(\mathcal{A}) \leq \Phi(f^\dagger, \mu^\dagger) \leq \mathcal{U}(\mathcal{A}), \quad (6.4)$$

to be optimal given the available information $(f^\dagger, \mu^\dagger) \in \mathcal{A}$ as follows: It is simple to see that the inequality (6.4) follows from the assumption that $(f^\dagger, \mu^\dagger) \in \mathcal{A}$. Moreover, for any $\varepsilon > 0$, there exists a $(f, \mu) \in \mathcal{A}$ such that

$$\mathcal{U}(\mathcal{A}) - \varepsilon < \Phi(f, \mu) \leq \mathcal{U}(\mathcal{A}).$$

Consequently since all that is known about (f^\dagger, μ^\dagger) is that $(f^\dagger, \mu^\dagger) \in \mathcal{A}$, it follows that the upper bound $\Phi(f^\dagger, \mu^\dagger) \leq \mathcal{U}(\mathcal{A})$ is the best obtainable given that information, and the lower bound is optimal in the same sense.

Although the OUQ optimization problems (6.2) and (6.3) are extremely large and although some are computationally intractable, an important subclass enjoys significant and practical finite-dimensional reduction properties [114]. First, by [114, Cor. 4.4], although the optimization variables (f, μ) lie in a product space of functions and probability measures, for OUQ problems governed by linear

inequality constraints on generalized moments, the search can be reduced to one over probability measures that are products of finite convex combinations of Dirac masses with explicit upper bounds on the number of Dirac masses.

Furthermore, in the special case that all constraints are generalized moments of functions of f , the dependency on the coordinate positions of the Dirac masses is eliminated by observing that the search over admissible functions reduces to a search over functions on an m -fold product of finite discrete spaces, and the search over m -fold products of finite convex combinations of Dirac masses reduces to a search over the products of probability measures on this m -fold product of finite discrete spaces [114, Thm. 4.7]. Finally, by [114, Thm. 4.9], using the lattice structure of the space of functions, the search over these functions can be reduced to a search over a finite set.

Fundamental to this development is Winkler's [169] generalization of the characterization of the extreme points of compact (in the weak topology) sets of probability measures constrained by a finite number of generalized moment inequalities defined by continuous functions to *non-compact sets of tight measures*, in particular probability measures on Borel subsets of Polish metric spaces, defined by Borel measurable moment functions, along with his [168] development of Choquet theory for weakly closed convex *non-compact* sets of tight measures. These results are based on Kendall's [71] equivalence between a linearly compact Choquet simplex and a vector lattice and results of Dubins [31] concerning the extreme points of affinely constrained convex sets in terms of the extreme points of the unconstrained set. It is interesting to note that Winkler [169] uses Kendall's result to derive a strong sharpening of Dubins result [31]. Winkler's results allow the extension of existing optimization results over measures on compact metric spaces constrained by continuous generalized moment functions to optimization over measures on Borel subsets of Polish spaces constrained by Borel measurable moment functions. For systems with symmetry, the Choquet theorem of Varadarajan [151] can be used to show that the Dirac masses can be replaced by the ergodic measures in these results. The inclusion of sets of functions along with sets of measures in the optimization problems facilitates the application to systems with imprecisely known response functions. In particular, a result of Ressel [121], providing conditions under which the map $(f, \mu) \rightarrow f_*\mu$ from function/measure pairs to the induced law is Borel measurable, facilitates the extension of these techniques from sets of *measures* to sets of *random variables*. In general, the inclusion of functions in the domain of optimization requires the development of generalized programming techniques such as generalized Benders decompositions described in Geoffrion [46]. Moreover, as has been so successful in machine learning, it will be convenient to approximate the space of measurable functions $\mathcal{F}(\mathcal{X})$ by some reproducing kernel Hilbert space $\mathcal{H}(\mathcal{X}) \subset \mathcal{F}(\mathcal{X})$ producing an approximation $\mathcal{H}(\mathcal{X}) \times \mathcal{M}(\mathcal{X}) \subset \mathcal{F}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$ to the full base space. Under the mild assumption that \mathcal{X} is an analytic subset of a Polish space and $\mathcal{H}(\mathcal{X})$ possesses a measurable feature map, it has recently been shown in [111] that $\mathcal{H}(\mathcal{X})$ is separable. Consequently, since all separable Hilbert spaces are isomorphic with ℓ^2 , it follows that the space $\ell^2 \times \mathcal{M}(\mathcal{X})$ is a universal representation space for the approximation of $\mathcal{F}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$. Moreover, in that case, since \mathcal{X} is separable and metric, so is $\mathcal{M}(\mathcal{X})$ in the weak topology, and since

$\mathcal{H}(\mathcal{X})$ is Polish, it follows that the approximation space $\mathcal{H}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$ is the product of a Polish space and a separable metric space. When furthermore \mathcal{X} is Polish, it follows that the approximation space is the product of Polish spaces and therefore Polish.

Example 1. A classic example is $\Phi(f, \mu) := \mu[f \geq a]$ where a is a safety margin. In the certification context, one is interested in showing that $\mu^\dagger[f^\dagger \geq a] \leq \varepsilon$, where ε is a safety certification threshold (i.e., the maximum acceptable μ^\dagger -probability of the system f^\dagger exceeding the safety margin a). If $\mathcal{U}(\mathcal{A}) \leq \varepsilon$, then the system associated with (f^\dagger, μ^\dagger) is safe even in the worst-case scenario (given the information represented by \mathcal{A}). If $\mathcal{L}(\mathcal{A}) > \varepsilon$, then the system associated with (f^\dagger, μ^\dagger) is unsafe even in the best-case scenario (given the information represented by \mathcal{A}). If $\mathcal{L}(\mathcal{A}) \leq \varepsilon < \mathcal{U}(\mathcal{A})$, then the safety of the system cannot be decided (although one could declare the system to be unsafe due to lack of information).

2.3 Worst-Case Analysis

The proposed solutions to Problems 1 and 2 are particular instances of worst-case analysis that, as noted by [135] and [127, p.5], is an old concept that could be summarized by the popular adage *When in doubt, assume the worst!* or:

The gods to-day stand friendly, that we may,
 Lovers of peace, lead on our days to age
 But, since the affairs of men rests still uncertain,
 Lets reason with the worst that may befall.

Julius Caesar, Act 5, Scene 1
 William Shakespeare (1564–1616)

As noted in [114], an example of worst-case analysis in seismic engineering is that of Drenick's *critical excitation* [30] which seeks to quantify the safety of a structure to the worst earthquake given a constraint on its magnitude. The combination of structural optimization (in various fields of engineering) to produce an optimal design given the (deterministic) worst-case scenario has been referred to as *optimization and anti-optimization* [35]. The main difference between OUQ and anti-optimization lies in the fact that the former is based on an optimization over (admissible) functions and measures (f, μ) , while the latter only involves an optimization over f . Because of its robustness, many engineers have adopted the (deterministic) worst-case scenario approach to UQ [35, Chap. 10] when a high reliability is required.

2.4 Stochastic and Robust Optimization

Robust control [176] and robust optimization [7, 14] have been founded upon the worst-case approach to uncertainty. Recall that robust optimization describes optimization involving uncertain parameters. While these uncertain

parameters are modeled as random variables (of known distribution) in stochastic programming [26], robust optimization only assumes that they are contained in known (ambiguity) sets. Although, as noted in [35], *probabilistic methods do not find appreciation among theoreticians and practitioners alike* because “probabilistic reliability studies involve assumptions on the probability densities, whose knowledge regarding relevant input quantities is central,” the deterministic worst-case approach (limited to optimization problems over f) is sometimes “too pessimistic to be practical” [30, 35] because “it does not take into account the *improbability* that (possibly independent or weakly correlated) random variables conspire to produce a failure event” [114] (which constitutes one motivation for considering ambiguity sets involving both measures and functions). Therefore OUQ and *distributionally robust optimization* (DRO) [7, 14, 49, 53, 166, 174, 177] could be seen as middle ground between the deterministic worst-case approach of robust optimization [7, 14] and approaches of *stochastic programming* and *chance-constrained optimization* [19, 24] by defining optimization objectives and constraints in terms of expected values and probabilities with respect to imperfectly known distributions.

Although stochastic optimization has different objectives than OUQ and DRO, many of its optimization results, such as those found in Birge and Wets [16], and Ermoliev [36] and Gaivoronski, [44], are useful. In particular, the well-developed subject of Edmundson and Madansky bounds such as Edmundson [34]; Madansky [91, 92]; Gassman and Ziemba [45]; Huang, Ziemba, and Ben-Tal [57]; Frauendorfer [41]; Ben-Tal and Hochman [8]; Huang, Vertinsky, and Ziemba [56]; and Kall [67] provide powerful results. Recently Hanasusanto, Roitch, Kuhn, and Wiesemann [53] derive explicit conic reformulations for tractable problem classes and suggest efficiently computable conservative approximations for intractable ones. In some cases, e.g., Bertsimas and Popescu’s [15] and Han et al. [52], DRO/OUQ optimization problems can be reduced to convex optimization.

2.5 Čebyšev Inequalities and Optimization Theory

As noted in [114], inequalities (6.4) can be seen as a generalization of Čebyšev inequalities. The history of classical inequalities can be found in [70], and some generalizations in [15] and [150]; in the latter works, the connection between Čebyšev inequalities and optimization theory is developed based on the work of Mulholland and Rogers [98], Godwin [48], Isii [60–62], Olkin and Pratt [106], Marshall and Olkin [94], and the classical Markov–Krein theorem [70, pages 82 & 157], among others. We also refer to the field of majorization, as discussed in Marshall and Olkin [95], the inequalities of Anderson [5], Hoeffding [54], Joe [64], Bentkus et al. [12], Bentkus [10, 11], Pinelis [118, 119], and Boucheron, Lugosi, and Massart [20]. Moreover, the solution of the resulting nonconvex optimization problems benefit from duality theories for nonconvex optimization problems such as Rockafellar [123] and the development of convex envelopes for them, as can be found, for example, in Rikun [122] and Sherali [131].

3 The UQ Problem with Sample Data

3.1 From Game Theory to Decision Theory

To motivate the general formulation in the presence of sample data, consider another simple warm-up problem.

Problem 3. Let \mathcal{A} be the set of measures of probability on $[0, 1]$ having mean less than $m \in (0, 1)$. Let μ^\dagger be an unknown element of \mathcal{A} and let $a \in (m, 1)$. You observe $d := (d_1, \dots, d_n)$, n i.i.d. samples from μ^\dagger . What is the *sharpest estimate* of $\mu^\dagger[X \geq a]$?

The only difference between Problems 3 and 1 lies in the availability of data sampled from the underlying unknown distribution. Observe that, in presence of this sample data, the notions of *sharpest estimate* or *smallest interval of confidence* are far from being transparent and call for clear and precise definitions. Note also that if the constraint $\mathbb{E}_{\mu^\dagger}[X] \leq m$ is ignored, and the number n of sample data is large, then one could use the central limit theorem or a concentration inequality (such as Hoeffding's inequality) to derive an interval of confidence for $\mu^\dagger[X \geq a]$. A nontrivial question of practical importance is what to do when n is not large.

Writing $\Phi(\mu^\dagger) := \mu^\dagger[X \geq a]$ as the quantity of interest, observe that an estimation of $\Phi(\mu^\dagger)$ is a function (which will be written θ) of the data d . Ideally one would like to choose θ so that the estimation error $\theta(d) - \Phi(\mu^\dagger)$ is as close as possible to zero. Since d is random, a more robust notion of error is that of a statistical error $\mathcal{E}(\theta, \mu^\dagger)$ defined by weighting the error with respect to a real measurable positive loss function $V: \mathbb{R} \rightarrow \mathbb{R}$ and the distribution of the data, i.e.,

$$\mathcal{E}(\theta, \mu^\dagger) := \mathbb{E}_{d \sim (\mu^\dagger)^n} [V[\theta(d) - \Phi(\mu^\dagger)]] \quad (6.5)$$

Note that for $V(x) = x^2$, the statistical error $\mathcal{E}(\theta, \mu^\dagger)$ defined in (6.5) is the mean squared error with respect to the distribution of the data d of the estimation error. For $V(x) = \mathbb{1}_{[\gamma, \infty)}(|x|)$ defined for some $\gamma > 0$, $\mathcal{E}(\theta, \mu^\dagger)$ is the probability with respect to the distribution of d that the estimation error is larger than γ .

Now since μ^\dagger is unknown, the statistical error $\mathcal{E}(\theta, \mu^\dagger)$ of any θ is also unknown. However one can still identify the least upper bound on that statistical error through a worst-case scenario with respect to all possible candidates for μ^\dagger , i.e.,

$$\sup_{\mu \in \mathcal{A}} \mathcal{E}(\theta, \mu). \quad (6.6)$$

The sharpest estimator (possibly within a given class) is then naturally obtained as the minimizer of (6.6) over all functions θ of the data d within that class, i.e., as the minimizer of

$$\inf_{\theta} \sup_{\mu \in \mathcal{A}} \mathcal{E}(\theta, \mu). \quad (6.7)$$

Observe that the optimal estimator is identified independently from the observation/realization of the data and if the minimum of (6.7) is not achieved then one can still use a near-optimal θ . Then, when the data is observed, the estimate of the quantity of interest $\Phi(\mu^\dagger)$ is then derived by evaluating the near-optimal θ on the data d . The notion of optimality described here is that of Wald's statistical decision theory [156–158, 160, 161], evidently influenced by Von Neumann's game theory [153, 155]. In Wald's formulation [157], which cites both Von Neumann [153] and Von Neumann and Morgenstern [155], the statistician finds himself in an adversarial game played against the Universe in which he tries to minimize a risk function $\mathcal{E}(\theta, \mu)$ with respect to θ in a worst-case scenario with respect to what the Universe's choice of μ could be.

3.2 The Optimization Approach to Statistics

The optimization approach to statistics is not new and this section will now give a short, albeit incomplete, description of its development, primarily using Lehmann's account [87]. Accordingly, it began with Gauss and Laplace with the nonparametric result referred to as the Gauss-Markov theorem, asserting that the least squares estimates are the linear unbiased estimates with minimum variance. Then, in Fisher's fundamental paper [39], for parametric models, he proposes the maximum likelihood estimator and claims (but does not prove) that such estimators are consistent and asymptotically efficient. According to Lehmann, "the situation is complex, but under suitable restrictions Fisher's conjecture is essentially correct . . ." The Fisher's maximum likelihood principle was first proposed on intuitive grounds and then its optimality properties developed. However, according to Lehmann [86, Pg. 1011], Pearson then asked Neyman "Why these tests rather than any of the many other that could be proposed? This question resulted in Neyman and Pearson's 1928 papers [104] on the likelihood ratio method, which gives the same answer as Fisher's tests under normality assumptions. However, Neyman was not satisfied. He agreed that the likelihood ratio principle was appealing but felt that it was lacking a logically convincing justification. This then led to the publication of Neyman and Pearson [105], containing their now famous Neyman-Pearson lemma, which according to Lehmann [87], "In a certain sense this is the true start of optimality theory." In a major extension of the Neyman-Pearson work, Huber [58] proves a *robust* version of the Neyman-Pearson lemma, in particular, providing an optimality criteria defining the robust estimator, giving rise to a rigorous theory of robust statistics based on optimality; see Huber's Wald lecture [59]. Robustness is particularly suited to the Wald framework since robustness considerations are easily formulated with the proper choices of admissible functions and measures in the Wald framework. Another example is Kiefer's introduction of optimality into experimental design, resulting in Kiefer's 40 papers on Optimum Experimental Designs [74].

Not everyone was happy with "optimality" as a guiding principle. For example, Lehmann [87] states that at a 1958 meeting of the Royal Statistical Society at which Kiefer presented a survey talk [73] on Optimum Experimental Designs, Barnard

quotes Kiefer as saying of procedures proposed in a paper by Box and Wilson that they “often [are] not even well-defined rules of operation.” Barnard’s reply:

in the field of practical human activity, rules of operation which are not well-defined may be preferable to rules which are.

Wynn [173], in his introduction to a reprint of Kiefer’s paper, calls this “a clash of statistical cultures.” Indeed, it is interesting to read the generally negative responses to Kiefer’s article [73] and the remarkable rebuttal by Kiefer therein. Tukey had other criticisms regarding “The tyranny of the best” in [147] and “The dangers of optimization” in [148]. In the latter he writes:

Some [statisticians] seem to equate [optimization] to statistics an attitude which, if widely adopted, is guaranteed to produce a dried-up, encysted field with little chance of real growth.

For an account of how the Exploratory Data Analysis approach of Tukey fits within the Fisher/Neyman–Pearson debate, see Lehnard [88].

Let us also remark on the influence that Student – William Sealy Gosset – had on both Fisher and Pearson. As presented in Lehmann’s [85] “‘Student’ and small-sample theory,” quoting F. N. David [79]: “I think he [Gosset] was really the big influence in statistics. . . He asked the questions and Pearson or Fisher put them into statistical language and then Neyman came to work with the mathematics. But I think most of it stems from Gosset.” The aim of Lehmann’s paper [85] is to consider to what extent David’s conclusion is justified. Indeed, the claim is surprising since Gosset is mainly known for only one contribution, that is, Student [141], with the introduction of Student’s t-test and its analysis under the normal distribution. According to Lehmann, “Today the pathbreaking nature of this paper is generally recognized and has been widely commented upon, . . .” Gosset’s primary concern in communicating with both Fisher and Pearson was the robustness of the test to non-normality. Lehmann concludes that “the main ideas leading to Pearson’s research were indeed provided by Student.” See Lehmann [85] for the full account, including Gosset’s relationship to the Fisher–Pearson debate, Pearson [116] for a statistical biography of Gosset, and Fisher [40] for a eulogy. Consequently, modern statistics appears to owe a lot to Gosset. Moreover, the reason for the pseudonym was a policy by Gosset’s employer, the brewery Arthur Guinness, Sons, and Co., against work done for the firm being made public. Allowing Gosset to publish under a pseudonym was a concession that resulted in the birth of the statistician Student. Consequently, the authors would like to take this opportunity to thank the Guinness Brewery for its influence on statistics today, and for such a fine beer.

3.3 Abraham Wald

Following Neyman and Pearson’s breakthrough, a different approach to optimality was introduced in Wald [156] and then developed in a sequence of papers culminating in Wald’s [161] book Statistical Decision Functions. Evidence of the influence of Neyman on Wald can be found in the citation of Neyman [102] in

the introduction of Wald [156]. Brown [22] quotes the students of Neyman in 1966 from [103]:

The concepts of confidence intervals and of the Neyman-Pearson theory have proved immensely fruitful. A natural but far reaching extension of their scope can be found in Abraham Wald's theory of statistical decision functions. The elaboration and application of the statistical tools related to these ideas has already occupied a generation of statisticians. It continues to be the main lifestream of theoretical statistics.

Brown's purpose was to address if the last sentence in the quote was still true in 2000.

Wolfowitz [170] describes the primary accomplishments of Wald's statistical decision theory as follows:

Wald's greatest achievement was the theory of statistical decision functions, which includes almost all problems which are the *raison d'être* of statistics.

Leonard [89, Chp. 12] portrays Von Neumann's return to game theory as "partly an early reaction to upheaval and war." However he adds that eventually Von Neumann became personally involved in the war effort and "with that involvement came a significant, unforeseeable moment in the history of game theory: this new mathematics made its wartime entrance into the world, not as the abstract theory of social order central to the book, but as a problem solving technique." Moreover, on pages 278–280, Leonard discusses the statistical research groups at Berkeley, Columbia, and Princeton, in particular Wald at Columbia, and how the effort to develop inspection and testing procedures leads Wald to the development of sequential methods "apparently yielding significant economies in inspection in the Navy," leading to the publication of Wald and Wolfowitz' [162] proof of the optimality of the sequential probability ratio test and Wald's book [159] *Sequential Analysis*. Leonard's claim essentially is that the war stimulated these fine theoretical minds to pursue activities with real application value. In this regard, it is relevant to note Mangel and Samaniego's [93] stimulating description of Wald's work on aircraft survivability, along with the contemporary, albeit somewhat vague, description of "How a Story from World War II shapes Facebook today" by Wilson [167]. Indeed, in the problem of how to allocate armoring to the allied bombers based on their condition upon return from their missions, it was discovered that armoring where the previous planes had been hit was not improving their rate of return. Wald's ingenious insight was that these were the returning bombers not the ones which had been shot down. So the places where the returning bombers were hit are more likely to be the places where one *does not* need to add armoring. Evidently, his rigorous and unconventional innovations to transform this intuition into a real methodology saved many lives. Wolfowitz [170] states:

Wald not only posed his statistical problems clearly and precisely, but he posed them to fit the practical problem and to accord with the decisions the statistician was called on to make. This, in my opinion, was the key to his success—a high level of mathematical talent of the most abstract sort, and a true feeling for, and insight into, practical problems. The combination of the two in his person at such high levels was what gave him his outstanding character.

The section on Von Neumann and Remak (along with the Notes that follows it) in Kurz and Salvadori [78] describes Wald and Von Neumann's relations. Brown [21] credits Wald as the creator of the minmax idea in statistics, evidently given axiomatic justification by Gilboa and Schmeidler [47]. This certainly had something to do with his friendship with Morgenstern and his relationship with Von Neumann, who together authored the famous book [155], but this influence can be explicitly seen in Wald's [157] citation of Von Neumann [153] and Von Neumann and Morgenstern [155] in his introduction [157] of the minmax idea in statistical decision theory. Indeed, Wolfowitz states that:

... he was also spurred on by the connection between the newly announced results of [Von Neumann and Morgenstern] [155] and his own theory, and by the general interest among economists and others aroused by the theory of games.

Wolfowitz asserts that Wald's work [156] Contributions to the Theory of Statistical Estimation and Testing Hypotheses is “probably his most important paper” but that it “went almost completely unnoticed,” possibly because “The use of Bayes solutions was deterrent” and “Wald did not really emphasize that he was using Bayes solutions only as a tool.” Moreover, although Wolfowitz considered Wald's Statistical Decision Functions [161] his greatest achievement, he also says:

The statistician who wants to apply the results of [161] to specific problems is likely to be disappointed. Except for special problems, the complete classes are difficult to characterize in a simple manner and have not yet been characterized. Satisfactory general methods are not yet known for obtaining minimax solutions. If one is not always going to use a minimax solution (to which serious objections have been raised) or a solution satisfying some given criterion, then the statistician should have the opportunity to choose from among “representative” decision functions on the basis of their risk functions. These are not available except for the simplest cases. It is clear that much remains to be done before the use of decision functions becomes common. The theory provides a rational basis for attacking almost any statistical problem, and, when some computational help is available and one makes some reasonable compromises in the interest of computational feasibility, one can obtain a practical answer to many problems which the classical theory is unable to answer or answers in an unsatisfactory manner.

Wolfowitz [170], Morgenstern [97], and Hotelling [55] provide a description of Wald's impact at the time of his passing. The influence of Wald's minimax paradigm can also be observed on (1) decision making under severe uncertainty [134–136], (2) stochastic programming [130] (minimax analysis of stochastic problems), (3) minimax solutions of stochastic linear programming problems [175], (4) robust convex optimization [9] (where one must find the best decision in view of the worst-case parameter values within deterministic uncertainty sets), (4) econometrics [143], and (5) Savage's minimax regret model [128].

3.4 Generalization to Unknown Pairs of Functions and Measures and to Arbitrary Sample Data

In practice, complex systems of interest may involve, both an imperfectly known response function f^\dagger and an imperfectly known probability measure μ^\dagger as illustrated in the following problem.

Problem 4. Let \mathcal{A} be a set of real functions and measures of probability on $[0, 1]$ such that $(f, \mu) \in \mathcal{A}$ if and only if $\mathbb{E}_\mu[X] \leq m$ and $\sup_{x \in [0, 1]} |g(x) - f(x)| \leq 0.1$ where g is some given real function on $[0, 1]$. Let (f^\dagger, μ^\dagger) be an unknown element of \mathcal{A} and let $a \in \mathbb{R}$. Let (X_1, \dots, X_n) be n i.i.d. samples from μ^\dagger , you observe (d_1, \dots, d_n) with $d_i = (X_i, f^\dagger(X_i))$. What is the “sharpest” estimate of $\mu^\dagger[f(X) \geq a]$?

Problem 4 is an illustration of a situation in which the response function f^\dagger and the probability measure μ^\dagger are not directly observed and the sample data arrives in the form of realizations of random variables, the distribution of which is related to (f^\dagger, μ^\dagger) . To simplify the current presentation, assume that this relation is, in general, determined by a function of (f^\dagger, μ^\dagger) and use the following notation: \mathbb{D} will denote the observable space (i.e., the space in which the sample data d take values, assumed to be a metrizable Suslin space) and d will denote the \mathbb{D} -valued random variable corresponding to the observed sample data. To represent the dependence of the distribution of d on the unknown state $(f^\dagger, \mu^\dagger) \in \mathcal{A}$, introduce a measurable function

$$\mathbb{D}: \mathcal{A} \rightarrow \mathcal{M}(\mathcal{D}), \quad (6.8)$$

where $\mathcal{M}(\mathcal{D})$ is given the Borel structure corresponding to the weak topology, to define this relation. The idea is that $\mathbb{D}(f, \mu)$ is the probability distribution of the observed sample data d if $(f^\dagger, \mu^\dagger) = (f, \mu)$, and for this reason it may be called the *data map* (or, even more loosely, the *observation operator*). Now consider the following problem.

Problem 5. Let \mathcal{A} be a known subset of $\mathcal{F}(\mathcal{X}) \times \mathcal{M}(\mathcal{X})$ as in Problem 2 and let \mathbb{D} be a known data map as in (6.8). Let Φ be a known measurable semi-bounded function mapping \mathcal{A} onto \mathbb{R} . Let (f^\dagger, μ^\dagger) be an unknown element of \mathcal{A} . You observe $d \in \mathcal{D}$ sampled from the distribution $\mathbb{D}(f^\dagger, \mu^\dagger)$. What is the sharpest estimation of $\Phi(f^\dagger, \mu^\dagger)$?

3.5 Model Error and Optimal Models

As in Wald’s statistical decision theory [157], a natural notion of optimality can be obtained by formulating Problem 5 as an adversarial game in which player A

chooses $(f^\dagger, \mu^\dagger) \in \mathcal{A}$ and player B (knowing \mathcal{A} and \mathbb{D}) chooses a function θ of the observed data d . As in (6.5) this notion of optimality requires the introduction of a risk function:

$$\mathcal{E}(\theta, (f, \mu)) := \mathbb{E}_{d \sim \mathbb{D}(f, \mu)} [V[\theta(d) - \Phi(f, \mu)]] \quad (6.9)$$

where $V: \mathbb{R} \rightarrow \mathbb{R}$ is a real positive measurable loss function. As in (6.6) the least upper bound on that statistical error $\mathcal{E}(\theta, (f, \mu))$ is obtained as through a worst-case scenario with respect to all possible candidates for (f, μ) (player's A choice), i.e.,

$$\sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}(\theta, (f, \mu)) \quad (6.10)$$

and an optimal estimator/model (possibly within a given class) is then naturally obtained as the minimizer of (6.10) over all functions θ of the data d in that class (player's B choice), i.e., as the minimizer of

$$\inf_{\theta} \sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}(\theta, (f, \mu)). \quad (6.11)$$

Since in real applications true optimality will never be achieved, it is natural to generalize to considering near-minimizers of (6.11) as near-optimal models/estimators.

Remark 1. In situations where the data map is imperfectly known (e.g., when the data d is corrupted by some noise of imperfectly known distribution), one has to include a supremum over all possible candidates $\mathbb{D} \in \mathfrak{D}$ in the calculation of the least upper bound on the statistical error.

3.6 Mean Squared Error, Variance, and Bias

For $(f, \mu) \in \mathcal{A}$ write $\text{Var}_{d \sim \mathbb{D}(f, \mu)} [\theta(d)]$ the variance of the random variable $\theta(d)$ when d is distributed according to $\mathbb{D}(f, \mu)$, i.e.,

$$\text{Var}_{d \sim \mathbb{D}(f, \mu)} [\theta(d)] := \mathbb{E}_{d \sim \mathbb{D}(f, \mu)} [(\theta(d))^2] - [\mathbb{E}_{d \sim \mathbb{D}(f, \mu)} [\theta(d)]]^2$$

The following equation, whose proof is straightforward, shows that for $V(x) = x^2$, the least upper bound on the mean squared error of θ is equal to the least upper bound on the sum of the variance of θ and the square of its bias:

$$\sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}(\theta, (f, \mu)) = \sup_{(f, \mu) \in \mathcal{A}} \left[\text{Var}_{d \sim \mathbb{D}(f, \mu)} [\theta(d)] + (\mathbb{E}_{d \sim \mathbb{D}(f, \mu)} [\theta(d)] - \Phi(f, \mu))^2 \right]$$

Therefore, for $V(x) = x^2$, the bias/variance tradeoff is made explicit.

3.7 Optimal Interval of Confidence

Although \mathcal{E} can a priori be defined to be any risk function, taking $V(x) = \mathbb{1}_{[\gamma, \infty]}(|x|)$ (for some $\gamma > 0$) in (6.5) allows for a transparent and objective identification of optimal intervals of confidence. Indeed, writing,

$$\mathcal{E}_\gamma(\theta, (f, \mu)) := \mathbb{P}_{d \sim \mathbb{D}(f, \mu)} \left[|\theta(d) - \Phi(f, \mu)| \geq \gamma \right]$$

note that $\sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}_\gamma(\theta, (f, \mu))$ is the least upper bound on the probability (with respect to the distribution of d) that the difference between the true value of the quantity of interest $\Phi(f^\dagger, \mu^\dagger)$ and its estimated value $\theta(d)$ is larger than γ . Let $\epsilon \in [0, 1]$. Define

$$\gamma_\epsilon := \inf \left\{ \gamma > 0 \mid \inf_{\theta} \sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}_\gamma(\theta, (f, \mu)) \leq \epsilon \right\},$$

and observe that if θ_ϵ is a minimizer of $\inf_{\theta} \sup_{(f, \mu) \in \mathcal{A}} \mathcal{E}_{\gamma_\epsilon}(\theta, (f, \mu))$ then $[\theta_\epsilon(d) - \gamma_\epsilon, \theta_\epsilon(d) + \gamma_\epsilon]$ is the smallest interval of confidence (random interval obtained as a function of the data) containing $\Phi(f^\dagger, \mu^\dagger)$ with probability at least $1 - \epsilon$. Observe also that this formulation is a natural extension of the OUQ formulation as described in [114]. Indeed, in the absence of sample data, it is easy to show that θ_1 is the midpoint of the optimal interval $[\mathcal{L}(\mathcal{A}), \mathcal{U}(\mathcal{A})]$.

Remark 2. We refer to [37, 38, 137] and in particular to Stein's notorious paradox [138] for the importance of a careful choice for loss function.

3.8 Ordering the Space of Experiments

A natural objective of UQ and statistics is the design of experiments, their comparisons, and the identification of optimal ones. Introduced in Blackwell [17] and Kiefer [73], with a more modern perspective in Le Cam [83] and Strasser [139], here observe that (6.11), as a function of \mathbb{D} , induces an order (transitive, total, but not antisymmetric) on the space of data maps that has a natural experimental design interpretation. More precisely if the data maps \mathbb{D}_1 and \mathbb{D}_2 are interpreted as the distribution of the outcome of two possible experiments, and if the value of (6.11) is smaller for \mathbb{D}_2 than \mathbb{D}_1 , then \mathbb{D}_2 is a preferable experiment.

3.9 Mixing Models

Given estimators $\theta_1, \dots, \theta_n$ can one obtain a better estimator by mixing those estimators? If V is convex (or quasi-convex), then the problem of finding an $\alpha \in [0, 1]^n$ minimizing the statistical error of $\sum_{i=1}^n \alpha_i \theta_i$ under the constraint

$\sum_{i=1}^n \alpha_i = 1$ is a finite-dimensional convex optimization problem in α . If estimators are seen as models of reality, then this observation supports the idea that one can obtain improved models by mixing them (with the goal of achieving minimal statistical errors).

4 The Complete Class Theorem and Bayesian Inference

4.1 The Bayesian Approach

The Bayesian answer to Problem 5 is to assume that (f^\dagger, μ^\dagger) is a sample from some (prior) measure π on \mathcal{A} and then condition the expectation of $\Phi(f, \mu)$ with respect to the observation of the data, i.e., use

$$\mathbb{E}_{(f, \mu) \sim \pi, d \sim \mathbb{D}(f, \mu)} [\Phi(f, \mu) | d] \quad (6.12)$$

as the estimator $\theta(d)$. This requires giving \mathcal{A} the structure of a measurable space such that important quantities of interest such as $(f, \mu) \rightarrow \mu[f(X) \geq a]$ and $(f, \mu) \rightarrow \mathbb{E}_\mu[f]$ are measurable. This can be achieved using results of Ressel [121] providing conditions under which the map $(f, \mu) \rightarrow f_*\mu$ from function/measure pairs to the induced law is Borel measurable. We will henceforth assume \mathcal{A} to be a Suslin space and proceed to construct the measure of probability $\pi \odot \mathbb{D}$ of $((f, \mu), d)$ on $\mathcal{A} \times \mathcal{D}$ via a natural generalization of the Campbell measure and Palm distribution associated with a random measure as described in [68]; see also [25, Ch. 13] for a more current treatment. We refer to Sect. 6 of the appendix for the details of the construction of the distribution $\pi \odot \mathbb{D}$ of $((f, \mu), d)$ when $(f, \mu) \sim \pi$ and $d \sim \mathbb{D}(f, \mu)$, and of the marginal distribution $\pi \cdot \mathbb{D}$ of $\pi \odot \mathbb{D}$ on the data space \mathcal{D} , and the resulting regular conditional expectation (6.12). Consequently, the nested expectation $\mathbb{E}_{(f, \mu) \sim \pi, d \sim \mathbb{D}(f, \mu)}$ appearing in (6.12) will from now on be rigorously written as the expectation $\mathbb{E}_{((f, \mu), d) \sim \pi \odot \mathbb{D}}$.

Statistical error when (f^\dagger, μ^\dagger) is random. When (f^\dagger, μ^\dagger) is a random realization of π^\dagger , one may consider averaging the statistical error (6.9) with respect to π^\dagger and introduce

$$\mathcal{E}(\theta, \pi^\dagger) := \mathbb{E}_{((f, \mu), d) \sim \pi^\dagger \odot \mathbb{D}} [V[\theta(d) - \Phi(f, \mu)]] \quad (6.13)$$

When π^\dagger is an unknown element of a subset Π of $\mathcal{M}(\mathcal{A})$, the least upper bound on the statistical error (6.13) given the available information is obtained by taking the sup of (6.13) with respect to all possible candidates for π^\dagger , i.e.,

$$\sup_{\pi \in \Pi} \mathcal{E}(\theta, \pi) \quad (6.14)$$

When \mathcal{A} is Suslin and when (f^\dagger, μ^\dagger) is not a random sample from π^\dagger but simply an unknown element of \mathcal{A} , then a straightforward application of the reduction theorems of [114] implies that when $\Pi = \mathcal{M}(\mathcal{A})$, then (6.14) is equal to (6.11), i.e.,

$$\sup_{(f,\mu) \in \mathcal{A}} \mathcal{E}(\theta, (f, \mu)) = \sup_{\pi \in \mathcal{M}(\mathcal{A})} \mathcal{E}(\theta, \pi) \quad (6.15)$$

4.2 Relation Between Adversarial Model Error and Bayesian Error

When Φ has a second moment with respect to π , one can utilize the classical variational description of conditional expectation as follows: Letting $L^2_{\pi \cdot \mathbb{D}}(\mathcal{D})$ denote the space of $(\pi \cdot \mathbb{D}$ a.e. equivalence classes of) real-valued measurable functions on \mathcal{D} that are square-integrable with respect to the measure $\pi \cdot \mathbb{D}$, one has (see Sect. 6)

$$\mathbb{E}_{\pi \odot \mathbb{D}}[\Phi | d] := \arg \min_{h \in L^2_{\pi \cdot \mathbb{D}}(\mathcal{D})} \mathbb{E}_{(f,\mu,d) \sim \pi \odot \mathbb{D}} \left[(\Phi(f, \mu) - h(d))^2 \right].$$

In other words, if (f, μ) is sampled from the measure π , $\mathbb{E}_{\pi \odot \mathbb{D}}[\Phi(f, \mu) | d]$ is the best mean-square approximation of $\Phi(f, \mu)$ in the space of square-integrable functions of d . As with the regular conditional probabilities, the real-valued function on \mathcal{D}

$$\theta_\pi(d) = \mathbb{E}_{(f,\mu,D) \sim \pi \odot \mathbb{D}} [\Phi(f, \mu) | D = d], \quad d \in \mathcal{D} \quad (6.16)$$

is uniquely defined up to subsets of \mathcal{D} of $(\pi \cdot \mathbb{D})$ -measure zero.

Using the orthogonal projection property of the conditional expectation, one obtains that if $V(x) = x^2$, then for arbitrary θ ,

$$\mathcal{E}(\theta, \pi) = \mathcal{E}(\theta_\pi, \pi) + \mathbb{E}_{d \sim \pi \cdot \mathbb{D}} \left[\theta(d) - \theta_\pi(d) \right]^2 \quad (6.17)$$

Therefore, if $\Pi \subset \mathcal{M}(\mathcal{A})$ is an admissible set of priors, then (6.17) implies that

$$\inf_{\theta} \sup_{\pi \in \Pi} \mathcal{E}(\theta, \pi) \geq \sup_{\pi \in \Pi} \mathcal{E}(\theta_\pi, \pi).$$

In particular, when $\Pi = \mathcal{M}(\mathcal{A})$ (6.15) implies that

$$\inf_{\theta} \sup_{(f,\mu) \in \mathcal{A}} \mathcal{E}(\theta, (f, \mu)) \geq \sup_{\pi \in \mathcal{M}(\mathcal{A})} \mathcal{E}(\theta_\pi, \pi). \quad (6.18)$$

Therefore, the mean squared error of the best estimator assuming $(f^\dagger, \mu^\dagger) \in \mathcal{A}$ to be unknown is bounded below by the largest mean squared error of the Bayesian estimator obtained by assuming that (f^\dagger, μ^\dagger) is distributed according to some $\pi \in \mathcal{M}(\mathcal{A})$. In the next section, it will be shown that a complete class theorem can be used to obtain that (6.18) is actually an equality. In that case, (6.18) can be used to

quantify the approximate optimality of an estimator by comparing the least upper bound $\sup_{(f,\mu) \in \mathcal{M}(\mathcal{A})} \mathcal{E}(\theta, (f, \mu))$ on the error of that estimator with $\mathcal{E}(\theta_\pi, \pi)$ for a carefully chosen π .

4.3 Complete Class Theorem

A fundamental question is whether (6.18) is an equality: is the adversarial error of the best estimator equal to the non-adversarial error of the worst Bayesian estimator? Is the best estimator Bayesian or an approximation thereof? A remarkable result of decision theory [156–158, 160, 161] is the complete class theorem which states (in the formulation of this paper) that if (1) the admissible measures μ are absolutely continuous with respect the Lebesgue measure, (2) the loss function V in the definition of $\mathcal{E}(\theta, (f, \mu))$ is convex in θ and (3) the decision space is compact, then optimal estimators live in the Bayesian class and non-Bayesian estimators cannot be optimal. The idea of the proof of this result is to use the compactness of the decision space and the continuity of the loss function to approximate the decision theory game by a finite game and recall that optimal strategies of adversarial finite zero-sum games are mixed strategies [99, 100].

Le Cam [81], see also [83], has substantially extended Wald's theory in the sense that requirements of boundedness, or even finiteness, of the loss function are replaced by a requirement of lower semicontinuity, and the requirements of the compactness of the decision space and the absolute continuity of the admissible measures with respect the Lebesgue measure are removed. These vast generalizations come at some price of abstraction yet reveal the relevance and utility of an appropriate complete Banach lattice of measures. In particular, this framework of Le Cam appears to facilitate efficient concrete approximation.

As an illustration, let us describe a complete class theorem on a space of admissible measures, without the inclusion of functions, where the observation map consists of taking n -i.i.d. samples, as in Eq. (6.5). Let $\mathcal{A} \subset \mathcal{M}(\mathcal{X})$ be a subset of the Borel probability measures on a topological space \mathcal{X} and consider a quantity of interest $\Phi : \mathcal{A} \rightarrow \mathbb{R}$. For $\mu \in \mathcal{A}$, the data d is generated by i.i.d. sampling with respect to μ^n . That is $d \sim \mu^n$. For $\mu^\dagger \in \mathcal{A}$, the statistical error $\mathcal{E}(\theta, \mu^\dagger)$ of an estimator $\theta : \mathcal{X}^n \rightarrow \mathbb{R}$ of $\Phi(\mu^\dagger)$ is defined in terms of a loss function $V : \mathbb{R} \rightarrow \mathbb{R}$ as in (6.5). Define the least upper bound on that statistical error and the sharpest estimator as in (6.6) and (6.7).

Let $\Theta := \{\theta : \mathcal{X}^n \rightarrow \mathbb{R}, \theta \text{ measurable}\}$ denote the space of estimators. Since, in general, the game $\mathcal{E}(\theta, \mu)$, $\theta \in \Theta$, $\mu \in \mathcal{A}$ will not have a value, that is, one will have a strict inequality:

$$\sup_{\mu \in \mathcal{A}} \inf_{\theta \in \Theta} \mathcal{E}(\theta, \mu) < \inf_{\theta \in \Theta} \sup_{\mu \in \mathcal{A}} \mathcal{E}(\theta, \mu),$$

classical arguments in game theory suggest that one extend to random estimators and random selection in \mathcal{A} . To that end, let the set of randomized estimators $\mathcal{R} := \{\hat{\theta} : \mathcal{X}^n \times \mathcal{B}(\mathbb{R}) \rightarrow [0, 1], \hat{\theta} \text{ Markov}\}$ be the set of Markov kernels.

To define a topology for \mathcal{R} , define a linear space of measures as follows. Let $\mathcal{A}^n := \{\mu^n \in \mathcal{M}(\mathcal{X}^n) : \mu \in \mathcal{A}\}$ denote the corresponding set of measures generating sample data. Say that \mathcal{A}^n is dominated if there exists an $\omega \in \mathcal{M}(\mathcal{X}^n)$ such that every $\mu^n \in \mathcal{A}^n$ is absolutely continuous with respect to ω . According to the Halmos–Savage lemma [50], see also Strasser [139, Lem. 20.3], the set \mathcal{A}^n is dominated if and only if there exists a countable mixture $\mu^* := \sum_{i=1}^{\infty} \alpha_i \mu_i^n$, with $\alpha_i \geq 0, \mu_i \in \mathcal{A}, i = 1, \dots, \infty$, and $\sum_{i=1}^{\infty} \alpha_i = 1$, such that $\mu^n \sim \mu^*, \mu \in \mathcal{A}$. A construct at the heart of Le Cam's approach is a natural linear space notion of a mixture space of \mathcal{A} , called the L -space $L(\mathcal{A}^n) := L^1(\mu^*)$. It follows easily, see [139, Lem. 41.1], that $L(\mathcal{A}^n)$ is the set of signed measures which are absolutely continuous with respect to μ^* . When \mathcal{A} is not dominated, a natural generalization of this construction [139, Def. 41.3] due to Le Cam [81] is used. A crucial property of the L -space $L(\mathcal{A}^n)$ is that not only is it a Banach lattice (see Strasser [139, Cor. 41.4]), but by [139, Lem. 41.5] it is a complete lattice. The utility of the concept of a complete lattice to the complete class theorems can clearly be seen in the proof of the lemma in Section 2 of Wald and Wolfowitz' [163] proof of the complete class theorem when the number of decisions and the number of distributions is finite. Then, the natural action of a randomized estimator on the bounded continuous function/mixture pairs $C_b(R) \times L(\mathcal{A}^n)$ is

$$f \hat{\theta} v := \int \int f(r) \hat{\theta}(x^n, dr) v(dx^n), \quad f \in C_b(R), v \in L(\mathcal{A}^n).$$

Let \mathcal{R} be endowed with the topology of pointwise convergence with respect to this action, that is, the weak topology with respect to integration against $C_b(R) \times L(\mathcal{A}^n)$. Moreover, this weak topology also facilitates a definition of the space $\overline{\mathcal{R}}$ of *generalized* random estimators as bilinear real-valued maps $\vartheta : C_b(R) \times L(\mathcal{A}^n) \rightarrow \mathbb{R}$ satisfying $|\vartheta(f, \mu)| \leq \|f\|_{\infty} \|\mu\|$, $\vartheta(f, \mu) \geq 0$ for $f \geq 0, \mu \geq 0$, and $\vartheta(1, \mu) = \mu(\mathcal{X})$. By [139, Thm. 42.3], the set of generalized random estimators $\overline{\mathcal{R}}$ is compact and convex, and by [139, Thm. 42.5] of Le Cam [82], \mathcal{R} is dense in $\overline{\mathcal{R}}$ in the weak topology. Moreover, when \mathcal{A}^n is dominated and one can restrict to a compact subset $C \in \mathbb{R}$ of the decision space, then Strasser [139, Cor. 42.8] asserts that $\overline{\mathcal{R}} = \mathcal{R}$.

Returning to our illustration, if one let $W_\mu, \mu \in \mathcal{A}$ be defined by $W_\mu(r) := V(r - \Phi(\mu))$, $r \in \mathbb{R}, \mu \in \mathcal{A}$ denote the associated family of loss functions, one can now define a generalization of the statistical error function $\mathcal{E}(\theta, \mu)$ of (6.5) to randomized estimators $\hat{\theta}$ by

$$\mathcal{E}(\hat{\theta}, \mu) := \int \int W_\mu(r) \hat{\theta}(x^n, dr) \mu^n(dx^n), \quad \hat{\theta} \in \mathcal{R}, \mu \in \mathcal{A}.$$

This definition reduces to the previous one (6.5) when the random estimator $\hat{\theta}$ corresponds to a point estimator θ and extends naturally to $\overline{\mathcal{R}}$. Finally, one says that an estimator $\vartheta^* \in \overline{\mathcal{R}}$ is Bayesian if there exists a probability measure m with finite support on \mathcal{A} such that

$$\int \mathcal{E}(\vartheta^*, \mu) m(d\mu) \leq \int \mathcal{E}(\vartheta, \mu) m(d\mu), \quad \vartheta \in \overline{\mathcal{R}}.$$

The following complete class theorem follows from Strasser [139, Thm. 47.9, Cor. 42.8] since one can naturally compactify the decision space \mathbb{R} when the quantity of interest Φ is bounded and the loss function V is sublevel compact, that is has compact sublevel sets.

Theorem 1. *Suppose that the loss function V is sublevel compact and the quantity of interest $\Phi : \mathcal{A} \rightarrow \mathbb{R}$ is bounded. Then, for each generalized randomized estimator $\vartheta \in \overline{\mathcal{R}}$, there exists a weak limit $\vartheta^* \in \overline{\mathcal{R}}$ of Bayesian estimators such that*

$$\mathcal{E}(\vartheta^*, \mu) \leq \mathcal{E}(\vartheta, \mu), \quad \mu \in \mathcal{A}.$$

If, in addition, \mathcal{A} is dominated, then there exists such a $\vartheta^* \in \mathcal{R}$.

A comprehensive connection of these results, where Bayesian estimators are defined only in terms of measures of finite support on \mathcal{A} , with the framework of Sect. 4 where Bayesian estimators are defined in terms of Borel measures on \mathcal{A} , is not available yet. Nevertheless it appears that much can be done in this regard. In particular, one can suspect that when \mathcal{A} is a closed convex set of probability measures equipped with the weak topology and \mathcal{X} is a Borel subset of a Polish space, that if the loss function V is convex and Φ is affine and measurable, the Choquet theory of Winkler [168, 169] can be used to facilitate this connection. Indeed, as mentioned above, complete class theorems are available for much more general loss functions than continuous or convex, more general decision spaces than \mathbb{R} , and without absolute continuity assumptions. Moreover, it is interesting to note that, although randomization was introduced to obtain minmax results, when the loss function V is strictly convex, Bayesian estimators can be shown to be *non-random*. This can be explicitly observed in the definition (6.16) of Bayesian estimators when $V(x) := x^2$ and is understood much more generally in Dvoretzky, Wald, and Wolfowitz [33]. We conjecture that further simplifications can be obtained by allowing *approximate* versions of complete class theorems, Bayesian estimators, optimality, and saddle points, as in Scovel, Hush, and Steinwart's [129] extension of classical Lagrangian duality theory to include approximations.

5 Incorporating Complexity and Computation

Although decision theory provides well-posed notions of optimality and performance in statistical estimation, it does not address the complexity of the actual computation of optimal or nearly optimal estimators and their evaluation against the data. Indeed, although the abstract identification of an optimal estimator as the solution of an optimization problem provides a clear objective, practical applications

require the actual implementation of the estimator on a machine and its numerical evaluation against the data.

5.1 Machine Wald

The simultaneous emphasis on *performance* and *computation* can be traced back to PAC (probably approximately correct) learning initiated by Valiant [149] which has laid down the foundations of machine learning (ML). Indeed, as asserted by Wasserman in his 2013 lecture, “The Rise of the Machines” [164, Sec. 1.5]:

There is another interesting difference that is worth pondering. Consider the problem of estimating a mixture of Gaussians. In Statistics we think of this as a solved problem. You use, for example, maximum likelihood which is implemented by the EM algorithm. But the EM algorithm does not solve the problem. There is no guarantee that the EM algorithm will actually find the MLE; it's a shot in the dark. The same comment applies to MCMC methods. In ML, when you say you've solved the problem, you mean that there is a polynomial time algorithm with provable guarantees.

That is, on even par with the rigorous performance analysis, machine learning also requires that solutions be efficiently implementable on a computer, and often such efficiency is established by proving bounds on the amount of computation required to produce such a solution with a given algorithm. Although Wald’s theory of optimal statistical decisions has resulted in many important statistical discoveries, looking through the three Lehmann symposia of Rojo and Pérez–Abreu [126] in 2004 and Rojo [124, 125] in 2006 and 2009, it is clear that the incorporation of the analysis of the computational algorithm, both in terms of its computational efficiency and its statistical optimality, has not begun. Therefore a natural answer to fundamental challenges in UQ appears to be the full incorporation of *computation* into a natural generalization of Wald’s statistical decision function framework, producing a framework one might call *Machine Wald*.

5.2 Reduction Calculus

The resolution of minimax problems (6.11) require, at an abstract level, searching in the space of all possible functions of the data. By restricting models to the Bayesian class, the complete class theorem allows to limit this search to prior distributions on \mathcal{A} , i.e., to measure over spaces of measures and functions. To enable the computation of these models, it is therefore necessary to identify conditions under which Minimax problems over measures over spaces of measures and functions can be reduced to the manipulation of finite-dimensional objects and develop the associated reduction calculus. For min or max problems over measures over spaces of measures (and possibly functions), this calculus can take the form of a reduction to a nesting of optimization problems over measures (and possibly functions for the inner part) [109, 112, 113], which, in turn, can be reduced to searches over extreme points [51, 110, 114, 142].

5.3 Stopping Conditions

Many of these optimization problems will not be tractable. However even in the tractable case, which has rigorous guarantees on the amount of computation required to obtain an approximate optima, it will be useful to have stopping criteria for the specific algorithm and the specific problem instance under consideration, which can be used to guarantee when an approximate optima has been achieved. Although in the intractable case no such guarantee will exist in general, intelligent choices of algorithms may result in the attainment of approximate optima and such tests guarantee that fact. Ermoliev, Gaivoronski, and Nedeva [36] successfully develop such stopping criteria using Lagrangian duality and generalized Bender's decompositions by Geoffrion [46] for certain stochastic optimization problems which are also relevant here. In addition, the approximation of intractable problems by tractable ones will be important. Recently, Hanasusanto, Roitch, Kuhn, and Wiesemann [53] derive explicit conic reformulations for tractable problem classes and suggest efficiently computable conservative approximations for intractable ones.

5.4 On the Borel-Kolmogorov Paradox

An oftentimes overlooked difficulty with Bayesian estimators lies in the fact that for a prior $\pi \in \mathcal{M}(\mathcal{A})$, the posterior (6.12) is not a measurable function of d but a convex set $\Theta(\pi)$ of measurable functions θ of d that are almost surely equal to each other under the measure $\pi \cdot \mathbb{D}$ on \mathcal{D} .

A notorious pathological consequence is the Borel–Kolmogorov paradox (see Chapter 5 of [76] and Section 15.7 of [63]). Recall that in the formulation of this paradox, one considers the uniform distribution on the two-dimensional sphere and one is interested in obtaining the conditional distribution associated with a great circle of that sphere. If the problem is parameterized in spherical coordinates, then the resulting conditional distribution is uniform for the equator but nonuniform for the longitude corresponding to the prime meridian. The following theorem suggests that this paradox is generic and dissipates the idea that it could be limited to fabricated toy examples. See also Singpurwalla and Swift [132] for implications of this paradox in modeling and inference.

Recall that for $\pi \in \mathcal{M}(\mathcal{A})$, that $\Theta(\pi)$ is defined as the convex set of measurable functions which are equal to $\pi \cdot \mathbb{D}$ -everywhere to the regular conditional expectation (6.12). Despite this indeterminateness, it is comforting to know that

$$\mathcal{E}(\theta_2, \pi) = \mathcal{E}(\theta_1, \pi), \quad \theta_1, \theta_2 \in \Theta(\pi).$$

Moreover, it is also easy to see that if π^\dagger is absolutely continuous with respect to π , then $\theta_1(d) = \theta_2(d)$ with $\pi^\dagger \cdot \mathbb{D}$ probability one for all $\theta_1, \theta_2 \in \Theta(\pi)$, and consequently

$$\mathcal{E}(\theta_2, \pi^\dagger) = \mathcal{E}(\theta_1, \pi^\dagger), \quad \theta_1, \theta_2 \in \Theta(\pi), \quad \pi^\dagger \prec \pi,$$

where the notation $\pi^\dagger \prec \pi$ means that π^\dagger is absolutely continuous with respect to π . The following theorem shows that this requirement of absolute continuity is necessary for all versions of conditional expectations $\theta \in \Theta(\pi)$ to share the same risk. See Sect. 6 for its proof.

Theorem 2. *Assume that $V(x) = x^2$ and that the image $\Phi(\mathcal{A})$ is a nontrivial interval. If π^\dagger is not absolutely continuous with respect to π then*

$$\frac{1}{4} \leq \frac{\sup_{\theta_1, \theta_2 \in \Theta(\pi)} (\mathcal{E}(\theta_2, \pi^\dagger) - \mathcal{E}(\theta_1, \pi^\dagger))}{(\mathcal{U}(\mathcal{A}) - \mathcal{L}(\mathcal{A}))^2 \sup_{B \in \mathcal{B}(\mathcal{D}): (\pi \cdot \mathbb{D})[B] = 0} (\pi^\dagger \cdot \mathbb{D})[B]} \leq 1 \quad (6.19)$$

where $\mathcal{U}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$ are defined by (6.2) and (6.3).

Remark 3. If moreover $\pi^\dagger \cdot \mathbb{D}$ is orthogonal to $\pi \cdot \mathbb{D}$, that is, there exists a set $B \in \mathcal{B}(\mathcal{D})$ such that $(\pi \cdot \mathbb{D})[B] = 0$ and $(\pi^\dagger \cdot \mathbb{D})[B] = 1$, then Theorem 2 implies that $\sup_{\theta_1, \theta_2 \in \Theta(\pi)} (\mathcal{E}(\theta_2, \pi^\dagger) - \mathcal{E}(\theta_1, \pi^\dagger))$ is larger than the statistical error of the midpoint estimator

$$\theta := \frac{\mathcal{L}(\mathcal{A}) + \mathcal{U}(\mathcal{A})}{2}.$$

As a remedy, one can try (see [144, 145] and [117]) constructing conditional expectations as disintegration or derivation limits defined as

$$\mathbb{E}_{\pi \odot \mathbb{D}} [\Phi(f, \mu) | D = d] = \lim_{B \downarrow \{d\}} \mathbb{E}_{\pi \odot \mathbb{D}} [\Phi(f, \mu) | D \in B] \quad (6.20)$$

where the limit $B \downarrow \{d\}$ is taken over a net of open neighborhoods of d . But as shown in [66], the limit generally depends on the net $B \downarrow \{d\}$ and the resulting conditional expectations can be distinctly different for different nets. Furthermore the limit (6.20) may exist/not exist on subsets of \mathcal{D} of $(\pi \cdot \mathbb{D})$ -measure zero (which, as shown above, can lead to the inconsistency of the estimator). A related important issue is that conditional probabilities can in general not be computed [2]. Observe that if the limit (6.20) does not exist, then Bayesian estimation of $\Phi(f, \mu)$ may have significant oscillations as the precise measurement of d becomes sharper.

5.5 On Bayesian Robustness/Brittleness

As much as classical numerical analysis shows that there are stable and unstable ways to discretize a partial differential equation, positive [13, 23, 28, 75, 80, 140, 152] and negative results [6, 27, 42, 43, 65, 84, 108, 109, 112, 113] are forming an emerging understanding of stable and unstable ways to apply Bayes' rule in practice. One

aspect of stability concerns the sensitivity of posterior conclusions with respect to the underlying models and prior beliefs.

Most statisticians would acknowledge that an analysis is not complete unless the sensitivity of the conclusions to the assumptions is investigated. Yet, in practice, such sensitivity analyses are rarely used. This is because sensitivity analyses involve difficult computations that must often be tailored to the specific problem. This is especially true in Bayesian inference where the computations are already quite difficult. [165]

Another aspect concerns situations where Bayes' rule is applied iteratively and posterior values become prior values for the next iteration. Observe in particular that when posterior distributions (which are later on used as prior distributions) are only approximated (e.g., via MCMC methods), stability requires the convergence of the MCMC method in the same metric used to quantify the sensitivity of posterior with respect to the prior distributions.

In the context of the framework being developed here, recent results [108, 109, 112, 113] on the extreme sensitivity (brittleness) of Bayesian inference in the TV and Prokhorov metrics appear to suggest that robust inference, in a continuous world under finite-information, should perhaps be done with reduced/coarse models rather than highly sophisticated/complex models (with a level of coarseness/reduction depending on the available finite information) [113].

5.6 Information-Based Complexity

From the point of view of practical applications, it is clear that the set of possible models entering in the minimax problem 6.11 must be restricted by introducing constraints on computational complexity. For example, finding optimal models of materials in extreme environments is not the correct objective when these models require full quantum mechanics calculations. A more productive approach is to search for computationally tractable optimal models in a given complexity class. Here one may wonder if Bayesian models remain a complete class for the resulting complexity constrained minimax problems. It is also clear that computationally tractable optimal models may not use all the available information, for instance, a material model of bounded complexity should not use the state of every atom. The idea that fast computation requires computation with partial information forms the core of information-based complexity, the branch of computational complexity that studies the complexity of approximating continuous mathematical operations with discrete and finite ones up to a specified level of accuracy [101, 115, 146, 171, 172], where it is also augmented by concepts of contaminated and priced information associated with, for example, truncation errors and the cost of numerical operations. Recent results [107] suggest that decision theory concepts could be used, not only to identify reduced models but also algorithms of near-optimal complexity by reformulating the process of computing with partial information and limited resources as that of playing underlying hierarchies of adversarial information games.

6 Conclusion

Although uncertainty quantification is still in its formative stage, much like the state of probability theory before its rigorous formulation by Kolmogorov in the 1930s, it has the potential to have an impact on the process of scientific discovery that is similar to the advent of scientific computing. Its emergence remains sustained by the persistent need to make critical decisions with partial information and limited resources. There are many paths to its development, but one such path appears to be the incorporation of notions of computation and complexity in a generalization of Wald's decision framework built on Von Neumann's theory of adversarial games.

Appendix

Construction of $\pi \odot \mathbb{D}$

The below construction works when $\mathcal{A} \subseteq \mathcal{G} \times \mathcal{M}(\mathcal{X})$ for some Polish subset $\mathcal{G} \subset \mathcal{F}(\mathcal{X})$ and \mathcal{X} is Polish. Observe that since \mathcal{D} is metrizable, it follows from [4, Thm. 15.13], that, for any $B \in \mathcal{B}(\mathcal{D})$, the evaluation $v \mapsto v(B)$, $v \in \mathcal{M}(\mathcal{D})$, is measurable. Consequently, the measurability of \mathbb{D} implies that the mapping

$$\widehat{\mathbb{D}}: \mathcal{A} \times \mathcal{B}(\mathcal{D}) \rightarrow R$$

defined by

$$\widehat{\mathbb{D}}((f, \mu), B) := \mathbb{D}(f, \mu)[B], \quad \text{for } (f, \mu) \in \mathcal{A}, B \in \mathcal{B}(\mathcal{D})$$

is a transition function in the sense that, for fixed $(f, \mu) \in \mathcal{A}$, $\widehat{\mathbb{D}}((f, \mu), \cdot)$ is a probability measure, and, for fixed $B \in \mathcal{B}(\mathcal{D})$, $\widehat{\mathbb{D}}(\cdot, B)$ is Borel measurable. Therefore, by [18, Thm. 10.7.2], any $\pi \in \mathcal{M}(\mathcal{A})$ defines a probability measure

$$\pi \odot \mathbb{D} \in \mathcal{M}(\mathcal{B}(\mathcal{A}) \times \mathcal{B}(\mathcal{D}))$$

through

$$\pi \odot \mathbb{D}[A \times B] := \mathbb{E}_{(f, \mu) \sim \pi} [\mathbb{1}_A(f, \mu) \mathbb{D}(f, \mu)[B]], \quad \text{for } A \in \mathcal{B}(\mathcal{A}), B \in \mathcal{B}(\mathcal{D}), \quad (6.21)$$

where $\mathbb{1}_A$ is the indicator function of the set A :

$$\mathbb{1}_A(f, \mu) := \begin{cases} 1, & \text{if } (f, \mu) \in A, \\ 0, & \text{if } (f, \mu) \notin A. \end{cases}$$

It is easy to see that π is the \mathcal{A} -marginal of $\pi \odot \mathbb{D}$. Moreover, when \mathcal{X} is Polish, [4, Thm. 15.15] implies that $\mathcal{M}(\mathcal{X})$ is Polish, and when \mathcal{G} is Polish, it follows that $\mathcal{A} \subseteq \mathcal{G} \times \mathcal{M}(\mathcal{X})$ is second countable. Consequently, since \mathcal{D} is Suslin and hence second countable, it follows from [32, Prop. 4.1.7] that

$$\mathcal{B}(\mathcal{A} \times \mathcal{D}) = \mathcal{B}(\mathcal{A}) \times \mathcal{B}(\mathcal{D})$$

and hence $\pi \odot \mathbb{D}$ is a probability measure on $\mathcal{A} \times \mathcal{D}$. That is,

$$\pi \odot \mathbb{D} \in \mathcal{M}(\mathcal{A} \times \mathcal{D}).$$

Henceforth denote $\pi \cdot \mathbb{D}$ the corresponding Bayes' sampling distribution defined by the \mathcal{D} -marginal of $\pi \odot \mathbb{D}$, and note that by (6.21), one has

$$\pi \cdot \mathbb{D}[B] := \mathbb{E}_{(f,\mu) \sim \pi} [\mathbb{D}(f, \mu)[B]], \quad \text{for } B \in \mathcal{B}(\mathcal{D}).$$

Since both \mathcal{D} and \mathcal{A} are Suslin, it follows that the product $\mathcal{A} \times \mathcal{D}$ is Suslin. Consequently, [18, Cor. 10.4.6] asserts that regular conditional probabilities exist for any sub- σ -algebra of $\mathcal{B}(\mathcal{A} \times \mathcal{D})$. In particular, the product theorem of [18, Thm. 10.4.11] asserts that product regular conditional probabilities

$$(\pi \odot \mathbb{D})|_d \in \mathcal{M}(\mathcal{A}), \quad \text{for } d \in \mathcal{D}$$

exist and that they are $\pi \cdot \mathbb{D}$ -a.e. unique.

Proof of Theorem 2

If $\pi^\dagger \cdot \mathbb{D}$ is not absolutely continuous with respect to $\pi \cdot \mathbb{D}$, then there exists $B \in \mathcal{B}(\mathcal{D})$ such that $(\pi \cdot \mathbb{D})[B] = 0$ and $(\pi^\dagger \cdot \mathbb{D})[B] > 0$. Let $\theta \in \Theta(\pi)$. Define

$$\theta_y(d) := \theta(d)1_{B^c}(d) + y1_B(d) \tag{6.22}$$

Then it is easy to see that if y is in the range of Φ , then $\theta_y \in \Theta(\pi)$. Now observe that for $y, z \in \text{Image}(\Phi)$,

$$\mathcal{E}(\theta_y, \pi^\dagger) - \mathcal{E}(\theta_z, \pi^\dagger) = \mathbb{E}_{(f,\mu,d) \sim \pi^\dagger \odot \mathbb{D}} \left[1_B(d) \left(V(y - \Phi(f, \mu)) - V(z - \Phi(f, \mu)) \right) \right]$$

Hence, for $V(x) = x^2$, it holds true that

$$\mathcal{E}(\theta_y, \pi^\dagger) - \mathcal{E}(\theta_z, \pi^\dagger) = [(y - \gamma)^2 - (z - \gamma)^2](\pi^\dagger \cdot \mathbb{D})[B]$$

with

$$\gamma := \mathbb{E}_{\pi^\dagger \odot \mathbb{D}}[\Phi | D \in B]$$

which proves

$$\begin{aligned} \sup_{\theta_2 \in \Theta(\pi)} \mathcal{E}(\theta_2, \pi^\dagger) - \inf_{\theta_1 \in \Theta(\pi)} \mathcal{E}(\theta_1, \pi^\dagger) &\geq \sup_{B \in \mathcal{B}(\mathcal{D}) : (\pi \cdot \mathbb{D})[B] = 0, y, z \in \text{Image}(\Phi)} \\ &\quad \left[(y - \mathbb{E}_{\pi^\dagger \odot \mathbb{D}}[\Phi | D \in B])^2 - (z - \mathbb{E}_{\pi^\dagger \odot \mathbb{D}}[\Phi | D \in B])^2 \right] (\pi^\dagger \cdot \mathbb{D})[B], \end{aligned}$$

and,

$$\sup_{\theta_2 \in \Theta(\pi)} \mathcal{E}(\theta_2, \pi^\dagger) - \inf_{\theta_1 \in \Theta(\pi)} \mathcal{E}(\theta_1, \pi^\dagger) \leq (\mathcal{U}(\mathcal{A}) - \mathcal{L}(\mathcal{A}))^2 \sup_{B \in \mathcal{B}(\mathcal{D}) : (\pi \cdot \mathbb{D})[B] = 0} (\pi^\dagger \cdot \mathbb{D})[B].$$

To obtain the right hand side of (6.19) observe that (see for instance [29, Sec. 5]) there exists $B^* \in \mathcal{B}(\mathcal{D})$ such that

$$(\pi^\dagger \cdot \mathbb{D})[B^*] = \sup_{B \in \mathcal{B}(\mathcal{D}) : (\pi \cdot \mathbb{D})[B] = 0} (\pi^\dagger \cdot \mathbb{D})[B]$$

and (since $\theta_2 = \theta_1$ on the complement of B^*)

$$\begin{aligned} \sup_{\theta_1, \theta_2 \in \Theta(\pi)} (\mathcal{E}(\theta_2, \pi^\dagger) - \mathcal{E}(\theta_1, \pi^\dagger)) \\ = \sup_{\theta_1, \theta_2 \in \Theta(\pi)} \mathbb{E}_{(f, \mu, d) \sim \pi^\dagger \odot \mathbb{D}} \left[1_{B^*}(d) \left(V(\theta_2 - \Phi(f, \mu)) - V(\theta_1 - \Phi(f, \mu)) \right) \right]. \end{aligned}$$

We conclude by observing that for $V(x) = x^2$,

$$\sup_{\theta_1, \theta_2 \in \Theta(\pi)} \left(V(\theta_2 - \Phi(f, \mu)) - V(\theta_1 - \Phi(f, \mu)) \right) \leq (\mathcal{U}(\mathcal{A}) - \mathcal{L}(\mathcal{A}))^2.$$

Conditional Expectation as an Orthogonal Projection

It easily follows from Tonelli's Theorem that

$$\mathbb{E}_{\pi \cdot \mathbb{D}}[h^2] = \mathbb{E}_{\pi \odot \mathbb{D}}[h^2] = \mathbb{E}_{(f, \mu) \sim \pi} \mathbb{E}_{\mathbb{D}(f, \mu)}[h^2].$$

By considering the sub σ -algebra $\mathcal{A} \times \mathcal{B}(\mathcal{D}) \subset \mathcal{B}(\mathcal{A} \times \mathcal{D}) = \mathcal{B}(\mathcal{A}) \times \mathcal{B}(\mathcal{D})$, it follows from, e.g., Theorem 10.2.9 of [32], that $L^2_{\pi \cdot \mathbb{D}}(\mathcal{D})$ is a closed Hilbert subspace of the Hilbert space $L^2_{\pi \odot \mathbb{D}}(\mathcal{A} \times \mathcal{D})$ and the conditional expectation of Φ given the random variable D is the orthogonal projection from $L^2_{\pi \odot \mathbb{D}}(\mathcal{A} \times \mathcal{D})$ to $L^2_{\pi \cdot \mathbb{D}}(\mathcal{D})$.

References

1. Richardson, L.F.: Weather Prediction by Numerical Process. Cambridge Mathematical Library. Cambridge University Press, Cambridge (1922)
2. Ackerman, N.L., Freer, C.E., Roy, D.M.: On the computability of conditional probability. arXiv:1005.3014 (2010)
3. Adams, M., Lashgari, A., Li, B., McKerns, M., Mihaly, J.M., Ortiz, M., Owhadi, H., Rosakis, A.J., Stalzer, M., Sullivan, T.J.: Rigorous model-based uncertainty quantification with application to terminal ballistics. Part II: systems with uncontrollable inputs and large scatter. *J. Mech. Phys. Solids* **60**(5), 1002–1019 (2012)
4. Aliprantis, C.D., Border, K.C.: Infinite Dimensional Analysis: A Hitchhiker’s Guide, 3rd edn. Springer, Berlin (2006)
5. Anderson, T.W.: The integral of a symmetric unimodal function over a symmetric convex set and some probability inequalities. *Proc. Am. Math. Soc.* **6**(2), 170–176 (1955)
6. Belot, G.: Bayesian orgulity. *Philos. Sci.* **80**(4), 483–503 (2013)
7. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (2009)
8. Ben-Tal, A., Hochman, E.: More bounds on the expectation of a convex function of a random variable. *J. Appl. Probab.* **9**, 803–812 (1972)
9. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. *Math. Oper. Res.* **23**(4), 769–805 (1998)
10. Bentkus, V.: A remark on the inequalities of Bernstein, Prokhorov, Bennett, Hoeffding, and Talagrand. *Liet. Mat. Rink.* **42**(3), 332–342 (2002)
11. Bentkus, V.: On Hoeffding’s inequalities. *Ann. Probab.* **32**(2), 1650–1673 (2004)
12. Bentkus, V., Geuze, G.D.C., van Zuijlen, M.C.A.: Optimal Hoeffding-like inequalities under a symmetry assumption. *Statistics* **40**(2), 159–164 (2006)
13. Bernstein, S.N.: Collected Works. Izdat. “Nauka”, Moscow (1964)
14. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. *SIAM Rev.* **53**(3), 464–501 (2011)
15. Bertsimas, D., Popescu, I.: Optimal inequalities in probability theory: a convex optimization approach. *SIAM J. Optim.* **15**(3), 780–804 (electronic) (2005)
16. Birge, J.R., Wets, R.J.-B.: Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse. *Math. Prog. Stud.* **27**, 54–102 (1986)
17. Blackwell, D.: Equivalent comparisons of experiments. *Ann. Math. Stat.* **24**(2), 265–272 (1953)
18. Bogachev, V.I.: Measure Theory, vol. II. Springer, Berlin (2007)
19. Bot, R.I., Lorenz, N., Wanka, G.: Duality for linear chance-constrained optimization problems. *J. Korean Math. Soc.* **47**(1), 17–28 (2010)
20. Boucheron, S., Lugosi, G., Massart, P.: A sharp concentration inequality with applications. *Random Struct. Algorithms* **16**(3), 277–292 (2000)
21. Brown, L.D.: Minimaxity, more or less. In: Gupta, S.S., Berger, J.O. (eds.) Statistical Decision Theory and Related Topics V, pp. 1–18. Springer, New York (1994)
22. Brown, L.D.: An essay on statistical decision theory. *J. Am. Stat. Assoc.* **95**(452), 1277–1281 (2000)
23. Castillo, I., Nickl, R.: Nonparametric Bernstein–von Mises theorems in Gaussian white noise. *Ann. Stat.* **41**(4), 1999–2028 (2013)
24. Chen, W., Sim, M., Sun, J., Teo, C.-P.: From CVaR to uncertainty set: implications in joint chance-constrained optimization. *Oper. Res.* **58**(2), 470–485 (2010)
25. Daley, D.J., Vere-Jones, D.: An Introduction to the Theory of Point Processes.: General Theory and Structure. Probability and its Applications (New York), vol. II, 2nd edn. Springer, New York (2008)
26. Dantzig, G.B.: Linear programming under uncertainty. *Manag. Sci.* **1**, 197–206 (1955)

27. Diaconis, P., Freedman, D.A.: On the consistency of Bayes estimates. *Ann. Stat.* **14**(1), 1–67 (1986). With a discussion and a rejoinder by the authors
28. Doob, J.L.: Application of the theory of martingales. In: *Le Calcul des Probabilités et ses Applications, Colloques Internationaux du Centre National de la Recherche Scientifique*, vol. 13, pp. 23–27. Centre National de la Recherche Scientifique, Paris (1949)
29. Doob, J.L.: *Measure Theory*. Graduate Texts in Mathematics, vol. 143. Springer, New York (1994)
30. Drenick, R.F.: Aseismic design by way of critical excitation. *J. Eng. Mech. Div. Am. Soc. Civ. Eng.* **99**(4), 649–667 (1973)
31. Dubins, L.E.: On extreme points of convex sets. *J. Math. Anal. Appl.* **5**(2), 237–244 (1962)
32. Dudley, R.M.: *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics, vol. 74. Cambridge University Press, Cambridge (2002). Revised reprint of the 1989 original
33. Dvoretzky, A., Wald, A., Wolfowitz, J.: Elimination of randomization in certain statistical decision procedures and zero-sum two-person games. *Ann. Math. Stat.* **22**(1), 1–21 (1951)
34. Edmundson, H.P.: Bounds on the expectation of a convex function of a random variable. Technical report, DTIC Document (1957)
35. Elishakoff, I., Ohsaki, M.: *Optimization and Anti-optimization of Structures Under Uncertainty*. World Scientific, London (2010)
36. Ermoliev, Y., Gaivoronski, A., Nedeva, C.: Stochastic optimization problems with incomplete information on distribution functions. *SIAM J. Control Optim.* **23**(5), 697–716 (1985)
37. Fisher, R.: *The Design of Experiments*. Oliver and Boyd, Edinburgh (1935)
38. Fisher, R.: Statistical methods and scientific induction. *J. R. Stat. Soc. Ser. B.* **17**, 69–78 (1955)
39. Fisher, R.A.: On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc. Lond. Ser. A* **222**, 309–368 (1922)
40. Fisher, R.A.: “Student”. *Ann. Eugen.* **9**(1), 1–9 (1939)
41. Frauendorfer, K.: Solving SLP recourse problems with arbitrary multivariate distributions—the dependent case. *Math. Oper. Res.* **13**(3), 377–394 (1988)
42. Freedman, D.A.: On the asymptotic behavior of Bayes’ estimates in the discrete case. *Ann. Math. Stat.* **34**, 1386–1403 (1963)
43. Freedman, D.A.: On the Bernstein-von Mises theorem with infinite-dimensional parameters. *Ann. Stat.* **27**(4), 1119–1140 (1999)
44. Gaivoronski, A.A.: A numerical method for solving stochastic programming problems with moment constraints on a distribution function. *Ann. Oper. Res.* **31**(1), 347–369 (1991)
45. Gassmann, H., Ziembka, W.T.: A tight upper bound for the expectation of a convex function of a multivariate random variable. In: *Stochastic Programming 84 Part I. Mathematical Programming Study*, vol. 27, pp. 39–53. Springer, Berlin (1986)
46. Geoffrion, A.M.: Generalized Benders decomposition. *JOTA* **10**(4), 237–260 (1972)
47. Gilboa, I., Schmeidler, D.: Maxmin expected utility with non-unique prior. *J. Math. Econ.* **18**(2), 141–153 (1989)
48. Godwin, H.J.: On generalizations of Tchebychef’s inequality. *J. Am. Stat. Assoc.* **50**(271), 923–945 (1955)
49. Goh, J., Sim, M.: Distributionally robust optimization and its tractable approximations. *Oper. Res.* **58**(4, part 1), 902–917 (2010)
50. Halmos, P.R., Savage, L.J.: Application of the Radon-Nikodym theorem to the theory of sufficient statistics. *Ann. Math. Stat.* **20**(2), 225–241 (1949)
51. Han, S., Tao, M., Topcu, U., Owhadi, H., Murray, R.M.: Convex optimal uncertainty quantification. *SIAM J. Optim.* **25**(23), 1368–1387 (2015). arXiv:1311.7130
52. Han, S., Topcu, U., Tao, M., Owhadi, H., Murray, R.: Convex optimal uncertainty quantification: algorithms and a case study in energy storage placement for power grids. In: *American Control Conference (ACC), 2013*, Washington, DC, pp. 1130–1137. IEEE (2013)
53. Hanusanto, G.A., Roitch, V., Kuhn, D., Wiesemann, W.: A distributionally robust perspective on uncertainty quantification and chance constrained programming. *Math. Program.* **151**(1), 35–62 (2015)

54. Hoeffding, W.: On the distribution of the number of successes in independent trials. *Ann. Math. Stat.* **27**(3), 713–721 (1956)
55. Hotelling, H.: Abraham Wald. *Am. Stat.* **5**(1), 18–19 (1951)
56. Huang, C.C., Vertinsky, I., Ziembka, W.T.: Sharp bounds on the value of perfect information. *Oper. Res.* **25**(1), 128–139 (1977)
57. Huang, C.C., Ziembka, W.T., Ben-Tal, A.: Bounds on the expectation of a convex function of a random variable: with applications to stochastic programming. *Oper. Res.* **25**(2), 315–325 (1977)
58. Huber, P.J.: Robust estimation of a location parameter. *Ann. Math. Stat.* **35**, 73–101 (1964)
59. Huber, P.J.: The 1972 Wald lecture- Robust statistics: a review. *Ann. Math. Stat.* 1041–1067 (1972)
60. Isii, K.: On a method for generalizations of Tchebycheff's inequality. *Ann. Inst. Stat. Math. Tokyo* **10**(2), 65–88 (1959)
61. Isii, K.: The extrema of probability determined by generalized moments. I. Bounded random variables. *Ann. Inst. Stat. Math.* **12**(2), 119–134; errata, 280 (1960)
62. Isii, K.: On sharpness of Tchebycheff-type inequalities. *Ann. Inst. Stat. Math.* **14**(1):185–197, 1962/1963.
63. Jaynes, E.T.: Probability Theory. Cambridge University Press, Cambridge (2003)
64. Joe, H.: Majorization, randomness and dependence for multivariate distributions. *Ann. Probab.* **15**(3), 1217–1225 (1987)
65. Johnstone, I.M.: High dimensional Bernstein–von Mises: simple examples. In Borrowing Strength: Theory Powering Applications—A Festschrift for Lawrence D. Brown, volume 6 of *Inst. Math. Stat. Collect.*, pages 87–98. Inst. Math. Statist., Beachwood, OH (2010)
66. Kac, M., Slepian, D.: Large excursions of Gaussian processes. *Ann. Math. Stat.* **30**, 1215–1228 (1959)
67. Kall, P.: Stochastic programming with recourse: upper bounds and moment problems: a review. *Math. Res.* **45**, 86–103 (1988)
68. Kallenberg, O.: Random Measures. Akademie-Verlag, Berlin (1975) Schriftenreihe des Zentralinstituts für Mathematik und Mechanik bei der Akademie der Wissenschaften der DDR, Heft 23.
69. Kamga, P.-H.T., Li, B., McKerns, M., Nguyen, L.H., Ortiz, M., Owhadi, H., Sullivan, T.J.: Optimal uncertainty quantification with model uncertainty and legacy data. *J. Mech. Phys. Solids* **72**, 1–19 (2014)
70. Karlin, S., Studden, W.J.: Tchebycheff Systems: With Applications in Analysis and Statistics. Pure and Applied Mathematics, vol. XV. Interscience Publishers/Wiley, New York/London/Sydney (1966)
71. Kendall, D.G.: Simplexes and vector lattices. *J. Lond. Math. Soc.* **37**(1), 365–371 (1962)
72. Kidane, A.A., Lashgari, A., Li, B., McKerns, M., Ortiz, M., Owhadi, H., Ravichandran, G., Stalzer, M., Sullivan, T.J.: Rigorous model-based uncertainty quantification with application to terminal ballistics. Part I: Systems with controllable inputs and small scatter. *J. Mech. Phys. Solids* **60**(5), 983–1001 (2012)
73. Kiefer, J.: Optimum experimental designs. *J. R. Stat. Soc. Ser. B* **21**, 272–319 (1959)
74. Kiefer, J.: Collected Works, vol. III. Springer, New York (1985)
75. Kleijn, B.J.K., van der Vaart, A.W.: The Bernstein-Von-Mises theorem under misspecification. *Electron. J. Stat.* **6**, 354–381 (2012)
76. Kolmogorov, A.N.: Foundations of the Theory of Probability. Chelsea Publishing Co., New York (1956). Translation edited by Nathan Morrison, with an added bibliography by A. T. Bharucha-Reid
77. Krein, M.G.: The ideas of P. L. Čebyšev and A. A. Markov in the theory of limiting values of integrals and their further development. In: Dynkin, E.B. (ed.) Eleven Papers on Analysis, Probability, and Topology, American Mathematical Society Translations, Series 2, vol. 12, pp. 1–122. American Mathematical Society, New York (1959)
78. Kurz, H.D., Salvadori, N.: Understanding ‘Classical’ Economics: Studies in Long Period Theory. Routledge, London/New York (2002)

79. Laird, N.M.: A conversation with F. N. David. *Stat. Sci.* **4**, 235–246 (1989)
80. Le Cam, L.: On some asymptotic properties of maximum likelihood estimates and related Bayes' estimates. *Univ. Calif. Publ. Stat.* **1**, 277–329 (1953)
81. Le Cam, L.: An extension of Wald's theory of statistical decision functions. *Ann. Math. Stat.* **26**, 69–81 (1955)
82. Le Cam, L.: Sufficiency and approximate sufficiency. *Ann. Math. Stat.* **35**, 1419–1455 (1964)
83. Le Cam, L.: Asymptotic Methods in Statistical Decision Theory. Springer, New York (1986)
84. Leahu, H.: On the Bernstein–von Mises phenomenon in the Gaussian white noise model. *Electron. J. Stat.* **5**, 373–404 (2011)
85. Lehmann, E.L.: "Student" and small-sample theory. *Stat. Sci.* **14**(4), 418–426 (1999)
86. Lehmann, E.L.: Optimality and symposia: some history. *Lect. Notes Monogr. Ser.* **44**, 1–10 (2004)
87. Lehmann, E.L.: Some history of optimality. *Lect. Notes Monogr. Ser.* **57**, 11–17 (2009)
88. Lenhard, J.: Models and statistical inference: the controversy between Fisher and Neyman–Pearson. *Br. J. Philos. Sci.* **57**(1), 69–91 (2006)
89. Leonard, R.: Von Neumann, Morgenstern, and the Creation of Game Theory: From Chess to Social Science, 1900–1960. Cambridge University Press, New York (2010)
90. Lynch, P.: The origins of computer weather prediction and climate modeling. *J. Comput. Phys.* **227**(7), 3431–3444 (2008)
91. Madansky, A.: Bounds on the expectation of a convex function of a multivariate random variable. *Ann. Math. Stat.* 743–746 (1959)
92. Madansky, A.: Inequalities for stochastic linear programming problems. *Manag. Sci.* **6**(2), 197–204 (1960)
93. Mangel, M., Samaniego, F.J.: Abraham Wald's work on aircraft survivability. *J. A. S. A.* **79**(386), 259–267 (1984)
94. Marshall, A.W., Olkin, I.: Multivariate Chebyshev inequalities. *Ann. Math. Stat.* **31**(4), 1001–1014 (1960)
95. Marshall, A.W., Olkin, I.: Inequalities: Theory of Majorization and Its Applications. Mathematics in Science and Engineering, vol. 143. Academic [Harcourt Brace Jovanovich Publishers], New York (1979)
96. McKerns, M.M., Strand, L., Sullivan, T.J., Fang, A., Aivazis, M.A.G.: Building a framework for predictive science. In: Proceedings of the 10th Python in Science Conference (SciPy 2011) (2011)
97. Morgenstern, O.: Abraham Wald, 1902–1950. *Econometrica: J. Econom. Soci.* 361–367 (1951)
98. Mulholland, H.P., Rogers, C.A.: Representation theorems for distribution functions. *Proc. Lond. Math. Soc.* (3) **8**(2), 177–223 (1958)
99. Nash, J.: Non-cooperative games. *Ann. Math.* (2) **54**, 286–295 (1951)
100. Nash, J.F. Jr.: Equilibrium points in n -person games. *Proc. Natl. Acad. Sci. U. S. A.* **36**, 48–49 (1950)
101. Nemirovsky, A.S.: Information-based complexity of linear operator equations. *J. Complex.* **8**(2), 153–175 (1992)
102. Neyman, J.: Outline of a theory of statistical estimation based on the classical theory of probability. *Philos. Trans. R. Soc. Lond. Ser. A* **236**(767), 333–380 (1937)
103. Neyman, J.: A Selection of Early Statistical Papers of J. Neyman. University of California Press, Berkeley (1967)
104. Neyman, J., Pearson, E.S.: On the use and interpretation of certain test criteria for purposes of statistical inference. *Biometrika* **20A**, 175–240, 263–294 (1928)
105. Neyman, J., Pearson, E.S.: On the problem of the most efficient tests of statistical hypotheses. *Philos. Trans. R. Soc. Lond. Ser. A* **231**, 289–337 (1933)
106. Olkin, I., Pratt, J.W.: A multivariate Tchebycheff inequality. *Ann. Math. Stat.* **29**(1), 226–234 (1958)

107. Owhadi, H.: Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games. *SIAM Rev. (Research spotlights)* (2016, to appear). arXiv:1503.03467
108. Owhadi, H., Scovel, C.: Qualitative robustness in Bayesian inference. arXiv:1411.3984 (2014)
109. Owhadi, H., Scovel, C.: Brittleness of Bayesian inference and new Selberg formulas. *Commun. Math. Sci.* **14**(1), 83–145 (2016)
110. Owhadi, H., Scovel, C.: Extreme points of a ball about a measure with finite support. *Commun. Math. Sci.* (2015, to appear). arXiv:1504.06745
111. Owhadi, H., Scovel, C.: Separability of reproducing kernel Hilbert spaces. *Proc. Am. Math. Soc.* (2015, to appear). arXiv:1506.04288
112. Owhadi, H., Scovel, C., Sullivan, T.J.: Brittleness of Bayesian inference under finite information in a continuous world. *Electron. J. Stat.* **9**, 1–79 (2015)
113. Owhadi, H., Scovel, C., Sullivan, T.J.: On the Brittleness of Bayesian Inference. *SIAM Rev. (Research Spotlights)* (2015)
114. Owhadi, H., Scovel, C., Sullivan, T.J., McKerns, M., Ortiz, M.: Optimal Uncertainty Quantification. *SIAM Rev.* **55**(2), 271–345 (2013)
115. Packel, E.W.: The algorithm designer versus nature: a game-theoretic approach to information-based complexity. *J. Complex.* **3**(3), 244–257 (1987)
116. Pearson, E.S.: ‘Student’ A Statistical Biography of William Sealy Gosset. Clarendon Press, Oxford (1990)
117. Pfanzagl, J.: Conditional distributions as derivatives. *Ann. Probab.* **7**(6), 1046–1050 (1979)
118. Pinelis, I.: Exact inequalities for sums of asymmetric random variables, with applications. *Probab. Theory Relat. Fields* **139**(3-4):605–635 (2007)
119. Pinelis, I.: On inequalities for sums of bounded random variables. *J. Math. Inequal.* **2**(1), 1–7 (2008)
120. Platzman, G.W.: The ENIAC computations of 1950-gateway to numerical weather prediction. *Bull. Am. Meteorol. Soc.* **60**, 302–312 (1979)
121. Ressel, P.: Some continuity and measurability results on spaces of measures. *Mathematica Scandinavica* **40**, 69–78 (1977)
122. Rikun, A.D.: A convex envelope formula for multilinear functions. *J. Global Optim.* **10**(4), 425–437 (1997)
123. Rockafellar, R.T.: Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM J. Control* **12**(2), 268–285 (1974)
124. Rojo, J.: Optimality: The Second Erich L. Lehmann Symposium. IMS, Beachwood (2006)
125. Rojo, J.: Optimality: The Third Erich L. Lehmann Symposium. IMS, Beachwood (2009)
126. Rojo, J., Pérez-Abreu, V.: The First Erich L. Lehmann Symposium: Optimality. IMS, Beachwood (2004)
127. Rustem, B., Howe, M.: Algorithms for Worst-Case Design and Applications to Risk Management. Princeton University Press, Princeton (2002)
128. Savage, L.J.: The theory of statistical decision. *J. Am. Stat. Assoc.* **46**, 55–67 (1951)
129. Scovel, C., Hush, D., Steinwart, I.: Approximate duality. *J. Optim. Theory Appl.* **135**(3), 429–443 (2007)
130. Shapiro, A., Kleywegt, A.: Minimax analysis of stochastic problems. *Optim. Methods Softw.* **17**(3), 523–542 (2002)
131. Sherali, H.D.: Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta Math. Vietnam.* **22**(1), 245–270 (1997)
132. Singpurwalla, N.D., Swift, A.: Network reliability and Borel’s paradox. *Am. Stat.* **55**(3), 213–218 (2001)
133. Smith, J.E.: Generalized Chebychev inequalities: theory and applications in decision analysis. *Oper. Res.* **43**(5), 807–825 (1995)

134. Sniedovich, M.: The art and science of modeling decision-making under severe uncertainty. *Decis. Mak. Manuf. Serv.* **1**(1–2), 111–136 (2007)
135. Sniedovich, M.: A classical decision theoretic perspective on worst-case analysis. *Appl. Math.* **56**(5), 499–509 (2011)
136. Sniedovich, M.: Black Swans, new Nostradamuses, Voodoo decision theories, and the science of decision making in the face of severe uncertainty. *Int. Trans. Oper. Res.* **19**(1–2), 253–281 (2012)
137. Spanos, A.: Why the Decision-Theoretic Perspective Misrepresents Frequentist Inference (2014). <https://secure.hosting.vt.edu/www.econ.vt.edu/directory/spanos/spanos10.pdf>
138. Stein, C.: Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In: Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, 1954–1955, vol. I, pp. 197–206. University of California Press, Berkeley/Los Angeles (1956)
139. Strasser, H.: Mathematical Theory of Statistics: Statistical Experiments and Asymptotic Decision Theory, vol. 7. Walter de Gruyter, Berlin/New York (1985)
140. Stuart, A.M.: Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559 (2010)
141. Student: The probable error of a mean. *Biometrika* 1–25 (1908)
142. Sullivan, T.J., McKerns, M., Meyer, D., Theil, F., Owhadi, H., Ortiz, M.: Optimal uncertainty quantification for legacy data observations of Lipschitz functions. *ESAIM Math. Model. Numer. Anal.* **47**(6), 1657–1689 (2013)
143. Tintner, G.: Abraham Wald's contributions to econometrics. *Ann. Math. Stat.* **23**, 21–28 (1952)
144. Tjur, T.: Conditional Probability Distributions, Lecture Notes, No. 2. Institute of Mathematical Statistics, University of Copenhagen, Copenhagen (1974)
145. Tjur, T.: Probability Based on Radon Measures. Wiley Series in Probability and Mathematical Statistics. Wiley, Chichester (1980)
146. Traub, J.F., Wasilkowski, G.W., Woźniakowski, H.: Information-Based Complexity. Computer Science and Scientific Computing. Academic, Boston (1988). With contributions by A. G. Werschulz and T. Boult
147. Tukey, J.W.: Statistical and Quantitative Methodology. Trends in Social Science, pp. 84–136. Philosophical Library, New York (1961)
148. Tukey, J.W.: The future of data analysis. *Ann. Math. Stat.* **33**, 1–67 (1962)
149. Valiant, L.G.: A theory of the learnable. *Commun. ACM* **27**(11), 1134–1142 (1984)
150. Vandenberghe, L., Boyd, S., Comanor, K.: Generalized Chebyshev bounds via semidefinite programming. *SIAM Rev.* **49**(1), 52–64 (electronic) (2007)
151. Varadarajan, V.S.: Groups of automorphisms of Borel spaces. *Trans. Am. Math. Soc.* **109**(2), 191–220 (1963)
152. von Mises, R.: Mathematical Theory of Probability and Statistics. Edited and Complemented by Hilda Geiringer. Academic, New York (1964)
153. Von Neumann, J.: Zur Theorie der Gesellschaftsspiele. *Math. Ann.* **100**(1), 295–320 (1928)
154. Von Neumann, J., Goldstine, H.H.: Numerical inverting of matrices of high order. *Bull. Am. Math. Soc.* **53**, 1021–1099 (1947)
155. Von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
156. Wald, A.: Contributions to the theory of statistical estimation and testing hypotheses. *Ann. Math. Stat.* **10**(4), 299–326 (1939)
157. Wald, A.: Statistical decision functions which minimize the maximum risk. *Ann. Math. (2)* **46**, 265–280 (1945)
158. Wald, A.: An essentially complete class of admissible decision functions. *Ann. Math. Stat.* **18**, 549–555 (1947)
159. Wald, A.: Sequential Analysis. 1947.
160. Wald, A.: Statistical decision functions. *Ann. Math. Stat.* **20**, 165–205 (1949)
161. Wald, A.: Statistical Decision Functions. Wiley, New York (1950)

-
- 162. Wald, A., Wolfowitz, J.: Optimum character of the sequential probability ratio test. *Ann. Math. Stat.* **19**(3), 326–339 (1948)
 - 163. Wald, A., Wolfowitz, J.: Characterization of the minimal complete class of decision functions when the number of distributions and decisions is finite. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 149–157. University of California Press, Berkeley (1951)
 - 164. Wasserman, L.: Rise of the Machines. Past, Present and Future of Statistical Science. CRC Press, Boca Raton (2013)
 - 165. Wasserman, L., Lavine, M., Wolpert, R.L.: Linearization of Bayesian robustness problems. *J. Stat. Plann. Inference* **37**(3), 307–316 (1993)
 - 166. Wiesemann, W., Kuhn, D., Sim, M.: Distributionally robust convex optimization. *Oper. Res.* **62**(6), 1358–1376 (2014)
 - 167. Wilson, M.: How a story from World War II shapes Facebook today. IBM Watson (2012). <http://www.fastcodesign.com/1671172/how-a-story-from-world-war-ii-shapes-facebook-today>.
 - 168. Winkler, G.: On the integral representation in convex noncompact sets of tight measures. *Mathematische Zeitschrift* **158**(1), 71–77 (1978)
 - 169. Winkler, G.: Extreme points of moment sets. *Math. Oper. Res.* **13**(4), 581–587 (1988)
 - 170. Wolfowitz, J.: Abraham Wald, 1902–1950. *Ann. Math. Stat.* **23**, 1–13 (1952)
 - 171. Woźniakowski, H.: Probabilistic setting of information-based complexity. *J. Complex.* **2**(3), 255–269 (1986)
 - 172. Woźniakowski, H.: What is information-based complexity? In Essays on the Complexity of Continuous Problems, pp. 89–95. European Mathematical Society, Zürich (2009)
 - 173. Wynn, H.P.: Introduction to Kiefer (1959) Optimum Experimental Designs. In Breakthroughs in Statistics, pp. 395–399. Springer, New York (1992)
 - 174. Xu, L., Yu, B., Liu, W.: The distributionally robust optimization reformulation for stochastic complementarity problems. *Abstr. Appl. Anal.* **2014**, Art. ID 469587, (2014)
 - 175. Žáčková, J.: On minimax solutions of stochastic linear programming problems. *Časopis Pěst. Mat.* **91**, 423–430 (1966)
 - 176. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)
 - 177. Zymler, S., Kuhn, D., Rustem, B.: Distributionally robust joint chance constraints with second-order moment information. *Math. Program.* **137**(1–2, Ser. A), 167–198 (2013)

Hierarchical Models for Uncertainty Quantification: An Overview

7

Christopher K. Wikle

Abstract

Analyses of complex processes should account for the uncertainty in the data, the processes that generated the data, and the models that are used to represent the processes and data. Accounting for these uncertainties can be daunting in traditional statistical analyses. In recent years, hierarchical statistical models have provided a coherent probabilistic framework that can accommodate these multiple sources of quantifiable uncertainty. This overview describes a science-based hierarchical statistical modeling approach and the associated Bayesian inference. In addition, given that many complex processes involve the dynamical evolution of spatial processes, an overview of hierarchical dynamical spatio-temporal models is also presented. The hierarchical and spatio-temporal modeling frameworks are illustrated with a problem concerned with assimilating ocean vector wind observations from satellite and weather center analyses.

Keywords

Bayesian • Basis functions • BHM • Integro-difference equations • Latent process • Quadratic nonlinearity • MCMC • Multivariate • Ocean • Reduced-rank representation • Spatio-temporal • Wind

Contents

1	Introduction	194
2	Hierarchical Modeling in the Presence of Uncertainty	195
2.1	Basic Hierarchical Structure	196
2.2	Data Models	198
2.3	Process Models	198
2.4	Parameter Models	199
2.5	Bayesian Formulation	200

C.K. Wikle (✉)

Department of Statistics, University of Missouri, Columbia, MO, USA

e-mail: wiklec@missouri.edu

3	Dynamical Spatio-temporal Process Models.....	201
3.1	Linear DSTM Process Models.....	202
3.2	Nonlinear DSTM Process Models.....	203
3.3	Multivariate DSTM Process Models.....	204
3.4	Process and Parameter Reduction.....	205
4	Example: Near-Surface Winds Over the Ocean.....	207
4.1	Surface Vector Wind Background.....	208
4.2	Ocean SVW BHM.....	209
4.3	Implementation.....	213
4.4	Results.....	214
5	Conclusion.....	215
	References.....	217

1 Introduction

Scientists and engineers are increasingly aware of the importance of accurately characterizing various sources of uncertainty when trying to understand complex systems. When performing statistical modeling on complex phenomena, the goal is typically either inference, prediction, or forecasting. To accomplish these goals through modeling, one must synthesize information. This information can come from a variety of sources, including direct (*in situ*) observations, indirect (remotely sensed) observations, surrogate observations, previous empirical results, expert opinion, and scientific principles. In order to make inferential or predictive decisions with a statistical model, one must consider these sources of information in a coherent manner that accounts adequately for the various sources of uncertainty that are present. That is, there may be measurement error, model representativeness error, error associated with differing levels of support between observations and process, parameterization error, and parameter uncertainty. Over the last 20 years or so, one of the most useful statistical paradigms in which to consider complex models in the presence of uncertainty is *hierarchical modeling* (HM). The purpose of this overview is to outline the general principles of science-based statistical HM and its utility to a wide class of processes.

Hierarchical modeling is, at its core, just a system of coherently linked probability relationships. In this sense, it is certainly not a new idea, and from a modeling perspective, such ideas have been at the core of fundamental statistical methods such as mixed models, structural equation models, spatial models, directed acyclic graph models, among others. This class of models might be referred to as “little h” hierarchical models. That is, one is either focused on a data model (i.e., “likelihood”) and parameters, with the process considered a nuisance, or a data model and process model, with the parameters considered a nuisance. The perspective presented in this overview follows more closely the perspective originally outlined by Mark Berliner [4] in a somewhat obscure conference proceedings paper written while he was the director of the Geophysical Statistics Project at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado, USA. In this seminal paper, Berliner presents a simple, yet fundamentally important, way to think about partitioning uncertainty associated with data, processes, and parameters in complex

systems. As described below, the basic tenet of this modeling paradigm is to characterize uncertainty in the joint model of data, process, and parameters in terms of component conditional and marginal distributions, which is often facilitated by the inclusion of scientific information. The advent of this formulation coincided with the so-called computational Bayesian revolution, specifically in terms of Markov chain Monte Carlo (MCMC) methods that were facilitated by the classic paper of [10]. This understanding provided the practical tools necessary to implement these models in the Bayesian context. One of the key components of thinking about models from this perspective is that one deliberately pushes complexity into the conditional mean, in which case subprocesses and parameters are often modeled with fairly complex dependence structures. This [4] hierarchical modeling paradigm might be referred to as a “big H” hierarchical model (HM), to emphasize that the conditional structure and parameter models are fundamental to the HM, not just a nuisance, and that scientific/mechanistic information is included in the various components of the HM.

The chapter begins with a general overview of hierarchical modeling and its Bayesian implementation. This is then followed by an overview of discrete-time spatio-temporal dynamical processes, given their importance as component models in many complex hierarchical modeling applications. A discussion of process and parameter space reduction is included in this overview of spatio-temporal processes. A simple illustrative example based on blending different sources of information for ocean surface vector winds is presented to highlight some of the important components of hierarchical modeling. Finally, a brief conclusion is presented that outlines the trade-offs one has to consider when building complex BHMs.

2 Hierarchical Modeling in the Presence of Uncertainty

This section presents a broad overview of statistical hierarchical modeling. The focus of this presentation is on the role of conditional models and, specifically, the separation of the joint model into coherently linked models for data, process, and parameters. This discussion follows similar discussions in [7, 8, 38], and [43].

To motivate the discussion of HMs, consider a problem in which one has observations of near-surface winds over the ocean from a satellite scatterometer and wishes to “predict” the distribution of complete spatial fields of the true wind across time. That is, there are satellite observations of east-west and north-south wind components that occur at a fairly fine spatial resolution, but are incomplete spatially due to the polar orbit of the satellite, and the goal is to interpolate the observations to form complete spatial fields for a regular sequence of times. In this case, the “process” corresponds to the wind components and, potentially, other relevant atmospheric state variables (e.g., sea-level pressure). In addition to the satellite observations, there is additional information from weather center “analysis” fields (i.e., model output from operational data assimilation systems that combine worldwide weather observations and deterministic weather forecasting models).

It is reasonable to assume that the satellite-derived scatterometer observations have not been used to produce the weather center data assimilation products.

For purposes of this general exposition, let the wind observations (data) be denoted by D . One possible approach to solving the aforementioned interpolation problem is to apply some deterministic curve-fitting interpolation algorithm $\hat{D} = f(D)$ (e.g., linear, polynomial, or spline interpolation). However, such approaches do not account for the uncertainty associated with the observations and, more importantly, do not utilize scientific knowledge to help fill in the data gaps in a physically plausible manner.

A more traditional statistical modeling alternative to this curve-fitting interpolation approach might consider a distribution for the data conditioned on some parameters, say θ_o , which is denoted by $[D | \theta_o]$. Note the use of a bracket notation for distribution, “[],” which is common in the hierarchical modeling literature, where the vertical bar, “|,” denotes conditioning, $[A, B]$ represents the joint distribution of A and B , and $[A | B]$ represents the conditional distribution of A “given” B . In the traditional statistical model, one would seek the parameters, θ_o , that maximize the likelihood of observing the data D . Of course, this assumes that the distributional assumption adequately captures all of the variability (spatial, temporal, multivariate, etc.) in the data, subject to the correct specification of the parameters. Although this is very much a reasonable paradigm in many traditional statistical modeling problems, it is extremely tenuous in the example considered here (and, indeed, most complex physical, biological, or engineering problems) because it is typically not possible to adequately represent the complexity in the data via a single distributional assumption. In particular, this approach does not consider the fact that much of the complexity of the data arises from the scientific *process* (e.g., the atmospheric state variables in the wind example).

2.1 Basic Hierarchical Structure

A scientific modeling approach considers a model for the process of interest, say W here for “wind.” Recognizing the fact that one’s understanding of such scientific processes is always limited, this uncertainty is accounted for via a stochastic representation, denoted by the distribution $[W | \theta_W]$, where θ_W are parameters. The traditional statistical approach described above does not explicitly account for this uncertainty nor the uncertainty about the relationship between D and W . To see this more clearly, one might decompose the joint distribution of the data and the process given the associated parameters as

$$[D, W | \theta_D, \theta_W] = [D | W, \theta_D][W | \theta_W], \quad (7.1)$$

where the parameters in the conditional distribution of D given the process W are denoted by θ_D , which are different than the parameters θ_o for the marginal distribution of the data described above. That is, integrating out the process, W , from (7.1), gives $[D | \theta_o = \{\theta_D, \theta_W\}]$, which implies that the complexity associated

with the process W is present in the marginal form of this data distribution and the associated parameters. Typically, this integration cannot be done analytically, and so one does not know the actual form of this marginal data likelihood nor could one generally account for the complicated multivariate spatio-temporal dependence in such a parameterization for real-world processes (e.g., nonlinearity, nonstationarity, non-Gaussianity). Even in the rare situation where one can do the integration analytically (e.g., Gaussian data and process models), the marginal dependence structure is typically more complicated than can be captured by traditional spatio-temporal parameterizations, that is, the dependence is some complicated function of the parameters θ_D and θ_W . Perhaps more importantly, in the motivating application considered here, the interest is with W , so one does not want to integrate it out of (7.1). Indeed, one typically wants to predict this process distribution. This separation between the data model conditional on the process and the process model is exactly the paradigm in traditional state-space models in engineering and time-series applications [e.g., 29]. More generally, the trade-off between considering a statistical model from the marginal perspective, in which the random process (parameters) are integrated out, and the conditional perspective, in which complicated dependence structures must be parameterized, is just the well-known trade-off that occurs in traditional mixed-model analysis in statistics [e.g., 31].

The decomposition given in (7.1) above is powerful in the sense that it separates the uncertainty associated with the process and the uncertainty associated with the observation of the process. However, it does not factor in the uncertainty associated with the parameters themselves. Utilizing basic probability, one can always decompose a joint distribution into a sequence of conditional and marginal distributions. For example, $[A, B, C] = [A | B, C][B | C][C]$. Thus, the hierarchical decomposition can be written as

$$[D, W, \theta] = [D | W, \theta][W | \theta][\theta], \quad (7.2)$$

where $\theta = \{\theta_D, \theta_W\}$. This hierarchical decomposition is not unique, e.g., it is equally valid probabilistically to write $[A, B, C] = [C | B, A][A | B][B]$, but the decomposition in (7.2) is meaningful scientifically as it implies causality in the sense that the parameters drive the process and the process generates the data, etc. In addition, note that the distributions on the right-hand side (RHS) of (7.2) could be simplified such that $[D | W, \theta] = [D | W, \theta_D]$ and $[W | \theta_W]$, that is, it might be reasonable to assume conditional independence in the parameter decomposition. This is a modeling choice, but it is reasonable in this case based on how the individual data and process distributions were specified above. More generally, it is helpful to consider the following schematic representation of [4] when partitioning uncertainty in hierarchical decompositions as it provides a framework for building probabilistically consistent models:

$$\begin{aligned} [\text{data, process, parameters}] &= [\text{data} | \text{process, parameters}] \\ &\times [\text{process} | \text{parameters}] \times [\text{parameters}]. \end{aligned} \quad (7.3)$$

2.2 Data Models

Each of the stages of the hierarchy given in (7.3) can be decomposed into products of distributions or submodels. For example, say there are three datasets for the near-ocean surface wind process (W) denoted by $D^{(1)}$, $D^{(2)}$, and $D^{(3)}$. These might correspond to the satellite scatterometer data mentioned previously, ocean buoy data, and the weather center analysis data product. These observations need not be coincident nor even of the same spatial or temporal support as the other data nor the process. In this case, the data model might be represented as

$$[D^{(1)}, D^{(2)}, D^{(3)} | W, \theta_D] = [D^{(1)} | W, \theta_D^{(1)}][D^{(2)} | W, \theta_D^{(2)}][D^{(3)} | W, \theta_D^{(3)}], \quad (7.4)$$

where the parameters for each submodel are given by $\theta_D = \{\theta_D^{(1)}, \theta_D^{(2)}, \theta_D^{(3)}\}$. The RHS of (7.4) makes use of the assumption that the three datasets are all conditionally independent given the true process. This is not to say that the data are independent marginally, as they surely are not. Yet, the assumption of conditional independence is a powerful simplifying modeling assumption that is often reasonable in complex systems, but must be justified in practice. It is important to emphasize that the specific forms of the component distributions on the RHS of (7.4) can be quite different from each other, accounting for the differing support and measurement properties associated with the specific dataset. For example, satellite scatterometer wind observations have fairly well-known measurement-error properties and are associated with fairly small areal “footprints” (depending on the specific instrument), but wind observations from an ocean buoy are best considered point-level support with well-calibrated measurement-error properties.

2.3 Process Models

Typically, the process model in the hierarchical decomposition can also be further decomposed into component distributions. For example, in the case of the wind example described here, the wind process is a vector composed of two components, speed and direction or, equivalently, north-south and east-west components that depend on pressure. That is, one might write

$$[W^{(1)}, W^{(2)}, W^{(3)} | \theta_W] = [W^{(1)}, W^{(2)} | W^{(3)}, \theta_W^{(1,2)}][W^{(3)} | \theta_W^{(3)}], \quad (7.5)$$

where $W^{(1)}$ and $W^{(2)}$ correspond to the east-west and north-south wind components (typically denoted by u and v , respectively) and $W^{(3)}$ corresponds to the near-surface atmospheric pressure (typically denoted P). The decomposition in (7.5) is not unique, but is sensible in this case because there is strong scientific justification for conditioning the wind on the pressure [e.g., 14]. The process parameters are again decomposed into those components associated with each distribution, $\theta_W = \{\theta_W^{(1,2)}, \theta_W^{(3)}\}$. The decomposition in (7.5) simplifies the joint dependence structure

between the various process components by utilizing simplifying assumptions based on scientific input. It is important to recognize that these components are still distributions, so that the uncertainties in the relationships (say, between wind and pressure) can be accommodated through appropriate modeling components (e.g., bias and error terms).

Other types of joint interactions in the process can also be simplified through such conditional probability relationships. For example, given that the wind process is time varying, one might be able to make Markov assumptions in time. For example, if W_t corresponds to the wind process at time t for $t = 0, \dots, T$, then

$$[W_0, W_1, \dots, W_T | \theta_W] = \prod_{t=1}^T [W_t | W_{t-1}, \theta_W][W_0], \quad (7.6)$$

represents a first-order Markov assumption, that is, the process is independent of the past if conditioned on the most recent past. This is a significant simplifying assumption, and must be justified in practice, but such assumptions are often very realistic for real-world time-varying processes. Similar sorts of conditioning arguments can be made for networks, spatial processes (e.g., Markov random fields), and spatio-temporal processes (e.g., spatio-temporal dynamical models) as described in [7].

2.4 Parameter Models

An important consequence of the hierarchical modeling paradigm described above is the recognition that additional complexity can be accommodated by allowing the parameters to be random and endowing them with dependence structures (e.g., multivariate, spatial, temporal, etc.). That is, the parameter models can themselves be quite complex and can incorporate additional information, whether that be through exogenous data sources (e.g., a sea-surface temperature index corresponding to the El Niño/La Niña phenomenon) or scientific knowledge (e.g., spatial turbulent scaling relationships). For example, one might write $[\theta_W | X, \theta_X]$, where X is some exogenous covariate and θ_X are parameters. It can be very difficult, if not impossible, to account for such complex parameter dependence structures in the classical modeling approach discussed above.

Now, one must decide how to account for the uncertainty in X and θ_X , which often leads to yet another data or parameter level of the model hierarchy. Typically, at some point, there is no more information that can assist the specification of these distributions, and one either assigns some sort of non-informative distribution to the parameters or, in some cases, estimates them through some other means.

It is apparent that the distinction between “process” and “parameter” may not always be precise. This can be the case in some applications, but the strength of the hierachal paradigm is that it is the complete sequence of the hierarchical decomposition that is important, *not* what one calls “process” or “parameter.”

This suggests that one requires a flexible inferential paradigm that allows one to perform inference and prediction on both process and parameters and even their joint interaction.

2.5 Bayesian Formulation

The Bayesian paradigm fits naturally with hierarchical models because the posterior distribution is proportional to the product distributions in the hierarchical decomposition. For example, in the schematic representation of [4] given in (7.3), the posterior distribution can be written via Bayes' rule as

$$\begin{aligned} [\text{process, parameters} | \text{data}] &\propto [\text{data} | \text{process, parameters}] \\ &\quad \times [\text{process} | \text{parameters}] \times [\text{parameters}], \end{aligned} \quad (7.7)$$

where the normalizing constant is the integral (in the case of continuous distributions) of (7.3) with respect to the process and parameters (i.e., the marginal distribution of the data). In the context of the wind example, the posterior distribution can be written

$$[W, \theta_W, \theta_D | D] \propto [D | W, \theta_D][W | \theta_W][\theta_D, \theta_W]. \quad (7.8)$$

In practice, it is not typically possible to calculate the normalizing constant ($1/[D]$) analytically. With the understanding that Markov chain Monte Carlo (MCMC) methods could be used generally for such purposes (i.e., after the seminal paper of [10]), this has not been a serious limitation.

MCMC methods seek to draw simulation samples from a distribution that coincides with the posterior distribution of interest. In particular, a Markov chain is constructed algorithmically such that samples from the stationary distribution of the Markov chain correspond to samples from the desired posterior distribution. Details of the implementation of such algorithms are beyond the scope of this overview, but they can be found in references such as [25] and [6]. Alternatively, approximate solutions can sometimes be found with less computational burden, such as with variational methods, approximate Bayesian computation (ABC), and integrated nested Laplace approximations (INLA) [e.g., 21, 27, 30]. In general, one must find trade-offs between model complexity and computational complexity when building complex statistical models in the presence of uncertainty (see the Conclusion of this chapter).

In some simpler modeling situations (e.g., state-space models), one might be content with assuming the parameters are fixed but unknown rather than assign them distributions. In that case, one could write (7.8) as

$$[W | D, \theta_W, \theta_D] \propto [D | W, \theta_D][W | \theta_W]. \quad (7.9)$$

In applications where the component models are not too complex, these parameters can be estimated using classical statistical approaches, and then the parameters are used in a “plug-in” fashion in the model. For example, in state-space modeling, one might estimate the parameters through an E-M algorithm and then evaluate the process distributions through a Kalman filter/smooth [e.g., 29]. Such an approach is sometimes called “empirical Bayes” or, in the context of hierarchical models, empirical hierarchical modeling (EHM) [e.g., 7]. A potential concern using such an approach is accounting for the uncertainty in the parameter estimation. In some cases, this uncertainty can be accounted for by Taylor approximations or bootstrap resampling methods [e.g., 29]. Typically, in complex models, the BHM framework provides a more sensible approach to uncertainty quantification than EHM approaches.

3 Dynamical Spatio-temporal Process Models

The motivating wind example discussed above can be thought of as a data assimilation (DA) problem. [33] characterize DA as a set of methods that blend observations with prior system knowledge in an optimal way in order to obtain a distributional summary of a process of interest. In this context, “system knowledge” can correspond to deterministic models, scientific/mechanistic relationships, model output, and expert opinion. As summarized in [33], there is a large literature in the physical sciences dedicated to various methods for DA. In many ways, this is just a type of inverse modeling, and many different solution approaches are possible. However, if DA is considered from a BHM perspective, then one can gain a more comprehensive characterization of the uncertainty associated with the data, process, and parameters. From a statistical perspective, these methods typically require a dynamical spatio-temporal model (DSTM) of some sort. Hence, this section gives a brief overview of hierarchical DSTMs. More complete details can be found in [7] and [40]. This overview considers only DSTMs from a discrete-time perspective for the sake of brevity. However, it should be noted that many science-oriented process models are specified from a continuous time perspective (e.g., differential equations) and these can be used either to motivate HMs or can be implemented directly within the HM framework (e.g., [4]).

The data model in a general DSTM can be written

$$Z_t(\cdot) = \mathcal{H}(Y_t(\cdot), \theta_d(t), \epsilon_t(\cdot)), \quad t = 1, \dots, T,$$

where $Z_t(\cdot)$ corresponds to the data at time t and $Y_t(\cdot)$ is the corresponding latent process of interest, with a linear or nonlinear mapping function, $\mathcal{H}(\cdot)$, that relates the data to the latent process. The data model error is given by $\epsilon_t(\cdot)$, and data model parameters are represented by $\theta_d(t)$. These parameters may vary spatially and/or temporally in general. As discussed more generally above, an important assumption that is present here, and in many hierarchical representations of DSTMs, is that the data $Z_t(\cdot)$ are independent in time when conditioned on the true process, $Y_t(\cdot)$

and parameters $\boldsymbol{\theta}_d(t)$. Thus, the observations conditioned on the true process and parameters can be represented

$$\prod_{t=1}^T [Z_t(\cdot) \mid Y_t(\cdot), \boldsymbol{\theta}_d(t)].$$

The key component of the DSTM is the dynamical process model. As discussed above, one can simplify this by making use of conditional independence through Markov assumptions. For example, a first-order Markov process can be written as

$$[Y_t(\cdot) | Y_{t-1}(\cdot), \dots, Y_0(\cdot), \{\boldsymbol{\theta}_p(t), t = 0, \dots, T\}] = [Y_t(\cdot) | Y_{t-1}(\cdot), \boldsymbol{\theta}_p(t)],$$

for $t = 1, 2, \dots$ so that

$$\begin{aligned} [Y_0(\cdot), Y_1(\cdot), \dots, Y_T(\cdot) | \{\boldsymbol{\theta}_p(t), t = 0, \dots, T\}] &= \prod_{t=1}^T [Y_t(\cdot) | Y_{t-1}(\cdot), \boldsymbol{\theta}_p(t)] \\ &\quad \times [Y_0(\cdot) | \boldsymbol{\theta}_p(0)]. \end{aligned} \quad (7.10)$$

Higher-order Markov assumptions could be considered if warranted by the specific problem of interest. Such relationships are critical for real-world spatio-temporal processes because they follow the etiology of process development.

Now, the modeling focus is on the component Markov models in (7.10). For example, a first-order process can be written generally as

$$Y_t(\cdot) = \mathcal{M}(Y_{t-1}(\cdot), \boldsymbol{\theta}_p(t), \eta_t(\cdot)), \quad t = 1, 2, \dots, \quad (7.11)$$

where $\mathcal{M}(\cdot)$ is the evolution operator (linear or nonlinear), $\eta_t(\cdot)$ is the noise (error) process, and $\boldsymbol{\theta}_p(t)$ are process model parameters that may vary with time and/or space. Typically, one would also specify a distribution for the initial state, $[Y_0(\cdot) | \boldsymbol{\theta}_p(0)]$.

The hierarchical model then requires distributions to be assigned to the parameters $\{\boldsymbol{\theta}_d(t), \boldsymbol{\theta}_p(t), t = 0, \dots, T\}$. Specific distributional forms for the parameters (e.g., spatially or temporally varying, dependence on auxiliary covariate information, etc.) depend strongly on the problem of interest. Indeed, as mentioned above, one of the most critical aspects of complex hierarchical modeling is the specification of these distributions. This is illustrated below with regard to linear and nonlinear DSTMs.

3.1 Linear DSTM Process Models

In the case where one has a discrete set of spatial locations $D_s = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$ of interest (e.g., a lattice or grid), the first-order evolution process model (7.11) can be written as

$$Y_t(\mathbf{s}_i) = \sum_{j=1}^n m_{ij}(\boldsymbol{\theta}_m) Y_{t-1}(\mathbf{s}_j) + \eta_t(\mathbf{s}_i), \quad (7.12)$$

for $t = 1, 2, \dots$, with redistribution (transition) components $m_{ij}(\boldsymbol{\theta}_m)$ that depend on parameters $\boldsymbol{\theta}_m$. If interest is in continuous space and discrete time, one can also write this in terms of an *integro-difference equation (IDE)*

$$Y_t(\mathbf{s}) = \int_{D_s} m(\mathbf{s}, \mathbf{x}; \boldsymbol{\theta}_m) Y_{t-1}(\mathbf{x}) d\mathbf{x} + \eta_t(\mathbf{s}), \quad \mathbf{s}, \mathbf{x} \in D_s, \quad (7.13)$$

for $t = 1, 2, \dots$, where $m(\mathbf{s}, \mathbf{x}; \boldsymbol{\theta}_m)$ is a *transition kernel* that gives redistribution weights for process at the previous time and $\eta_t(\mathbf{s})$ is a time-varying (continuous) spatial error process. Analogous stochastic partial differential equation models could be specified for continuous time and space.

Now, denoting the process vector $\mathbf{Y}_t \equiv (Y_t(\mathbf{s}_1), \dots, Y_t(\mathbf{s}_n))'$, (7.12) can be written in vector/matrix form as a first-order vector autoregression (VAR(1)) DSTM

$$\mathbf{Y}_t = \mathbf{M}\mathbf{Y}_{t-1} + \boldsymbol{\eta}_t, \quad (7.14)$$

where the $n \times n$ transition matrix is given by \mathbf{M} with elements $\{m_{ij}\}$ with the associated time-varying spatial error process given by $\boldsymbol{\eta}_t \equiv (\eta_t(\mathbf{s}_1), \dots, \eta_t(\mathbf{s}_n))'$, which is typically specified to be zero mean and Gaussian, with spatial variance-covariance matrix \mathbf{C}_η . Usually, \mathbf{M} and \mathbf{C}_η are assumed to depend on parameters $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_\eta$, respectively, to mitigate the curse of dimensionality that often occurs in spatio-temporal modeling. As discussed below, the parameterization of these matrices is one way that additional mechanistic information can be incorporated into the HM framework.

3.2 Nonlinear DSTM Process Models

Many mechanistic processes are best modeled nonlinearly, at least at some spatial and temporal scales of variability. A class of nonlinear statistical DSTMs can be specified to accommodate such processes with quadratic interactions. Such a *general quadratic nonlinear (GQN)* DSTM [35] can be written as

$$Y_t(\mathbf{s}_i) = \sum_{j=1}^n m_{ij} Y_{t-1}(\mathbf{s}_j) + \sum_{k=1}^n \sum_{\ell=1}^n b_{i,k\ell} Y_{t-1}(\mathbf{s}_k) g(Y_{t-1}(\mathbf{s}_\ell); \boldsymbol{\theta}_g) + \eta_t(\mathbf{s}_i), \quad (7.15)$$

where m_{ij} are the linear transition coefficients seen previously and quadratic interaction transition coefficients are denoted by $b_{i,k\ell}$. A transformation of one of the components of the quadratic interaction is specified through the function $g(\cdot)$, which can depend on parameters $\boldsymbol{\theta}_g$. This function $g(\cdot)$ is responsible for the “general” in GQN, and such transformations are critical for many processes such as density-dependent growth that one may see in an epidemic or invasive species

population process. The spatio-temporal error process is again typically assumed to be independent in time and Gaussian with mean zero and a spatial covariance matrix. Note that the conditional GQN model is Gaussian, but the marginal model will not in general be Gaussian because of the nonlinear interactions.

3.3 Multivariate DSTM Process Models

There are three primary approaches to modeling multivariate spatio-temporal dynamical processes in statistics. An obvious approach is to simply *augment* the process vector (e.g., concatenating the process vectors for a given time) and then using one of the univariate models (such as described above) to model the evolution of the process. That is, if there are J processes given by $\{\mathbf{Y}_t^{(j)}\}$, $j = 1, \dots, J$, then for time t one could write $\mathbf{W}_t \equiv (\mathbf{Y}_t^{(1)'}, \dots, \mathbf{Y}_t^{(J)'})'$ and then evolve \mathbf{W}_t as above. The simplicity of this approach is appealing, but it is often more difficult to incorporate scientific information into the process evolution. Perhaps more critically, this often leads to very high-dimensional process vectors, which compounds the curse of dimensionality issue that is endemic in spatio-temporal statistical modeling.

As discussed generally above, multivariate processes can be modeled hierarchically by using the law of total probability and applying some conditional independence assumptions. As a simple example, consider $J = 3$ processes for the component conditional distribution for time t given time $t - 1$ might be written as

$$\begin{aligned} [\mathbf{Y}_t^{(1)}, \mathbf{Y}_t^{(2)}, \mathbf{Y}_t^{(3)} | \mathbf{Y}_{t-1}^{(1)}, \mathbf{Y}_{t-1}^{(2)}, \mathbf{Y}_{t-1}^{(3)}] &= [\mathbf{Y}_t^{(1)} | \mathbf{Y}_t^{(3)}, \mathbf{Y}_{t-1}^{(1)}, \mathbf{Y}_{t-1}^{(2)}] \\ &\quad \times [\mathbf{Y}_t^{(2)} | \mathbf{Y}_t^{(3)}, \mathbf{Y}_{t-1}^{(1)}, \mathbf{Y}_{t-1}^{(2)}] [\mathbf{Y}_t^{(3)} | \mathbf{Y}_{t-1}^{(3)}]. \end{aligned}$$

That is, processes 1 and 2 are conditionally independent at time t given process 3 at time t and previous values of processes 1 and 2 at time $t - 1$, and process 3 at time t is conditionally independent of the others given its previous values. Such a model formulation has the advantage of being able to match up to mechanistic knowledge about the processes and their interactions. However, if such knowledge is not available, this conditional formulation is arbitrary (or there is no basis for the conditional independence assumptions), and such an approach is not recommended.

The third primary approach for modeling multivariate dynamical spatio-temporal processes is to condition the J processes on one or more latent processes, much like what is done in multivariate factor analysis. For a set of $K \leq J$ common latent dynamical processes, $\{\alpha_{\ell,t}^{(k)}\}$, which may or may not be spatially referenced, consider

$$Y_t^{(j)}(\mathbf{s}_t) = \sum_{\ell=1}^{n_\alpha} \sum_{k=1}^K h_{i,\ell}^{(jk)} \alpha_{\ell,t}^{(k)} + \eta_t^{(j)}(\mathbf{s}_t), \quad (7.16)$$

for $i = 1, \dots, n$, $j = 1, \dots, J$, where $h_{i,\ell}^{(jk)}$ are interaction coefficients that account for how the ℓ th element of the k th latent process influences the j th process at

location i . This is a powerful modeling framework, but the curse of dimensionality in parameter space can easily make this impractical. In addition, care must be taken when modeling the latent processes, which is typically done at the next level of the model hierarchy, as there are identifiability problems between the h parameters at this level and potential dynamical evolution parameters for the α processes at the next level [see 7, Section 7.4.2, for more discussion].

3.4 Process and Parameter Reduction

As mentioned above, one of the greatest challenges when considering DSTMs in hierarchical settings is the curse of dimensionality associated with the process and parameter space. For the fairly common situation where the number of spatial locations (n) is much larger than the number of time replicates (T), even the fairly simple linear VAR(1) model (7.14) is problematic as there are on order n^2 parameters to estimate. This is compounded for the GQN model (7.15), which has on order n^3 free parameters and similarly for the multivariate model. To proceed, one must reduce the number of free parameters to be estimated in the model and/or reduce the dimension of the dynamical process. These two approaches are discussed briefly below.

3.4.1 Parameter Reduction

Very seldom would one estimate the full variance/covariance matrix (\mathbf{C}_η) in the DSTM. Rather, given that these are spatial covariance matrices, one would either use one of the common spatial covariance function representations (e.g., Matérn, conditional autoregressive, etc.; see Cressie and Wikle [7, Chapter 4]) or a spatial random effect representation (see the “Process Reduction” section below). Generally, the transition parameters in the DSTM require the most care. For example, in the case of the simple VAR model (7.14), one could parameterize the transition matrix \mathbf{M} simply as a random walk (i.e., $\mathbf{M} = \mathbf{I}$), a spatially homogeneous autoregressive process (i.e., $\mathbf{M} = \theta \mathbf{I}$), or a spatially varying autoregressive process ($\mathbf{M} = \text{diag}(\theta_m)$). The first two parameterizations are somewhat unrealistic for most real-world dynamical processes, and the latter, although able to accommodate non-separable spatio-temporal dependence, does not account for interactions dynamically across space and time. Although in the context of evolving a spectral latent process (see below), such models can be very effective.

More mechanistically realistic dynamical parameterizations in the context of physical space representations recognize that spatio-temporal interactions are crucial for dynamical propagation. For example, in the linear case, the asymmetry and rate of decay of the transition parameters relative to a location (say, \mathbf{s}_i) control propagation (linear advection) and spread (diffusion). This suggests that a simple *lagged-nearest-neighbor* parameterization can be quite effective. For example,

$$Y_t(\mathbf{s}_i) = \sum_{j \in \mathcal{N}_i} m_{ij} Y_{t-1}(\mathbf{s}_j) + \eta_t(\mathbf{s}_i), \quad (7.17)$$

where \mathcal{N}_i corresponds to a prespecified neighborhood of location $s_i, i = 1, \dots, n$ and $m_{ij} = 0$ for all $s_j \notin \mathcal{N}_i$. Such a parameterization reduces the number of parameters from $O(n^2)$ to $O(n)$. It can be shown that such a parameterization can be motivated by many mechanistic models, such as those suggested by standard discretization of differential equations (e.g., finite difference, Galerkin, spectral) [e.g., see 7, 35]. In these cases, the m_{ij} parameters in (7.17) can be parameterized in terms of other mechanistically motivated knowledge, such as spatially varying diffusion or advection coefficients [e.g., 16, 17, 32, 37, 45]. Mechanistically motivated parameterizations can also be applied to nonlinear and multivariate processes [35].

3.4.2 Process Rank Reduction

Useful process reductions can be formulated with the realization that the essential dynamics for spatio-temporal processes typically exist on a relatively low-dimensional manifold [e.g., 41]. This is helpful because instead of having to model the evolution of the n -dimensional process $\{\mathbf{Y}_t\}$, one can model the evolution of a much lower-dimensional (n_α) process $\{\boldsymbol{\alpha}_t\}$, where $n_\alpha \ll n$. Thus, consider a decomposition of \mathbf{Y}_t [36] such that

$$\mathbf{Y}_t = \boldsymbol{\mu}_t + \boldsymbol{\Phi}\boldsymbol{\alpha}_t + \boldsymbol{\Psi}\boldsymbol{\xi}_t + \boldsymbol{\nu}_t, \quad (7.18)$$

where $\boldsymbol{\mu}_t$ is an n -dimensional time-varying (potentially) spatial mean corresponding to large-scale non-dynamic features and/or covariate effects; $\boldsymbol{\Phi}$ is an $n \times n_\alpha$ matrix of basis vectors corresponding to the latent dynamical expansion coefficient process, $\{\boldsymbol{\alpha}_t\}$; and $\boldsymbol{\Psi}$ can either be an $n \times n_\xi$ basis function matrix corresponding to the latent process, $\{\boldsymbol{\xi}_t\}$, which typically is assumed to have different dynamical characteristics than $\{\boldsymbol{\alpha}_t\}$ or this component might account for non-dynamical spatial variability. The error process $\{\boldsymbol{\nu}_t\}$ is typically Gaussian and assumed to be mean zero with simple dependence structure. Note that a continuous space representation of this decomposition can be expressed in terms of IDEs [e.g., see 7, Section 7.1.3].

The evolution of the latent $\boldsymbol{\alpha}_t$ process can proceed according to the basic linear or nonlinear models described above. Even in this low-dimensional context, parameter space reduction may still be necessary, particularly the case in nonlinear models (e.g., there are on the order of n_α^3 free parameters to estimate in the GQN model). Mechanistic knowledge can again be used to motivate such parameterizations in some cases [11, 36], and/or model selection approaches can be used to reduce the parameter space, such as stochastic search variable selection [e.g., 34].

There are many choices for the basis vectors that make up $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. It has become quite common in recent years to represent spatial processes in terms of basis decompositions, and there are many choices for these, such as orthogonal polynomials, empirical spectral decompositions (i.e., empirical orthogonal functions (EOFs)), stochastic optimals, balanced truncations, wavelets, splines, bisquare bases, Wendland bases, Moran's I bases, kernel convolution and “predictive process” bases, and dynamic factor bases [e.g., see the discussion in 7, 39]. Each has its proponents, although it does not seem to matter too much in the spatial context which basis is used, so long as it can accommodate the appropriate variability. In the

context of DSTMs, it can make more of a difference because it is important that interactions across spatial scales be allowed [e.g., 40]. This is more difficult to do in the standard “knot-based” representations (e.g., splines, kernel convolutions, predictive processes) in which case the α_t coefficients are spatially referenced, but not necessarily multi-resolution. Most of the other basis representations are in some sense multi-scale, and the associated expansion coefficients α_t are not indexed in space. However, the product of the basis times the expansion coefficients is spatially referenced, and more importantly, dynamical evolution in the DSTM can accommodate scale interactions. Note that the coefficients ξ_t associated with the matrix Ψ are typically specified to have much-simpler dynamical structure (if at all) as the assumption is that the controlling dynamics are associated with α_t . In general, the ξ_t coefficient portion of the expansion is used to accommodate extra-dynamical spatial variability and/or exogenous effects.

The projection of the process $\{\mathbf{Y}_t\}$ to the lower-dimensional manifold need not be linear as shown in (7.18). There are a bewildering number of choices for nonlinear dimension reduction, and some of them could potentially represent the dynamics more realistically (e.g., Laplacian eigenmaps [2], kernel principal components [e.g., 44], etc.). However, these methods are somewhat limited by a lack of uniqueness in the back projection of the expansion coefficients into physical space, which requires either some sort of ad hoc procedure or an additional modeling component in the HM.

In some cases, the process is so complicated that it might be very difficult to specify an adequate process model. If deterministic simulation models are available, it can sometimes be easier to incorporate the mechanistic information through a surrogate model or statistical emulator. That is, much like the design and analysis of computer modeling experiment literature [e.g., 12, 18, 28], one builds a statistical model for the fairly rich simulation output (in terms of spatio-temporal behavior) and uses that either as a black box [e.g., 15, 22] or to inform prior distributions for simpler mechanistically motivated DSTMs [e.g., 19]. In other cases, one can build simpler lower-dimensional emulators and link them together hierarchically to represent the dynamical process [e.g., 20]. It is important to note that emulators in the context of dynamical spatio-temporal processes typically are built from what [15] call the “first-order” perspective. That is, process evolution is accounted for explicitly in the conditional mean structure, following the etiology of the real-world process. This is unlike the design and analysis of computer modeling literature, which typically considers so-called “second-order” emulators, in which the focus is on the covariance structure. Such an approach is well suited for the model calibration problem.

4 Example: Near-Surface Winds Over the Ocean

To illustrate many of the HM concepts described above, consider the motivating example of prediction of complete near-surface wind fields from a blend of weather center analysis winds and satellite scatterometer winds (so-called *surface vector*

winds (SVW)). Some relevant background on the problem is presented, followed by a fairly simple BHM illustration applied to this problem.

4.1 Surface Vector Wind Background

Near-surface ocean winds are a critical component of the atmosphere/ocean interface as they are directly responsible for the transfer momentum to the ocean and the wind speed modulates the exchanges of heat and freshwater to and from the upper ocean. The advent of spaceborne scatterometer instruments in the 1990s provided the first global wind fields, on daily timescales, from observations. Prior to these scatterometer instruments, ocean winds were largely inferred from global weather forecast models (so-called *analyses*). These analyses depend on sparse global network of in situ wind observations from buoys and ships of opportunity and blend them with a mechanistic model of the atmosphere. The practical spatial resolution of such winds is limited to the relatively large spatial and temporal scales of variability.

Scatterometer SVW observations are not direct measures of the wind [see [23](#), for a more detailed description]. The winds are derived from complicated (“geophysical model function”) relationships concerning the roughness imparted on the ocean by surface capillary waves in response to the shear stress vector at the air-sea interface. Depending on the specific sensor, SVW estimates from scatterometers are accurate to within at least 2 ms^{-1} in speed and 30° in direction, and resolutions are on the order of 12.5–50 km for up to 90% global coverage on daily timescales. The SVW retrievals occur in swaths along the polar-orbiting satellite ground track, with varying swath widths depending on the instrument system. For the purposes of predicting complete spatial fields, it is important to note that because of the polar orbit (approximately 14 polar orbits per day), the swaths overlap at high latitudes and are separated by gaps in coverage at low latitudes. So, although there are gaps over a day, areas in which there are SVWs exhibit much finer spatial resolution of atmospheric wind features than the analysis wind products from the same period, which are complete in space but have much lower effective spatial feature resolution in general (i.e., an unrealistic kinetic energy spatial spectrum). The goal of a statistical data assimilation is then to blend the complete, but energy-deficient, weather center analyses with the incomplete, yet energy-realistic, SVW in order to provide spatially complete wind fields at sub-daily intervals while managing the uncertainties associated with the different data sources and the blending procedure.

Uncertainty management via BHM with process dynamics motivated by mechanistic models (i.e., leading order terms and/or approximations of the primitive equations) has been shown to be a very effective approach for this wind data assimilation problem [e.g., see the sequence of papers: [5](#), [13](#), [23](#), [24](#), [26](#), [36](#), [42](#)]. In particular, these methods have been shown to be quite useful in the context of providing inputs to ocean forecasting systems such as the Mediterranean Forecast System (MFS) [[23](#), [24](#)].

The MFS produces 10-day forecasts for upper ocean fields every day. This forecast model resolves medium-scale (in time and space) features (e.g., *synoptic scale*) in the upper ocean, and the most uncertain parts of the forecast fields correspond to so-called *mesoscales* (i.e., hourly and 10–50 km scales). These are the primary scales of the upper ocean hydrodynamic instabilities driven by the surface wind. Thus, modeling uncertainty in the surface wind field can be an important means of quantifying uncertainty in the MFS ocean forecasts on the scales that are most important to daily users.

4.2 Ocean SVW BHM

[23] describe the details of a SVW BHM for the MFS, and [24] discuss the impacts of the resulting BHM SVW fields in an ensemble forecast methodology built around realizations from the posterior distribution for SVW from the BHM. The process model in [23] involves the leading-order terms in a Rayleigh friction equation (RFE) approximation at synoptic scales, with extra-spatial variability added to account for turbulent scaling relationships in the wind field. A critical component of the [23] model is that it is multivariate in terms of modeling the east-west (u) and north-south (v) wind components and surface pressure (all of which are spatio-temporal processes) such that the wind components are independently conditioned on the pressure, which is a reasonable and justifiable assumption to first order. However, higher-order interactions of wind components are most likely important even after conditioning on the pressure field, so the model presented here considers a multivariate low-rank representation of the residual wind components after accounting for potential pressure gradient effects as suggested by the RFE. The data, process, and parameter models are described below.

4.2.1 Data Models

Two sources of wind data are considered, along with sea-level pressure data. In particular, there are satellite wind observations from the QuikSCAT scatterometer and surface winds and pressures from an analysis by the European Centre for Medium-Range Weather Forecasts (ECMWF). In this simple illustrative application, the pressure will be considered “known,” and only the wind components are modeled as a process, i.e., the pressure is used as an exogenous variable here. The wind data models are then:

$$\mathbf{d}_t^{Q_u} | \mathbf{u}_t, \sigma_Q^2 \sim \text{ind. Gau}(\mathbf{H}_t^Q \mathbf{u}_t, \sigma_Q^2 \mathbf{I}),$$

$$\mathbf{d}_t^{Q_v} | \mathbf{v}_t, \sigma_Q^2 \sim \text{ind. Gau}(\mathbf{H}_t^Q \mathbf{v}_t, \sigma_Q^2 \mathbf{I}),$$

$$\mathbf{d}_t^{E_u} | \mathbf{u}_t, \sigma_E^2 \sim \text{ind. Gau}(\mathbf{H}_t^E \mathbf{u}_t, \sigma_E^2 \mathbf{I}),$$

$$\mathbf{d}_t^{E_v} | \mathbf{v}_t, \sigma_E^2 \sim \text{ind. Gau}(\mathbf{H}_t^E \mathbf{v}_t, \sigma_E^2 \mathbf{I}),$$

where $\mathbf{d}_t^{Q_u}$ and $\mathbf{d}_t^{Q_v}$ are m_t -dimensional vectors of scatterometer u -wind and v -wind observations, respectively, within a specified time window indexed by t ; and $\mathbf{d}_t^{E_u}$ and $\mathbf{d}_t^{E_v}$ are ECMWF u -wind and v -wind component observations, respectively, within the same window. The spatially vectorized true wind process components are given by the n -dimensional vectors \mathbf{u}_t , \mathbf{v}_t . The mapping matrices for the scatterometer and ECMWF observations are given by \mathbf{H}_t^Q and \mathbf{H}_t^E , respectively. In this case, these are just incidence matrices that map the observations to the nearest process grid location [see 7, Chapter 7 for details]. The measurement errors are assumed to have Gaussian distributions that are independent in space and time, conditioned upon the true process values. The measurement-error variances, σ_Q^2 and σ_E^2 , correspond to scatterometer and ECMWF wind components, respectively. The conditional independence of these data models follows from the more general discussion above concerning the relative ease of incorporating multiple data sources in the BHM framework.

The wind data for February 2, 2005, are shown in Figs. 7.1 and 7.2. These plots show the QuikSCAT scatterometer and ECMWF analysis observations available within a window of ± 3 h of $t = 00:00, 06:00, 12:00$, and $18:00$ UTC (“Coordinated Universal Time”). The ECMWF analysis winds and pressures are specified on a $0.5^\circ \times 0.5^\circ$ spatial grid, and they are available at each time period for all locations. This grid is also used for the process vectors, \mathbf{u}_t and \mathbf{v}_t . As described above, the QuikSCAT observations are available intermittently in space due to the polar orbit of the satellite, but at much higher spatial resolution (25 km) when they are available. Thus, the mapping matrices for the scatterometer data, \mathbf{H}_t^Q , are defined as incidence matrices such that all scatterometer observations within 0.25° of a process grid point, and within 3 h of time t , are associated with the wind process at that grid point and time.

4.2.2 Process Model

The wind component process models are specified as

$$\mathbf{u}_t \equiv \mathbf{D}_x \mathbf{p}_t \theta_{ux} + \mathbf{D}_y \mathbf{p}_t \theta_{uy} + \boldsymbol{\Phi}_u \boldsymbol{\alpha}_t \quad (7.19)$$

$$\mathbf{v}_t \equiv \mathbf{D}_x \mathbf{p}_t \theta_{vx} + \mathbf{D}_y \mathbf{p}_t \theta_{vy} + \boldsymbol{\Phi}_v \boldsymbol{\alpha}_t, \quad (7.20)$$

where \mathbf{D}_x and \mathbf{D}_y are matrix operators that give the x -direction- and y -direction-centered differences of the spatial field vector on which they operate, respectively, and \mathbf{p}_t is the vectorized gridded ECMWF pressure data (assumed known here). In the context of the process rank reduction decomposition given in (7.18), $\boldsymbol{\Phi}_u$ and $\boldsymbol{\Phi}_v$ correspond to $n \times n_\alpha$ matrices of basis functions for the u - and v -wind components, respectively, with the common random reduced-rank latent process expansion coefficients, $\boldsymbol{\alpha}_t$. In addition, relative to (7.18), let $\boldsymbol{\mu}_{u,t} \equiv \mathbf{D}_x \mathbf{p}_t \theta_{ux} + \mathbf{D}_y \mathbf{p}_t \theta_{uy}$ and $\boldsymbol{\mu}_{v,t} \equiv \mathbf{D}_x \mathbf{p}_t \theta_{vx} + \mathbf{D}_y \mathbf{p}_t \theta_{vy}$, where these terms represent the importance of winds on the gradient of pressure, as controlled by the parameters $\boldsymbol{\theta} \equiv \{\theta_{ux}, \theta_{uy}, \theta_{vx}, \theta_{vy}\}$. This particular formulation does not include a separate small-scale spatial variability component (e.g., $\boldsymbol{\Psi} \boldsymbol{\beta}_t$ in (7.18)) for simplicity. [36] and [23] include such a term

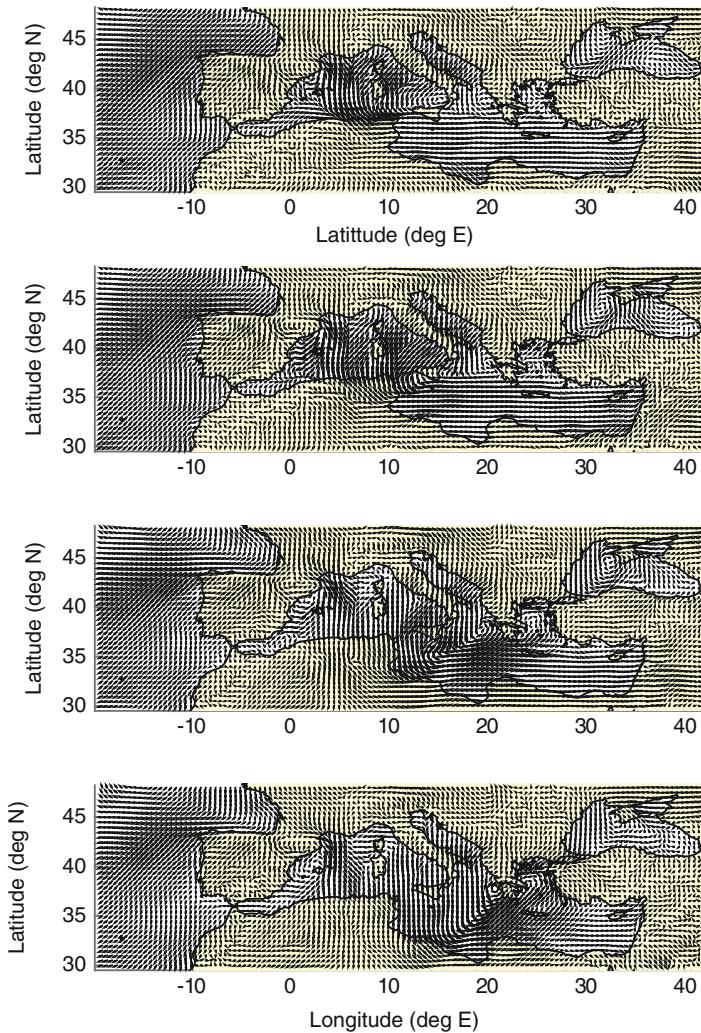


Fig. 7.1 Wind observations from February 2, 2005. From *top* to *bottom*, the panels correspond to the available data at 00:00, 06:00, 12:00, and 18:00 UTC (Universal Coordinated Time). The panels correspond to the ECMWF analysis winds on a $0.5^\circ \times 0.5^\circ$ grid. The length of the wind quiver (arrow) corresponds to speed, where the smallest is 0.06 m/s and the largest is 17.7 m/s.

and parameterize it in terms of two-dimensional spatial wavelet basis functions to account for the turbulent scaling relationships that are inherent in the SVW.

Note that the basis function matrices, Φ_u and Φ_v , are constructed from multivariate empirical orthogonal functions (EOFs) of the joint ECMWF u - and v -wind components [see 7, for an overview of EOF basis functions]. The advantage of such bases in this context is that they are constructed multivariately, so that the joint dependence of the wind components is considered in their construction. In

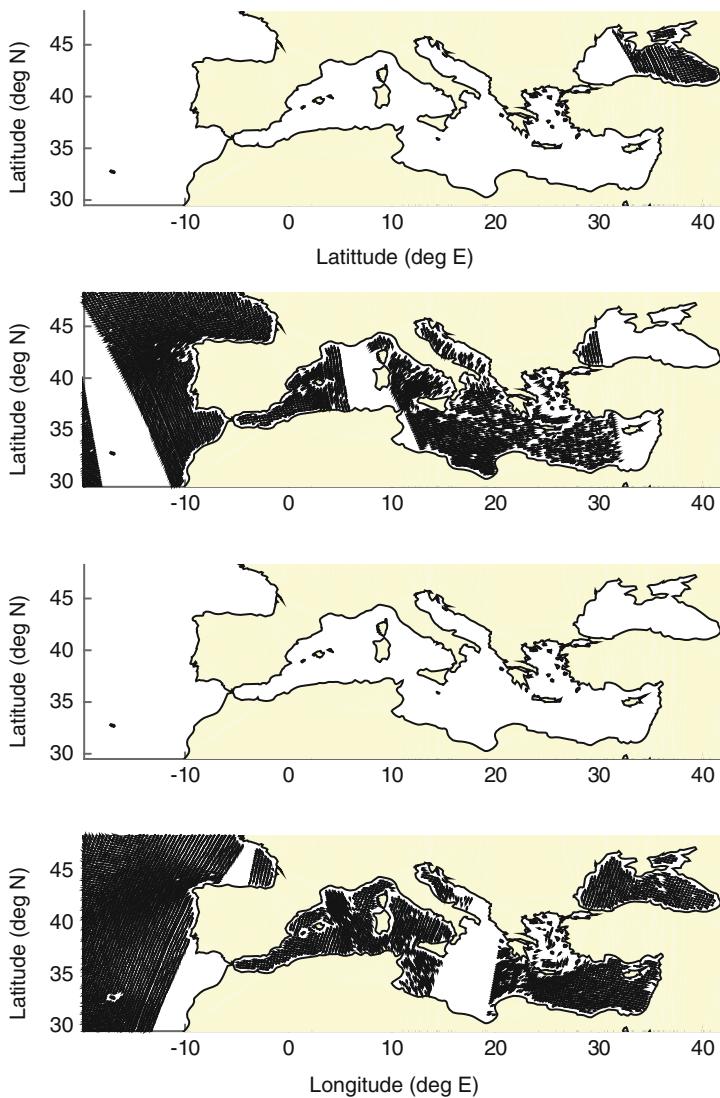


Fig. 7.2 Wind observations from February 2, 2005. From *top* to *bottom*, the panels correspond to the available data at 00:00, 06:00, 12:00, and 18:00 UTC (Universal Coordinated Time). The panels correspond to the high-resolution (25 km), but spatially intermittent, QuickSCAT scatterometer wind retrievals. The length of the wind quiver (arrow) corresponds to speed, where the smallest is 0.2 m/s and the largest is 21.3 m/s.

addition, EOFs generally are useful for dynamical reduced-rank modeling because the dimension reduction is quite significant (in the case here, $n = 4096$ and $n_\alpha = 32$, which accounts for approximately 98% of the variability in the ECMWF wind data). Although they can be quite useful for DSTM rank reduction, given

that EOFs are essentially spatial principal component loadings, they are optimal for variance reduction but not typically for dynamical propagation. Regarding the notions of inclusion of mechanistic information in BHMs, note that by conditioning the wind components on common processes \mathbf{p}_t and $\boldsymbol{\alpha}_t$, the process decomposition in (7.19) and (7.20) allows a reasonable mechanistic-based approach for building in the conditional independence between the wind components.

The dynamical evolution of the common latent process coefficients is specified fairly simply in this illustrative example as

$$\boldsymbol{\alpha}_t = \text{diag}(\mathbf{m}_\alpha) \boldsymbol{\alpha}_{t-1} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim Gau(\mathbf{0}, \mathbf{C}_\eta), \quad (7.21)$$

for $t = 1, \dots, T$, where $\text{diag}(\mathbf{m}_\alpha)$ corresponds to an n_α -dimensional diagonal matrix with \mathbf{m}_α on the main diagonal and zeros elsewhere. The initial condition is specified as $\boldsymbol{\alpha}_0 \sim Gau(\mathbf{0}, \mathbf{C}_0)$. Note that this fairly simple dynamical structure is motivated by the components of the RFE described in [23] that do not depend on pressure. Marginal dependence between the elements that make up $\boldsymbol{\alpha}_t$ is accommodated by a non-diagonal variance-covariance matrix, \mathbf{C}_η .

4.2.3 Parameter Models

To facilitate computation for this simple illustrative example, the parameters in the previous stages are given conjugate prior distributions. In particular, specify $\theta_k \sim N(\mu_i, \sigma_i^2)$ for $i = \{ux, uy, vx, vy\}$, $\mathbf{m}_\alpha \sim N(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha)$, and $\mathbf{C}_\eta^{-1} \sim W((d_\eta \mathbf{S}_\eta)^{-1}, d_\eta)$, where $W(\cdot)$ corresponds to a Wishart distribution. The remaining parameters and hyperparameters are fixed at scientifically plausible values (e.g., σ_Q^2 , σ_E^2 , μ_i , and $\boldsymbol{\mu}_\alpha$ as described in [23]) or given values to suggest vague (non-informative) priors (e.g., \mathbf{C}_0 , \mathbf{S}_η , d_η , σ_i^2 , \mathbf{C}_α).

4.3 Implementation

The posterior distribution for the random components of the model is given by

$$\begin{aligned} [\{\boldsymbol{\alpha}_t\}_{t=0}^T, \boldsymbol{\theta}, \mathbf{m}_\alpha, \mathbf{C}_\eta \mid \{\mathbf{d}_t^{Q_u}\}_{t=1}^T, \{\mathbf{d}_t^{Q_v}\}_{t=1}^T, \{\mathbf{d}_t^{E_u}\}_{t=1}^T, \{\mathbf{d}_t^{E_v}\}_{t=1}^T] &\propto \\ &\prod_{t=1}^T [\mathbf{d}_t^{Q_u} \mid \boldsymbol{\alpha}_t, \boldsymbol{\theta}] \prod_{t=1}^T [\mathbf{d}_t^{Q_v} \mid \boldsymbol{\alpha}_t, \boldsymbol{\theta}] \\ &\times \prod_{t=1}^T [\mathbf{d}_t^{E_u} \mid \boldsymbol{\alpha}_t, \boldsymbol{\theta}] \prod_{t=1}^T [\mathbf{d}_t^{E_v} \mid \boldsymbol{\alpha}_t, \boldsymbol{\theta}] \\ &\times \prod_{t=1}^T [\boldsymbol{\alpha}_t \mid \boldsymbol{\alpha}_{t-1}, \mathbf{m}_\alpha, \mathbf{C}_\eta][\boldsymbol{\alpha}_0][\mathbf{m}_\alpha][\mathbf{C}_\eta][\boldsymbol{\theta}]. \end{aligned}$$

Although an analytical posterior distribution is not available in this case, given the conjugate distributional forms, the required full-conditional distributions for a Gibbs sampler can all be derived analytically [e.g., see 7, for the details of a similar DSTM example]. In the example given here, the spatial grid sizes are $n = 4096$ and $T = 57$ times and were considered corresponding to the period from 12:00 UTC January 25, 2005, through 12:00 UTC February 8, 2005, at 6-h intervals. The reduced-rank vectors were of dimension $n_\alpha = 32$. The MCMC was run for 100,000 iterations after a 20,000-iteration burn-in period. The algorithm is quite efficient given the number of prediction locations 4096×57 and large amount of data (e.g., the MCMC was run in less than 4 h on a standard 2014 vintage laptop computer).

4.4 Results

The posterior mean wind fields for 12:00 UTC on February 2, 2005, are shown in Fig. 7.3. In addition, Fig. 7.4 shows a portion of the prediction domain, in which ten samples widely separated in the MCMC chain are plotted. One can see from this plot that the uncertainty associated with the spatial prediction of the wind fields is not homogeneous in the domain. For example, the strong flow off the south coast of France into the Gulf of Lion (so-called mistral winds) shows more variability in wind direction than areas over land. Note that this area of increased posterior variability is over a fairly small spatial region, which is important when one is using winds to force an ocean model such as with the MFS. That is, the small-scale variations in the wind forcing can lead to similar-scale uncertainties in the ocean state variables, which can make a substantial difference in ocean forecasts [see 24, for an in-depth discussion].

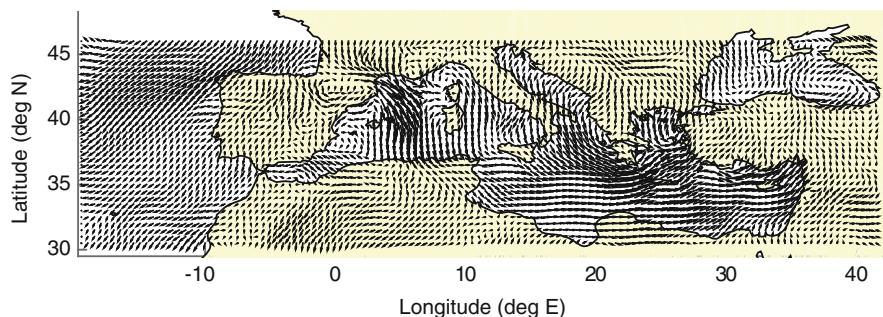


Fig. 7.3 Posterior mean wind vectors for 12:00 UTC on February 2, 2005, on the prediction grid. Wind speed is proportional to the length of the vectors, with the direction of wind toward the “arrowhead” on the wind quiver. The length of the wind quiver corresponds to speed, where the smallest is 0.0 m/s and the largest is 18.0 m/s.

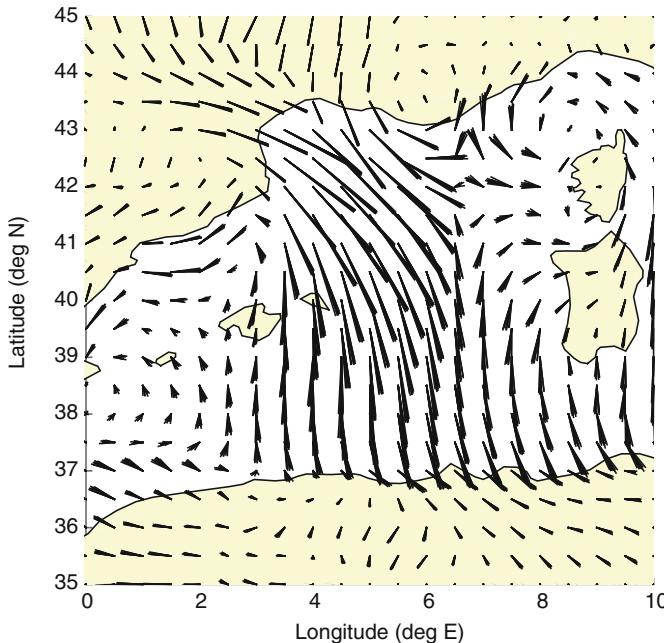


Fig. 7.4 Ten samples of wind vectors taken from the posterior distribution for 12:00 UTC on February 2, 2005, over the western Mediterranean basins. Wind speed is proportional to the length of the vectors. The direction of the vector is away from the vertex at the center of its grid cell. In this case, the “arrowhead” on the wind quivers is suppressed so that the variability in the wind corresponds to the width of the downwind portion of the vector.

5 Conclusion

This chapter has presented a brief summary of hierarchical modeling in the context of complex processes, typically those with mechanistically motivated spatio-temporal dependence. When modeling the complex processes one sees in many science and engineering applications, hierarchical modeling is a coherent approach to accommodate uncertainty in the observations (measurement error and sampling error), in the process specification, and in the knowledge of the parameters and potential additional forcings. The approach is very flexible, but with that flexibility comes potentially significant challenges and compromises when it comes to implementation.

Consider what [7] call the “data/model compromise.” Even for complicated spatio-temporal processes, if one has *enough* (whatever that may be) high-quality observations, then the model can be fairly simple since the complex dependence structure is already contained in the observations and, presumably, can be “learned” by the statistical model. In many respects, this was the case with the SVW example

presented above. There are a lot of observations from two different observation sources in this example, so the wind component process model is actually fairly simple relative to a mechanistic model that would be used with little or no data. In other SVW implementations, more sophisticated process models may have to be used depending on the data coverage and complexity of the dynamic environment [e.g., 36]. On the other hand, if one specifies a very complex mechanistic process model, but has very little data, there may not be enough information in the data to inform the posterior distributions associated with the parameters and process [e.g., 9]. That is, when the data are not rich enough to learn about the process and parameters, then one effectively has a practical lack of identifiability that may inhibit fitting the BHM. In practice, one tries to strike a balance between these two competing data/model trade-offs.

Perhaps the greatest challenge with implementing complex BHMs is recognizing the need to trade the complexity of the model for computational simplicity or what [7] call the “computing/model compromise.” Despite the ever-advancing state of statistical computation for HMs, the algorithms can still be difficult to implement, both in terms of time required to code and the effort required to tune the algorithm. Software packages to implement BHMs are increasing in number and quality, but it is still often difficult to implement very complex BHMs with these packages. Thus, one is often faced with the dilemma of either simplifying the model and sacrificing some realism or utilizing an approximate estimation/inference approach (e.g., ABC, INLA, variational Bayes, etc.) and either limiting the sorts of inference that can be accomplished or accepting some inaccuracy relative to the true posterior distribution of interest. Thus, when implementing a complex BHM, one must always consider the difference between what one *wants* to do and what one *can* do and whether it is best for the particular problem at hand to sacrifice model complexity or computational efficiency. Regardless, the BHM paradigm still remains one of the most powerful frameworks in which to quantify uncertainty.

This chapter is concerned with science-based hierarchical modeling, in which one has mechanistic information available to inform the model components (either data, model, or parameters). In recent years, alternative hierarchical modeling approaches have been developed from the statistical learning perspective [e.g., see the review outlined in 3], which typically do not make use of scientific/mechanistic information, but seek to build multilayer models (e.g., “deep learning”) through nonparametric approaches. These approaches can sometimes be useful in situations where subject-matter knowledge is not readily available yet can overfit in situations with complex spatial and temporal dependencies. In both the science-based and statistical learning-based HM approaches, much more work remains to be done on the theoretical properties of the estimators and predictors under various amounts of uncertainty in observations, process models and parameter structure, as well as data volume. Promising approaches are being developed [e.g., 1], but to date, these approaches have not been able to speak to the multilevel-dependent parameter structures common in the science-based BHM setting.

References

1. Agapiou, S., Stuart, A., Zhang, Y.X.: Bayesian posterior contraction rates for linear severely ill-posed inverse problems. *J. Inverse Ill-Posed Probl.* **22**, 297–321 (2014). doi:10.1515/jip-2012-0071
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neur. Comput.* **15**(6), 1373–1396 (2003)
3. Bengio, S., Deng, L., Larochelle, H., Lee, H., Salakhutdinov, R.: Guest editors’ introduction: special section on learning deep architectures. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1795–1797 (2013). doi:10.1109/TPAMI.2013.118
4. Berliner, L.: Hierarchical Bayesian time-series models. *Fund. Theor. Phys.* **79**, 15–22 (1996)
5. Berliner, L.M., Milliff, R.F., Wikle, C.K.: Bayesian hierarchical modeling of air-sea interaction. *J. Geophys. Res. Oceans* (1978–2012) **108**(C4), 2156–2202 (2003)
6. Brooks, S., Gelman, A., Jones, G., Meng, X.L.: *Handbook of Markov Chain Monte Carlo*. CRC Press, Boca Raton (2011)
7. Cressie, N., Wikle, C.: *Statistics for Spatio-Temporal Data*, vol. 465. Wiley, Hoboken (2011)
8. Cressie, N., Calder, C., Clark, J., Hoef, J., Wikle, C.: Accounting for uncertainty in ecological analysis: the strengths and limitations of hierarchical statistical modeling. *Ecol. Appl.* **19**(3), 553–570 (2009)
9. Fiechter, J., Herbei, R., Leeds, W., Brown, J., Milliff, R., Wikle, C., Moore, A., Powell, T.: A Bayesian parameter estimation method applied to a marine ecosystem model for the coastal gulf of Alaska. *Ecol. Model.* **258**, 122–133 (2013)
10. Gelfand, A.E., Smith, A.F.: Sampling-based approaches to calculating marginal densities. *J. Am. Stat. Assoc.* **85**(410), 398–409 (1990)
11. Gladish, D., Wikle, C.: Physically motivated scale interaction parameterization in reduced rank quadratic nonlinear dynamic spatio-temporal models. *Environmetrics* **25**(4), 230–244 (2014)
12. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
13. Hoar, T.J., Milliff, R.F., Nychka, D., Wikle, C.K., Berliner, L.M.: Winds from a Bayesian hierarchical model: computation for atmosphere-ocean research. *J. Comput. Graph. Stat.* **12**(4), 781–807 (2003)
14. Holton, J.: *Dynamic Meteorology*. Elsevier, Burlington (2004)
15. Hooten, M., Leeds, W., Fiechter, J., Wikle, C.: Assessing first-order emulator inference for physical parameters in nonlinear mechanistic models. *J. Agric. Biolog. Environ. Stat.* **16**(4), 475–494 (2011)
16. Hooten, M.B., Wikle, C.K.: A hierarchical Bayesian non-linear spatio-temporal model for the spread of invasive species with application to the eurasian collared-dove. *Environ. Ecol. Stat.* **15**(1), 59–70 (2008)
17. Hooten, M.B., Wikle, C.K., Dorazio, R.M., Royle, J.A.: Hierarchical spatiotemporal matrix models for characterizing invasions. *Biometrics* **63**(2), 558–567 (2007)
18. Kennedy, M.C., O’Hagan, A.: Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **63**(3), 425–464 (2001)
19. Leeds, W., Wikle, C., Fiechter, J.: Emulator-assisted reduced-rank ecological data assimilation for nonlinear multivariate dynamical spatio-temporal processes. *Stat. Methodol.* (2013). doi:10.1016/j.stamet.2012.11.004
20. Leeds, W., Wikle, C., Fiechter, J., Brown, J., Milliff, R.: Modeling 3-D spatio-temporal biogeochemical processes with a forest of 1-D statistical emulators. *Environmetrics* **24**, 1–12 (2013). doi:10.1002/env.2187
21. Marin, J.M., Pudlo, P., Robert, C.P., Ryder, R.J.: Approximate Bayesian computational methods. *Stat. Comput.* **22**(6), 1167–1180 (2012)

22. van der Merwe, R., Leen, T., Lu, Z., Frolov, S., Baptista, A.: Fast neural network surrogates for very high dimensional physics-based models in computational oceanography. *Neur. Netw.* **20**(4), 462–478 (2007)
23. Milliff, R., Bonazzi, A., Wikle, C., Pinardi, N., Berliner, L.: Ocean ensemble forecasting. Part I: ensemble mediterranean winds from a Bayesian hierarchical model. *Q. J. R. Meteorol. Soc.* **137**(657), 858–878 (2011)
24. Pinardi, N., Bonazzi, A., Dobricic, S., Milliff, R., Wikle, C., Berliner, L.: Ocean ensemble forecasting. Part II: mediterranean forecast system response. *Q. J. R. Meteorol. Soc.* **137**(657), 879–893 (2011)
25. Robert, C., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer New York (2004)
26. Royle, J., Berliner, L., Wikle, C., Milliff, R.: A Hierarchical Spatial Model for Constructing Wind Fields from Scatterometer Data in the Labrador Sea. *Lecture Notes in Statistics*, pp. 367–382. Springer, New York (1999)
27. Rue, H., Martino, S., Chopin, N.: Approximate Bayesian inference for latent gaussian models by using integrated nested laplace approximations. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **71**(2), 319–392 (2009)
28. Sacks, J., Welch, W., Mitchell, T., Wynn, H.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
29. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and its Applications: With R Examples. Springer, New York (2010)
30. Šmíd, V., Quinn, A.: The Variational Bayes Method in Signal Processing. Springer, Berlin/New York (2006)
31. Verbeke, G., Molenberghs, G.: Linear Mixed Models for Longitudinal Data. Springer, New York (2009)
32. Wikle, C.: Hierarchical Bayesian models for predicting the spread of ecological processes. *Ecol.* **84**(6), 1382–1394 (2003)
33. Wikle, C., Berliner, L.: A Bayesian tutorial for data assimilation. *Phys. D: Nonlinear Phenom.* **230**(1), 1–16 (2007)
34. Wikle, C., Holan, S.: Polynomial nonlinear spatio-temporal integro-difference equation models. *J. Time Ser. Anal.* **32**(4), 339–350 (2011)
35. Wikle, C., Hooten, M.: A general science-based framework for dynamical spatio-temporal models. *Test* **19**(3), 417–451 (2010)
36. Wikle, C., Milliff, R., Nychka, D., Berliner, L.: Spatiotemporal hierarchical Bayesian modeling tropical ocean surface winds. *J. Am. Stat. Assoc.* **96**(454), 382–397 (2001)
37. Wikle, C.K.: A kernel-based spectral model for non-gaussian spatio-temporal processes. *Stat. Model.* **2**(4), 299–314 (2002)
38. Wikle, C.K.: Hierarchical models in environmental science. *Int. Stat. Rev.* **71**(2), 181–199 (2003)
39. Wikle, C.K.: Low-Rank Representations for Spatial Processes. *Handbook of Spatial Statistics*, pp. 107–118. CRC, Boca Raton (2010)
40. Wikle, C.K.: Modern perspectives on statistics for spatio-temporal data. *Wiley Interdiscip. Rev. Comput. Stat.* **7**(1), 86–98 (2015)
41. Wikle, C.K., Cressie, N.: A dimension-reduced approach to space-time Kalman filtering. *Biometrika* **86**(4), 815–829 (1999)
42. Wikle, C.K., Berliner, L.M., Milliff, R.F.: Hierarchical Bayesian approach to boundary value problems with stochastic boundary conditions. *Mon. Weather Rev.* **131**(6), 1051–1062 (2003)
43. Wikle, C.K., Milliff, R.F., Herbei, R., Leeds, W.B.: Modern statistical methods in oceanography: a hierarchical perspective. *Stat. Sci.* **28**(4), 466–486 (2013). doi:10.1214/13-STS436, <http://dx.doi.org/10.1214/13-STS436>
44. Wu, G., Holan, S.H., Wikle, C.K.: Hierarchical Bayesian spatio-temporal Conway–Maxwell poisson models with dynamic dispersion. *Jo. Agri. Biol. Environ. Stat.* **18**(3), 335–356 (2013)
45. Xu, K., Wikle, C.K., Fox, N.I.: A kernel-based spatio-temporal dynamical model for nowcasting weather radar reflectivities. *J. Am. Stat. Assoc.* **100**(472), 1133–1144 (2005)

Random Matrix Models and Nonparametric Method for Uncertainty Quantification

8

Christian Soize

Abstract

This chapter deals with the fundamental mathematical tools and the associated computational aspects for constructing the stochastic models of random matrices that appear in the nonparametric method of uncertainties and in the random constitutive equations for multiscale stochastic modeling of heterogeneous materials. The explicit construction of ensembles of random matrices but also the presentation of numerical tools for constructing general ensembles of random matrices are presented and can be used for high stochastic dimension. The developments presented are illustrated for the nonparametric method for multiscale stochastic modeling of heterogeneous linear elastic materials and for the nonparametric stochastic models of uncertainties in computational structural dynamics.

Keywords

Random matrix • Symmetric random matrix • Positive-definite random matrix • Nonparametric uncertainty • Nonparametric method for uncertainty quantification • Random vector • Maximum entropy principle • Non-Gaussian • Generator • Random elastic medium • Uncertainty quantification in linear structural dynamics • Uncertainty quantification in nonlinear structural dynamics • Parametric-nonparametric uncertainties • Identification • Inverse problem • Statistical inverse problem

Contents

1	Introduction	221
2	Notions on Random Matrices and on the Nonparametric Method for Uncertainty Quantification	222
2.1	What Is a Random Matrix?	222
2.2	What Is the Nonparametric Method for Uncertainty Quantification?	223

C. Soize (✉)

Laboratoire Modélisation et Simulation Multi Echelle (MSME), Université Paris-Est, Marne-la-Vallée, France

e-mail: christian.soize@univ-paris-est.fr

3	A Brief History	224
3.1	Random Matrix Theory (RMT)	224
3.2	Nonparametric Method for UQ and Its Connection with the RMT	225
4	Overview	226
5	Notations	227
5.1	Euclidean and Hermitian Spaces	227
5.2	Sets of Matrices	227
5.3	Kronecker Symbol, Unit Matrix, and Indicator Function	227
5.4	Norms and Usual Operators	227
5.5	Order Relation in the Set of All the Positive-Definite Real Matrices	228
5.6	Probability Space, Mathematical Expectation, and Space of Second-Order Random Vectors	228
6	The MaxEnt for Constructing Random Matrices	228
6.1	Volume Element and Probability Density Function (PDF)	228
6.2	The Shannon Entropy as a Measure of Uncertainties	230
6.3	The MaxEnt Principle	230
7	A Fundamental Ensemble for the Symmetric Real Random Matrices with a Unit Mean Value	231
7.1	Classical Definition [74]	231
7.2	Definition by the MaxEnt and Calculation of the pdf	232
7.3	Decentering and Interpretation of Hyperparameter δ	232
7.4	Generator of Realizations	233
7.5	Use of the GOE Ensemble in Uncertainty Quantification	233
8	Fundamental Ensembles for Positive-Definite Symmetric Real Random Matrices	233
8.1	Ensemble SG_0^+ of Positive-Definite Random Matrices With a Unit Mean Value	234
8.2	Ensemble SG_ε^+ of Positive-Definite Random Matrices with a Unit Mean Value and an Arbitrary Positive-Definite Lower Bound	237
8.3	Ensemble SG_b^+ of Positive-Definite Random Matrices with Given Lower and Upper Bounds and with or without Given Mean Value	238
8.4	Ensemble SG_λ^+ of Positive-Definite Random Matrices with a Unit Mean Value and Imposed Second-Order Moments	241
9	Ensembles of Random Matrices for the Nonparametric Method in Uncertainty Quantification	242
9.1	Ensemble SE_0^+ of Positive-Definite Random Matrices with a Given Mean Value	243
9.2	Ensemble SE_ε^+ of Positive-Definite Random Matrices with a Given Mean Value and an Arbitrary Positive-Definite Lower Bound	244
9.3	Ensemble SE^{+0} of Semipositive-Definite Random Matrices with a Given Semipositive-Definite Mean Value	246
9.4	Ensemble SE^{rect} of Rectangular Random Matrices with a Given Mean Value	247
9.5	Ensemble SE^{HT} of a Pair of Positive-Definite Matrix-Valued Random Functions Related by a Hilbert Transform	248
10	MaxEnt as a Numerical Tool for Constructing Ensembles of Random Matrices	250
10.1	Available Information and Parameterization	251
10.2	Construction of the pdf of Random Vector Y Using the MaxEnt	252
11	MaxEnt for Constructing the pdf of a Random Vector	252
11.1	Existence and Uniqueness of a Solution to the MaxEnt	253
11.2	Numerical Calculation of the Lagrange Multipliers	256
11.3	Generator for Random Vector Y_λ and Estimation of the Mathematical Expectations in High Dimension	257
12	Nonparametric Stochastic Model For Constitutive Equation in Linear Elasticity	261
12.1	Positive-Definite Matrices Having a Symmetry Class	262

12.2	Representation Introducing a Positive-Definite Lower Bound	263
12.3	Introducing Deterministic Matrices $[A]$ and $[S]$	263
12.4	Nonparametric Stochastic Model for $[C]$	264
12.5	Construction of $[A]$ Using the MaxEnt Principle	265
13	Nonparametric Stochastic Model of Uncertainties in Computational Linear Structural Dynamics	267
13.1	Methodology	267
13.2	Mean Computational Model in Linear Structural Dynamics	268
13.3	Reduced-Order Model (ROM) of the Mean Computational Model	268
13.4	Nonparametric Stochastic Model of Both the Model-Parameter Uncertainties and the Model Uncertainties (Modeling Errors)	270
13.5	Case of Linear Viscoelastic Structures	271
13.6	Estimation of the Hyperparameters of the Nonparametric Stochastic Model of Uncertainties	273
14	Parametric-Nonparametric Uncertainties in Computational Nonlinear Structural Dynamics	274
14.1	Mean Nonlinear Computational Model in Structural Dynamics	274
14.2	Reduced-Order Model (ROM) of the Mean Nonlinear Computational Model	275
14.3	Parametric-Nonparametric Stochastic Modeling of Uncertainties	276
14.4	Estimation of the Hyperparameters of the Parametric-Nonparametric Stochastic Model of Uncertainties	278
15	Key Research Findings and Applications	279
15.1	Propagation of Uncertainties Using Nonparametric or Parametric-Nonparametric Stochastic Models of Uncertainties	279
15.2	Experimental Validations of the Nonparametric Method of Uncertainties	280
15.3	Additional Ingredients for the Nonparametric Stochastic Modeling of Uncertainties	280
15.4	Applications of the Nonparametric Stochastic Modeling of Uncertainties in Different Fields of Computational Sciences and Engineering	281
16	Conclusions	281
	References	282

1 Introduction

It is well known that the parametric method for uncertainty quantification consists in constructing stochastic models of the uncertain physical parameters of a computational model that results from the discretization of a boundary value problem. The parametric method is efficient for taking into account the variabilities of physical parameters, but has not the capability to take into account the model uncertainties induced by modeling errors that are introduced during the construction of the computational model. The nonparametric method for the uncertainty quantification is a way for constructing a stochastic model of the model uncertainties induced by the modeling errors. It is also an approach for constructing stochastic models of constitutive equations of materials involving some non-Gaussian tensor-valued random fields, such as in the framework of elasticity, thermoelasticity, electromagnetism, etc. The random matrix theory is a fundamental tool that is really efficient for performing stochastic modeling of matrices that appear in the nonparametric method of uncertainties and in the random constitutive equations for multiscale stochastic

modeling of heterogeneous materials. The applications of the nonparametric stochastic modeling of uncertainties and of the random matrix theory presented in this chapter have been developed and validated for many fields of computational sciences and engineering, in particular for dynamical systems encountered in aeronautics and aerospace engineering [7, 20, 78, 88, 91, 94], in biomechanics [30, 31], in environment [32], in nuclear engineering [9, 12, 13, 29], in soil-structure interaction and for the wave propagations in soils [4, 5, 26, 27], in rotor dynamics [79, 80, 82] and vibration of turbomachines [18, 19, 22, 70], in vibroacoustics of automotive vehicles [3, 38–40, 61], but also, in continuum mechanics for multiscale stochastic modeling of heterogeneous materials [48, 49, 51–53], for the heat transfer in complex composites and for their nonlinear thermomechanic analyses [97, 98].

The chapter is organized as follows:

- Notions on random matrices and on the nonparametric method for uncertainty quantification: What is a random matrix and what is the nonparametric method for uncertainty quantification?
- Brief history concerning the random matrix theory and the nonparametric method for UQ and its connection with the random matrix theory.
- Overview and mathematical notations used in the chapter.
- Maximum entropy principle (MaxEnt) for constructing random matrices.
- Fundamental ensemble for the symmetric real random matrices with a unit mean value.
- Fundamental ensembles for positive-definite symmetric real random matrices.
- Ensembles of random matrices for the nonparametric method in uncertainty quantification.
- The MaxEnt as a numerical tool for constructing ensembles of random matrices.
- The MaxEnt for constructing the pdf of a random vector.
- Nonparametric stochastic model for constitutive equation in linear elasticity.
- Nonparametric stochastic model of uncertainties in computational linear structural dynamics.
- Parametric-nonparametric uncertainties in computational nonlinear structural dynamics.
- Some key research findings and applications.

2 Notions on Random Matrices and on the Nonparametric Method for Uncertainty Quantification

2.1 What Is a Random Matrix?

A real (or complex) *matrix* is a rectangular or a square array of real (or complex) numbers, arranged in rows and columns. The individual items in a matrix are called its elements or its entries.

A real (or complex) *random matrix* is a matrix-valued random variable, which means that its entries are real (or complex) random variables. The *random matrix*

theory is related to the fundamental mathematical methods required for constructing the probability distribution of such a random matrix, for constructing a generator of independent realizations, for analyzing some algebraic properties and some spectral properties, etc.

Let us give an example for illustrating the types of problems related to the random matrix theory. Let us consider a random matrix $[A]$, defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in a set \mathbb{S}_n of matrices, which is a subset of the set $\mathbb{M}_n^S(\mathbb{R})$ of all the symmetric $(n \times n)$ real matrices. Thus, for θ in Θ , the realization $[A](\theta)$ is a deterministic matrix in $\mathbb{S}_n \subset \mathbb{M}_n^S(\mathbb{R})$. Fundamental questions are related to the definition and to the construction of the probability distribution $P_{[A]}$ of such a random matrix $[A]$. If this probability distribution is defined by a probability density function (pdf) with respect a volume element $d^S A$, which is a mapping $[A] \mapsto p_{[A]}([A])$ from $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{R}^+ = [0, +\infty[$, for which its support is \mathbb{S}_n (which implies that $p_{[A]}([A]) = 0$ if $[A] \notin \mathbb{S}_n$), then how must the volume element $d^S A$ be defined, how is the integration over $\mathbb{M}_n^S(\mathbb{R})$ defined, and what are the methods and tools for constructing pdf $p_{[A]}$ and its generator of independent realizations? For instance, such a pdf cannot simply be defined in giving the pdf of every entry $[A]_{jk}$ for many reasons among the following ones. As random matrix $[A]$ is symmetric, all the entries are not algebraically independent, and therefore, only the $n(n + 1)/2$ random variables $\{[A]_{1 \leq j \leq k \leq n}\}$ must be considered. In addition, if \mathbb{S}_n is the subset $\mathbb{M}_n^+(\mathbb{R})$ of all the positive-definite symmetric $(n \times n)$ real matrices, then there is an algebraic constraint that relates the random variables $\{[A]_{1 \leq j \leq k \leq n}\}$ in order that $[A]$ be with values in $\mathbb{M}_n^+(\mathbb{R})$, and such an algebraic constraint implies that all the random variables $\{[A]_{1 \leq j \leq k \leq n}\}$ are statistically dependent.

2.2 What Is the Nonparametric Method for Uncertainty Quantification?

The *parametric method* for uncertainty quantification consists in constructing stochastic models of the uncertain physical parameters (geometry, boundary conditions, material properties, etc) of a computational model that results from the discretization of a boundary value problem. The parametric method, which introduces prior and posterior stochastic models of the uncertain physical parameters of the computational model, has not the capability to take into account *model uncertainties* induced by *modeling errors* that are introduced during the construction of the computational model.

The *nonparametric method* for uncertainty quantification consists in constructing a stochastic model of both the uncertain physical parameters and the model uncertainties induced by the modeling errors, without separating the effects of the two types of uncertainties. Such an approach consists in directly constructing stochastic models of matrices representing operators of the problem considered and not in using the parametric method for the uncertain physical parameters whose matrices depend. Initially developed for uncertainty quantification in computational structural dynamics, the use of the nonparametric method has been extended for

constructing stochastic models of matrices of computational models, such as the nonparametric stochastic model for constitutive equation in linear elasticity.

The *parametric-nonparametric method* for uncertainty quantification consists in using simultaneously in a computational model, the parametric method for constructing stochastic models of certain of its uncertain physical parameters, and the nonparametric method for constructing a stochastic model of both, the other uncertain physical parameters and the model uncertainties induced by the modeling errors, in separating the effects of the two types of uncertainties.

Consequently, the nonparametric method for uncertainty quantification uses the random matrix theory.

3 A Brief History

3.1 Random Matrix Theory (RMT)

The *random matrix theory* (RMT) were introduced and developed in mathematical statistics by Wishart and others in the 1930s and was intensively studied by physicists and mathematicians in the context of nuclear physics. These works began with Wigner [125] in the 1950s and received an important effort in the 1960s by Dyson, Mehta, Wigner [36, 37, 126], and others. In 1965, Porter [92] published a volume of important papers in this field, followed, in 1967, by the first edition of the Mehta book [72] whose second edition [73] published in 1991 gives a synthesis of the random matrix theory. For applications in physics, an important ensemble of the random matrix theory is the Gaussian orthogonal ensemble (GOE) for which the elements are constituted of real symmetric random matrices with statistically independent entries and which are invariant under orthogonal linear transformations (this ensemble can be viewed as a generalization of a Gaussian real-valued random variable to a symmetric real square random matrix).

For an introduction to multivariate statistical analysis, we refer the reader to [2], for an overview on explicit probability distributions of ensembles of random matrices and their properties, to [55] and, for analytical mathematical methods devoted to the random matrix theory, to [74].

RMT has been used in other domains than nuclear physics. In 1984 and 1986, Bohigas et al. [14, 15] found that the level fluctuations of the quantum Sinai's billiard were able to predict with the GOE of random matrices. In 1989, Weaver [124] showed that the higher frequencies of an elastodynamic structure constituted of a small aluminum block had the behavior of the eigenvalues of a matrix belonging to the GOE. Then, Bohigas, Legrand, Schmidt, and Sornette [16, 65, 66, 99] studied the high-frequency spectral statistics with the GOE for elastodynamics and vibration problems in the high-frequency range. Langley [64] showed that, in the high-frequency range, the system of natural frequencies of linear uncertain dynamic systems is a non-Poisson point process. These results have been validated for the high-frequency range in elastodynamics. A synthesis of theses aspects related to

quantum chaos and random matrix theory, devoted to linear acoustics and vibration, can be found in the book edited by Wright and Weaver [127].

3.2 Nonparametric Method for UQ and Its Connection with the RMT

The nonparametric method was initially be introduced by Soize [106, 107] in 1999–2000 for uncertainty quantification in computational linear structural dynamics in order to take into account the model uncertainties induced by the modeling errors that could not be addressed by the parametric method. The concept of the nonparametric method then consisted in modeling the generalized matrices of the reduced-order model of the computational model by random matrices. It should be noted that the terminology “nonparametric” is not at all connected to the “nonparametric statistics” but was introduced to show the differences between the well-known parametric method consisting in constructing a stochastic model of uncertain physical parameters of the computational model, and the new proposed nonparametric method that consisted in modeling the generalized matrices of the reduced-order model by random matrices, related to the operators of the problem. Later, the parametric-nonparametric method has been introduced [113].

Early in the development of the concept of the nonparametric method, a problem has occurred in the choice of ensembles of random matrices. Indeed the ensembles of random matrices coming from the RMT were not adapted to stochastic modeling required by the nonparametric method. For instance, the GOE of random matrices could not be used for the generalized mass matrix, which must be positive definite, what is not the case for a random matrix belonging to GOE. Consequently, new ensembles of random matrices have had to be developed [76, 107, 108, 110, 115], using the maximum entropy (MaxEnt) principle, for implementing the concept of the nonparametric method for various computational models in mechanics, for which the matrices must verify various algebraic properties. In addition, parameterizations of the new ensembles of random matrices have been introduced in the different constructions in order to be in capability to quantify simply the level of uncertainties. These ensembles of random matrices have been constructed with a parameterization exhibiting a small number of hyperparameters, what allows for identifying the hyperparameters in using experimental data, solving a statistical inverse problems for random matrices that are, in general, in very high dimension. In these constructions, for certain types of available information, an explicit solution of the MaxEnt principle has been obtained, giving an explicit description of the ensembles of random matrices and of the corresponding generators of realizations. Nevertheless, for other cases of available information coming from computational models, there is no explicit solution of the MaxEnt, and therefore, a numerical tool adapted to the high dimension has had to be developed [112].

Finally, during these last 15 years the nonparametric method has extensively been used and extended, with experimental validations, to many problems in linear and nonlinear structural dynamics, in fluid-structure interaction and in vibroacoustics,

in unsteady aeroelasticity, in soil-structure interaction, in continuum mechanics of solids for the nonparametric stochastic modeling of the constitutive equations in linear and nonlinear elasticity, in thermoelasticity, etc. A brief overview on all the experimental validations and applications in different fields is given in the last Sect. 15.

4 Overview

This chapter is constituted of two main parts:

- The first one is devoted to the presentation of ensembles of random matrices that are explicitly described and also deals with an efficient numerical tool for constructing ensembles of random matrices when an explicit construction cannot be obtained. The presentation is focused to the fundamental results and to the fundamental tools related to ensembles of random matrices that are useful for constructing nonparametric stochastic models for uncertainty quantification in computational mechanics and in computational science and engineering, in such a framework, for the construction of nonparametric stochastic models of the random tensors or the tensor-valued random fields and also for the nonparametric stochastic models of uncertainties in linear and nonlinear structural dynamics.

All the ensembles of random matrices, which have been developed for the nonparametric method of uncertainties in computational sciences and engineering, are given hereinafter using a unified presentation based on the use of the MaxEnt principle, what allow us, not only to learn about the useful ensembles of random matrices for which the probability distributions and the associated generators of independent realizations are explicitly known but also to present a general tool for constructing any ensemble of random matrices, possibly using computation in high dimension.

- The second part deals with the nonparametric method for uncertainty quantification, which uses the new ensembles of random matrices that have been constructed in the context of the development of the nonparametric method and that are detailed in the first part. The presentation is limited to the nonparametric stochastic model for constitutive equation in linear elasticity, to the nonparametric stochastic model of uncertainties in computational linear structural dynamics for damped elastic structures but also for viscoelastic structures, and to the parametric-nonparametric uncertainties in computational nonlinear structural dynamics. In the last Sect. 15 brief bibliographical analysis is given concerning the propagation of uncertainties using nonparametric or parametric-nonparametric stochastic models of uncertainties, some additional ingredients useful for the nonparametric stochastic modeling of uncertainties, some experimental validations of the nonparametric method of uncertainties,

and finally some applications of the nonparametric stochastic modeling of uncertainties in different fields of computational sciences and engineering.

5 Notations

The following algebraic notations are used through all the developments devoted to this chapter.

5.1 Euclidean and Hermitian Spaces

Let $\mathbf{x} = (x_1, \dots, x_n)$ be a vector in \mathbb{K}^n with $\mathbb{K} = \mathbb{R}$ (the set of all the real numbers) or $\mathbb{K} = \mathbb{C}$ (the set of all the complex numbers). The Euclidean space \mathbb{R}^n (or the Hermitian space \mathbb{C}^n) is equipped with the usual inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^n x_j \bar{y}_j$ and the associated norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ in which \bar{y}_j is the complex conjugate of the complex number y_j and where $\bar{y}_j = y_j$ when y_j is a real number.

5.2 Sets of Matrices

$\mathbb{M}_{n,m}(\mathbb{R})$ be the set of all the $(n \times m)$ real matrices.

$\mathbb{M}_n(\mathbb{R}) = \mathbb{M}_{n,n}(\mathbb{R})$ the square matrices.

$\mathbb{M}_n(\mathbb{C})$ be the set of all the $(n \times m)$ complex matrices.

$\mathbb{M}_n^S(\mathbb{R})$ be the set of all the symmetric $(n \times n)$ real matrices.

$\mathbb{M}_n^{+0}(\mathbb{R})$ be the set of all the semipositive-definite symmetric $(n \times n)$ real matrices.

$\mathbb{M}_n^+(\mathbb{R})$ be the set of all the positive-definite symmetric $(n \times n)$ real matrices.

The ensembles of real matrices are such that

$$\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^{+0}(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R}) \subset \mathbb{M}_n(\mathbb{R}).$$

5.3 Kronecker Symbol, Unit Matrix, and Indicator Function

The Kronecker symbol is denoted as δ_{jk} and is such that $\delta_{jk} = 0$ if $j \neq k$ and $\delta_{jj} = 1$. The unit (or identity) matrix in $\mathbb{M}_n(\mathbb{R})$ is denoted as $[I_n]$ and is such that $[I_n]_{jk} = \delta_{jk}$. Let \mathbb{S} be any subset of any set \mathbb{M} , possibly with $\mathbb{S} = \mathbb{M}$. The indicator function $M \mapsto \mathbb{1}_{\mathbb{S}}(M)$ defined on set \mathbb{M} is such that $\mathbb{1}_{\mathbb{S}}(M) = 1$ if $M \in \mathbb{S} \subset \mathbb{M}$ and $\mathbb{1}_{\mathbb{S}}(M) = 0$ if $M \notin \mathbb{S}$.

5.4 Norms and Usual Operators

- (i) The determinant of a matrix $[G]$ in $\mathbb{M}_n(\mathbb{R})$ is denoted as $\det[G]$, and its trace is denoted as $\text{tr}[G] = \sum_{j=1}^n G_{jj}$.

- (ii) The transpose of a matrix $[G]$ in $\mathbb{M}_{n,m}(\mathbb{R})$ is denoted as $[G]^T$, which is in $\mathbb{M}_{m,n}(\mathbb{R})$.
- (iii) The operator norm of a matrix $[G]$ in $\mathbb{M}_{n,m}(\mathbb{R})$ is denoted as $\|G\| = \sup_{\|\mathbf{x}\| \leq 1} \| [G] \mathbf{x} \|$ for all \mathbf{x} in \mathbb{R}^m , which is such that $\| [G] \mathbf{x} \| \leq \|G\| \|\mathbf{x}\|$ for all \mathbf{x} in \mathbb{R}^m .
- (iv) For $[G]$ and $[H]$ in $\mathbb{M}_{n,m}(\mathbb{R})$, we denote $\ll [G], [H] \gg = \text{tr}\{[G]^T [H]\}$ and the Frobenius norm (or Hilbert-Schmidt norm) $\|G\|_F$ of $[G]$ is such that $\|G\|_F^2 = \ll [G], [G] \gg = \text{tr}\{[G]^T [G]\} = \sum_{j=1}^n \sum_{k=1}^m G_{jk}^2$, which is such that $\|G\| \leq \|G\|_F \leq \sqrt{n} \|G\|$.

5.5 Order Relation in the Set of All the Positive-Definite Real Matrices

Let $[G]$ and $[H]$ be two matrices in $\mathbb{M}_n^+(\mathbb{R})$. The notation $[G] > [H]$ means that the matrix $[G] - [H]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$.

5.6 Probability Space, Mathematical Expectation, and Space of Second-Order Random Vectors

The mathematical expectation relative to a probability space (Θ, \mathcal{T}, P) is denoted as E . The space of all the second-order random variables, defined on (Θ, \mathcal{T}, P) , with values in \mathbb{R}^n , equipped with the inner product $((\mathbf{X}, \mathbf{Y})) = E\{\langle \mathbf{X}, \mathbf{Y} \rangle\}$ and with the associated norm $\| \mathbf{X} \| = ((\mathbf{X}, \mathbf{X}))^{1/2}$, is a Hilbert space denoted as \mathcal{L}_n^2 .

6 The MaxEnt for Constructing Random Matrices

The measure of uncertainties using the entropy of information has been introduced by Shannon [103] in the framework of the development of information theory. The maximum entropy (MaxEnt) principle (that is to say, the maximization of the level of uncertainties) has been introduced by Jaynes [58] and allows a prior probability model of any random variables to be constructed, under the constraints defined by the available information. This principle appears as a major tool to construct the prior probability models. All the ensembles of random matrices presented hereinafter (including the well-known Gaussian Orthogonal Ensemble) are constructed in the framework of a unified presentation using the MaxEnt. This means that the probability distributions of the random matrices belonging to these ensembles are constructed using the MaxEnt.

6.1 Volume Element and Probability Density Function (PDF)

This section deals with the definition of a probability density function (pdf) of a random matrix $[G]$ with values in the Euclidean space $\mathbb{M}_n^S(\mathbb{R})$ (set of all the

symmetric $(n \times n)$ real matrices, equipped with the inner product $\ll [G], [H] \gg = \text{tr}\{[G]^T [H]\}$. In order to correctly defined the integration on Euclidean space $\mathbb{M}_n^S(\mathbb{R})$, it is necessary to define the volume element on this space.

6.1.1 Volume Element on the Euclidean Space of Symmetric Real Matrices

In order to well understand the principle of the construction of the volume element on Euclidean space $\mathbb{M}_n^S(\mathbb{R})$, the construction of the volume element on Euclidean spaces \mathbb{R}^n and $\mathbb{M}_n(\mathbb{R})$ is first introduced.

- (i) *Volume element on Euclidean space \mathbb{R}^n .* Let $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ be the orthonormal basis of \mathbb{R}^n such that $\mathbf{e}_j = (0, \dots, 1, \dots, 0)$ is the null vector with 1 in position j . Consequently, $\langle \mathbf{e}_j, \mathbf{e}_k \rangle = \delta_{jk}$. Any vector $\mathbf{x} = (x_1, \dots, x_n)$ in \mathbb{R}^n can then be written as $\mathbf{x} = \sum_{j=1}^n x_j \mathbf{e}_j$. This Euclidean structure on \mathbb{R}^n defines the volume element $d\mathbf{x}$ on \mathbb{R}^n such that $d\mathbf{x} = \prod_{j=1}^n dx_j$.
- (ii) *Volume element on Euclidean space $\mathbb{M}_n(\mathbb{R})$.* Similarly, let $\{[b_{jk}]\}_{jk}$ be the orthonormal basis of $\mathbb{M}_n(\mathbb{R})$ such that $[b_{jk}] = \mathbf{e}_j \mathbf{e}_k^T$. Consequently, we have $\ll [b_{jk}], [b_{j'k'}] \gg = \delta_{jj'} \delta_{kk'}$. Any matrix $[G]$ in $\mathbb{M}_n(\mathbb{R})$ can be written as $[G] = \sum_{j,k=1}^n G_{jk} [b_{jk}]$ in which $G_{jk} = [G]_{jk}$. This Euclidean structure on $\mathbb{M}_n(\mathbb{R})$ defines the volume element dG on $\mathbb{M}_n(\mathbb{R})$ such that $dG = \prod_{j,k=1}^n dG_{jk}$.
- (iii) *Volume element on Euclidean space $\mathbb{M}_n^S(\mathbb{R})$.* Let $\{[b_{jk}^S]\}, 1 \leq j \leq k \leq n$ be the orthonormal basis of $\mathbb{M}_n^S(\mathbb{R})$ such that $[b_{jj}^S] = \mathbf{e}_j \mathbf{e}_j^T$ and $[b_{jk}^S] = (\mathbf{e}_j \mathbf{e}_k^T + \mathbf{e}_k \mathbf{e}_j^T)/\sqrt{2}$ if $j < k$. We have $\ll [b_{jk}^S], [b_{j'k'}^S] \gg = \delta_{jj'} \delta_{kk'}$ for $j \leq k$ and $j' \leq k'$. Any symmetric matrix $[G]$ in $\mathbb{M}_n^S(\mathbb{R})$ can be written as $[G] = \sum_{1 \leq j \leq k \leq n} G_{jk}^S [b_{jk}^S]$ in which $G_{jj}^S = G_{jj}$ and $G_{jk}^S = \sqrt{2} G_{jk}$ if $j < k$. This Euclidean structure on $\mathbb{M}_n^S(\mathbb{R})$ defines the volume element $d^S G$ on $\mathbb{M}_n^S(\mathbb{R})$ such that $d^S G = \prod_{1 \leq j \leq k \leq n} dG_{jk}^S$. The volume element is then defined by

$$d^S G = 2^{n(n-1)/4} \prod_{1 \leq j \leq k \leq n} dG_{jk}. \quad (8.1)$$

6.1.2 Probability Density Function of a Symmetric Real Random Matrix

Let $[G]$ be a random matrix, defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_n^S(\mathbb{R})$ whose probability distribution $P_{[G]} = p_{[G]}([G])$ $d^S G$ is defined by a pdf $[G] \mapsto p_{[G]}([G])$ from $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{R}^+ = [0, +\infty[$ with respect to the volume element $d^S G$ on $\mathbb{M}_n^S(\mathbb{R})$. This pdf verifies the normalization condition,

$$\int_{\mathbb{M}_n^S(\mathbb{R})} p_{[G]}([G]) d^S G = 1, \quad (8.2)$$

in which the volume element $d^S G$ is defined by Eq. (8.1).

6.1.3 Support of the Probability Density Function

The support of pdf $p_{[G]}$, denoted as $\text{supp } p_{[G]}$, is any subset \mathbb{S}_n of $\mathbb{M}_n^S(\mathbb{R})$, possibly with $\mathbb{S}_n = \mathbb{M}_n^S(\mathbb{R})$. For instance, we can have $\mathbb{S}_n = \mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, which means that $[G]$ is a random matrix with values in the positive-definite symmetric $(n \times n)$ real matrices. Thus, $p_{[G]}([G]) = 0$ for $[G]$ not in \mathbb{S}_n , and Eq.(8.2) can be rewritten as

$$\int_{\mathbb{S}_n} p_{[G]}([G]) d^S G = 1. \quad (8.3)$$

It should be noted that, in the context of the construction of the unknown pdf $p_{[G]}$, it is assumed that support \mathbb{S}_n is a given (known) set.

6.2 The Shannon Entropy as a Measure of Uncertainties

The Shannon measure [103] of uncertainties of random matrix $[G]$ is defined by the entropy of information (*Shannon's entropy*), $\mathcal{E}(p_{[G]})$, of pdf $p_{[G]}$ whose support is $\mathbb{S}_n \subset \mathbb{M}_n^S(\mathbb{R})$, such that

$$\mathcal{E}(p_{[G]}) = - \int_{\mathbb{S}_n} p_{[G]}([G]) \log(p_{[G]}([G])) d^S G, \quad (8.4)$$

which can be rewritten as $\mathcal{E}(p_{[G]}) = -E\{\log(p_{[G]}([G]))\}$. For any pdf $p_{[G]}$ defined on $\mathbb{M}_n^S(\mathbb{R})$ and with support \mathbb{S}_n , entropy $\mathcal{E}(p_{[G]})$ is a real number. The uncertainty increases when the Shannon entropy increases. More the Shannon entropy is small and more the level of uncertainties is small. If $\mathcal{E}(p_{[G]})$ goes to $-\infty$, then the level of uncertainties goes to zero, and random matrix $[G]$ goes to a deterministic matrix for the convergence in probability distribution (in probability law).

6.3 The MaxEnt Principle

As explained before, the use of the MaxEnt principle requires to correctly defined the available information related to random matrix $[G]$ for which pdf $p_{[G]}$ (that is unknown with a given support \mathbb{S}_n) has to be constructed.

6.3.1 Available Information

It is assumed that the available information related to random matrix $[G]$ is represented by the following equation on \mathbb{R}^μ , where μ is a finite positive integer,

$$\mathbf{h}(p_{[G]}) = \mathbf{0}, \quad (8.5)$$

in which $p_{[G]} \mapsto \mathbf{h}(p_{[G]}) = (h_1(p_{[G]}), \dots, h_\mu(p_{[G]}))$ is a given functional of $p_{[G]}$, with values in \mathbb{R}^μ . For instance, if the mean value $E\{[G]\} = [\underline{G}]$ of $[G]$ is a

given matrix in \mathbb{S}_n , and if this mean value $[\underline{G}]$ corresponds to the only available information, then $h_\alpha(p_{[G]}) = \int_{\mathbb{S}_n} G_{jk} p_{[G]}([G]) d^S G - \underline{G}_{jk}$, in which $\alpha = 1, \dots, \mu$ is associated with the couple of indices (j, k) such as $1 \leq j \leq k \leq n$ and where $\mu = n(n + 1)/2$.

6.3.2 The Admissible Sets for the pdf

The following admissible sets $\mathcal{C}_{\text{free}}$ and \mathcal{C}_{ad} are introduced for defining the optimization problem resulting from the use of the MaxEnt principle in order to construct the pdf of random matrix $[G]$. The set $\mathcal{C}_{\text{free}}$ is made up of all the pdf $p : [G] \mapsto p([G])$, defined on $\mathbb{M}_n^S(\mathbb{R})$, with support $\mathbb{S}_n \subset \mathbb{M}_n^S(\mathbb{R})$,

$$\mathcal{C}_{\text{free}} = \{[G] \mapsto p([G]) : \mathbb{M}_n^S(\mathbb{R}) \rightarrow \mathbb{R}^+, \text{supp } p = \mathbb{S}_n, \int_{\mathbb{S}_n} p([G]) d^S G = 1\}. \quad (8.6)$$

The set \mathcal{C}_{ad} is the subset of $\mathcal{C}_{\text{free}}$ for which all the pdf p in $\mathcal{C}_{\text{free}}$ satisfy the constraint defined by

$$\mathcal{C}_{\text{ad}} = \{p \in \mathcal{C}_{\text{free}}, \mathbf{h}(p) = \mathbf{0}\}. \quad (8.7)$$

6.3.3 Optimization Problem for Constructing the pdf

The use of the MaxEnt principle for constructing the pdf $p_{[G]}$ of random matrix $[G]$ yields the following optimization problem:

$$p_{[G]} = \arg \max_{p \in \mathcal{C}_{\text{ad}}} \mathcal{E}(p). \quad (8.8)$$

The optimization problem defined by Eq. (8.8) on set \mathcal{C}_{ad} is transformed in an optimization problem on $\mathcal{C}_{\text{free}}$ in introducing the Lagrange multipliers associated with the constraints defined by Eqs. (8.5) [58, 60, 107]. This type of construction and the analysis of the existence and the uniqueness of a solution of the optimization problem defined by Eq. (8.8) are detailed in Sect. 10.

7 A Fundamental Ensemble for the Symmetric Real Random Matrices with a Unit Mean Value

A fundamental ensemble for the symmetric real random matrices is the Gaussian orthogonal ensemble (GOE) that is an ensemble of random matrices $[G]$, defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_n^S(\mathbb{R})$, defined by a pdf $p_{[G]}$ on $\mathbb{M}_n^S(\mathbb{R})$ with respect to the volume element $d^S G$, for which the support \mathbb{S}_n of $p_{[G]}$ is $\mathbb{M}_n^S(\mathbb{R})$, and satisfying the additional properties defined hereinafter.

7.1 Classical Definition [74]

The additional properties of a random matrix $[G]$ belonging to GOE are (i) invariance under any real orthogonal transformation, that is to say, for any orthogonal $(n \times n)$ real matrix $[R]$ such that $[R]^T [R] = [R] [R]^T = [I_n]$, the pdf (with respect

to $d^S G$) of the random matrix $[R]^T [\mathbf{G}] [R]$ is equal to pdf $p_{\mathbf{G}}$ of random matrix $[\mathbf{G}]$, and (ii) statistical independence of all the real random variables $\{\mathbf{G}_{jk}, 1 \leq j \leq k \leq n\}$.

7.2 Definition by the MaxEnt and Calculation of the pdf

Alternatively to the properties introduced in the classical definition, the additional properties of a random matrix $[\mathbf{G}]$ belonging to GOE are the following. For all $1 \leq j \leq k \leq n$,

$$E\{\mathbf{G}_{jk}\} = 0, \quad E\{\mathbf{G}_{jk}\mathbf{G}_{j'k'}\} = \delta_{jj'}\delta_{kk'}(1 + \delta_{jk}) \frac{\delta^2}{n+1}. \quad (8.9)$$

in which $\delta > 0$ is a given positive-valued hyperparameter whose interpretation is given after. The GOE is then defined using the MaxEnt principle for the available information given by Eq. (8.9), which defines mapping \mathbf{h} (see Eq. (8.5)). The corresponding ensemble is written as GOE_{δ} . In Eq. (8.9), the first equation means that the symmetric random matrix $[\mathbf{G}]$ is centered, and the second one means that its fourth-order covariance tensor is diagonal. Using the MaxEnt principle for random matrix $[\mathbf{G}]$ yields the following unique explicit expression for the pdf $p_{\mathbf{G}}$ with respect to the volume element $d^S G$:

$$p_{[\mathbf{G}]}([G]) = c_G \exp\left(-\frac{n+1}{4\delta^2} \text{tr}\{[G]^2\}\right), \quad G_{kj} = G_{jk}, \quad 1 \leq j \leq k \leq n, \quad (8.10)$$

in which c_G is the constant of normalization such that Eq. (8.2) is verified. It can then be deduced that $\{\mathbf{G}_{jk}, 1 \leq j \leq k \leq n\}$ are Gaussian independent real random variables such that Eq. (8.9) is verified. Consequently, for all $1 \leq j \leq k \leq n$, the pdf (with respect to dg on \mathbb{R}) of the Gaussian real random variable \mathbf{G}_{jk} is $p_{\mathbf{G}_{jk}}(g) = (\sqrt{2\pi}\sigma_{jk})^{-1} \exp\{-g^2/(2\sigma_{jk}^2)\}$ in which the variance of random variable \mathbf{G}_{jk} is $\sigma_{jk}^2 = (1 + \delta_{jk})\delta^2/(n+1)$.

7.3 Decentering and Interpretation of Hyperparameter δ

Let $[\mathbf{G}^{\text{GOE}}]$ be the random matrix with values in $\mathbb{M}_n^S(\mathbb{R})$ such that $[\mathbf{G}^{\text{GOE}}] = [I_n] + [\mathbf{G}]$ in which $[\mathbf{G}]$ is a random matrix belonging to the GOE_{δ} defined before. Therefore $[\mathbf{G}^{\text{GOE}}]$ is not centered and its mean value is $E\{[\mathbf{G}^{\text{GOE}}]\} = [I_n]$. The coefficient of variation of the random matrix $[\mathbf{G}^{\text{GOE}}]$ is defined [109] by

$$\delta_{\text{GOE}} = \left\{ \frac{E\{\|\mathbf{G}^{\text{GOE}} - E\{\mathbf{G}^{\text{GOE}}\}\|_F^2\}}{\|E\{\mathbf{G}^{\text{GOE}}\}\|_F^2} \right\}^{1/2} = \left\{ \frac{1}{n} E\{\|\mathbf{G}^{\text{GOE}} - I_n\|_F^2\} \right\}^{1/2}, \quad (8.11)$$

and $\delta_{\text{GOE}} = \delta$. The parameter $2\delta/\sqrt{n+1}$ can be used to specify a scale.

7.4 Generator of Realizations

For $\theta \in \Theta$, any realization $[\mathbf{G}^{\text{GOE}}(\theta)]$ is given by $[\mathbf{G}^{\text{GOE}}(\theta)] = [I_n] + [\mathbf{G}(\theta)]$ with, for $1 \leq j \leq k \leq n$, $\mathbf{G}_{kj}(\theta) = \mathbf{G}_{jk}(\theta)$ and $\mathbf{G}_{jk}(\theta) = \sigma_{jk} U_{jk}(\theta)$, in which $\{U_{jk}(\theta)\}_{1 \leq j \leq k \leq n}$ is the realization of $n(n+1)/2$ independent copies of a normalized (centered and unit variance) Gaussian real random variable.

7.5 Use of the GOE Ensemble in Uncertainty Quantification

The GOE can then be viewed as a generalization of the Gaussian real random variables to the Gaussian symmetric real random matrices. It can be seen that $[\mathbf{G}^{\text{GOE}}]$ is with values in $\mathbb{M}_n^S(\mathbb{R})$ but is not positive. In addition, for all fixed n ,

$$E\{\|[\mathbf{G}^{\text{GOE}}]^{-1}\|^2\} = +\infty. \quad (8.12)$$

- (i) It has been proved by Weaver [124] and others (see [127] and included references) that the GOE is well adapted for describing universal fluctuations of the eigenfrequencies for generic elastodynamical, acoustical, and elastoacoustical systems, in the high-frequency range corresponding to the asymptotic behavior of the largest eigenfrequencies.
- (ii) On the other hand, random matrix $[\mathbf{G}^{\text{GOE}}]$ cannot be used for stochastic modeling of a symmetric real matrix for which a positiveness property and an integrability of its inverse are required. Such a situation is similar to the following one that is well known for the scalar case. Let us consider the scalar equation in u : $(\underline{G} + G)u = v$ in which v is a given real number, \underline{G} is a given positive number, and G is a positive parameter. This equation has a unique solution $u = (\underline{G} + G)^{-1}v$. Let us assume that G is uncertain and is modeled by a centered random variable \mathbf{G} . We then obtain the random equation in U : $(\underline{G} + \mathbf{G})U = v$. If the random solution U must have finite statistical fluctuations, that is to say, U must be a second-order random variable (this is generally required due to physical considerations), then \mathbf{G} cannot be chosen as a Gaussian second-order centered real random variable, because with such a Gaussian stochastic modeling, the solution $U = (\underline{G} + \mathbf{G})^{-1}v$ is not a second-order random variable, because $E\{U^2\} = +\infty$ due to the non integrability of the function $G \mapsto (\underline{G} + G)^{-2}$ at point $G = -\underline{G}$.

8 Fundamental Ensembles for Positive-Definite Symmetric Real Random Matrices

In this section, we present fundamental ensembles of positive-definite symmetric real random matrices, SG_0^+ , SG_ε^+ , SG_b^+ , and SG_λ^+ , which have been developed and

analyzed for constructing other ensembles of random matrices used for the nonparametric stochastic modeling of matrices encountered in uncertainty quantification.

- The ensemble SG_0^+ is a subset of all the positive-definite symmetric real $(n \times n)$ random matrices for which the mean value is the unit matrix and for which the lower bound is the zero matrix. This ensemble has been introduced and analyzed in [107, 108] in the context of the development of the nonparametric method of model uncertainties induced by modeling errors in computational dynamics. This ensemble has later been used for constructing other ensembles of random matrices encountered in the nonparametric stochastic modeling of uncertainties [110].
- The ensemble SG_ε^+ is a subset of all the positive-definite symmetric real $(n \times n)$ random matrices for which the mean value is the unit matrix and for which there is an arbitrary lower bound that is a positive-definite matrix controlled by an arbitrary positive number ε that can be chosen as small as is desired [114]. In such an ensemble, the lower bound does not correspond to a given matrix that results from a physical model, but allows for assuring a uniform ellipticity for the stochastic modeling of elliptic operators encountered in uncertainty quantification of boundary value problems. The construction of this ensemble is directly derived from ensemble SG_0^+ ,
- The ensemble SG_b^+ is a subset of all the positive-definite random matrices for which the mean value is either not given or is equal to the unit matrix [28, 50] and for which a lower bound and an upper bound are given positive-definite matrices. In this ensemble, the lower bound and the upper bound are not arbitrary positive-definite matrices, but are given matrices that result from a physical model. The ensemble is interesting for the nonparametric stochastic modeling of tensors and tensor-valued random fields for describing uncertain physical properties in elasticity, poroelasticity, thermics, etc.
- The ensemble SG_λ^+ , introduced in [76], is a subset of all the positive-definite random matrices for which the mean value is the unit matrix, for which the lower bound is the zero matrix, and for which the second-order moments of diagonal entries are imposed. In the context of the nonparametric stochastic modeling of uncertainties, this ensemble allows for imposing the variances of certain random eigenvalues of stochastic generalized eigenvalue problems, such as the eigenfrequency problem in structural dynamics.

8.1 Ensemble SG_0^+ of Positive-Definite Random Matrices With a Unit Mean Value

8.1.1 Definition of SG_0^+ Using the MaxEnt and Expression of the pdf

The ensemble SG_0^+ of random matrices $[G_0]$, defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in the set $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, is constructed using the

MaxEnt with the following available information, which defines mapping \mathbf{h} (see Eq. (8.5)):

$$E\{\mathbf{G}_0\} = [I_n], \quad E\{\log(\det[\mathbf{G}_0])\} = v_{G_0}, \quad |v_{G_0}| < +\infty. \quad (8.13)$$

The support of the pdf is the subset $\mathbb{S}_n = \mathbb{M}_n^+(\mathbb{R})$ of $\mathbb{M}_n^S(\mathbb{R})$. This pdf $p_{[\mathbf{G}_0]}$ (with respect to the volume element $d^S G$ on the set $\mathbb{M}_n^S(\mathbb{R})$) verifies the normalization condition and is written as

$$p_{[\mathbf{G}_0]}([G]) = \mathbb{1}_{\mathbb{S}_n}([G]) c_{G_0} (\det[G])^{(n+1)\frac{(1-\delta^2)}{2\delta^2}} \exp(-\frac{n+1}{2\delta^2} \text{tr}[G]). \quad (8.14)$$

The positive parameter δ is such that $0 < \delta < (n+1)^{1/2}(n+5)^{-1/2}$, which allows the level of statistical fluctuations of random matrix $[\mathbf{G}_0]$ to be controlled and which is defined by

$$\delta = \left\{ \frac{E\{\|\mathbf{G}_0 - E\{\mathbf{G}_0\}\|_F^2\}}{\|E\{\mathbf{G}_0\}\|_F^2} \right\}^{1/2} = \left\{ \frac{1}{n} E\{\|\mathbf{G}_0 - [I_n]\|_F^2\} \right\}^{1/2}. \quad (8.15)$$

The normalization positive constant c_{G_0} is such that

$$c_{G_0} = (2\pi)^{-n(n-1)/4} \left(\frac{n+1}{2\delta^2} \right)^{n(n+1)(2\delta^2)^{-1}} \left\{ \prod_{j=1}^n \Gamma\left(\frac{n+1}{2\delta^2} + \frac{1-j}{2}\right) \right\}^{-1}, \quad (8.16)$$

where, for all $z > 0$, $\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt$. Note that $\{\mathbf{G}_0\}_{jk}$, $1 \leq j \leq k \leq n$ are dependent random variables. If $(n+1)/\delta^2$ is an integer, then this pdf coincides with the Wishart probability distribution [2, 107]. If $(n+1)/\delta^2$ is not an integer, then this probability density function can be viewed as a particular case of the Wishart distribution, in infinite dimension, for stochastic processes [104].

8.1.2 Second-Order Moments

Random matrix $[\mathbf{G}_0]$ is such that $E\{\|\mathbf{G}_0\|^2\} \leq E\{\|\mathbf{G}_0\|_F^2\} < +\infty$, which proves that $[\mathbf{G}_0]$ is a second-order random variable. The mean value of random matrix $[\mathbf{G}_0]$ is unit matrix $[I_n]$. The covariance $C_{jk,j'k'} = E\{[\mathbf{G}_0]_{jk} - [I_n]_{jk}\} ([\mathbf{G}_0]_{j'k'} - [I_n]_{j'k'})$ of the real-valued random variables $[\mathbf{G}_0]_{jk}$ and $[\mathbf{G}_0]_{j'k'}$ is $C_{jk,j'k'} = \delta^2(n+1)^{-1}\{\delta_{j'k} \delta_{jk'} + \delta_{jj'} \delta_{kk'}\}$. The variance of real-valued random variable $[\mathbf{G}_0]_{jk}$ is $\sigma_{jk}^2 = C_{jk,jk} = \delta^2(n+1)^{-1}(1 + \delta_{jk})$.

8.1.3 Invariance of Ensemble \mathbf{SG}_0^+ Under Real Orthogonal Transformations

Ensemble \mathbf{SG}_0^+ is invariant under real orthogonal transformations. This means that the pdf (with respect to $d^S G$) of the random matrix $[R]^T [\mathbf{G}_0] [R]$ is equal to the pdf

(with respect to $d^S G$) of random matrix $[G_0]$ for any real orthogonal matrix $[R]$ belonging to $\mathbb{M}_n(\mathbb{R})$.

8.1.4 Invertibility and Convergence Property When Dimension Goes to Infinity

Since $[G_0]$ is a positive-definite random matrix, $[G_0]$ is invertible almost surely, which means that for \mathcal{P} -almost θ in Θ , the inverse $[G_0(\theta)]^{-1}$ of the matrix $[G_0(\theta)]$ exists. This last property does not guarantee that $[G_0]^{-1}$ is a second-order random variable, that is to say, that $E\{\|[G_0]^{-1}\|_F^2\} = \int_{\Theta} \|[G_0(\theta)]^{-1}\|_F^2 d\mathcal{P}(\theta)$ is finite. However, it is proved [108] that

$$E\{\|[G_0]^{-1}\|_F^2\} \leq E\{\|[G_0]^{-1}\|_F^2\} < +\infty, \quad (8.17)$$

and that the following fundamental property holds:

$$\forall n \geq 2, \quad E\{\|[G_0]^{-1}\|_F^2\} \leq C_{\delta} < +\infty, \quad (8.18)$$

in which C_{δ} is a positive finite constant that is independent of n but that depends on δ . This means that $n \mapsto E\{\|[G_0]^{-1}\|_F^2\}$ is a bounded function from $\{n \geq 2\}$ into \mathbb{R}^+ .

It should be noted that the invertibility property defined by Eqs. (8.17) and (8.18) is due to the constraint $E\{\log(\det[G_0])\} = v_{G_0}$ with $|v_{G_0}| < +\infty$. This is the reason why the truncated Gaussian distribution restricted to $\mathbb{M}_n^+(\mathbb{R})$ does not satisfy this invertibility condition that is required for stochastic modeling in many cases.

8.1.5 Probability Density Function of the Random Eigenvalues

Let $\Lambda = (\Lambda_1, \dots, \Lambda_n)$ be the positive-valued random eigenvalues of the random matrix $[G_0]$ belonging to ensemble SG_0^+ , such that $[G_0] \Phi^j = \Lambda_j \Phi^j$ in which Φ^j is the random eigenvector associated with the random eigenvalue Λ_j . The joint probability density function $p_{\Lambda}(\lambda) = p_{\Lambda_1, \dots, \Lambda_n}(\lambda_1, \dots, \lambda_n)$ with respect to $d\lambda = d\lambda_1 \dots d\lambda_n$ of $\Lambda = (\Lambda_1, \dots, \Lambda_n)$ is written [107] as

$$p_{\Lambda}(\lambda) = \mathbb{1}_{[0, +\infty]^n}(\lambda) c_{\Lambda} \left\{ \prod_{j=1}^n \lambda_j^{(n+1)\frac{(1-\delta^2)}{2\delta^2}} \right\} \left\{ \prod_{\alpha < \beta} |\lambda_{\beta} - \lambda_{\alpha}| \right\} \exp \left\{ -\frac{(n+1)}{2\delta^2} \sum_{k=1}^n \lambda_k \right\}, \quad (8.19)$$

in which c_{Λ} is a constant of normalization defined by the equation $\int_0^{+\infty} \dots \int_0^{+\infty} p_{\Lambda}(\lambda) d\lambda = 1$. All the random eigenvalues Λ_j of random matrix $[G_0]$ in SG_0^+ are positive almost surely, while this assertion is not true for the random eigenvalues Λ_j^{GOE} of the random matrix $[G]^{\text{GOE}} = [I_n] + [G]$ in which $[G]$ is a random matrix belonging to the GOE_{δ} ensemble.

8.1.6 Algebraic Representation and Generator of Realizations

The generator of realizations of random matrix $[\mathbf{G}_0]$ whose pdf is defined by Eq. (8.14) is directly deduced from the following algebraic representation of $[\mathbf{G}_0]$ in SG_0^+ . Random matrix $[\mathbf{G}_0]$ is written as $[\mathbf{G}_0] = [\mathbf{L}]^T [\mathbf{L}]$ in which $[\mathbf{L}]$ is an upper triangular real $(n \times n)$ random matrix such that:

- (i) The random variables $\{[\mathbf{L}]_{jk}, j \leq k\}$ are independent;
- (ii) For $j < k$, the real-valued random variable $[\mathbf{L}]_{jk}$ is written as $[\mathbf{L}]_{jk} = \sigma_n U_{jk}$ in which $\sigma_n = \delta(n+1)^{-1/2}$ and where U_{jk} is a real-valued Gaussian random variable with zero mean and variance equal to 1;
- (iii) For $j = k$, the positive-valued random variable $[\mathbf{L}]_{jj}$ is written as $[\mathbf{L}]_{jj} = \sigma_n \sqrt{2V_j}$ in which σ_n is defined before and where V_j is a positive-valued gamma random variable whose pdf is $p_{V_j}(v) = \mathbb{1}_{\mathbb{R}^+}(v) \frac{1}{\Gamma(a_j)} v^{a_j-1} e^{-v}$, in which $a_j = \frac{n+1}{2\delta^2} + \frac{1-j}{2}$.

It should be noted that the set $\{\{U_{jk}\}_{1 \leq j < k \leq n}, \{V_j\}_{1 \leq j \leq n}\}$ of random variables are statistically independent, and the pdf of each diagonal element $[\mathbf{L}]_{jj}$ of random matrix $[\mathbf{L}]$ depends on the rank j of the entry.

For $\theta \in \Theta$, any realization $[\mathbf{G}_0(\theta)]$ is then deduced from the algebraic representation given before, using the realization $\{U_{jk}(\theta)\}_{1 \leq j < k \leq n}$ of $n(n-1)/2$ independent copies of a normalized (zero mean and unit variance) Gaussian real random variable and using the realization $\{V_j(\theta)\}_{1 \leq j \leq n}$ of the n independent positive-valued gamma random variable V_j with parameter a_j .

8.2 Ensemble SG_ε^+ of Positive-Definite Random Matrices with a Unit Mean Value and an Arbitrary Positive-Definite Lower Bound

The ensemble SG_ε^+ is a subset of all the positive-definite random matrices for which the mean value is the unit matrix and for which there is an arbitrary lower bound that is a positive-definite matrix controlled by an arbitrary positive number ε that can be chosen as small as is desired. In this ensemble, the lower bound does not correspond to a given matrix that results from a physical model.

Ensemble SG_ε^+ is the set of the random matrices $[\mathbf{G}]$ with values in $\mathbb{M}_n^+(\mathbb{R})$, which are written as

$$[\mathbf{G}] = \frac{1}{1+\varepsilon} \{[\mathbf{G}_0] + \varepsilon [I_n]\}, \quad (8.20)$$

in which $[\mathbf{G}_0]$ is a random matrix in SG_0^+ , with mean value $E\{[\mathbf{G}_0]\} = [I_n]$, and for which the level of statistical fluctuations is controlled by the hyperparameter δ defined by Eq. (8.15) and where ε is any positive number (note that for $\varepsilon = 0$, $\text{SG}_\varepsilon^+ = \text{SG}_0^+$ and then $[\mathbf{G}] = [\mathbf{G}_0]$). This definition shows that, almost surely,

$$[\mathbf{G}] - [G_\ell] = \frac{1}{1+\varepsilon} [\mathbf{G}_0] > 0, \quad (8.21)$$

in which the lower bound is the positive-definite matrix $[G_\ell] = c_\varepsilon[I_n]$ with $c_\varepsilon = \varepsilon/(1 + \varepsilon)$. For all $\varepsilon > 0$, we have

$$E\{[\mathbf{G}]\} = [I_n], \quad E\{\log(\det([\mathbf{G}] - [G_\ell]))\} = v_{G_\varepsilon}, \quad |v_{G_\varepsilon}| < +\infty, \quad (8.22)$$

with $v_{G_\varepsilon} = v_{G_0} - n \log(1 + \varepsilon)$. The coefficient of variation δ_G of random matrix $[\mathbf{G}]$, defined by

$$\delta_G = \left\{ \frac{E\{\|\mathbf{G} - E\{\mathbf{G}\}\|_F^2\}}{\|E\{\mathbf{G}\}\|_F^2} \right\}^{1/2} = \left\{ \frac{1}{n} E\{\|\mathbf{G} - [I_n]\|_F^2\} \right\}^{1/2}, \quad (8.23)$$

is such that

$$\delta_G = \frac{\delta}{1 + \varepsilon}, \quad (8.24)$$

where δ is the hyperparameter defined by Eq. (8.15).

8.2.1 Generator of Realizations

For $\theta \in \Theta$, any realization $[\mathbf{G}(\theta)]$ of $[\mathbf{G}]$ is given by $[\mathbf{G}(\theta)] = \frac{1}{1+\varepsilon}\{[\mathbf{G}_0(\theta)] + \varepsilon[I_n]\}$ in which $[\mathbf{G}_0(\theta)]$ is a realization of random matrix $[\mathbf{G}_0]$ constructed as explained before.

8.2.2 Lower Bound and Invertibility

For all $\varepsilon > 0$, the bilinear form $b(\mathbf{X}, \mathbf{Y}) = (([\mathbf{G}]\mathbf{X}, \mathbf{Y}))$ on $\mathcal{L}_n^2 \times \mathcal{L}_n^2$ is such that

$$b(\mathbf{X}, \mathbf{X}) \geq c_\varepsilon \|\mathbf{X}\|^2. \quad (8.25)$$

Random matrix $[\mathbf{G}]$ is invertible almost surely and its inverse $[\mathbf{G}]^{-1}$ is a second-order random variable, $E\{\|[\mathbf{G}]^{-1}\|_F^2\} < +\infty$.

8.3 Ensemble SG_b^+ of Positive-Definite Random Matrices with Given Lower and Upper Bounds and with or without Given Mean Value

The ensemble SG_b^+ is a subset of all the positive-definite random matrices for which the mean value is either the unit matrix or is not given and for which a lower bound and an upper bound are given positive-definite matrices. In this ensemble, the lower bound and the upper bound are not arbitrary positive-definite matrices, but are given matrices that result from a physical model.

The ensemble SG_b^+ is constituted of random matrices $[\mathbf{G}_b]$, defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in the set $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, such that

$$[0] < [G_\ell] < [\mathbf{G}_b] < [G_u], \quad (8.26)$$

in which the lower bound $[G_\ell]$ and the upper bound $[G_u]$ are given matrices in $\mathbb{M}_n^+(\mathbb{R})$ such that $[G_\ell] < [G_u]$. The support of the pdf $p_{[\mathbf{G}_b]}$ (with respect to the volume element $d^S G$ on $\mathbb{M}_n^S(\mathbb{R})$) of random matrix $[\mathbf{G}_b]$ is the subset \mathbb{S}_n of $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$ such that

$$\mathbb{S}_n = \{ [G] \in \mathbb{M}_n^+(\mathbb{R}) \mid [G_\ell] < [G] < [G_u] \}. \quad (8.27)$$

The available information associated with the presence of the lower and upper bounds is defined by

$$E\{\log(\det([\mathbf{G}_b] - [G_\ell]))\} = v_\ell, \quad E\{\log(\det([G_u] - [\mathbf{G}_b]))\} = v_u, \quad (8.28)$$

in which v_ℓ and v_u are two constants such that $|v_\ell| < +\infty$ and $|v_u| < +\infty$. The mean value $[\underline{G}_b] = E\{[\mathbf{G}_b]\}$ is given by

$$[\underline{G}_b] = \int_{\mathbb{S}_n} [G] p_{[\mathbf{G}_b]}([G]) d^S G. \quad (8.29)$$

The positive parameter δ_b , which allows the level of statistical fluctuations of random matrix $[\mathbf{G}_b]$ to be controlled, is defined by

$$\delta_b = \left\{ \frac{E\{\|\mathbf{G}_b - \underline{G}_b\|_F^2\}}{\|\underline{G}_b\|_F^2} \right\}^{1/2}. \quad (8.30)$$

8.3.1 Definition of SG_b^+ for a Non-given Mean Value Using the MaxEnt

The mean value $[\underline{G}_b]$ of random matrix $[\mathbf{G}_b]$ is not given and therefore does not constitute an available information. In this case, the ensemble SG_b^+ is constructed using the MaxEnt with the available information given by Eq. (8.28) (that defines mapping \mathbf{h} introduced in Eq. (8.5) and rewritten for $p_{[\mathbf{G}_b]}$). The pdf $p_{[\mathbf{G}_b]}$ is the generalized matrix-variate beta-type I pdf [55]:

$$p_{[\mathbf{G}_b]}([G]) = \mathbb{1}_{\mathbb{S}_n}([G]) c_{G_b} (\det[G - G_\ell])^{\alpha-(n+1)/2} (\det[G_u - G])^{\beta-(n+1)/2}, \quad (8.31)$$

in which c_{G_b} is the normalization constant and where $\alpha > (n-1)/2$ and $\beta > (n-1)/2$ are two real parameters that are unknown and that depend on the two unknown constants v_ℓ and v_u . The mean value $[\underline{G}_b]$ must be calculated using Eqs. (8.29) and (8.31), and the hyperparameter δ_b , which characterizes the level of statistical fluctuations, must be calculated using Eqs. (8.30) and (8.31). Consequently, $[\underline{G}_b]$ and δ_b depend on α and β . It can be seen that, for $n \geq 2$, the two scalar parameters α and β are not sufficient for identifying the mean value $[\underline{G}_b]$ that is in \mathbb{S}_n and the hyperparameter δ_b . An efficient algorithm for generating realizations of $[\mathbf{G}_b]$ can be found in [28].

8.3.2 Definition of SG_b^+ for a Given Mean Value Using the MaxEnt

The mean value $[\underline{G}_b]$ of random matrix $[\mathbf{G}_b]$ is given such that $[G_\ell] < [\underline{G}_b] < [G_u]$. In this case, the ensemble SG_b^+ is constructed using the MaxEnt with the available information given by Eqs. (8.28) and (8.29) that defines mapping \mathbf{h} introduced in Eq. (8.5). Following the construction proposed in [50], the following change of variable is introduced:

$$[\mathbf{A}_0] = ([\mathbf{G}_b] - [G_\ell])^{-1} - [G_{\ell u}]^{-1}, \quad [G_{\ell u}] = [G_u] - [G_\ell] \in \mathbb{M}_n^+(\mathbb{R}). \quad (8.32)$$

This equation shows that the random matrix $[\mathbf{A}_0]$ is with values in $\mathbb{M}_n^+(\mathbb{R})$. Introducing the mean value $[\underline{\mathbf{A}}_0] = E\{[\mathbf{A}_0]\}$ that belongs to $\mathbb{M}_n^+(\mathbb{R})$ and is Cholesky factorization $[\underline{\mathbf{A}}_0] = [\underline{\mathbf{L}}_0]^T [\underline{\mathbf{L}}_0]$ in which $[\underline{\mathbf{L}}_0]$ is an upper triangular real $(n \times n)$ matrix, random matrix $[\mathbf{A}_0]$ can be written as $[\mathbf{A}_0] = [\underline{\mathbf{L}}_0]^T [\mathbf{G}_0] [\underline{\mathbf{L}}_0]$ with $[\mathbf{G}_0]$ that belongs to ensemble SG_0^+ depending on the hyperparameter δ defined by Eq. (8.15). The inversion of Eq. (8.32) yields

$$[\mathbf{G}_b] = [G_\ell] + ([\underline{\mathbf{L}}_0]^T [\mathbf{G}_0] [\underline{\mathbf{L}}_0] + [G_{\ell u}]^{-1})^{-1}. \quad (8.33)$$

It can then be seen that for any arbitrary small $\varepsilon_0 > 0$ (for instance, $\varepsilon_0 = 10^{-6}$), we have

$$\|E\{([\mathbf{A}_0] + [G_{\ell u}]^{-1})^{-1}\} + [G_\ell] - [\underline{G}_b]\|_F \leq \varepsilon_0 \|\underline{G}_b\|_F. \quad (8.34)$$

For δ and $[\underline{\mathbf{L}}_0]$ fixed, for θ in Θ , the realization $[\mathbf{G}_0(\theta)]$ of random matrix $[\mathbf{G}_0]$ in SG_0^+ is constructed using the generator of $[\mathbf{G}_0]$, which has been detailed before. The mean value $E\{[\mathbf{G}_b]\}$ and the hyperparameter δ_b defined by Eq. (8.30) are estimated with the corresponding realization $[\mathbf{G}_b(\theta)] = [G_\ell] + ([\underline{\mathbf{L}}_0]^T [\mathbf{G}_0(\theta)] [\underline{\mathbf{L}}_0] + [G_{\ell u}]^{-1})^{-1}$ of random matrix $[\mathbf{G}_b]$. Let \mathcal{U}_L be the set of all the upper triangular real $(n \times n)$ matrices $[\underline{\mathbf{L}}_0]$ with positive diagonal entries. For a fixed value of δ , and for a given target value of $[\underline{G}_b]$, the value $[\underline{\mathbf{L}}_0^{\text{opt}}]$ of $[\underline{\mathbf{L}}_0]$ is calculated in solving the optimization problem

$$[\underline{\mathbf{L}}_0^{\text{opt}}] = \arg \min_{[\underline{\mathbf{L}}_0] \in \mathcal{U}_L} \mathcal{F}([\underline{\mathbf{L}}_0]), \quad (8.35)$$

in which the cost function \mathcal{F} is deduced from Eq. (8.34) and is written as

$$\mathcal{F}([\underline{\mathbf{L}}_0]) = \|E\{([\underline{\mathbf{L}}_0]^T [\mathbf{G}_0] [\underline{\mathbf{L}}_0] + [G_{\ell u}]^{-1})^{-1}\} + [G_\ell] - [\underline{G}_b]\|_F / \|\underline{G}_b\|_F. \quad (8.36)$$

8.4 Ensemble \mathbf{SG}_λ^+ of Positive-Definite Random Matrices with a Unit Mean Value and Imposed Second-Order Moments

The ensemble \mathbf{SG}_λ^+ is a subset of all the positive-definite random matrices for which the mean value is the unit matrix, for which the lower bound is the zero matrix, and for which the second-order moments of diagonal entries are imposed. In the context of nonparametric stochastic modeling of uncertainties, this ensemble allows for imposing the variances of certain random eigenvalues of stochastic generalized eigenvalue problems.

8.4.1 Definition of \mathbf{SG}_λ^+ Using the MaxEnt and Expression of the pdf

The ensemble \mathbf{SG}_λ^+ of random matrices $[\mathbf{G}_\lambda]$, defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in the set $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, is constructed using the MaxEnt with the following available information, which defines mapping \mathbf{h} (see Eq. (8.5)):

$$E\{[\mathbf{G}_\lambda]\} = [I_n], \quad E\{\log(\det[\mathbf{G}_\lambda])\} = v_{G_\lambda}, \quad E\{[\mathbf{G}_\lambda]_{jj}^2\} = s_j^2, \quad j = 1, \dots, m, \quad (8.37)$$

in which $|v_{G_\lambda}| < +\infty$, with $m < n$, and where s_1^2, \dots, s_m^2 are m given positive constants. The pdf $p_{[\mathbf{G}_\lambda]}$ (with respect to the volume element $d^S G$ on the set $\mathbb{M}_n^S(\mathbb{R})$) has a support that is $\mathbb{S}_n = \mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$ of $\mathbb{M}_n^S(\mathbb{R})$. The pdf verifies the normalization condition and is written [76] as

$$p_{[\mathbf{G}_\lambda]}([G]) = \mathbb{1}_{\mathbb{S}_n}([G]) \times C_{G_\lambda} \times (\det[G])^{\alpha-1} \times \exp\{-\text{tr}\{[\mu]^T[G]\} - \sum_{j=1}^m \tau_j G_{jj}^2\}, \quad (8.38)$$

in which C_{G_λ} is the normalization constant and α is a parameter such that $n + 2\alpha - 1 > 0$, where $[\mu]$ is a diagonal real $(n \times n)$ matrix such that $\mu_{jj} = (n + 2\alpha - 1)/2$ for $j > m$ and where $\mu_{11}, \dots, \mu_{mm}$ and τ_1, \dots, τ_m are $2m$ positive parameters, which are expressed as a function of α and s_1^2, \dots, s_m^2 . The level of statistical fluctuations of random matrix $[\mathbf{G}_\lambda]$ is controlled by the positive hyperparameter δ that is defined by

$$\delta = \left\{ \frac{E\{\|\mathbf{G}_\lambda - E\{\mathbf{G}_\lambda\}\|_F^2\}}{\|E\{\mathbf{G}_\lambda\}\|_F^2} \right\}^{1/2} = \left\{ \frac{1}{n} E\{\|\mathbf{G}_\lambda - [I_n]\|_F^2\} \right\}^{1/2}, \quad (8.39)$$

and where δ is such that

$$\delta^2 = \frac{1}{n} \sum_{j=1}^m s_j^2 + \frac{n + 1 - (m/n)(n + 2\alpha - 1)}{n + 2\alpha - 1}. \quad (8.40)$$

8.4.2 Generator of Realizations

For given $m < n$, δ , and s_1^2, \dots, s_m^2 , the explicit generator of realizations of random matrix $[\mathbf{G}_\lambda]$ whose pdf is defined by Eq. (8.38) is detailed in [76].

9 Ensembles of Random Matrices for the Nonparametric Method in Uncertainty Quantification

In this section, we present the ensembles SE_0^+ , SE_ε^+ , SE^{+0} , SE^{rect} , and SE^{HT} of random matrices which result from some transformations of the fundamental ensembles introduced before. These ensembles of random matrices are useful for performing the nonparametric stochastic modeling of matrices encountered in uncertainty quantification of computational models in structural dynamics, acoustics, vibroacoustics, fluid-structure interaction, unsteady aeroelasticity, soil-structure interaction, etc., but also in solid mechanics (elasticity tensors of random elastic continuous media, matrix-valued random fields for heterogeneous microstructures of materials), thermic (thermal conductivity tensor), electromagnetism (dielectric tensor), etc.

The ensembles of random matrices, devoted to the construction of nonparametric stochastic models of matrices encountered in uncertainty quantification, are briefly summarized below and then are mathematically detailed:

- The ensemble SE_0^+ is a subset of all the positive-definite random matrices for which the mean values are given and differ from the unit matrix (unlike to ensemble SG_0^+) and for which the lower bound is the zero matrix. This ensemble is constructed as a transformation of ensemble SG_0^+ in keeping all the mathematical properties of ensemble SG_0^+ such as the positiveness.
- The ensemble SE_ε^+ is a subset of all the positive-definite random matrices for which the mean value is a given positive-definite matrix and for which there is an arbitrary lower bound that is a positive-definite matrix controlled by an arbitrary positive number ε that can be chosen as small as is desired. In this ensemble, the lower bound does not correspond to a given matrix that results from a physical model. This ensemble is constructed as a transformation of ensemble SG_ε^+ and has the same area of use than ensemble SE_0^+ for stochastic modeling in uncertainty quantification but for which a lower bound is required in the stochastic modeling for mathematical reasons.
- The ensemble SE^{+0} is similar to ensemble SG_0^+ but is constituted of semipositive-definite ($m \times m$) real random matrices for which the mean value is a given semipositive-definite matrix. This ensemble is constructed as a transformation of positive-definite ($n \times n$) real random matrices belonging to ensemble SG_0^+ , with $n < m$, in which the dimension of the null space is $m - n$. Such an ensemble is useful for the nonparametric stochastic modeling of uncertainties such as those encountered in structural dynamics in presence of rigid body displacements.

- The ensemble SE^{rect} is an ensemble of rectangular random matrices for which the mean value is a given rectangular matrix and which is constructed using ensemble SE_ε^+ . This ensemble is useful for the nonparametric stochastic modeling of some uncertain coupling operators encountered, for instance, in fluid-structure interaction and in vibroacoustics.
- The ensemble SE^{HT} is a set of random functions with values in the set of the complex matrices such that the real part and the imaginary part are positive-definite random matrices that are constrained by an underlying Hilbert transform induced by a causality property. This ensemble allows for a nonparametric stochastic modeling in uncertainty quantification encountered, for instance, in linear viscoelasticity.

9.1 Ensemble SE_0^+ of Positive-Definite Random Matrices with a Given Mean Value

The ensemble SE_0^+ is a subset of all the positive-definite random matrices for which the mean values are given and differ from the unit matrix (unlike to ensemble SG_0^+). This ensemble is constructed as a transformation of ensemble SG_0^+ in keeping all the mathematical properties of ensemble SG_0^+ such as the positiveness [107].

9.1.1 Definition of Ensemble SE_0^+

Any random matrix $[\mathbf{A}_0]$ in ensemble SE_0^+ is defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, is with values in $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, and is such that

$$E\{[\mathbf{A}_0]\} = [A], \quad E\{\log(\det[\mathbf{A}_0])\} = v_{A_0}, \quad |v_{A_0}| < +\infty, \quad (8.41)$$

in which the mean value $[A]$ is a given matrix in $\mathbb{M}_n^+(\mathbb{R})$.

9.1.2 Expression of $[\mathbf{A}_0]$ as a Transformation of $[\mathbf{G}_0]$ and Generator of Realizations

Positive-definite mean matrix $[A]$ is factorized (Cholesky) as

$$[A] = [L_A]^T [L_A], \quad (8.42)$$

in which $[L_A]$ is an upper triangular matrix in $\mathbb{M}_n(\mathbb{R})$. Taking into account Eq. (8.41) and the definition of ensemble SG_0^+ , any random matrix $[\mathbf{A}_0]$ in ensemble SE_0^+ is written as

$$[\mathbf{A}_0] = [L_A]^T [\mathbf{G}_0] [L_A], \quad (8.43)$$

in which the random matrix $[\mathbf{G}_0]$ belongs to ensemble SG_0^+ , with mean value $E\{[\mathbf{G}_0]\} = [I_n]$, and for which the level of statistical fluctuations is controlled by the hyperparameter δ defined by Eq. (8.15).

Generator of realizations. For all θ in Θ , the realization $[\mathbf{G}_0(\theta)]$ of $[\mathbf{G}_0]$ is constructed as explained before. The realization $[\mathbf{A}_0(\theta)]$ of random matrix $[\mathbf{A}_0]$ is calculated by $[\mathbf{A}_0(\theta)] = [L_A]^T [\mathbf{G}_0(\theta)] [L_A]$.

Remark 1. It should be noted that the mean matrix $[A]$ could also been written as $[A] = [A]^{1/2} [A]^{1/2}$ in which $[A]^{1/2}$ is the square root of $[A]$ in $\mathbb{M}_n^+(\mathbb{R})$ and the random matrix $[\mathbf{A}_0]$ could then be written as $[\mathbf{A}_0] = [A]^{1/2} [\mathbf{G}_0] [A]^{1/2}$.

9.1.3 Properties of Random Matrix $[\mathbf{A}_0]$

Any random matrix $[\mathbf{A}_0]$ in ensemble SE_0^+ is a second-order random variable,

$$E\{\|\mathbf{A}_0\|^2\} \leq E\{\|\mathbf{A}_0\|_F^2\} < +\infty, \quad (8.44)$$

and its inverse $[\mathbf{A}_0]^{-1}$ exists almost surely and is a second-order random variable,

$$E\{\|[\mathbf{A}_0]^{-1}\|^2\} \leq E\{\|[\mathbf{A}_0]^{-1}\|_F^2\} < +\infty. \quad (8.45)$$

9.1.4 Covariance Tensor and Coefficient of Variation of Random Matrix $[\mathbf{A}_0]$

The covariance $C_{jk,j'k'} = E\{([\mathbf{A}_0]_{jk} - A_{jk})([\mathbf{A}_0]_{j'k'} - A_{j'k'})\}$ of random variables $[\mathbf{A}_0]_{jk}$ and $[\mathbf{A}_0]_{j'k'}$ is written as

$$C_{jk,j'k'} = \frac{\delta^2}{n+1} \{A_{j'k} A_{jk'} + A_{jj'} A_{kk'}\}, \quad (8.46)$$

and the variance $\sigma_{jk}^2 = C_{jk,jk}$ of random variable $[\mathbf{A}_0]_{jk}$ is

$$\sigma_{jk}^2 = \frac{\delta^2}{n+1} \{A_{jk}^2 + A_{jj} A_{kk}\}. \quad (8.47)$$

The coefficient of variation δ_{A_0} of random matrix $[\mathbf{A}_0]$ is defined by

$$\delta_{A_0} = \left\{ \frac{E\{\|\mathbf{A}_0 - A\|_F^2\}}{\|A\|_F^2} \right\}^{1/2}. \quad (8.48)$$

Since $E\{\|\mathbf{A}_0 - A\|_F^2\} = \sum_{j=1}^n \sum_{k=1}^n \sigma_{jk}^2$, we have

$$\delta_{A_0} = \frac{\delta}{\sqrt{n+1}} \left\{ 1 + \frac{(\text{tr}[A])^2}{\|A\|_F^2} \right\}^{1/2}. \quad (8.49)$$

9.2 Ensemble SE_ϵ^+ of Positive-Definite Random Matrices with a Given Mean Value and an Arbitrary Positive-Definite Lower Bound

The ensemble SE_ϵ^+ is a set of positive-definite random matrices for which the mean value is a given positive-definite matrix and for which there is an arbitrary lower

bound that is a positive-definite matrix controlled by an arbitrary positive number ε that can be chosen as small as is desired. In this ensemble, the lower bound does not correspond to a given matrix that results from a physical model. This ensemble is then constructed as a transformation of ensemble SG_ε^+ and has the same area of use than ensemble SE_0^+ for stochastic modeling in uncertainty quantification, but for which a lower bound is required in the stochastic modeling for mathematical reasons.

9.2.1 Definition of Ensemble SE_ε^+

For a fixed positive value of parameter ε (generally chosen very small, as 10^{-6}), any random matrix $[\mathbf{A}]$ in ensemble SE_ε^+ is defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, is with values in $\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$, and is such that

$$[\mathbf{A}] = [L_A]^T [\mathbf{G}] [L_A], \quad (8.50)$$

in which $[L_A]$ is the upper triangular matrix in $\mathbb{M}_n(\mathbb{R})$ corresponding by the Cholesky factorization $[L_A]^T [L_A] = [A]$ of the positive-definite mean matrix $[A] = E\{[\mathbf{A}]\}$ of random matrix $[\mathbf{A}]$, and where the random matrix $[\mathbf{G}]$ belongs to ensemble SG_ε^+ , with mean value $E\{[\mathbf{G}]\} = [I_n]$ and for which the coefficient of variation δ_G is defined by Eq. (8.24) as a function of the hyperparameter δ defined by Eq. (8.15), which allows the level of statistical fluctuations to be controlled. It should be noted that for $\varepsilon = 0$, $[\mathbf{G}] = [\mathbf{G}_0]$ that yields $[\mathbf{A}] = [\mathbf{A}_0]$, and consequently, the ensemble SE_ε^+ coincides with SE_0^+ (if $\varepsilon = 0$).

Generator of realizations. For all θ in Θ , the realization $[\mathbf{G}(\theta)]$ of $[\mathbf{G}]$ is constructed as explained before. The realization $[\mathbf{A}(\theta)]$ of random matrix $[\mathbf{A}]$ is calculated by $[\mathbf{A}(\theta)] = [L_A]^T [\mathbf{G}(\theta)] [L_A]$.

9.2.2 Properties of Random Matrix $[\mathbf{A}]$

Almost surely, we have

$$[\mathbf{A}] - [A_\ell] = \frac{1}{1 + \varepsilon} [\mathbf{A}_0] > 0, \quad (8.51)$$

in which $[\mathbf{A}_0]$ is defined by Eq. (8.43) and where the lower bound is the positive-definite matrix $[A_\ell] = c_\varepsilon [A]$ with $c_\varepsilon = \varepsilon/(1 + \varepsilon)$, and we have the following properties:

$$E\{[\mathbf{A}]\} = [A], \quad E\{\log(\det([\mathbf{A}] - [A_\ell]))\} = v_A, \quad |v_A| < +\infty, \quad (8.52)$$

with $v_A = v_{A_0} - n \log(1 + \varepsilon)$. For all $\varepsilon > 0$, random matrix $[\mathbf{A}]$ in ensemble SE_ε^+ is a second-order random variable,

$$E\{\|\mathbf{A}\|^2\} \leq E\{\|\mathbf{A}\|_F^2\} < +\infty, \quad (8.53)$$

and the bilinear form $b_A(\mathbf{X}, \mathbf{Y}) = (([\mathbf{A}] \mathbf{X}, \mathbf{Y}))$ on $\mathcal{L}_n^2 \times \mathcal{L}_n^2$ is such that

$$b_A(\mathbf{X}, \mathbf{X}) \geq c_\varepsilon (([\mathbf{A}] \mathbf{X}, \mathbf{X})) = c_\varepsilon \|\| [L_A] \mathbf{X} \||^2. \quad (8.54)$$

Random matrix $[\mathbf{A}]$ is invertible almost surely and its inverse $[\mathbf{A}]^{-1}$ is a second-order random variable,

$$E\{\|[\mathbf{A}]^{-1}\|^2\} \leq E\{\|[\mathbf{A}]^{-1}\|_F^2\} < +\infty. \quad (8.55)$$

The coefficient of variation δ_A of random matrix $[\mathbf{A}]$, defined by

$$\delta_A = \left\{ \frac{E\{\|\mathbf{A} - A\|_F^2\}}{\|A\|_F^2} \right\}^{1/2}. \quad (8.56)$$

is such that

$$\delta_A = \frac{1}{1 + \varepsilon} \delta_{A_0}, \quad (8.57)$$

in which δ_{A_0} is defined by Eq. (8.49).

9.3 Ensemble SE^{+0} of Semipositive-Definite Random Matrices with a Given Semipositive-Definite Mean Value

The ensemble SE^{+0} is similar to ensemble SG_0^+ but is constituted of semipositive-definite ($m \times m$) real random matrices $[\mathbf{A}]$ for which the mean value is a given semipositive-definite matrix. This ensemble is constructed [110] as a transformation of positive-definite ($n \times n$) real random matrices $[\mathbf{G}_0]$ belonging to ensemble SG_0^+ , with $n < m$.

9.3.1 Algebraic Structure of the Random Matrices in SE^{+0}

The ensemble SE^{+0} is constituted of random matrix $[\mathbf{A}]$ with values in the set $\mathbb{M}_m^{+0}(\mathbb{R})$ such that the null space of $[\mathbf{A}]$, denoted as $\text{null}([\mathbf{A}])$, is deterministic and is a subspace of \mathbb{R}^m with a fixed dimension $\mu_{\text{null}} < m$. This deterministic null space is defined as the null space of the mean value $[A] = E\{[\mathbf{A}]\}$ that is given in $\mathbb{M}_m^{+0}(\mathbb{R})$. We then have

$$[A] \in \mathbb{M}_m^{+0}(\mathbb{R}), \quad \dim \text{null}([A]) = \mu_{\text{null}} < m, \quad \text{null}([\mathbf{A}]) = \text{null}([A]). \quad (8.58)$$

There is a rectangular matrix $[R_A]$ in $\mathbb{M}_{n,m}(\mathbb{R})$, with $n = m - \mu_{\text{null}}$, such that

$$[A] = [R_A]^T [R_A]. \quad (8.59)$$

Such a factorization is performed using classical algorithms [47].

9.3.2 Definition and Construction of Ensemble SE^{+0}

The ensemble SE^{+0} is then defined as the subset of all the second-order random matrices $[\mathbf{A}]$, defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in the set $\mathbb{M}_m^{+0}(\mathbb{R})$, which are written as

$$[\mathbf{A}] = [R_A]^T [\mathbf{G}] [R_A], \quad (8.60)$$

in which $[\mathbf{G}]$ is a positive-definite symmetric $(n \times n)$ real random matrix belonging to ensemble SE_ε^+ , with mean value $E\{[\mathbf{G}]\} = [I_n]$ and for which the coefficient of variation δ_G is defined by Eq.(8.24) as a function of the hyperparameter δ defined by Eq.(8.15), which allows the level of statistical fluctuations to be controlled.

Generator of realizations. For all θ in Θ , the realization $[\mathbf{G}(\theta)]$ of $[\mathbf{G}]$ is constructed as explained before. The realization $[\mathbf{A}(\theta)]$ of random matrix $[\mathbf{A}]$ is calculated by $[\mathbf{A}(\theta)] = [R_A]^T [\mathbf{G}(\theta)] [R_A]$.

9.4 Ensemble SE^{rect} of Rectangular Random Matrices with a Given Mean Value

The ensemble SE^{rect} is an ensemble of rectangular random matrices for which the mean value is a given rectangular matrix and which is constructed with the MaxEnt. Such an ensemble depends on the available information and consequently, is not unique. We present hereinafter the construction proposed in [110], which is based on the use of a fundamental algebraic property for rectangular real matrices, which allows ensemble SE_ε^+ to be used.

9.4.1 Decomposition of a Rectangular Matrix

Let $[A]$ be a rectangular real matrix in $\mathbb{M}_{m,n}(\mathbb{R})$ for which its null space is reduced to $\{0\}$ ($[A] \mathbf{x} = \mathbf{0}$ yields $\mathbf{x} = \mathbf{0}$). Such a rectangular matrix $[A]$ can be written as

$$[A] = [U] [T], \quad (8.61)$$

in which the square matrix $[T]$ and the rectangular matrix $[U]$ are such that

$$[T] \in \mathbb{M}_n^+(\mathbb{R}) \quad \text{and} \quad [U] \in \mathbb{M}_{m,n}(\mathbb{R}) \quad \text{such that} \quad [U]^T [U] = [I_n]. \quad (8.62)$$

The construction of the decomposition defined by Eq.(8.61) can be performed, for instance, by using the singular value decomposition of $[A]$.

9.4.2 Definition of Ensemble SE^{rect}

Let $[A]$ be a given rectangular real matrix in $\mathbb{M}_{m,n}(\mathbb{R})$ with a null space reduced to $\{0\}$ and whose decomposition is given by Eqs.(8.61) and (8.62). Since symmetric real matrix $[T]$ is positive definite, there is an upper triangular matrix $[L_T]$ in $\mathbb{M}_n(\mathbb{R})$ such that $[T] = [L_T]^T [L_T]$ that corresponds to the Cholesky factorization of matrix $[T]$.

A random rectangular matrix $[\mathbf{A}]$ belonging to ensemble SE^{rect} is a second-order random matrix defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_{m,n}(\mathbb{R})$, whose mean value is the rectangular matrix $[A] = E\{[\mathbf{A}]\}$, and which is written as

$$[\mathbf{A}] = [U] [\mathbf{T}], \quad (8.63)$$

in which the random $(n \times n)$ matrix $[\mathbf{T}]$ belongs to ensemble SE_ε^+ and is then written as

$$[\mathbf{T}] = [L_T]^T [\mathbf{G}] [L_T]. \quad (8.64)$$

The random matrix $[\mathbf{G}]$ belongs to ensemble SG_ε^+ in which $[\mathbf{G}]$ is a positive-definite symmetric $(n \times n)$ real random matrix belonging to ensemble SE_ε^+ , with mean value $E\{[\mathbf{G}]\} = [I_n]$ and for which the coefficient of variation δ_G is defined by Eq. (8.24) as a function of hyperparameter δ defined by Eq. (8.15), which allows the level of statistical fluctuations to be controlled.

Generator of realizations. For all θ in Θ , the realization $[\mathbf{G}(\theta)]$ of $[\mathbf{G}]$ is constructed as explained before. The realization $[\mathbf{A}(\theta)]$ of random matrix $[\mathbf{A}]$ is calculated by $[\mathbf{A}(\theta)] = [U][L_T]^T[\mathbf{G}(\theta)][L_T]$.

9.5 Ensemble SE^{HT} of a Pair of Positive-Definite Matrix-Valued Random Functions Related by a Hilbert Transform

The ensemble SE^{HT} is a set of random functions $\omega \mapsto [\mathbf{Z}(\omega)] = [\mathbf{K}(\omega)] + i\omega [\mathbf{D}(\omega)]$ indexed by \mathbb{R} with values in a subset of all the $(n \times n)$ complex matrices such that $[\mathbf{K}(\omega)]$ and $[\mathbf{D}(\omega)]$ are positive-definite random matrices that are constrained by an underlying Hilbert transform induced by a causality property [115].

9.5.1 Defining the Deterministic Matrix Problem

We consider a family of complex $(n \times n)$ matrices $[Z(\omega)]$ depending on a parameter ω in \mathbb{R} , such that $[Z(\omega)] = i\omega[D(\omega)] + [K(\omega)]$ where i is the pure imaginary complex number ($i = \sqrt{-1}$) and where, for all ω in \mathbb{R} ,

- (i) $[D(\omega)]$ and $[K(\omega)]$ belong to $\mathbb{M}_n^+(\mathbb{R})$.
- (ii) $[D(-\omega)] = [D(\omega)]$ and $[K(-\omega)] = [K(\omega)]$.
- (iii) Matrices $[D(\omega)]$ and $[K(\omega)]$ are such that

$$\omega[D(\omega)] = [\widehat{N}^I(\omega)], \quad [K(\omega)] = [K_0] + [\widehat{N}^R(\omega)]. \quad (8.65)$$

The real matrices $[\widehat{N}^R(\omega)]$ and $[\widehat{N}^I(\omega)]$ are the real part and the imaginary part of the $(n \times n)$ complex matrix $[\widehat{N}(\omega)] = \int_{\mathbb{R}} e^{-i\omega t} [N(t)] dt$ that is the Fourier transform of an integrable function $t \mapsto [N(t)]$ from \mathbb{R} into $\mathbb{M}_n(\mathbb{R})$ such that $[N(t)] = [0]$ for $t < 0$ (causal function). Consequently, $\omega \mapsto [\widehat{N}^R(\omega)]$ and $\omega \mapsto [\widehat{N}^I(\omega)]$ are continuous functions on \mathbb{R} , which goes to $[0]$ as $|\omega| \rightarrow +\infty$ and which are related by the Hilbert transform [90],

$$[\widehat{N}^R(\omega)] = \frac{1}{\pi} \text{p.v.} \int_{-\infty}^{+\infty} \frac{1}{\omega - \omega'} [\widehat{N}^I(\omega')] d\omega', \quad (8.66)$$

in which p.v. denotes the Cauchy principal value. The real matrix $[K_0]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$ and can be written as

$$[K_0] = [K(0)] + \frac{2}{\pi} \int_0^{+\infty} [D(\omega)] d\omega = \lim_{|\omega| \rightarrow +\infty} [K(\omega)], \quad (8.67)$$

and consequently, we have the following equation:

$$[K(\omega)] = [K(0)] + \frac{\omega}{\pi} \text{p.v.} \int_{-\infty}^{+\infty} \frac{1}{\omega - \omega'} [D(\omega')] d\omega'. \quad (8.68)$$

9.5.2 Construction of a Nonparametric Stochastic Model

The construction of a nonparametric stochastic model then consists in modeling, for all real ω , the positive-definite symmetric $(n \times n)$ real matrices $[D(\omega)]$ and $[K(\omega)]$ by random matrices $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(\omega)]$ such that

$$E\{[\mathbf{D}(\omega)]\} = [D(\omega)], \quad E\{[\mathbf{K}(\omega)]\} = [K(\omega)], \quad (8.69)$$

$$[\mathbf{D}(-\omega)] = [\mathbf{D}(\omega)], \quad [\mathbf{K}(-\omega)] = [\mathbf{K}(\omega)]. \quad (8.70)$$

For $\omega \geq 0$, the construction of the stochastic model of the family of random matrices $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(\omega)]$ is carried out as follows:

- (i) Constructing the family $[\mathbf{D}(\omega)]$ of random matrices such that, for fixed ω , $[\mathbf{D}(\omega)] = [L_D(\omega)]^T [\mathbf{G}_D] [L_D(\omega)]$, where $[L_D(\omega)]$ is the upper triangular real $(n \times n)$ matrix resulting from the Cholesky decomposition of the positive-definite symmetric real matrix $[D(\omega)] = [L_D(\omega)]^T [L_D(\omega)]$ and where $[\mathbf{G}_D]$ is a $(n \times n)$ random matrix that belongs to ensemble SG_ε^+ , for which the hyperparameter δ is rewritten as δ_D . Hyperparameter δ_D allows the level of uncertainties to be controlled for random matrix $[\mathbf{D}(\omega)]$.
- (ii) Constructing the random matrix $[\mathbf{K}(0)] = [L_{K(0)}]^T [\mathbf{G}_{K(0)}] [L_{K(0)}]$ in which $[L_{K(0)}]$ is the upper triangular real $(n \times n)$ matrix resulting from the Cholesky decomposition of the positive-definite symmetric real matrix $[K(0)] = [L_{K(0)}]^T [L_{K(0)}]$ and where $[\mathbf{G}_{K(0)}]$ is a $(n \times n)$ random matrix that belongs to ensemble SG_ε^+ , for which the hyperparameter δ is rewritten as δ_K . Hyperparameter δ_K allows the level of uncertainties to be controlled for random matrix $[\mathbf{K}(0)]$.
- (iii) For fixed $\omega \geq 0$, constructing the random matrix $[\mathbf{K}(\omega)]$ using the equation,

$$[\mathbf{K}(\omega)] = [\mathbf{K}(0)] + \frac{\omega}{\pi} \text{p.v.} \int_{-\infty}^{+\infty} \frac{1}{\omega - \omega'} [\mathbf{D}(\omega')] d\omega', \quad (8.71)$$

or equivalently,

$$[\mathbf{K}(\omega)] = [\mathbf{K}(0)] + \frac{2\omega^2}{\pi} \text{p.v} \int_0^{+\infty} \frac{1}{\omega^2 - \omega'^2} [\mathbf{D}(\omega')] d\omega'. \quad (8.72)$$

The last equation can also be rewritten as the following equation recommended for computation (because the singularity in $u = 1$ is independent of ω):

$$\begin{aligned} [\mathbf{K}(\omega)] &= [\mathbf{K}(0)] + \frac{2\omega}{\pi} \text{p.v} \int_0^{+\infty} \frac{1}{1-u^2} [\mathbf{D}(\omega u)] du, \\ &= [\mathbf{K}(0)] + \frac{2\omega}{\pi} \lim_{\eta \rightarrow 0} \left\{ \int_0^{1-\eta} + \int_{1+\eta}^{+\infty} \right\}. \end{aligned} \quad (8.73)$$

- (iv) For fixed $\omega < 0$, $[\mathbf{K}(\omega)]$ is calculated using the even property, $[\mathbf{K}(\omega)] = [\mathbf{K}(-\omega)]$. With such a construction, it can be verified that, for all $\omega \geq 0$, $[\mathbf{K}(\omega)]$ is a positive-definite random matrix. The following sufficient condition is proved in [115]. If for all real vector $\mathbf{y} = (y_1, \dots, y_n)$, and if almost surely the random function $\omega \mapsto \langle [\mathbf{D}(\omega)] \mathbf{y}, \mathbf{y} \rangle$ is decreasing in ω for $\omega \geq 0$, then, for all $\omega \geq 0$, $[\mathbf{K}(\omega)]$ is a positive-definite random matrix.

10 MaxEnt as a Numerical Tool for Constructing Ensembles of Random Matrices

In the previous sections, we have presented fundamental ensembles of random matrices constructed with the MaxEnt principle. For these fundamental ensembles the optimization problem defined by Eq. (8.8) has been solved exactly, what has allowed us to explicitly construct the fundamental ensembles of random matrices and also to explicitly describe the generators of realizations. This was possible thanks to the type of the available information that was used to define the admissible set (see Eq. (8.7)). In many cases, the available information does not allow the Lagrange multipliers to be explicitly calculated and, thus, does not allow for solving explicitly the optimization problem defined by Eq. (8.8).

In this framework of the nonexistence of an explicit solution for constructing the pdf of random matrices using the MaxEnt principle under the constraints defined by the available information, the first difficulty consists of the computation of the Lagrange multipliers with an adapted algorithm that must be robust for the high dimension. In addition, the computation of the Lagrange multipliers requires the calculation of integrals in high dimension, which can be estimated only by the Monte Carlo method. Therefore a generator of realizations of the pdf, which is parameterized by the unknown Lagrange multipliers that are currently being calculated, must be constructed. This problem is particularly difficult for the high dimension. An advanced and efficient methodology is presented hereinafter for the case of the high dimension [112] (thus allows also for treating the cases of the small dimension and then for any dimension).

10.1 Available Information and Parameterization

Let $[A]$ be a random matrix defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in any subset \mathbb{S}_n of $\mathbb{M}_n^S(\mathbb{R})$, possibly with $\mathbb{S}_n = \mathbb{M}_n^S(\mathbb{R})$. For instance, \mathbb{S}_n can be $\mathbb{M}_n^+(\mathbb{R})$. Let $p_{[A]}$ be the pdf of $[A]$ with respect to the volume element $d^S A$ on $\mathbb{M}_n^S(\mathbb{R})$ (see Eq. (8.1)). The support, denoted as $\text{supp } p_{[A]}$ of pdf $[A]$, is \mathbb{S}_n . Thus, $p_{[A]}([A]) = 0$ for $[A]$ not in \mathbb{S}_n , and the normalization condition is written as

$$\int_{\mathbb{S}_n} p_{[A]}([A]) d^S A = 1. \quad (8.74)$$

The available information is defined by the following equation on \mathbb{R}^μ :

$$E\{\mathcal{G}([A])\} = \mathbf{f}, \quad (8.75)$$

in which $\mathbf{f} = (f_1, \dots, f_\mu)$ is a given vector in \mathbb{R}^μ with $\mu \geq 1$, where $[A] \mapsto \mathcal{G}([A]) = (\mathcal{G}_1([A]), \dots, \mathcal{G}_\mu([A]))$ is a given mapping from \mathbb{S}_n into \mathbb{R}^μ , and where E is the mathematical expectation. For instance, mapping \mathcal{G} can be defined by the mean value $E[A] = [A]$ in which $[A]$ is a given matrix in \mathbb{S}_n and by the condition $E\{\log(\det[A])\} = c_A$ in which $|c_A| < +\infty$. A parameterization of ensemble \mathbb{S}_n is introduced such that any matrix $[A]$ in \mathbb{S}_n is written as

$$[A] = [\mathcal{A}(\mathbf{y})], \quad (8.76)$$

in which $\mathbf{y} = (y_1, \dots, y_N)$ is a vector in \mathbb{R}^N and where $\mathbf{y} \mapsto [\mathcal{A}(\mathbf{y})]$ is a given mapping from \mathbb{R}^N into \mathbb{S}_n . Let $\mathbf{y} \mapsto \mathbf{g}(\mathbf{y}) = (g_1(\mathbf{y}), \dots, g_\mu(\mathbf{y}))$ be the mapping from \mathbb{R}^N into \mathbb{R}^μ such that

$$\mathbf{g}(\mathbf{y}) = \mathcal{G}([\mathcal{A}(\mathbf{y})]), \quad (8.77)$$

Let $\mathbf{Y} = (Y_1, \dots, Y_N)$ be a \mathbb{R}^N -valued second-order random variable for which the probability distribution on \mathbb{R}^N is represented by the pdf $\mathbf{y} \mapsto p_{\mathbf{Y}}(\mathbf{y})$ from \mathbb{R}^N into $\mathbb{R}^+ = [0, +\infty[$ with respect to $d\mathbf{y} = dy_1 \dots dy_N$. The support of function $p_{\mathbf{Y}}$ is \mathbb{R}^N . Function $p_{\mathbf{Y}}$ satisfies the normalization condition:

$$\int_{\mathbb{R}^N} p_{\mathbf{Y}}(\mathbf{y}) d\mathbf{y} = 1. \quad (8.78)$$

For random vector \mathbf{Y} , the available information is deduced from Eqs. (8.75) to (8.77) and is written as

$$E\{\mathbf{g}(\mathbf{Y})\} = \mathbf{f}. \quad (8.79)$$

10.1.1 Example of Parameterization

If $\mathbb{S}_n = \mathbb{M}_n^+(\mathbb{R})$, then the parameterization, $[A] = [\mathcal{A}(\mathbf{y})]$, of $[A]$ can be constructed in several ways. In order to obtain good properties for the random matrix $[A] = [\mathcal{A}(\mathbf{Y})]$ in which \mathbf{Y} is a \mathbb{R}^N -valued second-order random variable, deterministic matrix $[A]$ is written as

$$[A] = [L_{\underline{A}}]^T (\varepsilon[I_n] + [A_0]) [L_{\underline{A}}],$$

with $\varepsilon > 0$, where $[A_0]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$ and where $[L_{\underline{A}}]$ is the upper triangular $(n \times n)$ real matrix corresponding to the Cholesky factorization $[L_{\underline{A}}]^T [L_{\underline{A}}] = [A]$ of the mean matrix $[A] = E\{\mathbf{A}\}$ that is given in $\mathbb{M}_n^+(\mathbb{R})$. Positive-definite matrix $[A_0]$ can be written in two different forms (inducing different properties for random matrix $[\mathbf{A}]$):

- (i) Exponential-type representation [54, 86]. Matrix $[A_0]$ is written as $[A_0] = \exp_{\mathbb{M}}([G])$ in which the matrix $[G]$ belongs to $\mathbb{M}_n^S(\mathbb{R})$ and where $\exp_{\mathbb{M}}$ denotes the exponential of the symmetric real matrices.
- (ii) Square-type representation [86, 111]. Matrix $[A_0]$ is written as $[A_0] = [L]^T [L]$ in which $[L]$ belongs to the set \mathcal{U}_L of all the upper triangular $(n \times n)$ real matrices with positive diagonal entries and where $[L] = \mathcal{L}([G])$ in which \mathcal{L} is a given mapping from $\mathbb{M}_n^S(\mathbb{R})$ into \mathcal{U}_L .

For this two representations, the parameterization is constructed in taking for \mathbf{y} , the $N = n(n + 1)/2$ independent entries $\{[G]_{jk}, 1 \leq j \leq k \leq n\}$ of symmetric real matrix $[G]$. Then for all \mathbf{y} in \mathbb{R}^N , $[A] = [\mathcal{A}(\mathbf{y})]$ is in \mathbb{S}_n , that is to say, is a positive-definite matrix.

10.2 Construction of the pdf of Random Vector \mathbf{Y} Using the MaxEnt

The unknown pdf $p_{\mathbf{Y}}$ with support \mathbb{R}^N , whose normalization condition is given by Eq. (8.78), is constructed using the MaxEnt principle for which the available information is defined by Eq. (8.79). This construction is detailed in the next Sect. 11.

11 MaxEnt for Constructing the pdf of a Random Vector

Let $\mathbf{Y} = (Y_1, \dots, Y_N)$ be a \mathbb{R}^N -valued second-order random variable for which the probability distribution $P_{\mathbf{Y}}(d\mathbf{y})$ on \mathbb{R}^N is represented by the pdf $\mathbf{y} \mapsto p_{\mathbf{Y}}(\mathbf{y})$ from \mathbb{R}^N into $\mathbb{R}^+ = [0, +\infty[$ with respect to $d\mathbf{y} = dy_1 \dots dy_N$. The support of function $p_{\mathbf{Y}}$ is \mathbb{R}^N . Function $p_{\mathbf{Y}}$ satisfies the normalization condition

$$\int_{\mathbb{R}^N} p_{\mathbf{Y}}(\mathbf{y}) d\mathbf{y} = 1. \quad (8.80)$$

The unknown pdf $p_{\mathbf{Y}}$ is constructed using the MaxEnt principle for which the available information is

$$E\{\mathbf{g}(\mathbf{Y})\} = \mathbf{f}, \quad (8.81)$$

in which $\mathbf{y} \mapsto \mathbf{g}(\mathbf{y}) = (g_1(\mathbf{y}), \dots, g_\mu(\mathbf{y}))$ is a given mapping from \mathbb{R}^N into \mathbb{R}^μ . Equation (8.81) is rewritten as

$$\int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) p_{\mathbf{Y}}(\mathbf{y}) d\mathbf{y} = \mathbf{f}. \quad (8.82)$$

Let \mathcal{C}_p be the set of all the integrable positive-valued functions $\mathbf{y} \mapsto p(\mathbf{y})$ on \mathbb{R}^N , whose support is \mathbb{R}^N . Let \mathcal{C} be the set of all the functions p belonging to \mathcal{C}_p and satisfying the constraints defined by Eqs. (8.80) and (8.82),

$$\mathcal{C} = \left\{ p \in \mathcal{C}_p, \quad \int_{\mathbb{R}^N} p(\mathbf{y}) d\mathbf{y} = 1, \quad \int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) p(\mathbf{y}) d\mathbf{y} = \mathbf{f} \right\}. \quad (8.83)$$

The maximum entropy principle [58] consists in constructing $p_{\mathbf{Y}}$ in \mathcal{C} such that

$$p_{\mathbf{Y}} = \arg \max_{p \in \mathcal{C}} \mathcal{E}(p), \quad (8.84)$$

in which the Shannon entropy $\mathcal{E}(p)$ of p is defined [103] by

$$\mathcal{E}(p) = - \int_{\mathbb{R}^N} p(\mathbf{y}) \log(p(\mathbf{y})) d\mathbf{y}, \quad (8.85)$$

where \log is the Neperian logarithm. In order to solve the optimization problem defined by Eq. (8.84), a Lagrange multiplier $\lambda_0 \in \mathbb{R}^+$ (associated with the constraint defined by Eq. (8.80)) and a Lagrange multiplier $\boldsymbol{\lambda} \in \mathcal{C}_{\boldsymbol{\lambda}} \subset \mathbb{R}^\mu$ (associated with the constraint defined by Eq. (8.82)) are introduced, in which the admissible set $\mathcal{C}_{\boldsymbol{\lambda}}$ is defined by

$$\mathcal{C}_{\boldsymbol{\lambda}} = \{\boldsymbol{\lambda} \in \mathbb{R}^\mu, \quad \int_{\mathbb{R}^N} \exp(-\langle \boldsymbol{\lambda}, \mathbf{g}(\mathbf{y}) \rangle) d\mathbf{y} < +\infty\}. \quad (8.86)$$

The solution of Eq. (8.84) can be written (see the proof in the next section) as

$$p_{\mathbf{Y}}(\mathbf{y}) = c_0^{\text{sol}} \exp(-\langle \boldsymbol{\lambda}^{\text{sol}}, \mathbf{g}(\mathbf{y}) \rangle), \quad \forall \mathbf{y} \in \mathbb{R}^N, \quad (8.87)$$

in which the normalization constant c_0^{sol} is written as $c_0^{\text{sol}} = \exp(-\lambda_0^{\text{sol}})$ and where the method for calculating $(\lambda_0^{\text{sol}}, \boldsymbol{\lambda}^{\text{sol}}) \in \mathbb{R}^+ \times \mathcal{C}_{\boldsymbol{\lambda}}$ is presented in the next two sections.

11.1 Existence and Uniqueness of a Solution to the MaxEnt

The introduction of the Lagrange multipliers λ_0 and $\boldsymbol{\lambda}$ and the analysis of existence and uniqueness of the solution of the MaxEnt corresponding to the solution of the optimization problem defined by Eq. (8.84) are presented hereafter [53].

- The first step of the proof consists in assuming that there exists a unique solution (denoted as p_Y) to the optimization problem defined by Eq. (8.84). The functionals

$$p \mapsto \int_{\mathbb{R}^N} p(\mathbf{y}) d\mathbf{y} - 1 \quad \text{and} \quad p \mapsto \int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) p(\mathbf{y}) d\mathbf{y} - \mathbf{f}, \quad (8.88)$$

are continuously differentiable on \mathcal{C}_p and are assumed to be such that p_Y is a regular point (see p. 187 of [68]). The constraints appearing in set \mathcal{C} are taken into account by using the Lagrange multiplier method. Using the Lagrange multipliers $\lambda_0 \in \mathbb{R}^+$ and $\boldsymbol{\lambda} \in \mathcal{C}_{\lambda}$ defined by Eq. (8.86), the Lagrangian \mathcal{L} can be written, for all p in \mathcal{C}_p , as

$$\mathcal{L}(p; \lambda_0, \boldsymbol{\lambda}) = \mathcal{E}(p) - (\lambda_0 - 1) \left(\int_{\mathbb{R}^N} p(\mathbf{y}) d\mathbf{y} - 1 \right) - \langle \boldsymbol{\lambda}, \int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) p(\mathbf{y}) d\mathbf{y} - \mathbf{f} \rangle. \quad (8.89)$$

From Theorem 2, p. 188, of [68], it can be deduced that there exists $(\lambda_0^{\text{sol}}, \boldsymbol{\lambda}^{\text{sol}})$ such that the functional $(p, \lambda_0, \boldsymbol{\lambda}) \mapsto \mathcal{L}(p; \lambda_0, \boldsymbol{\lambda})$ is stationary at p_Y (given by Eq. (8.87)) for $\lambda_0 = \lambda_0^{\text{sol}}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{sol}}$.

- The second step deals with the explicit construction of a family \mathcal{F}_p of pdf indexed by $(\lambda_0, \boldsymbol{\lambda})$, which renders $p \mapsto \mathcal{L}(p; \lambda_0, \boldsymbol{\lambda})$ extremum. It is further proved that this extremum is unique and turns out to be a maximum. For any $(\lambda_0, \boldsymbol{\lambda})$ fixed in $\mathbb{R}^+ \times \mathcal{C}_{\lambda}$, it can first be deduced from the calculus of variations (Theorem 3.11.16, p. 341, in [101]) that the aforementioned extremum, denoted by $p_{\lambda_0, \boldsymbol{\lambda}}$, is written as

$$p_{\lambda_0, \boldsymbol{\lambda}}(\mathbf{y}) = \exp(-\lambda_0 - \langle \boldsymbol{\lambda}, \mathbf{g}(\mathbf{y}) \rangle), \quad \forall \mathbf{y} \in \mathbb{R}^N. \quad (8.90)$$

For any fixed value of λ_0 in \mathbb{R}^+ and $\boldsymbol{\lambda}$ in \mathcal{C}_{λ} , the uniqueness of this extremum directly follows from the uniqueness of the solution for the Euler equation that is derived from the calculus of variations. Upon calculating the second-order derivative with respect to p , at point $p_{\lambda_0, \boldsymbol{\lambda}}$, of the Lagrangian, it can be shown that this extremum is, indeed, a maximum.

- In a third step, using Eq. (8.90), it is proved that if there exists $(\lambda_0^{\text{sol}}, \boldsymbol{\lambda}^{\text{sol}})$ in $\mathbb{R}^+ \times \mathcal{C}_{\lambda}$ such that the solution of the constraint equations $\int_{\mathbb{R}^N} p_{\lambda_0, \boldsymbol{\lambda}}(\mathbf{y}) d\mathbf{y} = 1$ and $\int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) p_{\lambda_0, \boldsymbol{\lambda}}(\mathbf{y}) d\mathbf{y} = \mathbf{f}$, in $(\lambda_0, \boldsymbol{\lambda})$ and then $(\lambda_0^{\text{sol}}, \boldsymbol{\lambda}^{\text{sol}})$ is unique. These constraints are rewritten as

$$\int_{\mathbb{R}^N} \exp(-\lambda_0 - \langle \boldsymbol{\lambda}, \mathbf{g}(\mathbf{y}) \rangle) d\mathbf{y} = 1. \quad (8.91)$$

$$\int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) \exp(-\lambda_0 - \langle \boldsymbol{\lambda}, \mathbf{g}(\mathbf{y}) \rangle) d\mathbf{y} = \mathbf{f}. \quad (8.92)$$

Introducing the notations,

$$\boldsymbol{\Lambda} = (\lambda_0, \boldsymbol{\lambda}) \text{ and } \boldsymbol{\Lambda}^{\text{sol}} = (\lambda_0^{\text{sol}}, \boldsymbol{\lambda}^{\text{sol}}) \text{ that belong to } \mathcal{C}_{\boldsymbol{\Lambda}} = \mathbb{R}^+ \times \mathcal{C}_{\boldsymbol{\lambda}} \subset \mathbb{R}^{1+\mu},$$

$$\mathbf{F} = (1, \mathbf{f}) \text{ and } \mathbf{G}(\mathbf{y}) = (1, \mathbf{g}(\mathbf{y})) \text{ that belong to } \mathbb{R}^{1+\mu},$$

these constraint equations are written as

$$\int_{\mathbb{R}^N} \mathbf{G}(\mathbf{y}) \exp(-\langle \boldsymbol{\Lambda}, \mathbf{G}(\mathbf{y}) \rangle) d\mathbf{y} = \mathbf{F}. \quad (8.93)$$

It is assumed that the optimization problem stated by Eq. (8.84) is well posed in the sense that the constraints are algebraically independent, that is to say, that there exists a bounded subset \mathcal{S} of \mathbb{R}^N , with $\int_{\mathcal{S}} d\mathbf{y} > 0$, such that for any nonzero vector \mathbf{v} in $\mathbb{R}^{1+\mu}$,

$$\int_{\mathcal{S}} \langle \mathbf{v}, \mathbf{G}(\mathbf{y}) \rangle^2 d\mathbf{y} > 0. \quad (8.94)$$

Let $\boldsymbol{\Lambda} \mapsto H(\boldsymbol{\Lambda})$ be the function defined by

$$H(\boldsymbol{\Lambda}) = \langle \boldsymbol{\Lambda}, \mathbf{F} \rangle + \int_{\mathbb{R}^N} \exp(-\langle \boldsymbol{\Lambda}, \mathbf{G}(\mathbf{y}) \rangle) d\mathbf{y}. \quad (8.95)$$

The gradient $\nabla H(\boldsymbol{\Lambda})$ of $H(\boldsymbol{\Lambda})$ with respect to $\boldsymbol{\Lambda}$ is written as

$$\nabla H(\boldsymbol{\Lambda}) = \mathbf{F} - \int_{\mathbb{R}^N} \mathbf{G}(\mathbf{y}) \exp(-\langle \boldsymbol{\Lambda}, \mathbf{G}(\mathbf{y}) \rangle) d\mathbf{y}, \quad (8.96)$$

so that any solution of $\nabla H(\boldsymbol{\Lambda}) = \mathbf{0}$ satisfies Eq. (8.93) (and conversely). It is assumed that H admits at least one critical point. The Hessian matrix $[H''(\boldsymbol{\Lambda})]$ is written as

$$[H''(\boldsymbol{\Lambda})] = \int_{\mathbb{R}^N} \mathbf{G}(\mathbf{y}) \otimes \mathbf{G}(\mathbf{y}) \exp(-\langle \boldsymbol{\Lambda}, \mathbf{G}(\mathbf{y}) \rangle) d\mathbf{y}. \quad (8.97)$$

Since $\mathcal{S} \subset \mathbb{R}^N$, it turns out that, for any nonzero vector \mathbf{v} in $\mathbb{R}^{1+\mu}$,

$$\langle [H''(\boldsymbol{\Lambda})]\mathbf{v}, \mathbf{v} \rangle \geq \int_{\mathcal{S}} \langle \mathbf{v}, \mathbf{G}(\mathbf{y}) \rangle^2 \exp(-\langle \boldsymbol{\Lambda}, \mathbf{G}(\mathbf{y}) \rangle) d\mathbf{y} > 0, \quad (8.98)$$

Therefore, function $\boldsymbol{\Lambda} \mapsto H(\boldsymbol{\Lambda})$ is strictly convex that ensures the uniqueness of the critical point of H (should it exist). Under the aforementioned assumption of algebraic independence for the constraints, it follows that if $\boldsymbol{\Lambda}^{\text{sol}}$ (such that

the constraint defined by Eq. (8.93) is fulfilled) exists, then Λ^{sol} is unique and corresponds to the solution of the following optimization problem:

$$\Lambda^{\text{sol}} = \arg \min_{\Lambda \in \mathcal{C}_\Lambda} H(\Lambda), \quad (8.99)$$

where H is the strictly convex function defined by Eq. (8.95). The unique solution p_Y of the optimization problem defined by Eq. (8.84) is given by Eq. (8.87) with $(\lambda_0^{\text{sol}}, \lambda^{\text{sol}}) = \Lambda^{\text{sol}}$.

11.2 Numerical Calculation of the Lagrange Multipliers

When there is no explicit solution $(\lambda_0^{\text{sol}}, \lambda^{\text{sol}}) = \Lambda^{\text{sol}}$ of Eq. (8.93) in Λ , Λ^{sol} must be numerically calculated and the numerical method used must be robust for the high dimension. The numerical method could be based on the optimization problem defined by Eq. (8.99). Unfortunately, with such a formulation, the constant of normalization, $c_0 = \exp(-\lambda_0)$, is directly involved in the numerical calculations, what is not robust in high dimension. The numerical method proposed hereinafter [11] is based on the minimization of the convex objective function introduced in [1]. Using Eqs. (8.80) and (8.87), pdf p_Y can be rewritten as

$$p_Y(\mathbf{y}) = c_0(\lambda^{\text{sol}}) \exp(- < \lambda^{\text{sol}}, \mathbf{g}(\mathbf{y}) >), \quad \forall \mathbf{y} \in \mathbb{R}^N, \quad (8.100)$$

in which $c_0(\lambda)$ is defined by

$$c_0(\lambda) = \left\{ \int_{\mathbb{R}^N} \exp(- < \lambda, \mathbf{g}(\mathbf{y}) >) d\mathbf{y} \right\}^{-1}. \quad (8.101)$$

Since $\exp(-\lambda_0) = c_0(\lambda_0)$, and taking into account Eq. (8.101), the constraint equation defined by Eq. (8.92) can be rewritten as

$$\int_{\mathbb{R}^N} \mathbf{g}(\mathbf{y}) c_0(\lambda) \exp(- < \lambda, \mathbf{g}(\mathbf{y}) >) d\mathbf{y} = \mathbf{f}. \quad (8.102)$$

The optimization problem defined by Eq. (8.99), which allows for calculating $(\lambda_0^{\text{sol}}, \lambda^{\text{sol}}) = \Lambda^{\text{sol}}$, is replaced by the more convenient optimization problem that allows λ^{sol} to be computed,

$$\lambda^{\text{sol}} = \arg \min_{\lambda \in \mathcal{C}_\lambda \subset \mathbb{R}^\mu} \Gamma(\lambda), \quad (8.103)$$

in which the objective function Γ is defined by

$$\Gamma(\lambda) = < \lambda, \mathbf{f} > - \log(c_0(\lambda)). \quad (8.104)$$

Once λ^{sol} is calculated, c_0^{sol} is given by $c_0^{\text{sol}} = c_0(\lambda^{\text{sol}})$. Let $\{\mathbf{Y}_\lambda, \lambda \in \mathcal{C}_\lambda\}$ be the family of random variables with values in \mathbb{R}^N , for which pdf $p_{\mathbf{Y}_\lambda}$ is defined, for all λ in \mathcal{C}_λ , by

$$p_{\mathbf{Y}_\lambda}(\mathbf{y}) = c_0(\lambda) \exp(- < \lambda, \mathbf{g}(\mathbf{y}) >), \quad \forall \mathbf{y} \in \mathbb{R}^N. \quad (8.105)$$

The gradient vector $\nabla \Gamma(\lambda)$ and the Hessian matrix $[\Gamma''(\lambda)]$ of function $\lambda \mapsto \Gamma(\lambda)$ can be written as

$$\nabla \Gamma(\lambda) = \mathbf{f} - E\{\mathbf{g}(\mathbf{Y}_\lambda)\}, \quad (8.106)$$

$$[\Gamma''(\lambda)] = E\{\mathbf{g}(\mathbf{Y}_\lambda) \mathbf{g}(\mathbf{Y}_\lambda)^T\} - E\{\mathbf{g}(\mathbf{Y}_\lambda)\} E\{\mathbf{g}(\mathbf{Y}_\lambda)\}^T. \quad (8.107)$$

Matrix $[\Gamma''(\lambda)]$ is thus the covariance matrix of the random vector $\mathbf{g}(\mathbf{Y}_\lambda)$ and is positive definite (the constraints have been assumed to be algebraically independent). Consequently, function $\lambda \mapsto \Gamma(\lambda)$ is strictly convex and reaches its minimum for λ^{sol} which is such that $\nabla \Gamma(\lambda^{\text{sol}}) = \mathbf{0}$. The optimization problem defined by Eq. (8.103) can be solved using any minimization algorithm. Since function Γ is strictly convex, the Newton iterative method can be applied to the increasing function $\lambda \mapsto \nabla \Gamma(\lambda)$ for searching λ^{sol} such that $\nabla \Gamma(\lambda^{\text{sol}}) = \mathbf{0}$. This iterative method is not unconditionally convergent. Consequently, an under-relaxation is introduced and the iterative algorithm is written as

$$\lambda^{\ell+1} = \lambda^\ell - \alpha [\Gamma''(\lambda^\ell)]^{-1} \nabla \Gamma(\lambda^\ell), \quad (8.108)$$

in which α belongs to $[0, 1]$ in order to ensure the convergence. At each iteration ℓ , the error is calculated by

$$\text{err}(\ell) = \frac{\|\mathbf{f} - E\{\mathbf{g}(\mathbf{Y}_{\lambda^\ell})\}\|}{\|\mathbf{f}\|} = \frac{\|\nabla \Gamma(\lambda^\ell)\|}{\|\mathbf{f}\|}, \quad (8.109)$$

in order to control the convergence. The performance of the algorithm depends on the choice of the initial condition that can be found in [11]. For high dimension problem, the mathematical expectations appearing in Eqs. (8.106), (8.107), and (8.109) are calculated using a Markov chain Monte Carlo (MCMC) method that does not require the calculation of the normalization constant $c_0(\lambda)$ in the pdf defined by Eq. (8.105).

11.3 Generator for Random Vector \mathbf{Y}_λ and Estimation of the Mathematical Expectations in High Dimension

For λ fixed in $\mathcal{C}_\lambda \subset \mathbb{R}^\mu$, the pdf $p_{\mathbf{Y}_\lambda}$ on \mathbb{R}^N of the \mathbb{R}^N -valued random variable \mathbf{Y}_λ is defined by Eq. (8.105). Let w be a given mapping from \mathbb{R}^N into an Euclidean space such that $E\{w(\mathbf{Y}_\lambda)\} = \int_{\mathbb{R}^N} w(\mathbf{y}) p_{\mathbf{Y}_\lambda} d\mathbf{y}$ is finite. For instance, w can be

such that $w(\mathbf{Y}_\lambda) = \mathbf{g}(\mathbf{Y}_\lambda)$ or $w(\mathbf{Y}_\lambda) = \mathbf{g}(\mathbf{Y}_\lambda)\mathbf{g}(\mathbf{Y}_\lambda)^T$. These two choices allow for calculating the mathematical expectation in high dimension, $E\{\mathbf{g}(\mathbf{Y}_\lambda)\}$ and $E\{\mathbf{g}(\mathbf{Y}_\lambda)\mathbf{g}(\mathbf{Y}_\lambda)^T\}$, which are required for computing the gradient and the Hessian defined by Eqs. (8.106) and (8.107).

The estimation of $E\{w(\mathbf{Y}_\lambda)\}$ requires a generator of realizations of random vector \mathbf{Y}_λ , which is constructed using the Markov chain Monte Carlo method (MCMC) [59, 95, 117]. With the MCMC method, the transition kernel of the homogeneous Markov chain can be constructed using the Metropolis-Hastings algorithm [57, 75] (that requires the definition of a good proposal distribution), the Gibbs sampling [42] (that requires the knowledge of the conditional distribution), or the slice sampling [83] (that can exhibit difficulties related to the general shape of the probability distribution, in particular for multimodal distributions). In general, these algorithms are efficient, but can also be not efficient if there exist attraction regions which do not correspond to the invariant measure under consideration and tricky even in high dimension. These cases cannot easily be detected and are time consuming.

We refer the reader to the references given hereinbefore for the usual MCMC methods, and we present after a more advanced method that is very robust in high dimension, which have been introduced in [112] and used, for instance, in [11, 51]. The method presented looks like to the Gibbs approach but corresponds to a more direct construction of a random generator of realizations for random variable \mathbf{Y}_λ whose probability distribution is $p_{\mathbf{Y}_\lambda} d\mathbf{y}$. The difference between the Gibbs algorithm and the proposed algorithm is that the convergence in the proposed method can be studied with all the mathematical results concerning the existence and uniqueness of Itô stochastic differential equation (ISDE). In addition, a parameter is introduced which allows the transient part of the response to be killed in order to get more rapidly the stationary solution corresponding to the invariant measure. Thus, the construction of the transition kernel by using the detailed balance equation is replaced by the construction of an ISDE, which admits $p_{\mathbf{Y}_\lambda} d\mathbf{y}$ (defined by Eq. (8.105)) as a unique invariant measure. The ergodic method or the Monte Carlo method is used for estimating $E\{w(\mathbf{Y}_\lambda)\}$.

11.3.1 Random Generator and Estimation of Mathematical Expectations

It is assumed that λ is fixed in $\mathcal{C}_\lambda \subset \mathbb{R}^\mu$, and for simplifying the notation, λ is omitted. Let $\mathbf{u} \mapsto \Phi(\mathbf{u})$ be the function from \mathbb{R}^N into \mathbb{R} defined by

$$\Phi(\mathbf{u}) = \langle \lambda, \mathbf{g}(\mathbf{u}) \rangle, \quad (8.110)$$

Let $\{(\mathbf{U}(r), \mathbf{V}(r)), r \in \mathbb{R}^+\}$ be the Markov stochastic process defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by $\mathbb{R}^+ = [0, +\infty[$, with values in $\mathbb{R}^N \times \mathbb{R}^N$, satisfying, for all $r > 0$, the following ISDE with initial conditions:

$$d\mathbf{U}(r) = \mathbf{V}(r) dr, \quad (8.111)$$

$$d\mathbf{V}(r) = -\nabla_{\mathbf{u}}\Phi(\mathbf{U}(r)) dr - \frac{1}{2}f_0\mathbf{V}(r) dr + \sqrt{f_0} d\mathbf{W}(r), \quad (8.112)$$

$$\mathbf{U}(0) = \mathbf{u}_0, \quad \mathbf{V}(0) = \mathbf{v}_0 \quad a.s., \quad (8.113)$$

in which \mathbf{u}_0 and \mathbf{v}_0 are given vectors in \mathbb{R}^N (that will be taken as zero in the application presented later) and where $\mathbf{W} = (W_1, \dots, W_N)$ is the normalized Wiener process defined on $(\Theta, \mathcal{T}, \mathcal{P})$ indexed by \mathbb{R}^+ with values in \mathbb{R}^N . The matrix-valued autocorrelation function $[R_{\mathbf{W}}(r, r')] = E\{\mathbf{W}(r)\mathbf{W}(r')^T\}$ of \mathbf{W} is then written as $[R_{\mathbf{W}}(r, r')] = \min(r, r') [I_n]$. In Eq. (8.112), the free parameter $f_0 > 0$ allows a dissipation term to be introduced in the nonlinear second-order dynamical system (formulated in the Hamiltonian form with an additional dissipative term) in order to kill the transient part of the response and consequently to get more rapidly the stationary solution corresponding to the invariant measure. It is assumed that function \mathbf{g} is such that function $\mathbf{u} \mapsto \Phi(\mathbf{u})$ (i) is continuous on \mathbb{R}^N , (ii) is such that $\mathbf{u} \mapsto \|\nabla_{\mathbf{u}}\Phi(\mathbf{u})\|$ is a locally bounded function on \mathbb{R}^N (i.e., is bounded on all compact set in \mathbb{R}^N), and (iii) is such that

$$\inf_{\|\mathbf{u}\| > R} \Phi(\mathbf{u}) \rightarrow +\infty \quad \text{if} \quad R \rightarrow +\infty, \quad (8.114)$$

$$\inf_{\mathbf{u} \in \mathbb{R}^n} \Phi(\mathbf{u}) = \Phi_{\min} \quad \text{with} \quad \Phi_{\min} \in \mathbb{R}, \quad (8.115)$$

$$\int_{\mathbb{R}^n} \|\nabla_{\mathbf{u}}\Phi(\mathbf{u})\| e^{-\Phi(\mathbf{u})} d\mathbf{u} < +\infty. \quad (8.116)$$

Under hypotheses (i) to (iii), and using Theorems 4 to 7 in pages 211 to 216 of Ref. [105], in which the Hamiltonian is taken as $\mathbb{H}(\mathbf{u}, \mathbf{v}) = \|\mathbf{v}\|^2/2 + \Phi(\mathbf{u})$, and using [33, 62] for the ergodic property, it can be deduced that the problem defined by Eqs. (8.111), (8.112), and (8.113) admits a unique solution. This solution is a second-order diffusion stochastic process $\{(\mathbf{U}(r), \mathbf{V}(r)), r \in \mathbb{R}^+\}$, which converges to a stationary and ergodic diffusion stochastic process $\{(\mathbf{U}_{st}(r_{st}), \mathbf{V}_{st}(r_{st})), r_{st} \geq 0\}$, when r goes to infinity, associated with the invariant probability measure $P_{st}(d\mathbf{u}, d\mathbf{v}) = \rho_{st}(\mathbf{u}, \mathbf{v}) d\mathbf{u} d\mathbf{v}$. The probability density function $(\mathbf{u}, \mathbf{v}) \mapsto \rho_{st}(\mathbf{u}, \mathbf{v})$ on $\mathbb{R}^N \times \mathbb{R}^N$ is the unique solution of the steady-state Fokker-Planck equation associated with Eqs. (8.111)–(8.112) and is written (see pp. 120 to 123 in [105]), as

$$\rho_{st}(\mathbf{u}, \mathbf{v}) = c_N \exp \left\{ -\frac{1}{2} \|\mathbf{v}\|^2 - \Phi(\mathbf{u}) \right\}, \quad (8.117)$$

in which c_N is the constant of normalization. Equations (8.105), (8.110), and (8.117) yield

$$p_{\mathbf{Y}_{\lambda}}(\mathbf{y}) = \int_{\mathbb{R}^N} \rho_{st}(\mathbf{y}, \mathbf{v}) d\mathbf{v}, \quad \forall \mathbf{y} \in \mathbb{R}^N. \quad (8.118)$$

Random variable \mathbf{Y}_λ (for which the pdf $p_{\mathbf{Y}_\lambda}$ is defined by Eq. (8.105)) can then be written, for all fixed positive value of r_{st} , as

$$\mathbf{Y}_\lambda = \mathbf{U}_{\text{st}}(r_{\text{st}}) = \lim_{r \rightarrow +\infty} \mathbf{U}(r) \quad \text{in probability distribution.} \quad (8.119)$$

The free parameter $f_0 > 0$ introduced in Eq. (8.112) allows a dissipation term to be introduced in the nonlinear dynamical system for obtaining more rapidly the asymptotic behavior corresponding to the stationary and ergodic solution associated with the invariant measure. Using Eq. (8.119) and the ergodic property of stationary stochastic process \mathbf{U}_{st} yield

$$E\{w(\mathbf{Y}_\lambda)\} = \lim_{R \rightarrow +\infty} \frac{1}{R} \int_0^R w(\mathbf{U}(r, \theta)) dr, \quad (8.120)$$

in which, for $\theta \in \Theta$, $\mathbf{U}(\cdot, \theta)$ is any realization of \mathbf{U} .

11.3.2 Discretization Scheme and Estimating the Mathematical Expectations

A discretization scheme must be used for numerically solving Eqs. (8.111), (8.112), and (8.113). For general surveys on discretization schemes for ISDE, we refer the reader to [63, 118, 119] (among others). The present case, related to a Hamiltonian dynamical system, has also been analyzed using an implicit Euler scheme in [120]. Hereinafter, we present the Störmer-Verlet scheme, which is an efficient scheme that preserves energy for nondissipative Hamiltonian dynamical systems (see [56] for reviews about this scheme in the deterministic case, and see [17] and the references therein for the stochastic case).

Let $M \geq 1$ be an integer. The ISDE defined by Eqs. (8.111), (8.112), and (8.113) is solved on the finite interval $\mathcal{R} = [0, (M - 1) \Delta r]$, in which Δr is the sampling step of the continuous index parameter r . The integration scheme is based on the use of the M sampling points r_k such that $r_k = (k - 1) \Delta r$ for $k = 1, \dots, M$. The following notations are introduced: $\mathbf{U}^k = \mathbf{U}(r_k)$, $\mathbf{V}^k = \mathbf{V}(r_k)$, and $\mathbf{W}^k = \mathbf{W}(r_k)$, for $k = 1, \dots, M$, with $\mathbf{U}^1 = \mathbf{u}_0$, $\mathbf{V}^1 = \mathbf{v}_0$, and $\mathbf{W}^1 = \mathbf{0}$. Let $\{\Delta \mathbf{W}^{k+1} = \mathbf{W}^{k+1} - \mathbf{W}^k, k = 1, \dots, M-1\}$ be the family of the independent Gaussian second-order centered \mathbb{R}^N -valued random variables such that $E\{\Delta \mathbf{W}^{k+1} (\Delta \mathbf{W}^{k+1})^T\} = \Delta r [I_n]$. For $k = 1, \dots, M - 1$, the Störmer-Verlet scheme yields

$$\mathbf{U}^{k+\frac{1}{2}} = \mathbf{U}^k + \frac{\Delta r}{2} \mathbf{V}^k, \quad (8.121)$$

$$\mathbf{V}^{k+1} = \frac{1-b}{1+b} \mathbf{V}^k + \frac{\Delta r}{1+b} \mathbf{L}^{k+\frac{1}{2}} + \frac{\sqrt{f_0}}{1+b} \Delta \mathbf{W}^{k+1}, \quad (8.122)$$

$$\mathbf{U}^{k+1} = \mathbf{U}^{k+\frac{1}{2}} + \frac{\Delta r}{2} \mathbf{V}^{k+1}, \quad (8.123)$$

with the initial condition defined by (8.113), where $b = f_0 \Delta r / 4$ and where $\mathbf{L}^{k+\frac{1}{2}}$ is the \mathbb{R}^N -valued random variable such that $\mathbf{L}^{k+\frac{1}{2}} = -\{\nabla_{\mathbf{u}} \Phi(\mathbf{u})\}_{\mathbf{u}=\mathbf{U}^{k+\frac{1}{2}}}.$

For a given realization θ in Θ , the sequence $\{\mathbf{U}^k(\theta), k = 1, \dots, M\}$ is constructed using Eqs. (8.121), (8.122), and (8.123). The discretization of Eq. (8.120) yields the following estimation of the mathematical expectation:

$$E\{w(\mathbf{Y}_\lambda)\} = \lim_{M \rightarrow +\infty} \hat{w}_M, \quad \hat{w}_M = \frac{1}{M - M_0 + 1} \sum_{k=M_0}^M w(\mathbf{U}^k(\theta)), \quad (8.124)$$

in which, for f_0 fixed, the integer $M_0 > 1$ is chosen to remove the transient part of the response induced by the initial condition.

For details concerning the optimal choice of the numerical parameters, such as M_0 , M , f_0 , Δ_r , \mathbf{u}_0 , and \mathbf{v}_0 , we refer the reader to [11, 51, 54, 112].

12 Nonparametric Stochastic Model For Constitutive Equation in Linear Elasticity

This section deals with a nonparametric stochastic model for random elasticity matrices in the framework of the three-dimensional linear elasticity in continuum mechanics, using the methodologies and some of the results that have been given in the two previous sections: “Fundamental Ensembles for Positive-Definite Symmetric Real Random Matrices” and “MaxEnt as a Numerical Tool for Constructing Ensemble of Random Matrices.” The developments given hereinafter correspond to a synthesis of works detailed in [51, 53, 54].

From a continuum mechanics point of view, the framework is the 3D linear elasticity of a homogeneous random medium (material) at a given scale. Let $[\tilde{\mathbf{C}}]$ be the random elasticity matrix for which the nonparametric stochastic model has to be derived. Random matrix $[\tilde{\mathbf{C}}]$ is defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$ and is with values in $\mathbb{M}_n^+(\mathbb{R})$ with $n = 6$. This matrix corresponds to the so-called Kelvin’s matrix representation of the fourth-order symmetric elasticity tensor in 3D linear elasticity [71]. The symmetry classes for a linear elastic material, that is to say, the linear elastic symmetries, are [23] isotropic, cubic, transversely isotropic, trigonal, tetragonal, orthotropic, monoclinic, and anisotropic. From a stochastic modeling point of view, the random elasticity matrix $[\tilde{\mathbf{C}}]$ satisfies the following properties:

- (i) Random matrix $[\tilde{\mathbf{C}}]$ is assumed to have a mean value that belongs to $\mathbb{M}_n^+(\mathbb{R})$, but is, in mean, close to a given symmetry class induced by a material symmetry, denoted as $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ and which is a subset of $\mathbb{M}_n^+(\mathbb{R})$,

$$[\tilde{\mathbf{C}}] = E\{[\tilde{\mathbf{C}}]\} \in \mathbb{M}_n^+(\mathbb{R}). \quad (8.125)$$

- (ii) Random matrix $[\tilde{\mathbf{C}}]$ admits a positive-definite lower bound $[C_\ell]$ belonging to $\mathbb{M}_n^+(\mathbb{R})$

$$[\tilde{\mathbf{C}}] - [C_\ell] > 0 \quad a.s. \quad (8.126)$$

- (iii) The statistical fluctuations of random elasticity matrix $[\tilde{\mathbf{C}}]$ belong mainly to the symmetry class, but can be more or less anisotropic with respect to the above symmetry. The level of statistical fluctuations in the symmetry class must be controlled independently of the level of statistical anisotropic fluctuations.

12.1 Positive-Definite Matrices Having a Symmetry Class

For the positive-definite symmetric ($n \times n$) real matrices, a given symmetry class is defined by a subset $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$ such that any matrix $[M]$ exhibiting the above symmetry then belongs to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ and can be written as

$$[M] = \sum_{j=1}^N m_j [E_j^{\text{sym}}], \quad \mathbf{m} = (m_1, \dots, m_N) \in \mathcal{C}_{\mathbf{m}} \subset \mathbb{R}^N, \quad [E_j^{\text{sym}}] \in \mathbb{M}_n^S(\mathbb{R}), \quad (8.127)$$

in which $\{[E_j^{\text{sym}}], j = 1, \dots, N\}$ is the matrix algebraic basis of $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (Walpole's tensor basis [122]) and where the admissible subset $\mathcal{C}_{\mathbf{m}}$ of \mathbb{R}^N is such that

$$\mathcal{C}_{\mathbf{m}} = \{\mathbf{m} \in \mathbb{R}^N \mid \sum_{j=1}^N m_j [E_j^{\text{sym}}] \in \mathbb{M}_n^+(\mathbb{R})\}. \quad (8.128)$$

It should be noted that the basis matrices $[E_j^{\text{sym}}]$ are symmetric matrices belonging to $\mathbb{M}_n^S(\mathbb{R})$, but are not positive definite, that is to say, do not belong to $\mathbb{M}_n^+(\mathbb{R})$. The dimension N for all material symmetry classes is 2 for isotropic, 3 for cubic, 5 for transversely isotropic, 6 or 7 for trigonal, 6 or 7 for tetragonal, 9 for orthotropic, 13 for monoclinic, and 21 for anisotropic. The following properties are proved (see [54, 122]):

- (i) If $[M]$ and $[M']$ belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$, then

$$[M][M'] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}), \quad [M]^{-1} \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}), \quad [M]^{1/2} \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}). \quad (8.129)$$

- (ii) Any matrix $[N]$ belonging to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ can be written as

$$[N] = \exp_{\mathbb{M}}([\mathcal{N}]), \quad [\mathcal{N}] = \sum_{j=1}^N y_j [E_j^{\text{sym}}], \quad \mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N, \quad (8.130)$$

in which $\exp_{\mathbb{M}}$ is the exponential of symmetric real matrices. It should be noted that matrix $[\mathcal{N}]$ is a symmetric real matrix but does not belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (because \mathbf{y} is in \mathbb{R}^N and therefore $[\mathcal{N}]$ is not a positive-definite matrix).

12.2 Representation Introducing a Positive-Definite Lower Bound

Using Eq. (8.126), the representation of random elasticity matrix $[\tilde{\mathbf{C}}]$ is written as

$$[\tilde{\mathbf{C}}] = [C_\ell] + [\mathbf{C}], \quad (8.131)$$

in which the lower bound is the deterministic matrix $[C_\ell]$ belonging to $\mathbb{M}_n^+(\mathbb{R})$ and where $[\mathbf{C}] = [\tilde{\mathbf{C}}] - [C_\ell]$ is a random matrix with values in $\mathbb{M}_n^+(\mathbb{R})$. The mean value $[\underline{\mathbf{C}}] = E\{[\mathbf{C}]\}$ of $[\mathbf{C}]$ is written as

$$[\underline{\mathbf{C}}] = [\tilde{\mathbf{C}}] - [C_\ell] \in \mathbb{M}_n^+(\mathbb{R}), \quad (8.132)$$

in which $[\tilde{\mathbf{C}}]$ is defined by Eq. (8.125). Such a lower bound can be defined in two ways:

- (1) If some microstructural information is available, $[C_\ell]$ may be computed, either by using some well-known micromechanics-based bounds (such as the Reuss bound, for heterogeneous materials made up with ordered phases with deterministic properties) or by using a numerical approximation based on the realizations of the stochastic lower bound obtained from computational homogenization and invoking the Huet partition theorem (see the discussion in [49]).
- (2) In the absence of such information, a simple a priori expression for $[C_\ell]$ can be obtained as $[C_\ell] = \epsilon[\tilde{\mathbf{C}}]$ with $0 \leq \epsilon < 1$, from which it can be deduced that $[\underline{\mathbf{C}}] = (1 - \epsilon)[\tilde{\mathbf{C}}] > 0$.

12.3 Introducing Deterministic Matrices $[\underline{\mathbf{A}}]$ and $[\underline{\mathbf{S}}]$

Let $[\underline{\mathbf{A}}]$ be the deterministic matrix in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ defined by

$$[\underline{\mathbf{A}}] = P^{\text{sym}}([\underline{\mathbf{C}}]), \quad (8.133)$$

in which $[\underline{\mathbf{C}}] \in \mathbb{M}_n^+(\mathbb{R})$ is defined by Eq. (8.132) and where P^{sym} is the projection operator from $\mathbb{M}_n^+(\mathbb{R})$ on $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$.

- (i) For a given symmetry class with $N < 21$, if there is no anisotropic statistical fluctuations, then the mean matrix $[\underline{\mathbf{C}}]$ belongs to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ and consequently $[\underline{\mathbf{A}}]$ is equal to $[\underline{\mathbf{C}}]$.

- (ii) If the class of symmetry is anisotropic (thus $N = 21$), then $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ coincides with $\mathbb{M}_n^+(\mathbb{R})$ and again $[\underline{A}]$ is equal to the mean matrix $[\underline{C}]$ that belongs to $\mathbb{M}_n^+(\mathbb{R})$.
- (iii) In general, for a given symmetry class with $N < 21$, and due to the presence of anisotropic statistical fluctuations, the mean matrix $[\underline{C}]$ of random matrix $[\mathbf{C}]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$ but does not belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. For this case, an invertible deterministic $(n \times n)$ real matrix $[\underline{S}]$ is introduced such that

$$[\underline{C}] = [\underline{S}]^T [\underline{A}] [\underline{S}]. \quad (8.134)$$

The construction of $[\underline{S}]$ is performed as follows. Let $[L_{\underline{C}}]$ and $[L_{\underline{A}}]$ be the upper triangular real matrices with positive diagonal entries resulting from the Cholesky factorization of matrices $[\underline{C}]$ and $[\underline{A}]$,

$$[\underline{C}] = [L_{\underline{C}}]^T [L_{\underline{C}}], \quad [\underline{A}] = [L_{\underline{A}}]^T [L_{\underline{A}}]. \quad (8.135)$$

Therefore, the matrix $[\underline{S}]$ is written as

$$[\underline{S}] = [L_{\underline{A}}]^{-1} [L_{\underline{C}}]. \quad (8.136)$$

It should be noted that for cases (i) and (ii), Eq. (8.136) shows that $[\underline{S}] = [I_n]$.

12.4 Nonparametric Stochastic Model for $[\mathbf{C}]$

In order that the statistical fluctuations of random matrix $[\mathbf{C}]$ belong mainly to the symmetry class $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ and exhibit more or less some anisotropic fluctuations around this symmetry class, and in order that the level of statistical fluctuations in the symmetry class is controlled independently of the level of statistical anisotropic fluctuation, the use of the nonparametric method leads us to introduce the following representation:

$$[\mathbf{C}] = [\underline{S}]^T [\mathbf{A}]^{1/2} [\mathbf{G}_0] [\mathbf{A}]^{1/2} [\underline{S}], \quad (8.137)$$

in which:

- (1) The deterministic $(n \times n)$ real matrix $[\underline{S}]$ is defined by Eq. (8.136).
- (2) $[\mathbf{G}_0]$ belongs to ensemble SG_0^+ of random matrices and models the anisotropic statistical fluctuations. The mean value of random matrix $[\mathbf{G}_0]$ is matrix $[I_n]$ (see Eq. (8.13)). The level of the statistical fluctuations of $[\mathbf{G}_0]$ is controlled by the hyperparameter δ defined by Eq. (8.15).
- (3) The random matrix $[\mathbf{A}]^{1/2}$ is the square root of a random matrix $[\mathbf{A}]$ with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$, which models the statistical fluctuations in the given

symmetry class and which is statistically independent of random matrix $[G_0]$. The mean value of random matrix $[A]$ is the matrix $[\underline{A}]$ defined by Eq. (8.133),

$$E\{[A]\} = [\underline{A}] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R}). \quad (8.138)$$

The level of the statistical fluctuations of $[A]$ is controlled by the coefficient of variation δ_A defined by

$$\delta_A = \left\{ \frac{E\{\|A - \underline{A}\|_F^2\}}{\|\underline{A}\|_F^2} \right\}^{1/2}. \quad (8.139)$$

Taking into account the statistical independence of A and G_0 and taking the mathematical expectation of the two members of Eq. (8.137) yield Eq. (8.134).

12.4.1 Remarks Concerning the Control of the Statistical Fluctuations and the Limit Cases

- (1) For a given symmetry class with $N < 21$, if the level of anisotropic statistical fluctuations goes to zero, that is to say, if $\delta \rightarrow 0$ what implies that $[G_0]$ goes to $[I_n]$ (in probability distribution) and implies that $[\underline{A}]$ goes to $[\underline{C}]$ and thus $[\underline{S}]$ goes to $[I_n]$, then Eq. (8.137) shows that $[C]$ goes to $[A]$ (in probability distribution), which is a random matrix with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$.
- (2) If the given symmetry class is anisotropic ($N = 21$) and $\delta_A \rightarrow 0$, then $[\underline{A}]$ goes to the mean matrix $[\underline{C}]$, $[\underline{S}]$ goes to $[I_n]$, and $[A]$ goes to $[\underline{A}]$ that goes to $[\underline{C}]$ (in probability distribution). Then $[C]$ goes to $[\underline{C}]^{1/2} [G_0] [\underline{C}]^{1/2}$, which is the full anisotropic nonparametric stochastic modeling of $[C]$.

12.5 Construction of $[A]$ Using the MaxEnt Principle

In this section, random matrix $[A]$ that allows for describing the statistical fluctuations in the class of symmetry $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ with $N < 21$ is constructed using the MaxEnt principle and, in particular, using all the results and notations introduced in Sect. 10.

12.5.1 Defining the Available Information

Let $p_{[A]}$ be the unknown pdf of random matrix $[A]$, with respect to volume element $d^S A$ on $\mathbb{M}_n^S(\mathbb{R})$ (see Eq. (8.1)), with values in the given symmetry class $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R})$ with $N < 21$. The support, $\text{supp } p_{[A]}$, is the subset $\mathbb{S}_n = \mathbb{M}_n^{\text{sym}}(\mathbb{R})$, and the normalization condition is given by Eq. (8.74). The available information is defined by

$$E[A] = [\underline{A}], \quad E\{\log(\det[A])\} = c_A, \quad |c_A| < +\infty, \quad (8.140)$$

in which $[\underline{A}]$ is the matrix in \mathbb{S}_n , defined by Eq. (8.133), and where the second available information is introduced in order that pdf $[A] \mapsto p_{[\underline{A}]}([A])$ decreases toward zero when $\|A\|_F$ goes to zero. The constant c_A that has no physical meaning is re-expressed as a function of the hyperparameter δ_A defined by Eq. (8.139). This available information defines the vector $\mathbf{f} = (f_1, \dots, f_\mu)$ in \mathbb{R}^μ with $\mu = n(n+1)/2 + 1$ and defines the mapping $[A] \mapsto \mathcal{G}([A]) = (\mathcal{G}_1([A]), \dots, \mathcal{G}_\mu([A]))$ from \mathbb{S}_n into \mathbb{R}^μ , such that (see Eq. (8.75))

$$E\{\mathcal{G}([\mathbf{A}])\} = \mathbf{f}. \quad (8.141)$$

12.5.2 Defining the Parameterization

The objective is to construct the parameterization of ensemble $\mathbb{S}_n = \mathbb{M}_n^{\text{sym}}(\mathbb{R})$, such that any matrix $[A]$ in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ is written (see Eq. (8.76)) as

$$[A] = [\mathcal{A}(\mathbf{y})], \quad (8.142)$$

in which $\mathbf{y} = (y_1, \dots, y_N)$ is a vector in \mathbb{R}^N and where $\mathbf{y} \mapsto [\mathcal{A}(\mathbf{y})]$ is a given mapping from \mathbb{R}^N into $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. Let $[\underline{A}]^{1/2}$ be the square root of matrix $[\underline{A}] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$ that is defined by Eq. (8.133). Due to Eq. (8.129), $[\underline{A}]^{1/2}$ belongs to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. Any matrix $[A]$ in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ can then be written as

$$[A] = [\underline{A}]^{1/2} [N] [\underline{A}]^{1/2}, \quad (8.143)$$

in which, due to Eq. (8.129) and due to the invertibility of $[\underline{A}]^{1/2}$, $[N]$ is a unique matrix belonging to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. Using Eq. (8.130), matrix $[N]$ has the following representation:

$$[N] = \exp_{\mathbb{M}}([\mathcal{N}(\mathbf{y})]), \quad [\mathcal{N}(\mathbf{y})] = \sum_{j=1}^N y_j [E_j^{\text{sym}}], \quad \mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N, \quad (8.144)$$

Consequently, Eqs. (8.143) and (8.144) define the parameterization $[A] = [\mathcal{A}(\mathbf{y})]$.

12.5.3 Construction of $[\mathbf{A}]$ Using the Parameterization and Generator of Realizations

The random matrix $[\mathbf{A}]$ with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ is then written

$$[\mathbf{A}] = [\underline{A}]^{1/2} [\mathbf{N}] [\underline{A}]^{1/2}, \quad (8.145)$$

in which $[\mathbf{N}]$ is the random matrix with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$, which is written as

$$[\mathbf{N}] = \exp_{\mathbb{M}}([\mathcal{N}(\mathbf{Y})]), \quad [\mathcal{N}(\mathbf{Y})] = \sum_{j=1}^N Y_j [E_j^{\text{sym}}], \quad (8.146)$$

in which $\mathbf{Y} = (Y_1, \dots, Y_N)$ is the random vector with values in \mathbb{R}^N whose pdf $p_{\mathbf{Y}}$ on \mathbb{R}^N and the generator of realizations have been detailed in Sect. 10. Since $[\mathbf{N}]$ can be written as $[\mathbf{N}] = [\underline{\mathbf{A}}]^{-1/2} [\mathbf{A}] [\underline{\mathbf{A}}]^{-1/2}$, and since $E[\mathbf{A}] = [\underline{\mathbf{A}}]$ (see Eq. (8.140)), it can be deduced that

$$E\{[\mathbf{N}]\} = [I_n]. \quad (8.147)$$

13 Nonparametric Stochastic Model of Uncertainties in Computational Linear Structural Dynamics

The nonparametric method for stochastic modeling of uncertainties has been introduced in [106, 107] to take into account both the model-parameter uncertainties and the model uncertainties induced by modeling errors in computational linear structural dynamics, without separating the contribution of each one of these two types of uncertainties.

The nonparametric method is presented hereinafter for linear vibrations of fixed linear structures (no rigid body displacement, but only deformation), formulated in the frequency domain, and for which two cases are considered:

- The case of damped linear elastic structures for which the damping and the stiffness matrices of the computational model are independent of the frequency.
- The case of linear viscoelastic structures for which the damping and the stiffness matrices of the computational model depend on the frequency.

13.1 Methodology

The methodology of the nonparametric method consists in introducing:

- (i) A mean computational model for the linear dynamics of the structure,
- (ii) A reduced-order model (ROM) of the mean computational model,
- (iii) The nonparametric stochastic modeling of both the model-parameter uncertainties and the model uncertainties induced by modeling errors, consisting in modeling the mass, damping, and stiffness matrices of the ROM by random matrices,
- (iv) A prior probability model of the random matrices based on the use of the fundamental ensembles of random matrices introduced previously,
- (v) An estimation of the hyperparameters of the prior probability model of uncertainties if some experimental data are available.

The extension to the case of vibrations of free linear structures (presence of rigid body displacements and of elastic deformations) is straightforward, because it is sufficient to construct the ROM (which is then devoted only to the prediction of the

structural deformations) in projecting the response on the elastic structural modes (without including the rigid body modes) [89].

13.2 Mean Computational Model in Linear Structural Dynamics

The dynamical system is a damped fixed elastic structure for which the vibrations are studied around a static equilibrium configuration considered as a natural state without prestresses and which is subjected to an external load. For given nominal values of the parameters of the dynamical system, the finite element model [128] is called the mean computational model, which is written, in the time domain, as

$$[\mathbb{M}] \ddot{\mathbf{x}}(t) + [\mathbb{D}] \dot{\mathbf{x}}(t) + [\mathbb{K}] \mathbf{x}(t) = \mathbb{f}(t), \quad (8.148)$$

in which $\mathbf{x}(t)$ is the vector of the m degrees of freedom (DOF) (displacements and/or rotations); $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ are the velocity and acceleration vectors; $\mathbb{f}(t)$ is the external load vector of the m inputs (forces and/or moments); and $[\mathbb{M}]$, $[\mathbb{D}]$, and $[\mathbb{K}]$ are the mass, damping, and stiffness matrices of the mean computational model, respectively, which belong to $\mathbb{M}_m^+(\mathbb{R})$.

- The solution $\{\mathbf{x}(t), t > 0\}$ of the time evolution problem is constructed in solving Eq. (8.148) for $t > 0$ with the initial conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $\dot{\mathbf{x}}(0) = \mathbf{v}_0$.
- The forced response $\{\mathbf{x}(t), t \in \mathbb{R}\}$ is such that, for all t fixed in \mathbb{R} , $\mathbf{x}(t)$ verifies Eq. (8.148), and its Fourier transform $\hat{\mathbf{x}}(\omega) = \int_{\mathbb{R}} e^{-i\omega t} \mathbf{x}(t) dt$ is such that, for all ω in \mathbb{R} ,

$$(-\omega^2 [\mathbb{M}] + i\omega [\mathbb{D}] + [\mathbb{K}]) \hat{\mathbf{x}}(\omega) = \hat{\mathbf{f}}(\omega), \quad (8.149)$$

in which $\hat{\mathbf{f}}$ is the Fourier transform of \mathbb{f} . As $[\mathbb{M}]$, $[\mathbb{D}]$, and $[\mathbb{K}]$ are positive-definite matrices, the $\mathbb{M}_m(\mathbb{C})$ -valued frequency response function $\omega \mapsto [\hat{\mathbf{h}}(\omega)] = (-\omega^2 [\mathbb{M}] + i\omega [\mathbb{D}] + [\mathbb{K}])^{-1}$ is a bounded function on \mathbb{R} . From a point of view of the nonparametric stochastic modeling of uncertainties, it is equivalent of presenting the time evolution problem or the forced response problem expressed in the frequency domain. Nevertheless, for such a linear system, the analysis is mainly carried out in the frequency domain. In order to limit the developments, the forced response problem expressed in the frequency domain is presented.

13.3 Reduced-Order Model (ROM) of the Mean Computational Model

The ROM of the mean computational model is constructed for analyzing the response of the structure over a frequency band \mathcal{B} (bounded symmetric interval of pulsations in rad/s) such that

$$\mathcal{B} = [-\omega_{\max}, -\omega_{\min}] \cup [\omega_{\min}, \omega_{\max}], \quad 0 \leq \omega_{\min} < \omega_{\max} < +\infty, \quad (8.150)$$

and is obtained in using the method of modal superposition (or modal analysis) [8, 87]. The generalized eigenvalue problem associated with the mass and stiffness matrices of the mean computational model is written as

$$[\mathbb{K}] \boldsymbol{\phi} = \lambda [\mathbb{M}] \boldsymbol{\phi}, \quad (8.151)$$

for which the eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m$ and the associated elastic structural modes $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_m\}$ are such that

$$<[\mathbb{M}] \boldsymbol{\phi}_\alpha, \boldsymbol{\phi}_\beta> = \mu_\alpha \delta_{\alpha\beta}, \quad (8.152)$$

$$<[\mathbb{K}] \boldsymbol{\phi}_\alpha, \boldsymbol{\phi}_\beta> = \mu_\alpha \omega_\alpha^2 \delta_{\alpha\beta}, \quad (8.153)$$

in which $\omega_\alpha = \sqrt{\lambda_\alpha}$ is the eigenfrequency of elastic structural mode $\boldsymbol{\phi}_\alpha$ whose normalization is defined by the generalized mass μ_α . Let \mathbb{H}_n be the subspace of \mathbb{R}^m spanned by $\{\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n\}$ with $n \ll m$ and let \mathbb{H}_n^c be its complexified (i.e., $\mathbb{H}_n^c = \mathbb{H}_n + i \mathbb{H}_n$). Let $[\Phi]$ be the $(m \times n)$ real matrix whose columns are vectors $\{\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n\}$. The ROM of the mean computational model is obtained as the projection $\mathbf{x}^n(\omega)$ of $\hat{\mathbf{x}}(\omega)$ on \mathbb{H}_n^c , which is written as $\mathbf{x}^n(\omega) = [\Phi] \mathbf{q}(\omega)$ in which $\mathbf{q}(\omega)$ is the vector in \mathbb{C}^n of the generalized coordinates and is written, for all ω in \mathcal{B} , as

$$\mathbf{x}^n(\omega) = [\Phi] \mathbf{q}(\omega), \quad (8.154)$$

$$(-\omega^2[M] + i\omega[D] + [K]) \mathbf{q}(\omega) = \mathbf{f}(\omega), \quad (8.155)$$

in which $[M]$, $[D]$, and $[K]$ (generalized mass, damping, and stiffness matrices) belong to $\mathbb{M}_n^+(\mathbb{R})$ and are such that

$$[M]_{\alpha\beta} = \mu_\alpha \delta_{\alpha\beta}, \quad [D]_{\alpha\beta} = <[\mathbb{D}] \boldsymbol{\phi}_\beta, \boldsymbol{\phi}_\alpha>, \quad [K]_{\alpha\beta} = \mu_\alpha \omega_\alpha^2 \delta_{\alpha\beta}. \quad (8.156)$$

In general, $[D]$ is a full matrix. The generalized force $\mathbf{f}(\omega)$ is a \mathbb{C}^n -vector such that $\mathbf{f}(\omega) = [\Phi]^T \hat{\mathbf{f}}(\omega)$ in which $\hat{\mathbf{f}}$ is the Fourier transform of \mathbf{f} , which is assumed to be a bounded function on \mathbb{R} .

13.3.1 Convergence of the ROM with Respect to n Over Frequency Band of Analysis \mathcal{B}

For the given frequency band of analysis \mathcal{B} , and for a fixed value of the relative error ε_0 with $0 < \varepsilon_0 \ll 1$, let n_0 (depending on ε_0) be the smallest value of n such that $1 \leq n_0 < m$, for which, for all ω in \mathcal{B} , the convergence of the ROM (with respect to dimension n) is reached with relative error ε_0 (if n_0 was equal to m , then ε would be equal to 0). The value of n_0 is such that

$$\forall n \geq n_0, \quad \int_{\mathcal{B}} \|[\hat{\mathbb{H}}(\omega)] - [\hat{\mathbb{H}}^n(\omega)]\|_F^2 d\omega \leq \varepsilon_0 \int_{\mathcal{B}} \|[\hat{\mathbb{H}}(\omega)]\|_F^2 d\omega, \quad (8.157)$$

in which $[\hat{\mathbb{H}}^n(\omega)] = [\Phi](-\omega^2[M] + i\omega[D] + [K])^{-1}[\Phi]^T$. In practice, for large computational model, Eq. (8.157) is replaced by a convergence analysis of \mathbf{x}^n to \mathbf{x} on \mathcal{B} for a given subset of generalized forces \mathbf{f} .

13.4 Nonparametric Stochastic Model of Both the Model-Parameter Uncertainties and the Model Uncertainties (Modeling Errors)

For the given frequency band of analysis \mathcal{B} , and for n fixed to the value n_0 such that Eq. (8.157) is verified, the nonparametric stochastic model of uncertainties consists in replacing in Eq. (8.155) the deterministic matrices $[M]$, $[D]$, and $[K]$ by random matrices $[\mathbf{M}]$, $[\mathbf{D}]$, and $[\mathbf{K}]$ defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_n^+(\mathbb{R})$. The deterministic ROM defined by Eqs. (8.154) and (8.155) is then replaced by the following stochastic ROM:

$$\mathbf{X}^n(\omega) = [\Phi] \mathbf{Q}(\omega), \quad (8.158)$$

$$(-\omega^2[\mathbf{M}] + i\omega[\mathbf{D}] + [\mathbf{K}]) \mathbf{Q}(\omega) = \mathbf{f}(\omega), \quad (8.159)$$

in which, for all ω in \mathcal{B} , $\mathbf{X}^n(\omega)$ and $\mathbf{Q}(\omega)$ are \mathbb{C}^m - and \mathbb{C}^n -valued random vectors defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$.

13.4.1 Available Information for Constructing a Prior Probability Model of $[\mathbf{M}]$, $[\mathbf{D}]$, and $[\mathbf{K}]$

The available information for constructing the prior probability model of random matrices $[\mathbf{M}]$, $[\mathbf{D}]$, and $[\mathbf{K}]$ using the MaxEnt principle are the following:

- (i) Random matrices $[\mathbf{M}]$, $[\mathbf{D}]$, and $[\mathbf{K}]$ are with values in $\mathbb{M}_n^+(\mathbb{R})$.
- (ii) The mean values of these random matrices are chosen as the corresponding matrices in the ROM of the mean computational model,

$$E\{[\mathbf{M}]\} = [M], \quad E\{[\mathbf{D}]\} = [D], \quad E\{[\mathbf{K}]\} = [K]. \quad (8.160)$$

- (iii) The prior probability model of these random matrices must be chosen such that, for all ω in \mathcal{B} , the solution $\mathbf{Q}(\omega)$ of Eq. (8.159) is a second-order \mathbb{C}^n -valued random variable, that is to say, such that

$$E\{\|(-\omega^2[\mathbf{M}] + i\omega[\mathbf{D}] + [\mathbf{K}])^{-1}\|_F^2\} < +\infty, \quad \forall \omega \in \mathcal{B}. \quad (8.161)$$

13.4.2 Prior Probability Model of [M], [D], and [K], Hyperparameters and Generator of Realizations

The joint pdf of random matrices $[M]$, $[D]$, and $[K]$ is constructed using the MaxEnt principle under the constraints defined by the available information described before. Taking into account such an available information, it is proved [107] that these three random matrices are statistically independent. Taking into account Eqs. (8.52), (8.55), (8.160), and (8.161), each random matrix $[M]$, $[D]$, and $[K]$ is then chosen in ensemble SE_e^+ of the positive-definite random matrices with a given mean value and an arbitrary positive-definite lower bound. The level of uncertainties, for each type of forces (mass, damping, and stiffness), is controlled by the three hyperparameters δ_M , δ_D , and δ_K of the pdf of random matrices $[M]$, $[D]$, and $[K]$, which are defined by Eq. (8.56). The generator of realizations for ensemble SE_e^+ has explicitly been described.

13.5 Case of Linear Viscoelastic Structures

The dynamical system is a fixed viscoelastic structure for which the vibrations are studied around a static equilibrium configuration considered as a natural state without prestresses and which is subjected to an external load. Consequently, in the frequency domain, the damping and stiffness matrices depend on frequency ω , instead of to be independent of the frequency as in the previous analyzed case. Consequently, two aspects must be addressed. The first one is relative to the choice of the basis for constructing the ROM, and the second one is the nonparametric stochastic modeling of the frequency dependent damping and stiffness matrices which are related by a Hilbert transform; we then use for such a nonparametric stochastic modeling ensemble SE^{HT} of a pair of positive-definite matrix-valued random functions related to a Hilbert transform.

13.5.1 Mean Computational Model, ROM, and Convergence

In such a case, the mean computational model defined by Eq. (8.149) is replaced by the following:

$$(-\omega^2 [M] + i\omega[D(\omega)] + [K(\omega)]) \hat{x}(\omega) = \hat{f}(\omega), \quad (8.162)$$

For constructing the ROM, the projection basis is chosen as previously in taking the stiffness matrix at zero frequency. The generalized eigenvalue problem, defined by Eq. (8.151), is then rewritten as $[K(0)]\phi = \lambda[M]\phi$. With such a choice of a basis, Eqs. (8.154) to (8.156) that defined the ROM for all ω belonging to the frequency band of analysis \mathcal{B} are replaced by

$$\mathbf{x}^n(\omega) = [\Phi] \mathbf{q}(\omega), \quad (8.163)$$

$$(-\omega^2[M] + i\omega[D(\omega)] + [K(\omega)]) \mathbf{q}(\omega) = \mathbf{f}(\omega), \quad (8.164)$$

in which $[M]$, $[D(\omega)]$, and $[K(\omega)]$ belong to $\mathbb{M}_n^+(\mathbb{R})$ and are such that

$$[M]_{\alpha\beta} = \mu_\alpha \delta_{\alpha\beta}, [D(\omega)]_{\alpha\beta} = <[\mathbb{D}(\omega)] \phi_\beta, \phi_\alpha>, [K(\omega)]_{\alpha\beta} = <[\mathbb{K}(\omega)] \phi_\beta, \phi_\alpha>. \quad (8.165)$$

The matrices $[D(\omega)]$ and $[K(\omega)]$ are full matrices belonging to $\mathbb{M}_n^+(\mathbb{R})$, which verify (see [89]) all the mathematical properties introduced in the construction of ensemble SE^{HT} and, in particular, verify Eqs. (8.65) to (8.68). For ε_0 fixed, the value n_0 of the dimension n of the ROM is such that Eq. (8.157) holds (equation in which the frequency dependence of the damping and stiffness matrices is introduced). In practice, for large computational model, this criterion is replaced by a convergence analysis of \mathbf{x}^n to \mathbf{x} on \mathcal{B} for a given subset of generalized forces \mathbf{f} .

13.5.2 Nonparametric Stochastic Model of Both the Model-Parameter Uncertainties and the Model Uncertainties (Modeling Errors)

For the given frequency band of analysis \mathcal{B} , and for n fixed to the value n_0 , the nonparametric stochastic model of uncertainties consists in replacing in Eq. (8.164) the deterministic matrices $[M]$, $[D(\omega)]$, and $[K(\omega)]$ by random matrices $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(\omega)]$ defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_n^+(\mathbb{R})$. The deterministic ROM defined by Eqs. (8.163) and (8.164) is then replaced by the following stochastic ROM:

$$\mathbf{X}^n(\omega) = [\Phi] \mathbf{Q}(\omega), \quad (8.166)$$

$$(-\omega^2[\mathbf{M}] + i\omega[\mathbf{D}(\omega)] + [\mathbf{K}(\omega)]) \mathbf{Q}(\omega) = \mathbf{f}(\omega), \quad (8.167)$$

in which, for all ω in \mathcal{B} , $\mathbf{X}^n(\omega)$ and $\mathbf{Q}(\omega)$ are \mathbb{C}^m - and \mathbb{C}^n -valued random vectors defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$.

13.5.3 Available Information for Constructing a Prior Probability Model of $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(\omega)]$

The available information for constructing the prior probability model of random matrices $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(\omega)]$ using the MaxEnt principle are, for all ω in \mathcal{B} :

- (i) Random matrices $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(\omega)]$ are with values in $\mathbb{M}_n^+(\mathbb{R})$.
- (ii) The mean values of these random matrices are chosen as the corresponding matrices in the ROM of the mean computational model,

$$E\{[\mathbf{M}]\} = [M], E\{[\mathbf{D}(\omega)]\} = [D(\omega)], E\{[\mathbf{K}(\omega)]\} = [K(\omega)]. \quad (8.168)$$

- (iii) The random matrices $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(\omega)]$ are such that

$$[\mathbf{D}(-\omega)] = [\mathbf{D}(\omega)], [\mathbf{K}(-\omega)] = [\mathbf{K}(\omega)]. \quad (8.169)$$

- (iv) The prior probability model of these random matrices must be chosen for that, for all ω in \mathcal{B} , the solution $\mathbf{Q}(\omega)$ of Eq. (8.167) is a second-order \mathbb{C}^n -valued random variable, that is to say, for that

$$E\{\|(-\omega^2[\mathbf{M}] + i\omega[\mathbf{D}(\omega)] + [\mathbf{K}(\omega)])^{-1}\|_F^2\} < +\infty, \quad \forall \omega \in \mathcal{B}. \quad (8.170)$$

- (v) The algebraic dependence between $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(\omega)]$ induced by the causality must be preserved, which means that random matrix $[\mathbf{K}(\omega)]$ is given by Eq. (8.72) as a function of random matrix $[\mathbf{K}(0)]$ and the family of random matrices $\{\mathbf{D}(\omega), \omega \geq 0\}$,

$$[\mathbf{K}(\omega)] = [\mathbf{K}(0)] + \frac{2\omega^2}{\pi} \text{p.v.} \int_0^{+\infty} \frac{1}{\omega^2 - \omega'^2} [\mathbf{D}(\omega')] d\omega', \quad \forall \omega \geq 0. \quad (8.171)$$

13.5.4 Prior Probability Model of $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(0)]$, Hyperparameters, and Generator of Realizations

Taking into account the available information, the use of the MaxEnt principle yields that random matrices $[\mathbf{M}]$, $\{[\mathbf{D}(\omega)], \omega \geq 0\}$, and $[\mathbf{K}(0)]$ are statistically independent.

- As previously, random matrix $[\mathbf{M}]$ is chosen in ensemble SE_ϵ^+ of the positive-definite random matrices with a given mean value and an arbitrary positive-definite lower bound. The pdf is explicitly defined and depends on the hyperparameter δ_M defined by Eq. (8.56). The generator of realizations is the generator of the ensemble SE_ϵ^+ , which was explicitly defined.
- For all fixed ω , random matrices $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(0)]$ that are statistically independent are constructed as explained in the section devoted to ensemble SE^{HT} . The levels of uncertainties of random matrices $[\mathbf{D}(\omega)]$ and $[\mathbf{K}(0)]$ are controlled by the two frequency-independent hyperparameters δ_D and δ_K introduced in paragraphs (i) and (ii) located after Eq. (8.70). The generator of realizations is directly deduced from the generator of realizations of fundamental ensemble SG_ϵ^+ , which was explicitly defined.
- With such a nonparametric stochastic modeling, the level of uncertainties is controlled by hyperparameters δ_M , δ_D , and δ_K , and the generators of realizations of random matrices $[\mathbf{M}]$, $[\mathbf{D}(\omega)]$, and $[\mathbf{K}(0)]$ are explicitly described.

13.6 Estimation of the Hyperparameters of the Nonparametric Stochastic Model of Uncertainties

For the nonparametric stochastic model of uncertainties in computational linear structural dynamics, dimension n of the ROM is fixed to the value n_0 for which the response of the ROM of the mean computational model is converged with respect to n . The prior probability model of uncertainties then depends on the vector-valued hyperparameter $\boldsymbol{\delta}_{\text{npar}} = (\delta_M, \delta_D, \delta_K)$ belonging to an admissible set $\mathcal{C}_{\text{npar}}$.

- If no experimental data are available, then $\boldsymbol{\delta}_{\text{npar}}$ must be considered as a vector-valued parameter for performing a sensitivity analysis of the stochastic solution with respect to the level of uncertainties. Such a nonparametric stochastic model of both the model-parameter uncertainties and the model uncertainties then

allows the robustness of the solution to be analyzed as a function of the level of uncertainties which is controlled by δ_{npar} .

- If experimental data are available, an estimation of δ_{npar} can be carried out, for instance, using a least square method or the maximum likelihood method [102, 117, 123]. Let \mathbf{W} be the random real vector which is observed, which is independent of ω , but which depends on $\{\mathbf{X}^n(\omega), \omega \in \mathcal{B}\}$ where $\mathbf{X}^n(\omega)$ is the second-order random complex vector given by Eq. (8.158) or (8.166). For all $\delta_{\text{npar}} \in \mathcal{C}_{\text{npar}}$, the probability density function of \mathbf{W} is denoted as $\mathbf{w} \mapsto p_{\mathbf{W}}(\mathbf{w}; \delta_{\text{npar}})$. Using the maximum likelihood method, the optimal value $\delta_{\text{npar}}^{\text{opt}}$ of δ_{npar} is estimated by maximizing the logarithm of the likelihood function,

$$\delta_{\text{npar}}^{\text{opt}} = \arg \max_{\delta_{\text{npar}} \in \mathcal{C}_{\text{npar}}} \sum_{\ell=1}^{v_{\text{exp}}} \log p_{\mathbf{W}}(\mathbf{w}_{\ell}^{\text{exp}}; \delta_{\text{npar}}). \quad (8.172)$$

in which $\mathbf{w}_1^{\text{exp}}, \dots, \mathbf{w}_{v_{\text{exp}}}^{\text{exp}}$ are v_{exp} independent experimental data corresponding to \mathbf{W} .

14 Parametric-Nonparametric Uncertainties in Computational Nonlinear Structural Dynamics

The last two presented sections have been devoted to the nonparametric stochastic model of both the model-parameter uncertainties and the model uncertainties induced by the modeling errors, without separating the contribution of each one of these two types of uncertainties. Sometimes, there is an interest of separating the uncertainties for a small number of model parameters that exhibit an important sensitivity on the responses, from uncertainties induced by the model uncertainties and the uncertainties on other model parameters.

Such an objective requires to use a parametric-nonparametric stochastic model of uncertainties, also called the generalized probabilistic approach of uncertainties in computational structural dynamics, which has been introduced in [113].

As the nonparametric stochastic model of uncertainties has been presented in the previous sections for linear dynamical systems formulated in the frequency domain, in the present section, the parametric-nonparametric stochastic model of uncertainties is presented in computational nonlinear structural dynamics formulated in the time domain.

14.1 Mean Nonlinear Computational Model in Structural Dynamics

The dynamical system is a damped fixed structure for which the nonlinear vibrations are studied in the time domain around a static equilibrium configuration considered as a natural state without prestresses and subjected to an external load. For given

nominal values of the model parameters of the dynamical system, the basic finite element model is called the mean nonlinear computational model. In addition, it is assumed that a set of model parameters has been identified as sensitive parameters that are uncertain. These uncertain model parameters are the components of a vector $\tilde{\mathbf{y}}$ belonging to an admissible set \mathcal{C}_{par} which is a subset of \mathbb{R}^N . It is assumed that a parameterization is constructed such that $\tilde{\mathbf{y}} = \mathcal{Y}(\mathbf{y})$ in which $\mathbf{y} \mapsto \mathcal{Y}(\mathbf{y})$ is a given and known function from \mathcal{C}_{par} into \mathbb{R}^N . For instance, if the component \tilde{y}_j of $\tilde{\mathbf{y}}$ must belong to $[0, +\infty[$, then \tilde{y}_j could be defined as $\exp(y_j)$ with $y_j \in \mathbb{R}$, which yields $\mathcal{Y}_j(\mathbf{y}) = \exp(y_j)$. Hereinafter, it is then assumed that the uncertain model parameters are represented by vector $\mathbf{y} = (y_1, \dots, y_N)$ belonging to \mathbb{R}^N . The nonlinear mean computational model, depending on uncertain model parameter \mathbf{y} , is written as

$$[\mathbb{M}(\mathbf{y})]\ddot{\mathbf{x}}(t) + [\mathbb{D}(\mathbf{y})]\dot{\mathbf{x}}(t) + [\mathbb{K}(\mathbf{y})]\mathbf{x}(t) + \mathbb{f}_{\text{NL}}(\mathbf{x}(t), \dot{\mathbf{x}}(t); \mathbf{y}) = \mathbb{f}(t; \mathbf{y}), \quad (8.173)$$

in which $\mathbf{x}(t)$ is the unknown time response vector of the m degrees of freedom (DOF) (displacements and/or rotations); $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ are the velocity and acceleration vectors respectively; $\mathbb{f}(t; \mathbf{y})$ is the known external load vector of the m inputs (forces and/or moments); $[\mathbb{M}(\mathbf{y})]$, $[\mathbb{D}(\mathbf{y})]$, and $[\mathbb{K}(\mathbf{y})]$ are the mass, damping, and stiffness matrices of the linear part of the mean nonlinear computational model, respectively, which belong to $M_m^+(\mathbb{R})$; and $(\mathbf{x}(t), \dot{\mathbf{x}}(t)) \mapsto \mathbb{f}_{\text{NL}}(\mathbf{x}(t), \dot{\mathbf{x}}(t); \mathbf{y})$ is the nonlinear mapping that models the local nonlinear forces (such as nonlinear elastic barriers).

We are interested in the time evolution problem defined by Eq. (8.173) for $t > 0$ with the initial conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $\dot{\mathbf{x}}(0) = \mathbf{v}_0$.

14.2 Reduced-Order Model (ROM) of the Mean Nonlinear Computational Model

For all \mathbf{y} fixed in \mathbb{R}^N , let $\{\phi_1(\mathbf{y}), \dots, \phi_m(\mathbf{y})\}$ be an algebraic basis of \mathbb{R}^m constructed, for instance, either using the elastic structural modes of the linearized system, using the elastic structural modes of the underlying linear system, or using the POD (proper orthogonal decomposition) modes of the nonlinear system). Hereinafter, it is assumed that the elastic structural modes of the underlying linear system are used for constructing the ROM of the mean nonlinear computational model (such a choice is not intrusive with respect to a black-box software, but in counterpart requires a large parallel computation induced by all the sampling values of \mathbf{y} , which are considered by the stochastic solver).

For each value of \mathbf{y} given in \mathbb{R}^N , the generalized eigenvalue problem associated with the mean mass and stiffness matrices is written as

$$[\mathbb{K}(\mathbf{y})]\boldsymbol{\phi}(\mathbf{y}) = \lambda(\mathbf{y})[\mathbb{M}(\mathbf{y})]\boldsymbol{\phi}(\mathbf{y}), \quad (8.174)$$

for which the eigenvalues $0 < \lambda_1(\mathbf{y}) \leq \lambda_2(\mathbf{y}) \leq \dots \leq \lambda_m(\mathbf{y})$ and the associated elastic structural modes $\{\phi_1(\mathbf{y}), \phi_2(\mathbf{y}), \dots, \phi_m(\mathbf{y})\}$ are such that

$$\langle [\mathbb{M}(\mathbf{y})] \phi_\alpha(\mathbf{y}), \phi_\beta(\mathbf{y}) \rangle = \mu_\alpha(\mathbf{y}) \delta_{\alpha\beta}, \quad (8.175)$$

$$\langle [\mathbb{K}(\mathbf{y})] \phi_\alpha(\mathbf{y}), \phi_\beta(\mathbf{y}) \rangle = \mu_\alpha(\mathbf{y}) \omega_\alpha(\mathbf{y})^2 \delta_{\alpha\beta}, \quad (8.176)$$

in which $\omega_\alpha(\mathbf{y}) = \sqrt{\lambda_\alpha(\mathbf{y})}$ is the eigenfrequency of elastic structural mode $\phi_\alpha(\mathbf{y})$ whose normalization is defined by the generalized mass $\mu_\alpha(\mathbf{y})$. Let $[\phi(\mathbf{y})]$ be the $(m \times n)$ real matrix whose columns are vectors $\{\phi_1(\mathbf{y}), \dots, \phi_n(\mathbf{y})\}$. For \mathbf{y} fixed in \mathbb{R}^N and for all fixed $t > 0$, the ROM is obtained as the projection $\mathbf{x}^n(t)$ of $\mathbf{x}(t)$ on the subspace of \mathbb{R}^m spanned by $\{\phi_1(\mathbf{y}), \dots, \phi_n(\mathbf{y})\}$ with $n \ll m$, which is written as $\mathbf{x}^n(t) = [\phi(\mathbf{y})] \mathbf{q}(t)$ in which $\mathbf{q}(t)$ is the vector in \mathbb{R}^n of the generalized coordinates and is written, for all $t > 0$, as

$$\mathbf{x}^n(t) = [\phi(\mathbf{y})] \mathbf{q}(t), \quad (8.177)$$

$$[M(\mathbf{y})] \ddot{\mathbf{q}}(t) + [D(\mathbf{y})] \dot{\mathbf{q}}(t) + [K(\mathbf{y})] \mathbf{q}(t) + \mathbf{F}_{\text{NL}}(\mathbf{q}(t), \dot{\mathbf{q}}(t); \mathbf{y}) = \mathbf{f}(t; \mathbf{y}), \quad (8.178)$$

in which $[M(\mathbf{y})]$, $[D(\mathbf{y})]$, and $[K(\mathbf{y})]$ (generalized mass, damping, and stiffness matrices) belong to $\mathbb{M}_n^+(\mathbb{R})$ and are such that

$$[M(\mathbf{y})]_{\alpha\beta} = \mu_\alpha(\mathbf{y}) \delta_{\alpha\beta}, \quad [D(\mathbf{y})]_{\alpha\beta} = \langle [\mathbb{D}(\mathbf{y})] \phi_\beta(\mathbf{y}), \phi_\alpha(\mathbf{y}) \rangle, \quad (8.179)$$

$$[K(\mathbf{y})]_{\alpha\beta} = \mu_\alpha(\mathbf{y}) \omega_\alpha(\mathbf{y})^2 \delta_{\alpha\beta}. \quad (8.180)$$

In general, $[D(\mathbf{y})]$ is a full matrix. The generalized force $\mathbf{f}(t; \mathbf{y})$ is a \mathbb{R}^n -vector such that $\mathbf{f}(t; \mathbf{y}) = [\phi(\mathbf{y})]^T \mathbb{f}(t; \mathbf{y})$. The generalized nonlinear force is such that $\mathbf{F}_{\text{NL}}(\mathbf{q}(t), \dot{\mathbf{q}}(t); \mathbf{y}) = [\phi(\mathbf{y})]^T \mathbb{f}_{\text{NL}}([\phi(\mathbf{y})] \mathbf{q}(t), [\phi(\mathbf{y})] \dot{\mathbf{q}}(t); \mathbf{y})$.

Convergence of the ROM with respect to n . Let n_0 be the value of n , for which, for a given accuracy and for all \mathbf{y} in \mathbb{R}^N , the response \mathbf{x}^n is converged to \mathbf{x} for all $n > n_0$.

14.3 Parametric-Nonparametric Stochastic Modeling of Uncertainties

In all this section, the value of n is fixed to the value n_0 defined hereinbefore.

14.3.1 Methodology

- The parametric stochastic modeling of uncertainties consists in modeling uncertain model parameter \mathbf{y} by a second-order random variable $\mathbf{Y} = (Y_1, \dots, Y_N)$, defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in \mathbb{R}^N . Consequently,

deterministic matrices $[M(\mathbf{y})]$, $[D(\mathbf{y})]$, and $[K(\mathbf{y})]$ defined by Eqs. (8.179)–(8.180) become the second-order random matrices, $[M(\mathbf{Y})]$, $[D(\mathbf{Y})]$, and $[K(\mathbf{Y})]$, defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_n^+(\mathbb{R})$. The mean values of these random matrices are the matrices in $\mathbb{M}_n^+(\mathbb{R})$ such that

$$[\underline{M}] = E\{[M(\mathbf{Y})]\}, \quad [\underline{D}] = E\{[D(\mathbf{Y})]\}, \quad [\underline{K}] = E\{[K(\mathbf{Y})]\}, \quad (8.181)$$

- The nonparametric stochastic modeling of uncertainties consists, for all \mathbf{y} fixed in \mathbb{R}^N , in modeling matrices $[M(\mathbf{y})]$, $[D(\mathbf{y})]$, and $[K(\mathbf{y})]$ defined by Eqs. (8.179)–(8.180), by the second-order random matrices $[\mathbf{M}(\mathbf{y})] = \{\theta' \mapsto [\mathbf{M}(\theta'; \mathbf{y})]\}$, $[\mathbf{D}(\mathbf{y})] = \{\theta' \mapsto [\mathbf{D}(\theta'; \mathbf{y})]\}$, and $[\mathbf{K}(\mathbf{y})] = \{\theta' \mapsto [\mathbf{K}(\theta'; \mathbf{x})]\}$, defined on another probability space $(\Theta', \mathcal{T}', \mathcal{P}')$ (and thus independent of \mathbf{Y}), with values in $\mathbb{M}_n^+(\mathbb{R})$.
 - The parametric-nonparametric stochastic modeling of uncertainties consists, in Eq. (8.178)):
- (i) In modeling $[M(\mathbf{y})]$, $[D(\mathbf{y})]$, and $[K(\mathbf{y})]$ by random matrices $[\mathbf{M}(\mathbf{y})]$, $[\mathbf{D}(\mathbf{y})]$, and $[\mathbf{K}(\mathbf{y})]$,
 - (ii) In modeling uncertain model parameter \mathbf{y} by the \mathbb{R}^N -valued random variable \mathbf{Y} .

Consequently, the statistically dependent random matrices $[\mathbf{M}(\mathbf{Y})] = \{(\theta, \theta') \mapsto [\mathbf{M}(\theta'; \mathbf{Y}(\theta))]\}$, $[\mathbf{D}(\mathbf{Y})] = \{(\theta, \theta') \mapsto [\mathbf{D}(\theta'; \mathbf{Y}(\theta))]\}$ and $[\mathbf{K}(\mathbf{Y})] = \{(\theta, \theta') \mapsto [\mathbf{K}(\theta'; \mathbf{Y}(\theta))]\}$ are measurable mappings from $\Theta \times \Theta'$ into $\mathbb{M}_n^+(\mathbb{R})$. The deterministic ROM defined by Eqs. (8.177)–(8.178) is then replaced by the following stochastic ROM:

$$\mathbf{X}^n(t) = [\phi(\mathbf{Y})] \mathbf{Q}(t), \quad (8.182)$$

$$[\mathbf{M}(\mathbf{Y})] \ddot{\mathbf{Q}}(t) + [\mathbf{D}(\mathbf{Y})] \dot{\mathbf{Q}}(t) + [\mathbf{K}(\mathbf{Y})] \mathbf{Q}(t) + \mathbf{F}_{NL}(\mathbf{Q}(t), \dot{\mathbf{Q}}(t); \mathbf{Y}) = \mathbf{f}(t; \mathbf{Y}), \quad (8.183)$$

in which for all fixed t , $\mathbf{X}^n(t) = \{(\theta, \theta') \mapsto \mathbf{X}^n(\theta, \theta'; t)\}$ and $\mathbf{Q}(t) = \{(\theta, \theta') \mapsto \mathbf{Q}(\theta, \theta'; t)\}$ are \mathbb{R}^m - and \mathbb{R}^n -valued random vectors defined for (θ, θ') in $\Theta \times \Theta'$.

14.3.2 Prior Probability Model of \mathbf{Y} , Hyperparameters, and Generator of Realizations

The prior pdf $p_{\mathbf{Y}}$ on \mathbb{R}^N of random vector \mathbf{Y} is constructed using the MaxEnt principle under the constraints defined by the available information given by Eq. (8.81), as explained in Sect. 11, in which a generator of realizations $\{\mathbf{Y}(\theta), \theta \in \Theta\}$ has been detailed. Such a generator depends on the hyperparameters related to the available information. In general, the hyperparameters are the mean vector $\underline{\mathbf{y}} = E\{\mathbf{Y}\}$ belonging to \mathbb{R}^N and a vector-valued hyperparameter δ_{par} that belongs to an admissible set \mathcal{C}_{par} , which allows the level of parametric uncertainties to be controlled.

14.3.3 Prior Probability Model of $[M(y)]$, $[D(y)]$, and $[K(y)]$, Hyperparameters, and Generator of Realizations

Similarly to the construction given in section entitled “Nonparametric Stochastic Model of Uncertainties in Computational Linear Structural Dynamics”, for all \mathbf{y} fixed in \mathbb{R}^N , random matrices $[M(\mathbf{y})]$, $[D(\mathbf{y})]$, and $[K(\mathbf{y})]$, are statistically independent and written as

$$[M(\mathbf{y})] = [L_M(\mathbf{y})]^T [G_M] [L_M(\mathbf{y})], \quad (8.184)$$

$$[D(\mathbf{y})] = [L_D(\mathbf{y})]^T [G_D] [L_D(\mathbf{y})], \quad (8.185)$$

$$[K(\mathbf{y})] = [L_K(\mathbf{y})]^T [G_K] [L_K(\mathbf{y})], \quad (8.186)$$

in which, for all \mathbf{y} in \mathbb{R}^N , $[L_M(\mathbf{y})]$, $[L_D(\mathbf{y})]$, and $[L_K(\mathbf{y})]$ are the upper triangular matrices such that (Cholesky factorization) $[M(\mathbf{y})] = [L_M(\mathbf{y})]^T [L_M(\mathbf{y})]$, $[D(\mathbf{y})] = [L_D(\mathbf{y})]^T [L_D(\mathbf{y})]$, and $[K(\mathbf{y})] = [L_K(\mathbf{y})]^T [L_K(\mathbf{y})]$. In Eqs. (8.184) to (8.186), $[G_M]$, $[G_D]$, and $[G_K]$ are independent random matrices defined on probability space $(\Theta', \mathcal{T}', \mathcal{P}')$, with values in $\mathbb{M}_n^+(\mathbb{R})$, independent of \mathbf{y} , and belonging to fundamental ensemble SG_ε^+ of random matrices. The level of nonparametric uncertainties is controlled by the coefficients of variation δ_{G_M} , δ_{G_D} , and δ_{G_K} defined by Eq. (8.24) and the vector-valued parameter δ_{npar} is defined as $\delta_{\text{npar}} = (\delta_M, \delta_D, \delta_K)$ that belongs to an admissible set $\mathcal{C}_{\text{npar}}$. The generator of realizations $\{[G_M(\theta')], [G_D(\theta')], [G_K(\theta')]\}$ for θ' in Θ' is explicitly described in the section devoted to the construction of ensembles SG_ε^+ and SG_0^+ .

14.3.4 Mean Values of Random Matrices $[M(Y)]$, $[D(Y)]$, $[K(Z)]$ and Hyperparameters of the Parametric-Nonparametric Stochastic Model of Uncertainties

Taking into account the construction presented hereinbefore, we have

$$E\{[M(Y)]\} = [\underline{M}], \quad E\{[D(Y)]\} = [\underline{D}], \quad E\{[K(Y)]\} = [\underline{K}], \quad (8.187)$$

in which the matrices $[\underline{M}]$, $[\underline{D}]$, and $[\underline{K}]$ are the deterministic matrices defined by Eq. (8.181). The hyperparameters of the parametric-nonparametric stochastic model of uncertainties are

$$\underline{\mathbf{y}} \in \mathbb{R}^N, \quad \delta_{\text{par}} \in \mathcal{C}_{\text{par}}, \quad \delta_{\text{npar}} = (\delta_M, \delta_D, \delta_K) \in \mathcal{C}_{\text{npar}}. \quad (8.188)$$

14.4 Estimation of the Hyperparameters of the Parametric-Nonparametric Stochastic Model of Uncertainties

The value of n is fixed to the value n_0 that has been defined. The parametric-nonparametric stochastic model of uncertainties is controlled by the hyperparameters defined by Eq. (8.188).

- If no experimental data are available, then $\underline{\mathbf{y}}$ can be fixed to a nominal value \mathbf{y}_0 , and δ_{par} and δ_{npar} must be considered as parameters to perform a sensitivity analysis of the stochastic solution. Such a parametric-nonparametric stochastic model of uncertainties allows the robustness of the solution to be analyzed as a function of the level of uncertainties controlled by δ_{par} and δ_{npar} .
- If experimental data are available, an estimation of $\underline{\mathbf{y}}$, δ_{par} , and δ_{npar} can be carried out, for instance, using a least square method or the maximum likelihood method [102, 117, 123]. Let \mathbf{W} be the random real vector which is observed, which is independent of t , but which depends on $\{\mathbf{X}^n(t), t \geq 0\}$ where $\mathbf{X}^n(t)$ is the second-order stochastic solution of Eq. (8.182)–(8.183) for $t > 0$ with initial conditions for $t = 0$. Let $\mathbf{r} = (\underline{\mathbf{y}}, \delta_{\text{par}}, \delta_{\text{npar}})$ be the vector-valued hyperparameter belonging to the admissible set $\mathcal{C}_{\mathbf{r}} = \mathbb{R}^N \times \mathcal{C}_{\text{par}} \times \mathcal{C}_{\text{npar}}$. For all \mathbf{r} in $\mathcal{C}_{\mathbf{r}}$, the probability density function of \mathbf{W} is denoted as $\mathbf{w} \mapsto p_{\mathbf{W}}(\mathbf{w}; \mathbf{r})$. Using the maximum likelihood method, the optimal values \mathbf{r}^{opt} of \mathbf{r} are estimated by maximizing the logarithm of the likelihood function,

$$\mathbf{r}^{\text{opt}} = \arg \max_{\mathbf{r} \in \mathcal{C}_{\mathbf{r}}} \sum_{\ell=1}^{v_{\text{exp}}} \log p_{\mathbf{W}}(\mathbf{w}_{\ell}^{\text{exp}}; \mathbf{r}). \quad (8.189)$$

in which $\mathbf{w}_1^{\text{exp}}, \dots, \mathbf{w}_{v_{\text{exp}}}^{\text{exp}}$ are v_{exp} independent experimental data corresponding to \mathbf{W} .

15 Key Research Findings and Applications

15.1 Propagation of Uncertainties Using Nonparametric or Parametric-Nonparametric Stochastic Models of Uncertainties

The stochastic modeling introduces some random vectors and some random matrices in the stochastic computational models. Consequently, a stochastic solver is required. Two distinct classes of techniques can be used:

- The first one is constituted of the stochastic spectral methods, pioneered by Roger Ghanem in 1990–1991 [43, 44], consisting in performing a projection of the Galerkin type (see [45, 46, 67, 69, 84, 121]), and of separated representations methods [34, 85]. This class of techniques allows for obtaining a great precision for the approximated solution that is constructed.
- The second class is composed of methods based on a direct simulation of which the most popular is the Monte Carlo numerical simulation method (see, for instance, [41, 96]). With such a method, the convergence can be controlled during the computation, and its speed of convergence is independent of the stochastic dimension and can be improved using either advanced Monte Carlo simulation procedures [100], or a technique of subset simulation [6], or finally a method of

local simulation domain [93]. The Monte Carlo simulation method is a stochastic solver that is particularly well adapted to the high stochastic dimension induced by the random matrices introduced by the nonparametric method of uncertainties.

15.2 Experimental Validations of the Nonparametric Method of Uncertainties

The nonparametric stochastic modeling of uncertainties has been experimentally validated through applications in different domains of computational sciences and engineering, in particular:

- In linear dynamics, for the dynamics of complex structures in the low-frequency domain [7, 12, 13]; for the dynamics of structures with nonhomogeneous uncertainties, in the low-frequency domain [24] and in transient dynamics [35]; and finally for the dynamics of composite sandwich panels in low- and medium-frequency domains [25];
- In nonlinear dynamics, for nonlinear structural dynamics of fuel assemblies [9], for nonlinear post-buckling static and dynamical analyses of uncertain cylindrical shells [21], and for some nonlinear reduced-order models [81];
- In linear structural acoustics, for the vibroacoustic of complex structures in low- and medium-frequency domains [38], with sound-insulation layers [39], and for the wave propagation in multilayer live tissues in the ultrasonic domain [30];
- In continuum mechanics of solids, for the nonlinear thermomechanical analysis [97] and the heat transfer in complex composite panels [98] and for linear elasticity of composites reinforced with fibers at mesoscale [48].

15.3 Additional Ingredients for the Nonparametric Stochastic Modeling of Uncertainties

Some important ingredients have been developed for having the tools required for performing the nonparametric stochastic modeling of uncertainties in linear and nonlinear dynamics of mechanical systems, in particular:

- The dynamic substructuring with uncertain substructures which allows for the nonparametric modeling of nonhomogeneous uncertainties in different parts of a structure [116];
- The nonparametric stochastic modeling of uncertain structures with uncertain boundary conditions/coupling between substructures [78];
- The nonparametric stochastic modeling of matrices that depend on the frequency and that are related by a Hilbert transform due to the existence of causality properties, such as those encountered in the linear viscoelasticity theory [89, 115];
- The multi-body dynamics for which there are uncertain bodies (mass, center of mass, inertia tensor), for which the uncertainties in the bodies come from a lack

of knowledge of the distribution of the mass inside the bodies (for instance, the spatial distribution of the passengers inside a high-speed train) [10];

- The nonparametric stochastic modeling in vibroacoustics of complex systems for low- and medium-frequency domains, including the stochastic modeling of the coupling matrices between the structure and the acoustic cavities [38, 89, 110];
- The formulation of the nonparametric stochastic modeling of the nonlinear operators occurring in the static and the dynamics of uncertain geometrically nonlinear structures [21, 77, 81].

15.4 Applications of the Nonparametric Stochastic Modeling of Uncertainties in Different Fields of Computational Sciences and Engineering

- In dynamics:

Aeronautics and aerospace engineering systems [7, 20, 78, 88, 91]

Biomechanics [30, 31]

Environment for well integrity for geologic CO₂ sequestration [32]

Nuclear engineering [9, 12, 13, 29]

Pipe conveying fluid [94]

Rotordynamics [79, 80, 82]

Soil-structure interaction and wave propagation in soils [4, 5, 26, 27]

Vibration of turbomachines [18, 19, 22, 70]

Vibroacoustics of automotive vehicles [3, 38–40, 61]

- In continuum mechanics of heterogeneous materials:

Composites reinforced with fibers [48]

Heat transfer of complex composite panels [98]

Nonlinear thermomechanics in heterogeneous materials [97]

Polycrystalline microstructures [49]

Porous materials [52]

Random elasticity tensors of materials exhibiting symmetry properties [51, 53]

16 Conclusions

In this chapter, fundamental mathematical tools have been presented concerning the random matrix theory, which are useful for many problems encountered in uncertainty quantification, in particular for the nonparametric method of the multiscale stochastic modeling of heterogeneous elastic materials, and for the nonparametric stochastic models of uncertainties in computational structural dynamics. The explicit construction of ensembles of random matrices but also the presentation of numerical tools for constructing general ensembles of random matrices are presented and can be used in high dimension. Many applications and validations have

already been performed in many fields of computational sciences and engineering, but the methodologies and tools presented can be used and developed for many other problems for which uncertainties must be quantified.

References

1. Agmon, N., Alhassid, Y., Levine, R.D.: An algorithm for finding the distribution of maximal entropy. *J. Comput. Phys.* **30**, 250–258 (1979)
2. Anderson, T.W.: *An Introduction to Multivariate Statistical Analysis*, 3rd edn. John Wiley & Sons, New York (2003)
3. Arnoux, A., Batou, A., Soize, C., Gagliardini, L.: Stochastic reduced order computational model of structures having numerous local elastic modes in low frequency dynamics. *J. Sound Vib.* **332**(16), 3667–3680 (2013)
4. Arnst, M., Clouet, D., Chebli, H., Othman, R., Degrande, G.: A non-parametric probabilistic model for ground-borne vibrations in buildings. *Probab. Eng. Mech.* **21**(1), 18–34 (2006)
5. Arnst, M., Clouet, D., Bonnet, M.: Inversion of probabilistic structural models using measured transfer functions. *Comput. Methods Appl. Mech. Eng.* **197**(6–8), 589–608 (2008)
6. Au, S.K., Beck, J.L.: Subset simulation and its application to seismic risk based on dynamic analysis. *J. Eng. Mech. – ASCE* **129**(8), 901–917 (2003)
7. Avalos, J., Swenson, E.D., Mignolet, M.P., Lindsley, N.J.: Stochastic modeling of structural uncertainty/variability from ground vibration modal test data. *J. Aircr.* **49**(3), 870–884 (2012)
8. Bathe, K.J., Wilson, E.L.: *Numerical Methods in Finite Element Analysis*. Prentice-Hall, New York (1976)
9. Batou, A., Soize, C.: Experimental identification of turbulent fluid forces applied to fuel assemblies using an uncertain model and fretting-wear estimation. *Mech. Syst. Signal Pr.* **23**(7), 2141–2153 (2009)
10. Batou, A., Soize, C.: Rigid multibody system dynamics with uncertain rigid bodies. *Multibody Syst. Dyn.* **27**(3), 285–319 (2012)
11. Batou, A., Soize, C.: Calculation of Lagrange multipliers in the construction of maximum entropy distributions in high stochastic dimension. *SIAM/ASA J. Uncertain. Quantif.* **1**(1), 431–451 (2013)
12. Batou, A., Soize, C., Audebert, S.: Model identification in computational stochastic dynamics using experimental modal data. *Mech. Syst. Signal Pr.* **50–51**, 307–322 (2014)
13. Batou, A., Soize, C., Corus, M.: Experimental identification of an uncertain computational dynamical model representing a family of structures. *Comput. Struct.* **89**(13–14), 1440–1448 (2011)
14. Bohigas, O., Giannoni, M.J., Schmit, C.: Characterization of chaotic quantum spectra and universality of level fluctuation laws. *Phys. Rev. Lett.* **52**(1), 1–4 (1984)
15. Bohigas, O., Giannoni, M.J., Schmit, C.: Spectral fluctuations of classically chaotic quantum systems. In: Seligman, T.H., Nishioka, H. (eds.) *Quantum Chaos and Statistical Nuclear Physics*, pp. 18–40. Springer, New York (1986)
16. Bohigas, O., Legrand, O., Schmit, C., Sornette, D.: Comment on spectral statistics in elastodynamics. *J. Acoust. Soc. Am.* **89**(3), 1456–1458 (1991)
17. Burrage, K., Lenane, I., Lythe, G.: Numerical methods for second-order stochastic differential equations. *SIAM J. Sci. Comput.* **29**, 245–264 (2007)
18. Capiez-Lernout, E., Soize, C.: Nonparametric modeling of random uncertainties for dynamic response of mistuned bladed disks. *ASME J. Eng. Gas Turbines Power* **126**(3), 600–618 (2004)
19. Capiez-Lernout, E., Soize, C., Lombard, J.P., Dupont, C., Seinturier, E.: Blade manufacturing tolerances definition for a mistuned industrial bladed disk. *ASME J. Eng. Gas Turbines Power* **127**(3), 621–628 (2005)

20. Capiez-Lernout, E., Pellissetti, M., Pradlwarter, H., Schueller, G.I., Soize, C.: Data and model uncertainties in complex aerospace engineering systems. *J. Sound Vib.* **295**(3–5), 923–938 (2006)
21. Capiez-Lernout, E., Soize, C., Mignolet, M.: Post-buckling nonlinear static and dynamical analyses of uncertain cylindrical shells and experimental validation. *Comput. Methods Appl. Mech. Eng.* **271**(1), 210–230 (2014)
22. Capiez-Lernout, E., Soize, C., Mbaye, M.: Mistuning analysis and uncertainty quantification of an industrial bladed disk with geometrical nonlinearity. *J. Sound Vib.* **356**, 124–143 (2015)
23. Chadwick, P., Vianello, M., Cowin, S.C.: A new proof that the number of linear elastic symmetries is eight. *J. Mech. Phys. Solids* **49**, 2471–2492 (2001)
24. Chebli, H., Soize, C.: Experimental validation of a nonparametric probabilistic model of non homogeneous uncertainties for dynamical systems. *J. Acoust. Soc. Am.* **115**(2), 697–705 (2004)
25. Chen, C., Duhamel, D., Soize, C.: Probabilistic approach for model and data uncertainties and its experimental identification in structural dynamics: case of composite sandwich panels. *J. Sound Vib.* **294**(1–2), 64–81 (2006)
26. Cottereau, R., Clouet, D., Soize, C.: Construction of a probabilistic model for impedance matrices. *Comput. Methods Appl. Mech. Eng.* **196**(17–20), 2252–2268 (2007)
27. Cottereau, R., Clouet, D., Soize, C.: Probabilistic impedance of foundation, impact of the seismic design on uncertain soils. *Earthq. Eng. Struct. D.* **37**(6), 899–918 (2008)
28. Das, S., Ghanem, R.: A bounded random matrix approach for stochastic upscaling. *Multiscale Model. Simul.* **8**(1), 296–325 (2009)
29. Desceliers, C., Soize, C., Cambier, S.: Non-parametric – parametric model for random uncertainties in nonlinear structural dynamics – application to earthquake engineering. *Earthq. Eng. Struct. Dyn.* **33**(3), 315–327 (2004)
30. Desceliers, C., Soize, C., Grimal, Q., Talmant, M., Naili, S.: Determination of the random anisotropic elasticity layer using transient wave propagation in a fluid-solid multilayer: model and experiments. *J. Acoust. Soc. Am.* **125**(4), 2027–2034 (2009)
31. Desceliers, C., Soize, C., Naili, S., Haiat, G.: Probabilistic model of the human cortical bone with mechanical alterations in ultrasonic range. *Mech. Syst. Signal Pr.* **32**, 170–177 (2012)
32. Desceliers, C., Soize, C., Yanez-Godoy, H., Houdu, E., Poupart, O.: Robustness analysis of an uncertain computational model to predict well integrity for geologic CO₂ sequestration. *Comput. Mech.* **17**(2), 307–323 (2013)
33. Doob, J.L.: *Stochastic Processes*. John Wiley & Sons, New York (1990)
34. Doostan, A., Iaccarino, G.: A least-squares approximation of partial differential equations with high dimensional random inputs. *J. Comput. Phys.* **228**(12), 4332–4345 (2009)
35. Duchereau, J., Soize, C.: Transient dynamics in structures with nonhomogeneous uncertainties induced by complex joints. *Mech. Syst. Signal Pr.* **20**(4), 854–867 (2006)
36. Dyson, F.J.: Statistical theory of the energy levels of complex systems. Parts I, II, III. *J. Math. Phys.* **3**, 140–175 (1962)
37. Dyson, F.J., Mehta, M.L.: Statistical theory of the energy levels of complex systems. Parts IV, V. *J. Math. Phys.* **4**, 701–719 (1963)
38. Durand, J.F., Soize, C., Gagliardini, L.: Structural-acoustic modeling of automotive vehicles in presence of uncertainties and experimental identification and validation. *J. Acoust. Soc. Am.* **124**(3), 1513–1525 (2008)
39. Fernandez, C., Soize, C., Gagliardini, L.: Fuzzy structure theory modeling of sound-insulation layers in complex vibroacoustic uncertain systems – theory and experimental validation. *J. Acoust. Soc. Am.* **125**(1), 138–153 (2009)
40. Fernandez, C., Soize, C., Gagliardini, L.: Sound-insulation layer modelling in car computational vibroacoustics in the medium-frequency range. *Acta Acust. United Ac.* **96**(3), 437–444 (2010)
41. Fishman, G.S.: *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, New York (1996)

42. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian distribution of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAM I-6**, 721–741 (1984)
43. Ghanem, R., Spanos, P.D.: Polynomial chaos in stochastic finite elements. *J. Appl. Mech. Trans. ASME* **57**(1), 197–202 (1990)
44. Ghanem, R., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
45. Ghanem, R., Spanos, P.D.: *Stochastic Finite Elements: A spectral Approach (rev. edn.)*. Dover Publications, New York (2003)
46. Ghosh, D., Ghanem, R.: Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions. *Int. J. Numer. Methods Eng.* **73**(2), 162–184 (2008)
47. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, Fourth, The Johns Hopkins University Press, Baltimore (2013)
48. Guilleminot, J., Soize, C., Kondo, D.: Mesoscale probabilistic models for the elasticity tensor of fiber reinforced composites: experimental identification and numerical aspects. *Mech. Mater.* **41**(12), 1309–1322 (2009)
49. Guilleminot, J., Noshadravan, A., Soize, C., Ghanem, R.G.: A probabilistic model for bounded elasticity tensor random fields with application to polycrystalline microstructures. *Comput. Methods Appl. Mech. Eng.* **200**, 1637–1648 (2011)
50. Guilleminot, J., Soize, C.: Probabilistic modeling of apparent tensors in elastostatics: a MaxEnt approach under material symmetry and stochastic boundedness constraints. *Probab. Eng. Mech.* **28**(SI), 118–124 (2012)
51. Guilleminot, J., Soize, C.: Generalized stochastic approach for constitutive equation in linear elasticity: a random matrix model. *Int. J. Numer. Methods Eng.* **90**(5), 613–635 (2012)
52. Guilleminot, J., Soize, C., Ghanem, R.: Stochastic representation for anisotropic permeability tensor random fields. *Int. J. Numer. Anal. Met. Geom.* **36**(13), 1592–1608 (2012)
53. Guilleminot, J., Soize, C.: On the statistical dependence for the components of random elasticity tensors exhibiting material symmetry properties. *J. Elast.* **111**(2), 109–130 (2013)
54. Guilleminot, J., Soize, C.: Stochastic model and generator for random fields with symmetry properties: application to the mesoscopic modeling of elastic random media. *Multiscale Model. Simul. (A SIAM Interdiscip. J.)* **11**(3), 840–870 (2013)
55. Gupta, A.K., Nagar, D.K.: *Matrix Variate Distributions*. Chapman & Hall/CRC, Boca Raton (2000)
56. Hairer, E., Lubich, C., G. Wanner, G.: *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, Heidelberg (2002)
57. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **109**, 57–97 (1970)
58. Jaynes, E.T.: Information theory and statistical mechanics. *Phys. Rev.* **106**(4), 620–630 and **108**(2), 171–190 (1957)
59. Kaipio, J., Somersalo, E.: *Statistical and Computational Inverse Problems*. Springer, New York (2005)
60. Kapur, J.N., Kesavan, H.K.: *Entropy Optimization Principles with Applications*. Academic, San Diego (1992)
61. Kassem, M., Soize, C., Gagliardini, L.: Structural partitioning of complex structures in the medium-frequency range. An application to an automotive vehicle. *J. Sound Vib.* **330**(5), 937–946 (2011)
62. Khasminskii, R.: *Stochastic Stability of Differential Equations*, 2nd edn. Springer, Heidelberg (2012)
63. Kloeden, P.E., Platen, E.: *Numerical Solution of Stochastic Differential Equations*. Springer, Heidelberg (1992)
64. Langley, R.S.: A non-Poisson model for the vibration analysis of uncertain dynamic systems. *Proc. R. Soc. Ser. A* **455**, 3325–3349 (1999)
65. Legrand, O., Sornette, D.: Coarse-grained properties of the chaotic trajectories in the stadium. *Physica D* **44**, 229–235 (1990)

66. Legrand, O., Schmit, C., Sornette, D.: Quantum chaos methods applied to high-frequency plate vibrations. *Europhys. Lett.* **18**(2), 101–106 (1992)
67. Le Maître, O.P., Knio, O.M.: *Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics*. Springer, Heidelberg (2010)
68. Luenberger, D.G.: *Optimization by Vector Space Methods*. John Wiley & Sons, New York (2009)
69. Matthies, H.G., Kesse, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**(12–16), 1295–1331 (2005)
70. Mbaye, M., Soize, C., Ousty, J.P., Capiez-Lernout, E.: Robust analysis of design in vibration of turbomachines. *J. Turbomach.* **135**(2), 021008-1–021008-8 (2013)
71. Mehrabadi, M.M., Cowin, S.C.: Eigentensors of linear anisotropic elastic materials. *Q. J. Mech. Appl. Math.* **43**:15–41 (1990)
72. Mehta, M.L.: *Random Matrices and the Statistical Theory of Energy Levels*. Academic, New York (1967)
73. Mehta, M.L.: *Random Matrices*, Revised and Enlarged, 2nd edn. Academic Press, San Diego (1991)
74. Mehta, M.L.: *Random Matrices*, 3rd edn. Elsevier, San Diego (2014)
75. Metropolis, N., Ulam, S.: The Monte Carlo method. *J. Am. Stat. Assoc.* **49**, 335–341 (1949)
76. Mignolet, M.P., Soize, C.: Nonparametric stochastic modeling of linear systems with prescribed variance of several natural frequencies. *Probab. Eng. Mech.* **23**(2–3), 267–278 (2008)
77. Mignolet, M.P., Soize, C.: Stochastic reduced order models for uncertain nonlinear dynamical systems. *Comput. Methods Appl. Mech. Eng.* **197**(45–48), 3951–3963 (2008)
78. Mignolet, M.P., Soize, C., Avalos, J.: Nonparametric stochastic modeling of structures with uncertain boundary conditions/coupling between substructures. *AIAA J.* **51**(6), 1296–1308 (2013)
79. Murthy, R., Mignolet, M.P., El-Shafei, A.: Nonparametric stochastic modeling of uncertainty in rotordynamics-Part I: Formulation. *J. Eng. Gas Turb. Power* **132**, 092501-1–092501-7 (2009)
80. Murthy, R., Mignolet, M.P., El-Shafei, A.: Nonparametric stochastic modeling of uncertainty in rotordynamics-Part II: applications. *J. Eng. Gas Turb. Power* **132**, 092502-1–092502-11 (2010)
81. Murthy, R., Wang, X.Q., Perez, R., Mignolet, M.P., Richter, L.A.: Uncertainty-based experimental validation of nonlinear reduced order models. *J. Sound Vib.* **331**, 1097–1114 (2012)
82. Murthy, R., Tomei, J.C., Wang, X.Q., Mignolet, M.P., El-Shafei, A.: Nonparametric stochastic modeling of structural uncertainty in rotordynamics: Unbalance and balancing aspects. *J. Eng. Gas Turb. Power* **136**, 62506-1–62506-11 (2014)
83. Neal, R.M.: Slice sampling. *Ann. Stat.* **31**, 705–767 (2003)
84. Nouy, A.: Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations. *Arch. Comput. Methods Eng.* **16**(3), 251–285 (2009)
85. Nouy, A.: Proper Generalized Decomposition and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Eng.* **16**(3), 403–434 (2010)
86. Nouy, A., Soize, C.: Random fields representations for stochastic elliptic boundary value problems and statistical inverse problems. *Eur. J. Appl. Math.* **25**(3), 339–373 (2014)
87. Ohayon, R., Soize, C.: *Structural Acoustics and Vibration*. Academic, San Diego (1998)
88. Ohayon, R., Soize, C.: Advanced computational dissipative structural acoustics and fluid-structure interaction in low- and medium-frequency domains. Reduced-order models and uncertainty quantification. *Int. J. Aeronaut. Space Sci.* **13**(2), 127–153 (2012)
89. Ohayon, R., Soize, C.: *Advanced Computational Vibroacoustics. Reduced-Order Models and Uncertainty Quantification*. Cambridge University Press, New York (2014)
90. Papoulis, A.: *Signal Analysis*. McGraw-Hill, New York (1977)

91. Pellissetti, M., Capiez-Lernout, E., Pradlwarter, H., Soize, C., Schueller, G.I.: Reliability analysis of a satellite structure with a parametric and a non-parametric probabilistic model. *Comput. Methods Appl. Mech. Eng.* **198**(2), 344–357 (2008)
92. Peter, C.E.: Statistical Theories of Spectra: Fluctuations. Academic, New York (1965)
93. Pradlwarter, H.J., Schueller, G.I.: Local domain Monte Carlo simulation. *Struct. Saf.* **32**(5), 275–280 (2010)
94. Ritto, T.G., Soize, C., Rochinha, F.A., Sampaio, R.: Dynamic stability of a pipe conveying fluid with an uncertain computational model. *J. Fluid Struct.* **49**, 412–426 (2014)
95. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2005)
96. Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo Method, 2nd edn. John Wiley & Sons, New York (2008)
97. Sakji, S., Soize, C., Heck, J.V.: Probabilistic uncertainties modeling for thermomechanical analysis of plasterboard submitted to fire load. *J. Struct. Eng. – ASCE* **134**(10), 1611–1618 (2008)
98. Sakji, S., Soize, C., Heck, J.V.: Computational stochastic heat transfer with model uncertainties in a plasterboard submitted to fire load and experimental validation. *Fire Mater.* **33**(3), 109–127 (2009)
99. Schmit, C.: Quantum and classical properties of some billiards on the hyperbolic plane. In: Giannoni, M.J., Voros, A., Zinn-Justin, J. (eds.) *Chaos and Quantum Physics*, pp. 333–369. North-Holland, Amsterdam (1991)
100. Schueller, G.I.: Efficient Monte Carlo simulation procedures in structural uncertainty and reliability analysis – recent advances. *Struct. Eng. Mech.* **32**(1), 1–20 (2009)
101. Schwartz, L.: *Analyse II Calcul Différentiel et Equations Différentielles*. Hermann, Paris (1997)
102. Serfling, R.J.: Approximation Theorems of Mathematical Statistics. John Wiley & Sons, New York (1980)
103. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423, 623–659 (1948)
104. Soize, C.: Oscillators submitted to squared Gaussian processes. *J. Math. Phys.* **21**(10), 2500–2507 (1980)
105. Soize, C.: The Fokker-Planck Equation for Stochastic Dynamical Systems and its Explicit Steady State Solutions. World Scientific Publishing Co Pte Ltd, Singapore (1994)
106. Soize, C.: A nonparametric model of random uncertainties in linear structural dynamics. In: Bouc R., Soize, C. (eds.) *Progress in Stochastic Structural Dynamics*, pp. 109–138. Publications LMA-CNRS, Marseille (1999). ISBN 2-909669-16-5
107. Soize, C.: A nonparametric model of random uncertainties for reduced matrix models in structural dynamics. *Probab. Eng. Mech.* **15**(3), 277–294 (2000)
108. Soize, C.: Maximum entropy approach for modeling random uncertainties in transient elastodynamics. *J. Acoust. Soc. Am.* **109**(5), 1979–1996 (2001)
109. Soize, C.: Random matrix theory and non-parametric model of random uncertainties. *J. Sound Vib.* **263**(4), 893–916 (2003)
110. Soize, C.: Random matrix theory for modeling random uncertainties in computational mechanics. *Comput. Methods Appl. Mech. Eng.* **194**(12–16), 1333–1366 (2005)
111. Soize, C.: Non Gaussian positive-definite matrix-valued random fields for elliptic stochastic partial differential operators. *Comput. Methods Appl. Mech. Eng.* **195**(1–3), 26–64 (2006)
112. Soize, C.: Construction of probability distributions in high dimension using the maximum entropy principle. Applications to stochastic processes, random fields and random matrices. *Int. J. Numer. Methods Eng.* **76**(10), 1583–1611 (2008)
113. Soize, C.: Generalized Probabilistic approach of uncertainties in computational dynamics using random matrices and polynomial chaos decompositions. *Int. J. Numer. Methods Eng.* **81**(8), 939–970 (2010)
114. Soize, C.: *Stochastic Models of Uncertainties in Computational Mechanics*. American Society of Civil Engineers (ASCE), Reston (2012)

115. Soize, C., Poloskov, I.E.: Time-domain formulation in computational dynamics for linear viscoelastic media with model uncertainties and stochastic excitation. *Comput. Math. Appl.* **64**(11), 3594–3612 (2012)
116. Soize, C., Chebli, H.: Random uncertainties model in dynamic substructuring using a nonparametric probabilistic model. *J. Eng. Mech.-ASCE* **129**(4), 449–457 (2003)
117. Spall, J.C.: *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Hoboken (2003)
118. Talay, D., Tubaro, L.: Expansion of the global error for numerical schemes solving stochastic differential equation. *Stoch. Anal. Appl.* **8**(4), 94–120 (1990)
119. Talay, D.: Simulation and numerical analysis of stochastic differential systems. In: Kree, P., Wedig, W. (eds.) *Probabilistic Methods in Applied Physics*. Lecture Notes in Physics, vol. 451, pp. 54–96. Springer, Heidelberg (1995)
120. Talay, D.: Stochastic Hamiltonian system: exponential convergence to the invariant measure and discretization by the implicit Euler scheme. *Markov Process. Relat. Fields* **8**, 163–198 (2002)
121. Tipireddy, R., Ghanem, R.: Basis adaptation in homogeneous chaos spaces. *J. Comput. Phys.* **259**, 304–317 (2014)
122. Walpole, L.J.: Elastic behavior of composite materials: theoretical foundations. *Adv. Appl. Mech.* **21**, 169–242 (1981)
123. Walter, E., Pronzato, L.: *Identification of Parametric Models from Experimental Data*. Springer, Berlin (1997)
124. Weaver, R.L.: Spectral statistics in elastodynamics. *J. Acoust. Soc. Am.* **85**(3), 1005–1013 (1989)
125. Wigner, E.P.: On the statistical distribution of the widths and spacings of nuclear resonance levels. *Proc. Camb. Philos. Soc.* **47**, 790–798 (1951)
126. Wigner, E.P.: Distribution laws for the roots of a random Hermitian matrix In: Pöter, C.E. (ed.) *Statistical Theories of Spectra: Fluctuations*, pp. 446–461. Academic, New York (1965)
127. Wright, M., Weaver, R.: *New Directions in Linear Acoustics and Vibration. Quantum Chaos, Random Matrix Theory, and Complexity*. Cambridge University Press, New York (2010)
128. Zienkiewicz, O.C., Taylor, R.L.: *The Finite Element Method For Solid and Structural Mechanics*, Sixth edition. Elsevier, Butterworth-Heinemann, Amsterdam (2005)

Maximin Sliced Latin Hypercube Designs with Application to Cross Validating Prediction Error

9

Yan Chen, David M. Steinberg, and Peter Qian

Abstract

This paper introduces an approach to construct a new type of design, called a maximin sliced Latin hypercube design. This design is a special type of Latin hypercube design that can be partitioned into smaller slices of Latin hypercube designs, where both the whole design and each slice are optimal under the maximin criterion. To construct these designs, a two-step construction method for generating sliced Latin hypercubes is proposed. Several examples are presented to evaluate the performance of the algorithm. An application of this type of optimal design in estimating prediction error by cross validation is also illustrated here.

Keywords

Computer experiments • Maximin design • Enhanced stochastic evolutionary algorithm • Design of experiments

Contents

1	Introduction	290
2	Notation and Definition	291
2.1	Sliced Latin Hypercube	291
2.2	Optimality Criteria	293
2.3	The Enhanced Stochastic Evolutionary Algorithm	293
2.4	An Alternate Construction Method for SLHDs	294
3	A Motivating Example	295

Y. Chen (✉) • P. Qian
University of Wisconsin-Madison, Madison, WI, USA
e-mail: chenyan@stat.wisc.edu; peterq@stat.wisc.edu

D.M. Steinberg
Tel Aviv University, Tel Aviv, Israel
e-mail: dms@post.tau.ac.il

4 Construction of Maximin SLHD	297
5 Numerical Illustration	298
6 Application to the Estimation of Prediction Error	302
6.1 Evaluation of Multiple Computer Models	302
6.2 Cross Validation of Prediction Error	303
7 Conclusion	308
References	308

1 Introduction

Computer experiments are widely used in science and engineering to model complex physical phenomena. The Latin hypercube design (LHD), introduced in [8], is a popular choice for computer experiments. When an LHD with n points on the input space $(0, 1)^q$ is projected onto any single dimension, precisely one point falls into each of the n intervals: $(0, \frac{1}{n}], (\frac{1}{n}, \frac{2}{n}], \dots, (\frac{n-1}{n}, 1]$ for that direction. For simplicity, this paper only considers a special type of LHD, where the projection of any point is the midpoint of one of the evenly spaced intervals.

It is common to apply a secondary criterion in choosing a Latin hypercube design from among the many that exist. One popular criterion is called *maximin* [5], which evaluates a design in terms of the minimum distance between any pair of design points:

$$\min_{1 \leq i < j \leq n} d_{ij}, \quad (9.1)$$

where d_{ij} is the Euclidean distance between the i th point and the j th point of the design. The asymptotic optimality of maximin designs is also discussed in [5]. A related criterion, called ϕ_p , is proposed in [10], which is scalar valued and computationally cheaper than criterion (9.1).

The *sliced Latin hypercube design* (SLHD) was introduced in [13] for collective evaluation of computer models. An SLHD is a special Latin hypercube design that can be partitioned into slices of smaller Latin hypercube designs, which can be constructed by random permutation of each column in the design matrix [13]. As a result, the generated design is not optimal under any given criterion. This chapter focuses on constructing a “best” SLHD which preserves the maximin optimality not only as one LHD but also in each slice. Note that the existing criteria for optimal LHDs, such as the maximin criterion (9.1), are not directly applicable here, as it cannot control the distance in each slice and in the whole design simultaneously. Also, in algorithmic searches for optimal designs, new designs are often obtained by exchanging two elements in one column, which could destroy the sliced structure in SLHDs. These issues motivate us to define a new maximin criterion for SLHD and to modify the enhanced stochastic evolutionary algorithm [4] to construct this new maximin SLHD.

The chapter is organized as follows. Relevant definitions and notations are given in Sect. 2. An example that motivates the proposed maximin criterion for SLHDs is shown in Sect. 3. The algorithm for constructing maximin SLHDs is introduced in Sect. 4 and evaluated with a simulation study in Sect. 5. An application of maximin SLHDs in cross validation of prediction error for Gaussian process regression is discussed in Sect. 6. The conclusions can be found in Sect. 7.

2 Notation and Definition

2.1 Sliced Latin Hypercube

A Latin hypercube of n points in q dimensions, as defined in [8], is an $n \times q$ matrix with a permutation of the integers $\{1, 2, \dots, n\}$ in each column. Given an n by q Latin hypercube X , an LHD can be constructed as

$$(X - U)/n, \quad (9.2)$$

where the entries of U follow independent uniform distributions on $[0, 1)$. Similarly, SLHDs are generated using sliced Latin hypercubes in [13]. For any positive integers t and m with $n = mt$, an n by q matrix X is called a sliced Latin hypercube with t slices if two conditions are satisfied: (i) X is a Latin hypercube of size n ; (ii) when dividing X by rows into small blocks of $m \times q$ matrixes $\{X^{(1)}, X^{(2)}, \dots, X^{(t)}\}$, each $\lceil X^{(i)}/t \rceil$ is a smaller Latin hypercube of m runs, where $\lceil X^{(i)}/t \rceil$ is obtained by taking the smallest integer no less than the corresponding element of $X^{(i)}/t$.

As a stepping stone to generating an n by q sliced Latin hypercube with t slices, [13] proposed an algorithm to construct q independent m by t sliced permutation matrices, which they denote as $SPM(m, t)$. $SPM(m, t)$ is constructed by the following steps. First, divide the elements of $Z_n = \{1, 2, \dots, n\}$ into m blocks, b_1, \dots, b_m , where $b_i = \{a \in Z_n | \lceil a/t \rceil = i\}$. Next, generate $SPM(m, t)$ in two steps: (1) for $i = 1, \dots, m$, fill the i th row of an $m \times t$ empty matrix H with a uniform permutation on the set b_i , with the permutations carried out independently in each row; (2) for $j = 1, \dots, t$, randomly shuffle the entries in the j th column of H , with the permutations carried out independently in each column.

Let H_1, \dots, H_q be q m by t sliced permutation matrices. For $i = 1, 2, \dots, t$, $X^{(i)}$ can be constructed by letting its j th column be the i th column of H_j , for $j = 1, 2, \dots, q$. Then the sliced Latin hypercube X can be generated by combining $X^{(1)}, \dots, X^{(t)}$ row by row. Finally, an SLHD is constructed by (9.2). For simplicity, this paper only considers a special class of SLHDs where each element of U is fixed to be 0.5. Since the algorithm carries out m independent permutations on t numbers in step 1 and t independent permutations on m numbers in step 2, there are $[(m!)^t(t!)^m]^q$ sliced Latin hypercubes in total among $(n!)^q$ different Latin hypercubes. An alternative construction algorithm is discussed in Sect. 2.4.

2.1.1 Sampling Properties of SLHD

Consider an experiment using t similar computer models, f_1, f_2, \dots, f_t . Assume each f_i has input \mathbf{x} , which is uniformly distributed on $(0, 1]^q$. For $i = 1, \dots, t$, define $\mu_i = E[f_i(\mathbf{x})]$. Given the value of each f_i at m selected input values, the goal is to estimate μ_1, \dots, μ_t and a linear combination of μ_1, \dots, μ_t . For $0 \leq \lambda_i \leq 1$, $i = 1, \dots, t$, the linear combination is defined as

$$\eta = \sum_{i=1}^t \lambda_i \mu_i. \quad (9.3)$$

Consider three different schemes for selecting the input values: (i) IID, take an independent and identically distributed sample of m runs for each f_i , with the t samples generated independently; (ii) LHD, obtain t independent ordinary Latin hypercube designs of m runs, each of which is associated with one f_i ; (iii) SLHD, generate an $n \times q$ SLHD with t slices, where each slice is assigned to one f_i .

For any of these schemes, let $\mathbf{D}^{(i)}$ denote the design set for f_i , $i = 1, \dots, t$. Denote the k th row of $\mathbf{D}^{(i)}$ as $\mathbf{d}_k^{(i)}$. For $i = 1, \dots, t$, μ_i is estimated by

$$\hat{\mu}_i = \frac{1}{m} \sum_{k=1}^m f_i(\mathbf{d}_k^{(i)}), \quad (9.4)$$

and η is estimated by

$$\hat{\eta} = \sum_{i=1}^t \lambda_i \hat{\mu}_i. \quad (9.5)$$

The following theorem, taken from [13], shows the effect of the design set on the variance of the $\hat{\mu}_i$ and $\hat{\eta}$.

Theorem 1. Suppose that for $i = 1, \dots, t$, $f_i(\mathbf{x})$ is monotonic in each argument x_j of $\mathbf{x} = (x_1, \dots, x_q)$, and any pair of functions f_{i_1} and f_{i_2} , $i_1 \neq i_2$, are either both increasing or both decreasing in each argument x_j of \mathbf{x} . Then, it is derived in [13] that

(i) for $i = 1, \dots, t$ and $\hat{\mu}_i$ defined in (9.4),

$$\text{var}_{\text{SLHD}}(\hat{\mu}_i) \leq \text{var}_{\text{IID}}(\hat{\mu}_i); \text{ and} \quad (9.6)$$

(ii) for $\hat{\eta}$ defined in (9.5),

$$\text{var}_{\text{SLHD}}(\hat{\eta}) \leq \text{var}_{\text{LHD}}(\hat{\eta}) \leq \text{var}_{\text{IID}}(\hat{\eta}) \quad (9.7)$$

2.2 Optimality Criteria

The optimal design defined by the original maximin criterion (9.1) is not unique. To break ties among all the maximin designs, the following criterion is proposed in [10]. For a given design X , define a distance list $\mathbf{d} = (d_1, d_2, \dots, d_k)$ in which the elements are the distinct, increasing values of the distances between two rows in X . Let $\mathbf{J} = (J_1, J_2, \dots, J_k)$ be the corresponding index list, where J_i is the number of pairs of sites in the design separated by distance d_i . Under the generalized maximin criterion, design X is optimal if it sequentially maximizes d_i 's and minimizes J_i 's in the following order: $d_1, J_1, d_2, J_2, \dots, d_k, J_k$. A scalar-valued design criterion was also proposed in [10] as a computationally cheaper substitute for the maximin criterion:

$$\phi_p = \left[\sum_{i=1}^k J_i d_i^{-p} \right]^{1/p}, \quad (9.8)$$

where p is a positive integer. For large enough p , the designs that minimize ϕ_p are optimal designs under the generalized maximin criterion.

2.3 The Enhanced Stochastic Evolutionary Algorithm

To construct maximin SLHDs, the enhanced stochastic evolutionary algorithm in [4] needs to be modified. The original algorithm is a general method that can be applied to find a design X_{best} that is optimal with respect to a criterion $f(\cdot)$. The algorithm searches through the design space by exchanging elements within one column and deciding whether to accept the new design with a threshold-based criterion. As shown below, it consists of double loops in which the inner loop decides whether to accept a new design, while the outer loop controls the entire optimization process by adjusting the threshold T_h in the acceptance criterion of the inner loop.

Some guidelines are provided in [4] for the choices of J and M . Typically, J is set to be $n_e/5$ but no larger than 50, where n_e is the number of all possible distinct element changes in one column of the design. For an LHD of size n , it is $\binom{n}{2}$. The recommended choice for the parameter M is $2n_e q/J$ but no larger than 100, where q is the number of factors in the design.

The outer loop controls threshold T_h in the acceptance criterion of the inner loop. Initially, T_h is set to be a small value, then it will be adjusted based on whether one cycle of the inner loop could actually improve the current design. In the improving process, T_h is maintained on a small value so that only better design or slightly worse design will be accepted. In particular, T_h will be decreased if $n_{acpt}/M > 10\%$ and $n_{impt} < n_{acpt}$; T_h will be increased if $n_{acpt}/M \leq 10\%$; and T_h will remain the same otherwise. In the exploration process, if $n_{acpt}/M \leq 10\%$, T_h will be quickly increased until $n_{acpt}/M > 10\%$, after which T_h will be slowly decreased until $n_{acpt}/M > 10\%$.

Algorithm 1 The enhanced stochastic evolutionary algorithm

Input: An initial design X_0 , the optimality criterion $f(\cdot)$ to be minimized.
Initialization: $X = X_0$, $X_{best} = X$, $T_h = T_{h0}$, $i = 0$.

```

while  $i < N$  do
     $X_{old.best} = X_{best}$ ,
     $n_{acpt} = 0$ ,  $n_{impt} = 0$ ,  $flag_{imp} = 0$ ,  $j = 0$ .
    while  $j < M$  do
        Randomly pick  $J$  distinct element exchanges within the current column ( $j \bmod q$ ),
        and choose the best design  $X_{try}$  based on  $f(\cdot)$ .
        if  $f(X_{try}) - f(X) \leq T_h \cdot \text{uniform}[0, 1]$  then
             $X \leftarrow X_{try}$ ,  $n_{acpt} \leftarrow n_{acpt} + 1$ .
            if  $f(X) < f(X_{best})$  then
                 $|X_{best} \leftarrow X$ ,  $n_{impt} \leftarrow n_{impt} + 1$ .
            end
        end
         $j \leftarrow j + 1$ 
    end
    if  $f(X_{old.best}) - f(X_{best}) > tol$  then
         $|flag_{imp} = 1$ .
    end
    Update  $T_h$  based on  $n_{acpt}$ ,  $n_{impt}$  and  $flag_{imp}$ .
     $i \leftarrow i + 1$ .
end
```

Algorithm 2 Two-step Construction for Sliced Latin Hypercube

Initialize: n by q arrays \mathbf{B} , \mathbf{C} .
for $i \leftarrow 1$ **to** t **do**
 Generate $\mathbf{B}^{(i)}$ as an m by q Latin hypercube.
 Set the i th block of \mathbf{B} to be $\mathbf{B}^{(i)}$.
end
for $i \leftarrow 1$ **to** q **do**
for $j \leftarrow 1$ **to** m **do**
 Find the entries with value j in the i th column of \mathbf{B} ,
 set the corresponding entries in \mathbf{C} to be a permutation on $\{0, 1, \dots, t - 1\}$.
 end
end
Construct a sliced Latin hypercube $X = t\mathbf{B} - \mathbf{C}$.

2.4 An Alternate Construction Method for SLHDs

In order to break down the proposed algorithm for construction of maximin SLHDs into two stages, Algorithm 2 is introduced here as a new construction for an n by q sliced Latin hypercube with t slices and m points in each slice.

Lemma 1. *The array generated by Algorithm 2 is a sliced Latin hypercube of size n , with t slices and m points in each slice.*

Proof. Since \mathbf{B} consists of t Latin hypercubes of size m , there are t entries in each column of \mathbf{B} with value i , for $i \in \{1, 2, \dots, m\}$. By construction, the corresponding entries in \mathbf{C} are set to be a permutation on $\{0, 1, \dots, t-1\}$. As a result, each column of \mathbf{X} takes value from $\{ti - j \mid i = 1, 2, \dots, m; j = 0, 1, \dots, t-1\} = \{1, 2, \dots, n\}$. By definition, the resulting design \mathbf{X} is a Latin hypercube of size n .

Furthermore, the i th block of $\lceil \mathbf{X}/t \rceil$ is $\mathbf{B}^{(i)}$, an m by q Latin hypercube. Hence, design \mathbf{X} is a sliced Latin hypercube with t slices and m points in each slice. \square

This alternate method generates the design slice by slice, which provides flexibility for optimizing smaller designs as part of the construction of maximin SLHDs. The proposed algorithm in Sect. 4 is therefore based on this construction for SLHDs.

Example 1. To illustrate this construction method, consider the generation of an SLHD with $m = 3$, $t = 4$, and $q = 2$.

Step 1: Combine four randomly generated 3 by 2 Latin hypercubes to get \mathbf{B} (in transpose):

$$\mathbf{B}^T = \left(\begin{array}{ccc|cc|cc|cc} 1 & 2 & 3 & 2 & 1 & 3 & 1 & 2 & 3 & 1 & 2 \\ 2 & 3 & 1 & 1 & 3 & 2 & 1 & 3 & 2 & 3 & 2 \\ \end{array} \right).$$

Step 2: Each column of \mathbf{C} contains three permutations on $\{0, 1, 2, 3\}$, where each of the 12 combinations between $\{1, 2, 3\}$ and $\{0, 1, 2, 3\}$ appears exactly once for the pairing between one column of \mathbf{B} and \mathbf{C} . Here,

$$\mathbf{C}^T = \left(\begin{array}{ccc|cc|cc|cc} 0 & 1 & 3 & 3 & 1 & 1 & 2 & 2 & 0 & 2 & 3 & 0 \\ 0 & 2 & 2 & 0 & 0 & 1 & 3 & 3 & 3 & 1 & 0 & 2 & 1 \\ \end{array} \right).$$

$$\text{So } \mathbf{X}^T = 4 * \mathbf{B}^T - \mathbf{C}^T = \left(\begin{array}{ccc|cc|cc|cc} 4 & 7 & 9 & 5 & 3 & 11 & 2 & 6 & 12 & 10 & 1 & 8 \\ 8 & 10 & 2 & 4 & 11 & 5 & 1 & 9 & 7 & 12 & 6 & 3 \\ \end{array} \right).$$

Since \mathbf{B} contains t permutations on m integers in each column and \mathbf{C} contains m permutations on t integers in each column, this construction method can generate all $[(m!)^t (t!)^m]^q$ different sliced Latin hypercube designs defined in [13]. For any randomly generated \mathbf{X} , the corresponding pair (\mathbf{B}, \mathbf{C}) can be recovered by $\mathbf{B} = \lceil \mathbf{X}/t \rceil$, $\mathbf{C} = t\mathbf{B} - \mathbf{X}$.

3 A Motivating Example

In this section, the general ideas of maximin SLHDs are derived based on the comparison of four SLHDs with two slices, given in Fig. 9.1. Black circles and red squares are used to represent different slices. Design (a) is based on a sliced Latin hypercube generated by random permutation as in [13], while design (d) is constructed by the proposed algorithm in Sect. 4. Design (b) is obtained by

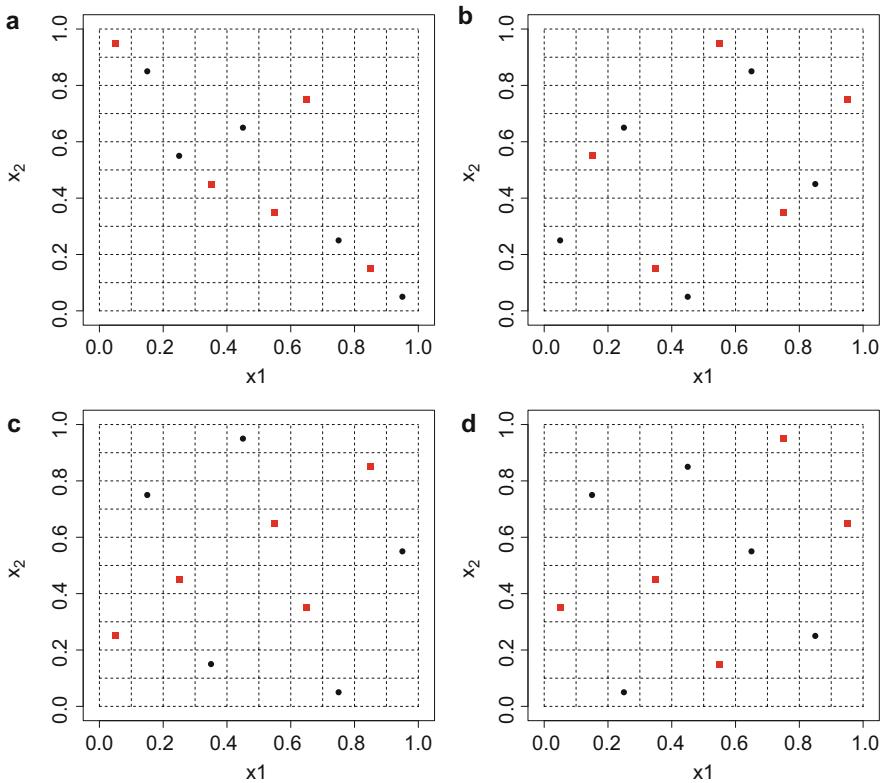


Fig. 9.1 Example SLHDs with ten runs, two factors, and two slices

generating two maximin LHDs by the enhanced stochastic evolutionary algorithm and then combining the two to get an LHD. Design (c) is constructed by splitting a 10-run maximin LHD into two slices arbitrarily.

The four designs are evaluated based on the maximin distance criterion evaluated on the whole design and also on each slice alone. Design (a) is undesirable, considering the whole design or one particular slice alone. Design (b) is better than (a) in the sense that both slices are good designs based on the maximin criterion. However, when putting the two slices together, some points lie very close to one another. This means maximin SLHDs cannot be constructed by working on different slices separately. Design (c) is the opposite case of (b): the whole design is not far from a maximin LHD, but for the slice in red squares, the inter-site distance is small for some points. This shows that the enhanced stochastic evolutionary algorithm cannot be directly used for the construction of maximin SLHDs because of the sliced structure. Design (d) is the best: not only the whole design but also each slice alone has good space-filling properties under the maximin criterion. Since SLHDs have a sliced structure, it is natural to define a maximin SLHD to be a design that has control of the minimum distance in each slice as well as in the whole design. To

construct such designs, it is necessary to define a new maximin criterion for SLHDs and to modify the enhanced stochastic evolutionary algorithm to optimize for such a design.

4 Construction of Maximin SLHD

A maximin SLHD should maximize the minimum distance in the whole design and in each slice simultaneously. The original ϕ_p criterion in (9.8) is generalized for SLHDs as follows:

$$\phi_p^s(\mathbf{X}) = w \left[\sum_{i=1}^t \phi_p(\mathbf{X}^{(i)})/t \right] + (1-w)\phi_p(\mathbf{X}), \quad (9.9)$$

where \mathbf{X} is an SLHD with t slices $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(t)}\}$, and w is the weight to be specified. An SLHD is defined to be a maximin SLHD if it minimizes criterion $\phi_p^s(\cdot)$ among all possible SLHDs of the same size.

Based on the alternative construction of sliced Latin hypercubes proposed in Sect. 2.4, the algorithm for constructing maximin SLHDs contains two steps: (1) generate different slices in $\mathbf{B} = \lceil \mathbf{X}/t \rceil$ as t independent maximin LHDs; (2) construct optimal \mathbf{C} while keeping the between-slice distance structure of $\lceil \mathbf{X}/t \rceil$. The optimal \mathbf{C} is defined by the following criterion:

$$\psi_p = \left[\sum_{1 \leq i, j \leq n, j-i > m} (d_{ij})^{-p} \right]^{1/p}, \quad (9.10)$$

where the constraint $\{j - i > m\}$ ensures that d_{ij} measures the distances between points from different slices in \mathbf{X} . This ψ_p works in the same way as ϕ_p in (9.8) but only controls the between-slice distance. The details of the proposed method for constructing maximin SLHDs are given in Algorithm 3.

A simple modification can be made for the enhanced stochastic evolutionary algorithm to construct optimal sliced Latin hypercubes. The element exchanges within one column could be restricted to those that could keep the sliced structure. There are two types of such “sliced exchanges” within one column: (i) exchange two elements within the same slice; (ii) exchange two elements in different slices if they correspond to the same level in $\lceil \mathbf{X}/t \rceil$. For example, for a sliced Latin hypercube with three slices, levels $\{1, 2, 3\}$ in the same column of the design can be switched, because $\lceil 1/3 \rceil = \lceil 2/3 \rceil = \lceil 3/3 \rceil = 1$. Algorithm 3 is therefore compared to this modified enhanced stochastic evolutionary algorithm called “SlicedEx,” where new designs are obtained by the two types of element exchanges that keep the structure in sliced Latin hypercubes.

The proposed maximin SLHD construction is closely related to a similar proposal in a recent article [1]. Both papers independently proposed the same maximin

Algorithm 3 Constructing Maximin Sliced Latin Hypercube.

Step 1:

Generate different slices of \mathbf{B} as t independent optimal Latin hypercubes under the ϕ_p criterion in (9.8) using the enhanced stochastic evolutionary algorithm.

Step 2:

- For $j = 1, \dots, q$, generate \mathbf{p}_j to be a random permutation on $\{0, 1, \dots, t - 1\}$. Set all elements of in the i th slice of the j th column of \mathbf{C} to be the i th element of \mathbf{p}_j , for $i = 1, 2, \dots, t$.
- Construct new \mathbf{C} by exchanging elements in two different slices that correspond to the same levels in \mathbf{B} , where the new design $\mathbf{X} = t\mathbf{B} - \mathbf{C}$ is evaluated by ψ_p in (9.10) in the enhanced stochastic evolutionary algorithm.

Obtain $\mathbf{X} = t\mathbf{B} - \mathbf{C}$ as the maximin sliced Latin hypercube.

criterion for SLHDs, as shown in (9.9), but used different construction algorithms. The main differences of the two algorithms are summarized as follows, where the actual numerical comparison is shown in Sect. 5. (i) The algorithm in [1] is based on a standard simulated annealing (SA) algorithm established in [7], which accepts a new design \mathbf{X}_{try} with probability $\exp\{-[f(\mathbf{X}_{try}) - f(\mathbf{X})]/T_h\}$, where T_h will be monotonically reduced by a cooling schedule. (ii) The first stage of Algorithm 3 constructs each slice of \mathbf{B} separately, whereas the method in [1] directly works on \mathbf{B} , avoiding duplicated rows by swapping elements. (iii) The second stage in Algorithm 3 uses (9.10) to control between-slice distances, whereas the optimality criterion used in [1] is still (9.9). (iv) The algorithm in [1] uses a more efficient way to update $\phi_p(\mathbf{X}_{try})$, which allows it to be faster for large designs. The application of the proposed maximin SLHDs in this chapter is mainly used for cross validating prediction errors, whereas the design constructed in [1] is used for fitting computer experiments with both continuous and categorical variables.

5 Numerical Illustration

The following examples are used for construction of an n by q maximin SLHD \mathbf{D} with t slices and m points in each slice in the following examples. To evaluate the performance of the proposed algorithm and “SlicedEx,” two other methods, called “Combine” and “Split,” are also performed. The procedure “Combine” simply puts t maximin Latin hypercubes together to get a Latin hypercube. This design would provide a reference for evaluating each slice of a maximin SLHD. Method “Split” randomly splits a maximin Latin hypercube into slices and is used to provide a reference value for the whole design. The four methods are coded by Matlab, with details listed below. In addition, the algorithm in [1] is also performed using R package “SLHD” (contributed by the authors of [1]), with results summarized in tables under the column named “BA-SA.”

- Proposed: Algorithm 3 in Sect. 4.
- SlicedEx: in the enhanced stochastic evolutionary algorithm, construct a new design by “sliced exchanges” that keep the sliced structure; evaluate new designs by ϕ_p defined in (9.8).
- Combine: combine t small maximin Latin hypercubes to get one Latin hypercube.
- Split: split a maximin Latin hypercube into t slices, where each slice is not necessarily a Latin hypercube.

This chapter considers a specific type of SLHD constructed by

$$\mathbf{D} = (X - 0.5)/n, \quad (9.11)$$

where X is a sliced Latin hypercube. Hence, the projection of the design onto any factor is the midpoints of n evenly spaced intervals on $[0, 1]$. Each of the five algorithms specified below is used to generate X first, then the resulting design \mathbf{D} is obtained by formula (9.11). The optimality criteria in (9.8), (9.9), and (9.10) are evaluated on \mathbf{D} in the search process of the algorithms. By adjusting parameters of the enhanced stochastic evolutionary algorithm, the time for constructing the designs is set to be close for all the algorithms. Let \mathbf{D}_1 denote the worst slice in \mathbf{D} under criterion ϕ_p in (9.8). Each procedure is replicated 50 times. The means of $\phi_p(\mathbf{D})$, $\phi_p^s(\mathbf{D})$, $\phi_p(\mathbf{D}_1)$, and CPU time are summarized in tables, where p is set to be 30 in each criterion, and the weight w is chosen to be 1/2. Boxplots of $\phi_p(\mathbf{D})$ and $\phi_p(\mathbf{D}_1)$ are also given. The “box” goes from the first quartile to the third quartile, and the line within the box indicates the median of the data set. Boxplots may also have lines extending vertically from the boxes (whiskers) showing variability outside the upper and lower quartiles.

Example 1. Consider the construction of a maximin SLHD with 50 runs, 2 factors, and 5 slices. The results are shown in Table 9.1 and Fig. 9.2.

Example 2. Consider the construction of a maximin SLHD with 80 runs, 2 factors, and 8 slices. The results are shown in Table 9.2 and Fig. 9.3.

Example 3. Consider the construction of a maximin SLHD with 120 runs, 2 factors, and 12 slices. The results are shown in Table 9.3 and Fig. 9.4.

Table 9.1 Construction of maximin SLHDs with 50 runs, 2 factors, and 5 slices

Method	Proposed	SlicedEx	Combine	Split	BA-SA
$\phi_p(\mathbf{D})$	12.41	9.688	34.20	9.183	10.21
$\phi_p(\mathbf{D}_1)$	4.474	8.375	3.629	8.454	4.517
$\phi_p^s(\mathbf{D})$	8.212	8.538	18.87	8.485	7.177
CPU Time(s)	6.700	6.436	6.457	6.906	6.01

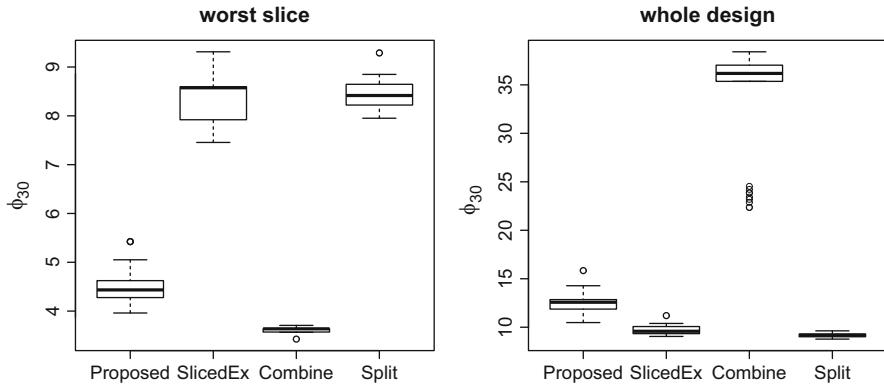


Fig. 9.2 Boxplot of $\phi_p(\mathbf{D}_1)$ and $\phi_p(\mathbf{D})$ for maximin SLHDs with 50 runs, 2 factors, and 5 slices

Table 9.2 Construction of maximin SLHDs with 80 runs, 2 factors, and 8 slices

Method	Proposed	SlicedEx	Combine	Split	BA-SA
$\phi_p(\mathbf{D})$	14.54	12.60	53.96	12.95	12.03
$\phi_p(\mathbf{D}_1)$	4.629	10.39	3.629	11.50	4.916
$\phi_p^s(\mathbf{D})$	9.277	10.17	28.74	11.23	8.167
CPU Time(s)	16.95	14.29	15.20	13.43	14.12

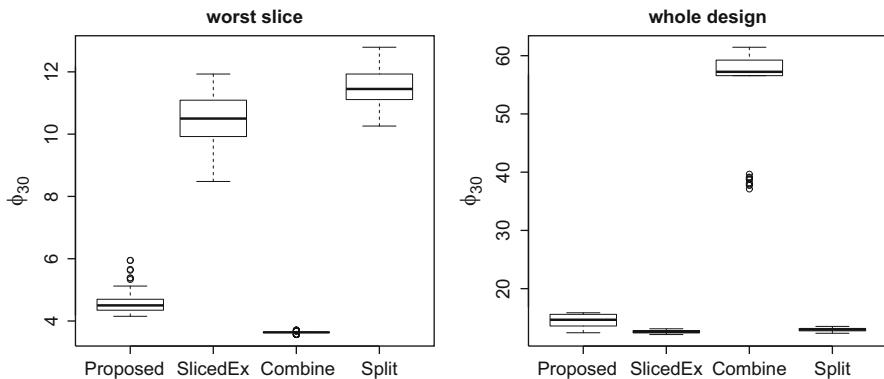


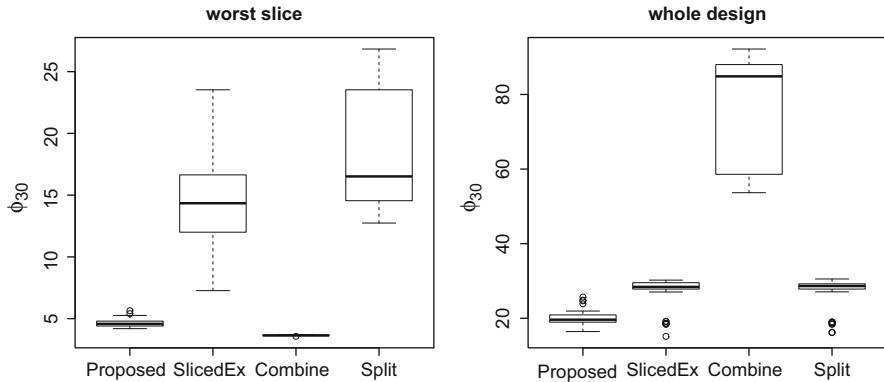
Fig. 9.3 Boxplot of $\phi_p(\mathbf{D}_1)$ and $\phi_p(\mathbf{D})$ for maximin SLHDs with 80 runs, 2 factors, and 8 slices

Example 4. Consider the construction of a maximin SLHD with 50 runs, 10 factors, and 5 slices. The results are shown in Table 9.4 and Fig. 9.5.

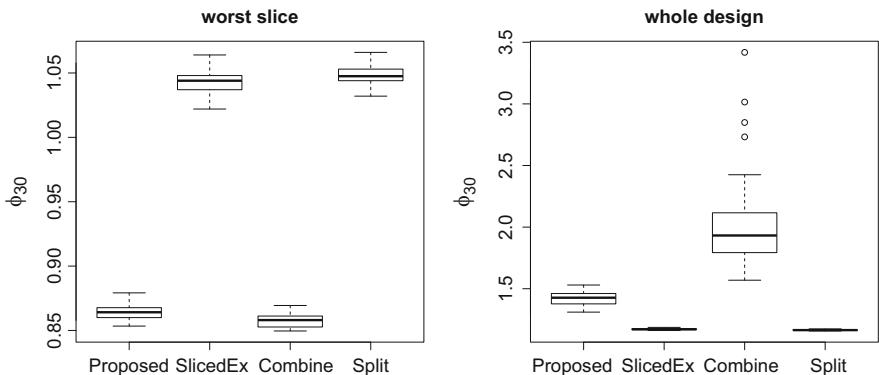
First, the comparison among methods “Proposed,” “SlicedEx,” “Combine,” and “Split” is summarized as follows. (i) Considering the worst slice in the design, the proposed method is significantly better than “SlicedEx” but slightly worse than

Table 9.3 Construction of maximin SLHDs with 120 runs, 2 factors, and 12 slices

Method	Proposed	SlicedEx	Combine	Split	BA-SA
$\phi_p(\mathbf{D})$	15.96	17.13	53.04	16.97	14.50
$\phi_p(\mathbf{D}_1)$	4.658	11.50	3.644	13.99	5.711
$\phi_p^s(\mathbf{D})$	11.99	17.79	41.07	19.23	9.565
CPU Time(s)	26.03	25.00	22.57	22.89	25.73

**Fig. 9.4** Boxplot of $\phi_p(\mathbf{D}_1)$ and $\phi_p(\mathbf{D})$ for maximin SLHDs with 120 runs, 2 factors, and 12 slices**Table 9.4** Construction of maximin SLHDs with 50 runs, 10 factors, and 5 slices

Method	Proposed	SlicedEx	Combine	Split	BA-SA
$\phi_p(\mathbf{D})$	1.419	1.171	2.031	1.165	1.171
$\phi_p(\mathbf{D}_1)$	0.8648	1.042	0.8578	1.048	0.8895
$\phi_p^s(\mathbf{D})$	1.136	1.098	1.441	1.098	1.024
CPU Time(s)	38.70	38.94	38.13	37.45	41.40

**Fig. 9.5** Boxplot of $\phi_p(\mathbf{D}_1)$ and $\phi_p(\mathbf{D})$ for maximin SLHDs with 50 runs, 10 factors, and 5 slices

“Combine.” (ii) The proposed method is the best under criterion ϕ_p^s in (9.9) except for Example 4, in which the design has ten points in ten dimensions for each of the five slices. In that example, the advantage of “Proposed” over “SlicedEx” in terms of between-slice distance is not that obvious, since the pairwise distances between ten points in ten-dimensional space would not be very small. (iii) Based on the results in Examples 1, 2, and 3, “SlicedEx” is slightly better than the two-step method in terms of the whole design when there are only five slices, but the difference tends to be smaller as more slices are added; eventually the proposed method outperforms “SlicedEx” when the number of slices is 12. This makes sense intuitively, since compared with working on the whole design, it is more efficient to optimize and evaluate different slices separately, especially when the number of slices is not small. Then, compared with the proposed method, the algorithm in [1] is worse in terms of $\phi_p(\mathbf{D}_1)$ but is better in terms of $\phi_p^s(\mathbf{D})$ and $\phi_p(\mathbf{D})$. This could be a result of a more efficient updating scheme of ϕ_p used in [1] or some subtle difference between the SA algorithm in [1] and the ESE algorithm, such as the parameter setting and the acceptance criterion.

6 Application to the Estimation of Prediction Error

6.1 Evaluation of Multiple Computer Models

Consider the collective evaluation of t similar computer models, f_1, f_2, \dots, f_t , where each f_i has inputs drawn from the uniform distribution on $(0, 1]^q$. Define f to be a linear combination of f_i 's, i.e., $\sum_{i=1}^t \lambda_i f_i$. To approximate f_i and f , the following ordinary Kriging model in [15] is used:

$$y = \beta + Z(\mathbf{x}), \quad (9.12)$$

where the Gaussian process $Z(\mathbf{x})$ is assumed to have mean 0 and covariance function:

$$R(\mathbf{x}, \mathbf{w}) = \sigma^2 \prod_{j=1}^q \exp\{-\theta_j(x_j - w_j)^2\}. \quad (9.13)$$

Given the set of responses $\{f_i(\mathbf{x}) : \mathbf{x} \in \mathbf{D}_i\}$, the computation of \hat{f}_i as the predictor is obtained by maximizing the likelihood of (9.12) or, equivalently, the BLUP estimator. Similarly, let \hat{f} denote the predictor of f using the response set from the whole design $\mathbf{D} = (\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_t)$.

In the simulation, the following five designs are used for $\mathbf{D} = (\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_t)$:

- LHD: \mathbf{D} is an LHD, and \mathbf{D}_i 's are obtained by randomly splitting \mathbf{D} into t slices.
- MLHD: \mathbf{D} is a maximin LHD, and \mathbf{D}_i 's are obtained by randomly splitting \mathbf{D} into t slices.
- IMLH: \mathbf{D} is an LHD, and \mathbf{D}_i 's are independent maximin LHDs.
- SLHD: \mathbf{D} is an LHD, and \mathbf{D}_i 's are independent LHDs.
- MSLH: \mathbf{D} is a maximin SLHD, and \mathbf{D}_i 's are its slices.

Table 9.5 Average of the mean squared prediction error using 100 Replications in Example 1

		LHD	MLHD	IMLH	SLHD	MSLH
f_1	$m = 5$	0.1483	0.1567	0.1224	0.1250	0.1214
	$m = 10$	0.0941	0.0894	0.0698	0.0781	0.0707
	$m = 20$	0.0678	0.0596	0.0465	0.0545	0.0464
	$m = 40$	0.0503	0.0438	0.0327	0.0411	0.0334
\hat{f}	$m = 5$	0.0483	0.0424	0.0588	0.0516	0.0567
	$m = 10$	0.0360	0.0302	0.0421	0.0406	0.0389
	$m = 20$	0.0289	0.0230	0.0329	0.0311	0.0288
	$m = 40$	0.0288	0.0212	0.0280	0.0290	0.0252

Example 1. Consider the following example from [13]:

$$\begin{aligned} f_1(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1}{\sqrt{x_2}}\right), \\ f_2(\mathbf{x}) &= \log\left(\frac{0.98}{\sqrt{x_1}} + \frac{0.95}{\sqrt{x_2}}\right), \\ f_3(\mathbf{x}) &= \log\left(\frac{1.02}{\sqrt{x_1}} + \frac{1.02}{\sqrt{x_2}}\right), \\ f_4(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1.03}{\sqrt{x_2}}\right), \end{aligned}$$

and $f(\mathbf{x}) = \frac{1}{4} \sum_{i=1}^4 f_i(\mathbf{x})$. For each choice of the size of D_i (denoted as m), the parameters for the algorithm are adjusted so that the construction time for the three types of optimal designs is close. Table 9.5 provides the comparison between the designs in terms of the mean squared prediction error on a separate 1000 run LHD test design. In this example, “MSLH” and “IMLH” achieve the lowest prediction error for f_1 ; “MLHD” is the best considering the prediction error for f . In this example, lower ϕ_p value of the design matrix is associated with smaller prediction error of the Kriging model.

6.2 Cross Validation of Prediction Error

In this section, the mean squared prediction error (MSPE) for Gaussian process regression is estimated by cross validation [16]. Given the value of an unknown function f on n_L data points, $y_{L,i} = f(x_{L,i})$, for $i = 1, 2, \dots, n_L$, let \hat{f} be the BLUP predictor obtained with MLE parameter estimates. Given a large testing data set $T = (\mathbf{x}_{T,i}, y_{T,i})$ of n_T observations, MSPE can be estimated by:

$$MSPE_{test} = \frac{1}{n_T} \sum_{i=1}^{n_T} (y_{T,i} - \hat{f}(\mathbf{x}_{T,i}))^2.$$

However, a testing data set is usually not available in practice, while K-fold cross validation can be easily implemented to find an estimate based on the learning data L only. The procedure starts by dividing L into t slices, L_1, L_2, \dots, L_t , with m points in each slice. For $k = 1, 2, \dots, K$, build a predictor \hat{f}_{-k} based on data $L \setminus L_k$ by Gaussian process regression, and then calculate $MSPE_{CV,k}$ as the mean squared prediction error of \hat{f}_{-k} on the remaining slice L_k . The cross-validation estimator is the average of the mean squared prediction errors on all t slices:

$$MSPE_{CV} = \frac{1}{t} \sum_{k=1}^t MSPE_{CV,k}.$$

In the simulation, the five designs discussed previously are still used to generate the learning data L for K-fold cross validation. The leave-one-out estimator is a special case of K-fold cross validation, where the number of slices t is equal to the size of the training data n_L . To simplify the computation, [15] proposes a pseudo version of the leave-one-out estimator, in which \hat{f}_{-k} uses the maximum likelihood estimates of the covariance parameters in (9.13) based on the complete data L . For reference, these two leave-one-out estimates of MSPE are also computed based on the optimal design MLHD.

The quantity $MSPE_{test}$ is close to the true squared prediction error given a large testing data set. Hence, $MSPE_{CV} - MSPE_{test}$ is considered as the bias of the cross-validation estimates to evaluate different designs. In addition, the standard deviation of the prediction error on t slices $\{MSPE_{CV,1}, MSPE_{CV,2}, \dots, MSPE_{CV,t}\}$ is also computed, which is denoted as s_{cv} . The computation time to construct optimal designs MLHD, IMLH, and MSLH is set to be close by adjusting the parameters of these algorithms.

Example 2. This example considers the Branin function [6] on the domain $[-5, 10] \times [0, 15]$. The response takes the form:

$$f(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10.$$

The testing set is generated by LHD with 5000 runs. The predictor \hat{f} is obtained by Gaussian process regression on a learning data set of size 36. The K-fold cross-validation estimate with six slices is obtained for each of the five sampling schemes. The leave-one-out estimate and its pseudo version are also calculated using a maximin LHD. The result of 50 replications is shown in Fig. 9.6 and Table 9.6.

Example 3. This example uses a multimodal six-hump camel back function from [14]:

$$f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$$

Fig. 9.6 Boxplot of $MSPE_{CV} - MSPE_{test}$ in Example 2

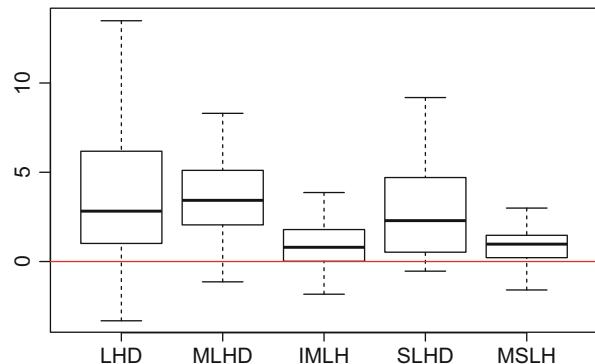
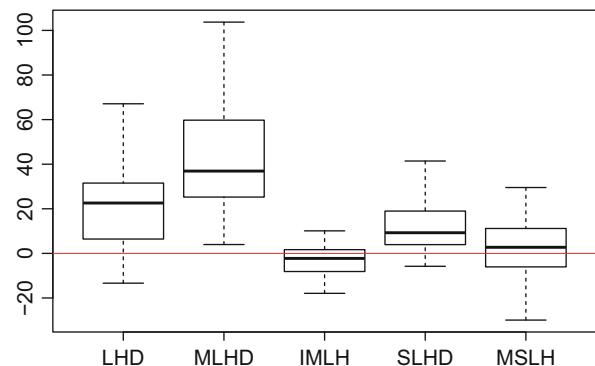


Table 9.6 Sample statistics for the cross-validation estimates using 50 replicates in Example 2

Method	K-Fold					Leave-one-out	
	LHD	MLHD	IMLH	SLHD	MSLH	true	pseudo
mean of $MSPE_{test}$	0.7599	0.31530	0.5364	1.217	0.4866	0.3153	0.3153
sd. of $MSPE_{test}$	1.948	0.8242	1.131	4.302	0.7816	0.8242	0.8242
mean of $MSPE_{CV}$	5.339	6.710	1.760	5.682	2.295	1.072	17.77
sd. of $MSPE_{CV}$	5.199	9.538	1.925	9.856	3.046	1.840	24.43
s_{cv}	9.437	12.00	2.978	11.28	3.655	4.852	75.39
Time		6.554	7.527		5.832		

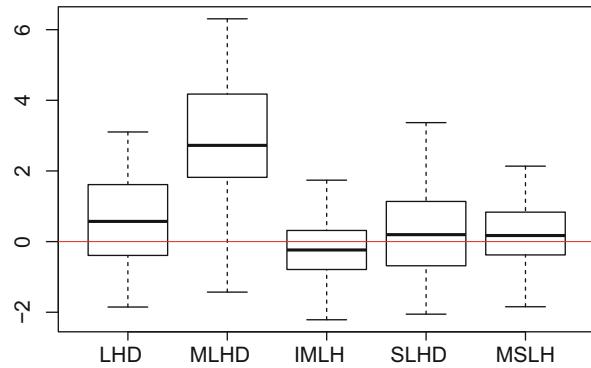
Fig. 9.7 Boxplot of $MSPE_{CV} - MSPE_{test}$ in Example 3



where $x_1 \in [-3, 3]$, $x_2 \in [-2, 2]$. The testing set is generated by an LHD with 5000 runs. The predictor \hat{f} is obtained by Gaussian process regression on a learning data set of size 49. The K-fold cross-validation estimate with seven slices is obtained for each of the five sampling schemes. The leave-one-out estimate and its pseudo version are also calculated using a maximin LHD. The result of 50 replications is shown in Fig. 9.7 and Table 9.7.

Table 9.7 Sample statistics for the cross-validation estimates using 50 replicates in Example 3

Method	K-Fold					Leave-one-out	
	LHD	MLHD	IMLH	SLHD	MSLH	true	pseudo
mean of $MSPE_{test}$	4.617	2.854	9.215	3.629	12.74	2.854	2.854
sd. of $MSPE_{test}$	7.738	3.900	9.155	5.237	13.33	3.900	3.900
mean of $MSPE_{CV}$	30.66	46.40	6.075	17.03	13.60	9.917	40.84
sd. of $MSPE_{CV}$	29.08	25.91	5.959	14.32	9.370	12.08	22.77
s_{cv}	49.473	62.72	9.278	27.97	14.38	47.67	134.7
Time		13.25	14.62		14.60		

Fig. 9.8 Boxplot of $MSPE_{CV} - MSPE_{test}$ in Example 4

Example 4. This example uses the 3 dimensional Ishigami function from [3]. The original function is defined on $[-\pi, \pi]^3$:

$$f(x_1, x_2, x_3) = \sin(x_1) + 7 \sin^2(x_2) + 0.1x_3^4 \sin(x_1).$$

The testing set is generated by an LHD with 10000 runs. The predictor \hat{f} is obtained by Gaussian process regression on a learning data set of size 80. The K-fold cross-validation estimate with eight slices is obtained for each of the five sampling schemes. The leave-one-out estimate and its pseudo version are also calculated using a maximin LHD. The result of 50 replications is shown in Fig. 9.8 and Table 9.8.

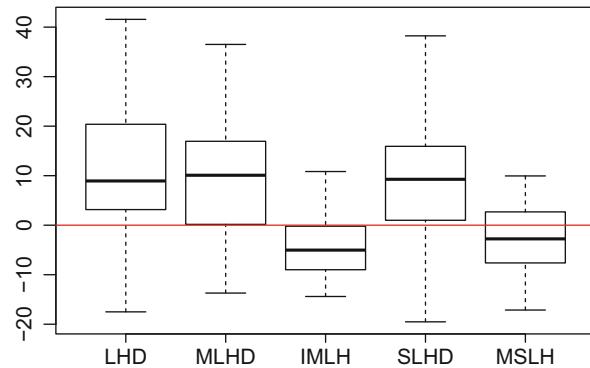
Example 5. This example uses the eight-dimensional borehole function investigated in [9], which models water flow rate through a borehole drilled from the ground surface through two aquifers. The flow rate is given by

$$f(r, r_w, T_u, T_l, H_u, H_l, L, K_w) = \frac{2\pi(H_u - H_l)}{\log(r/r_w) \left(1 + \frac{2LT_u}{\log(r/r_w)r_w^2K_w} + \frac{T_u}{T_l} \right)}.$$

The testing set is generated by an LHD with 10000 runs. The predictor \hat{f} is obtained by Gaussian process regression on a learning data set of size 90. The

Table 9.8 Sample statistics for the cross-validation estimates using 50 replicates in Example 4

Method	K-Fold					Leave-one-out	
	LHD	MLHD	IMLH	SLHD	MSLH	true	pseudo
mean of $MSPE_{test}$	2.668	2.074	2.378	2.662	2.185	2.074	2.074
sd. of $MSPE_{test}$	0.5678	0.4919	0.5508	0.4803	0.4179	0.4919	0.4919
mean of $MSPE_{CV}$	3.386	4.896	2.189	2.957	2.414	3.611	4.230
sd. of $MSPE_{CV}$	0.9334	1.189	0.6558	0.9348	0.7453	1.048	1.112
s_{cv}	2.680	3.081	1.403	2.060	1.590	8.739	9.010
Time		17.76	18.96		20.24		

Fig. 9.9 Boxplot of $MSPE_{CV} - MSPE_{test}$ in Example 5**Table 9.9** Sample statistics for the cross-validation estimates using 50 replicates in Example 5

Method	K-Fold					Leave-one-out	
	LHD	MLHD	IMLH	SLHD	MSLH	true	pseudo
mean of $MSPE_{test}$	18.23	23.67	18.59	17.37	18.79	23.67	23.67
sd. of $MSPE_{test}$	5.697	8.284	3.635	4.634	4.224	8.284	8.284
mean of $MSPE_{CV}$	31.10	34.04	14.02	28.15	16.65	11.48	44.04
sd. of $MSPE_{CV}$	13.73	12.55	4.101	12.91	6.763	5.972	13.36
s_{cv}	25.53	27.09	8.383	21.88	12.02	53.81	127.9
Time		49.84	47.81		51.30		

K-fold cross-validation estimate with five slices is obtained for each of the five sampling schemes. The leave-one-out estimate and its pseudo version are also calculated based on a maximin LHD. The result of 50 replications is shown in Fig. 9.9 and Table 9.9.

Based on the examples in this section, the conclusions are listed as follows.

- The true version of the leave-one-out estimate is the best based on the bias criterion: $MSPE_{CV} - MSPE_{test}$. However, this estimate is known to have large variation because of the similarity between the training data sets. Furthermore,

given a larger learning data set, the leave-one-out estimate is not practical because of the repeated calculation of the maximum likelihood estimate.

- The pseudo version of the leave-one-out estimate can be applied for large data sets, but it is not accurate.
- Randomly splitting an LHD or a maximin LHD to perform K-fold cross validation leads to large variation in the estimate.
- Designs “IMLH” and “MSLH” are the best among all five choices in terms of the bias of the K-fold cross validation. For the standard deviation of the estimate, “MSLH” is slightly better. In Example 5, K-fold cross validation using “IMLH” underestimates the prediction error, which could be a result of the small between-slice distances.

7 Conclusion

One important application of SLHD (and therefore maximin SLHD) is for fitting multiple computer models with similar response, which, as discussed in [13], has the advantage of reducing predictive variance. Other potential applications include computer models with qualitative and quantitative factors [2, 11, 12], cross validation, and stochastic optimization. It has been shown in [5] that the maximin design is asymptotically D-optimal under the Gaussian process regression model with correlation function $\text{corr}(\mathbf{x}_i, \mathbf{x}_j) = [\rho(\|\mathbf{x}_i - \mathbf{x}_j\|)]^k$, as $k \rightarrow \infty$, where $\rho(\cdot)$ is a decreasing function. With this asymptotic D-optimality, the maximin SLHD is expected to improve the performance of SLHD for these various applications.

References

1. Ba, S., Brenneman, W.A., Myers, W.R.: Optimal sliced Latin hypercube designs. *Technometrics* **57**, 479–487 (2015)
2. Han, G., Santner, T.J., Notz, W.I., Bartel, D.L.: Prediction for computer experiments having quantitative and qualitative input variables. *Technometrics* **51**, 278–288 (2009)
3. Ishigami, T., Homma, T.: An importance quantification technique in uncertainty analysis for computer models, In: Proceedings of the First International Symposium on Uncertainty Modeling and Analysis (ISUMA'90), University of Maryland, pp. 398–403 (1990)
4. Jin, R., Chen, W., Sudjianto, A.: An efficient algorithm for constructing optimal design of computer experiments. *J. Stat. Plann. Inference* **134**, 268–287 (2005)
5. Johnson, M.E., Moore, L.M., Ylvisaker, D.: Minimax and maximin distance designs. *J. Stat. Plan. Inference* **26**, 131–148 (1990)
6. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.* **79**, 157–181 (1993)
7. Lundy, M., Mees, A.: Convergence of an annealing algorithm. *Math. Prog.* **34**, 111–124 (1986)
8. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245 (1979)
9. Morris, M.D., Mitchell, T.J., Ylvisaker, D.: Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics* **35**, 243–255 (1993)
10. Morris M.D., Mitchell, T.J.: Exploratory designs for computational experiments. *J. Stat. Plan. Inference* **43**, 381–402 (1995)

11. Qian, P.Z.G., Wu, C.F.J.: Sliced space-filling designs. *Biometrika* **96**, 945–956 (2006)
12. Qian, P.Z.G., Wu, H., Wu, C.F.J.: Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**, 383–396 (2008)
13. Qian, P.Z.G.: Sliced Latin hypercube designs. *J. Am. Stat. Assoc.* **107**, 393–399 (2012)
14. Szegö, G. P.: *Towards Global Optimization II*. North-Holland, New York (1978)
15. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D.: Screening, predicting, and computer experiments. *Technometrics* **34**, 15–25 (1992)
16. Zhang, Q., Qian, P.Z.G.: Designs for crossvalidating approximation models. *Biometrika* **100**, 997–1004 (2013)

The Bayesian Approach to Inverse Problems 10

Masoumeh Dashti and Andrew M. Stuart

Abstract

These lecture notes highlight the mathematical and computational structure relating to the formulation of, and development of algorithms for, the Bayesian approach to inverse problems in differential equations. This approach is fundamental in the quantification of uncertainty within applications involving the blending of mathematical models with data. The finite-dimensional situation is described first, along with some motivational examples. Then the development of probability measures on separable Banach space is undertaken, using a random series over an infinite set of functions to construct draws; these probability measures are used as priors in the Bayesian approach to inverse problems. Regularity of draws from the priors is studied in the natural Sobolev or Besov spaces implied by the choice of functions in the random series construction, and the Kolmogorov continuity theorem is used to extend regularity considerations to the space of Hölder continuous functions. Bayes' theorem is derived in this prior setting, and here interpreted as finding conditions under which the posterior is absolutely continuous with respect to the prior, and determining a formula for the Radon-Nikodym derivative in terms of the likelihood of the data. Having established the form of the posterior, we then describe various properties common to it in the infinite-dimensional setting. These properties include well-posedness, approximation theory, and the existence of maximum a posteriori estimators. We then describe measure-preserving dynamics, again on the infinite-dimensional space, including Markov chain Monte Carlo and sequential Monte Carlo methods, and measure-preserving reversible stochastic

M. Dashti (✉)

Department of Mathematics, University of Sussex, Brighton, UK
e-mail: m.dashti@sussex.ac.uk

A.M. Stuart

Mathematics Institute, University of Warwick, Coventry, UK
e-mail: a.m.stuart@warwick.ac.uk

differential equations. By formulating the theory and algorithms on the underlying infinite-dimensional space, we obtain a framework suitable for rigorous analysis of the accuracy of reconstructions, of computational complexity, as well as naturally constructing algorithms which perform well under mesh refinement, since they are inherently well defined in infinite dimensions.

Keywords

Inverse problems • Bayesian inversion • Tikhonov regularization and MAP estimators • Markov chain Monte Carlo • Sequential Monte Carlo • Langevin stochastic partial differential equations

Contents

1	Introduction	313
1.1	Bayesian Inversion on \mathbb{R}^n	314
1.2	Inverse Heat Equation	316
1.3	Elliptic Inverse Problem	318
1.4	Bibliographic Notes	320
2	Prior Modeling	320
2.1	General Setting	321
2.2	Uniform Priors	322
2.3	Besov Priors	324
2.4	Gaussian Priors	330
2.5	Random Field Perspective	335
2.6	Summary	338
2.7	Bibliographic Notes	339
3	Posterior Distribution	340
3.1	Conditioned Random Variables	340
3.2	Bayes' Theorem for Inverse Problems	342
3.3	Heat Equation	343
3.4	Elliptic Inverse Problem	345
3.5	Bibliographic Notes	349
4	Common Structure	349
4.1	Well Posedness	350
4.2	Approximation	354
4.3	MAP Estimators and Tikhonov Regularization	358
4.4	Bibliographic Notes	364
5	Measure Preserving Dynamics	364
5.1	General Setting	365
5.2	Metropolis-Hastings Methods	367
5.3	Sequential Monte Carlo Methods	371
5.4	Continuous Time Markov Processes	379
5.5	Finite-Dimensional Langevin Equation	380
5.6	Infinite-Dimensional Langevin Equation	384
5.7	Bibliographic Notes	392
6	Conclusions	394
A	Appendix	394
A.1	Function Spaces	394
A.2	Probability and Integration In Infinite Dimensions	402

A.3 Gaussian Measures	414
A.4 Wiener Processes in Infinite-Dimensional Spaces	419
A.5 Bibliographical Notes	422
References	424

1 Introduction

Many uncertainty quantification problems arising in the sciences and engineering require the incorporation of data into a model; indeed doing so can significantly reduce the uncertainty in model predictions and is hence a very important step in many applications. Bayes' formula provides the natural way to do this. The purpose of these lecture notes is to develop the Bayesian approach to inverse problems in order to provide a rigorous framework for the development of uncertainty quantification in the presence of data. Of course it is possible to simply discretize the inverse problem and apply Bayes' formula on a finite-dimensional space. However, we adopt a different approach: we formulate Bayes' formula on a separable Banach space and study its properties in this infinite dimensional setting. This approach, of course, requires considerably more mathematical sophistication and it is important to ask whether this is justified. The answer, of course, is "yes." The formulation of the Bayesian approach on a separable Banach space has numerous benefits: (i) it reveals an attractive well-posedness framework for the inverse problem, allowing for the study of robustness to changes in the observed data, or to numerical approximation of the forward model; (ii) it allows for direct links to be established with the classical theory of regularization, which has been developed in a separable Banach space setting; (iii) and it leads to new algorithmic approaches which build on the full power of analysis and numerical analysis to leverage the structure of the infinite-dimensional inference problem.

The remainder of this section contains a discussion of Bayesian inversion in finite dimensions, for motivational purposes, and two examples of partial differential equation (PDE) inverse problems. In Sect. 2 we describe the construction of priors on separable Banach spaces, using random series and employing the random series to discuss various Sobolev, Besov and Hölder regularity results. Section 3 is concerned with the statement and derivation of Bayes' theorem in this separable Banach space setting. In Sect. 4, we describe various properties common to the posterior, including well posedness in the Hellinger metric, a related approximation theory which leverages well posedness to deliver the required stability estimate, and the existence of maximum a posteriori (MAP) estimators; these address points (i) and (ii) above, respectively. Then, in Sect. 5, we discuss various discrete and continuous time Markov processes which preserve the posterior probability measure, including Markov chain Monte Carlo methods (MCMC), sequential Monte Carlo methods (SMC) and reversible stochastic partial differential equations, addressing point (iii) above. The infinite-dimensional perspective on algorithms is beneficial as it provides a direct way to construct algorithms which behave well under refinement

of finite-dimensional approximations of the underlying separable Banach space. We conclude in Sect. 6 and then an appendix collects together a variety of basic definitions and results from the theory of differential equations and probability. Each section is accompanied by bibliographical notes connecting the developments herein to the wider literature. The notes complement and build on other overviews of Bayesian inversion, and its relations to uncertainty quantification, which may be found in [92, 93]. All results (lemmas, theorems, etc.) which are quoted without proof are given pointers to the literature, where proofs may be found, within the bibliography of the section containing the result.

1.1 Bayesian Inversion on \mathbb{R}^n

Consider the problem of finding $u \in \mathbb{R}^n$ from $y \in \mathbb{R}^J$ where u and y are related by the equation

$$y = G(u).$$

We refer to y as *observed data* and to u as the *unknown*. This problem may be difficult for a number of reasons. We highlight two of these, both particularly relevant to our future developments.

1. The first difficulty, which may be illustrated in the case where $n = J$, concerns the fact that often the equation is perturbed by noise and so we should really consider the equation

$$y = G(u) + \eta, \quad (10.1)$$

where $\eta \in \mathbb{R}^J$ represents the *observational noise* which enters the observed data. Assume further that G maps \mathbb{R}^J into a proper subset of itself, Im_G , and that G has a unique inverse as a map from Im_G into \mathbb{R}^J . It may then be the case that, because of the noise, $y \notin \text{Im}_G$ so that simply inverting G on the data y will not be possible. Furthermore, the specific instance of η which enters the data may not be known to us; typically, at best, only the statistical properties of a typical noise η are known. Thus we cannot subtract η from the observed data y to obtain something in Im_G . Even if $y \in \text{Im}_G$, the uncertainty caused by the presence of noise η causes problems for the inversion.

2. The second difficulty is manifest in the case where $n > J$ so that the system is *underdetermined*: the number of equations is smaller than the number of unknowns. How do we attach a sensible meaning to the concept of solution in this case where, generically, there will be many solutions?

Thinking probabilistically enables us to overcome both of these difficulties. We will treat u , y and η as random variables and determine the joint probability

distribution of (u, y) . We then define the “solution” of the inverse problem to be the probability distribution of u given y , denoted $u|y$. This allows us to model the noise via its statistical properties, even if we do not know the exact instance of the noise entering the given data. And it also allows us to specify a priori the form of solutions that we believe to be more likely, thereby enabling us to attach weights to multiple solutions which explain the data. This is the *Bayesian approach* to inverse problems.

To this end, we define a random variable $(u, y) \in \mathbb{R}^n \times \mathbb{R}^J$ as follows. We let $u \in \mathbb{R}^n$ be a random variable with (Lebesgue) density $\rho_0(u)$. Assume that $y|u$ (y given u) is defined via the formula (10.1) where $G : \mathbb{R}^n \rightarrow \mathbb{R}^J$ is measurable and η is independent of u (we sometimes write this as $\eta \perp u$) and distributed according to measure \mathbb{Q}_0 with Lebesgue density $\rho(\eta)$. Then $y|u$ is simply found by shifting \mathbb{Q}_0 by $G(u)$ to measure \mathbb{Q}_u with Lebesgue density $\rho(y - G(u))$. It follows that $(u, y) \in \mathbb{R}^n \times \mathbb{R}^J$ is a random variable with Lebesgue density $\rho(y - G(u))\rho_0(u)$.

The following theorem allows us to calculate the distribution of the random variable $u|y$:

Theorem 1 (Bayes’ Theorem). *Assume that*

$$Z := \int_{\mathbb{R}^n} \rho(y - G(u))\rho_0(u)du > 0.$$

Then $u|y$ is a random variable with Lebesgue density $\rho^y(u)$ given by

$$\rho^y(u) = \frac{1}{Z} \rho(y - G(u))\rho_0(u).$$

Remarks 1. The following remarks establish the nomenclature of Bayesian statistics and also frame the previous theorem in a manner which generalizes to the infinite-dimensional setting.

- $\rho_0(u)$ is the **prior density**.
- $\rho(y - G(u))$ is the **likelihood**.
- $\rho^y(u)$ is the **posterior density**.
- It will be useful in what follows to define

$$\Phi(u; y) = -\log \rho(y - G(u)).$$

We call Φ the **potential**. This is the **negative log likelihood**.

- Note that Z is the probability of y . Bayes’ formula expresses

$$\mathbb{P}(u|y) = \frac{1}{\mathbb{P}(y)} \mathbb{P}(y|u)\mathbb{P}(u).$$

- Let μ^y be a measure on \mathbb{R}^n with density ρ^y and μ_0 a measure on \mathbb{R}^n with density ρ_0 . Then the conclusion of Theorem 1 may be written as:

$$\begin{aligned}\frac{d\mu^y}{d\mu_0}(u) &= \frac{1}{Z} \exp(-\Phi(u; y)), \\ Z &= \int_{\mathbb{R}^n} \exp(-\Phi(u; y)) \mu_0(du).\end{aligned}\tag{10.2}$$

Thus the posterior is absolutely continuous with respect to the prior, and the Radon-Nikodym derivative is proportional to the likelihood. This is rewriting Bayes' formula in the form

$$\frac{1}{\mathbb{P}(u)} \mathbb{P}(u|y) = \frac{1}{\mathbb{P}(y)} \mathbb{P}(y|u).$$

- The expression for the Radon-Nikodym derivative is to be interpreted as the statement that, for all measurable $f : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$\mathbb{E}^{\mu^y} f(u) = \mathbb{E}^{\mu_0} \left(\frac{d\mu^y}{d\mu_0}(u) f(u) \right).$$

Alternatively we may write this in integral form as

$$\begin{aligned}\int_{\mathbb{R}^n} f(u) \mu^y(du) &= \int_{\mathbb{R}^n} \left(\frac{1}{Z} \exp(-\Phi(u; y)) f(u) \right) \mu_0(du) \\ &= \frac{\int_{\mathbb{R}^n} \exp(-\Phi(u; y)) f(u) \mu_0(du)}{\int_{\mathbb{R}^n} \exp(-\Phi(u; y)) \mu_0(du)}.\end{aligned}$$

□

1.2 Inverse Heat Equation

This inverse problem illustrates the first difficulty, labeled 1. in the previous subsection, which motivates the Bayesian approach to inverse problems. Let $D \subset \mathbb{R}^d$ be a bounded open set, with Lipschitz boundary ∂D . Then define the Hilbert space H and operator A as follows:

$$\begin{aligned}H &= \left(L^2(D), \langle \cdot, \cdot \rangle, \|\cdot\| \right); \\ A &= -\Delta, \quad \mathcal{D}(A) = H^2(D) \cap H_0^1(D).\end{aligned}$$

We make the following assumption about the spectrum of A which is easily verified for simple geometries, but in fact holds quite generally.

Assumption 1. *The eigenvalue problem*

$$A\varphi_j = \alpha_j \varphi_j,$$

has a countably infinite set of solutions, indexed by $j \in \mathbb{Z}^+$. They may be normalized to satisfy the L^2 -orthonormality condition

$$\langle \varphi_j, \varphi_k \rangle = \begin{cases} 1, & j = k \\ 0, & j \neq k, \end{cases}$$

and form a basis for H . Furthermore, the eigenvalues are positive and, if ordered to be increasing, satisfy $\alpha_j \asymp j^{\frac{2}{d}}$. \square

Here and in the remainder of the notes, the notation \asymp denotes the existence of constants $C^\pm > 0$ such that

$$C^- j^{2/d} \leq \alpha_j \leq C^+ j^{2/d} \quad (10.3)$$

for all $j \in \mathbb{N}$.

Any $w \in H$ can be written as

$$w = \sum_{j=1}^{\infty} \langle w, \varphi_j \rangle \varphi_j$$

and we can define the Hilbert scale of spaces $\mathcal{H}^t = \mathcal{D}(A^{t/2})$ as explained in Sect. A.1.3 for any $t > 0$ and with the norm

$$\|w\|_{\mathcal{H}^t}^2 = \sum_{j=1}^{\infty} j^{\frac{2t}{d}} |w_j|^2$$

where $w_j = \langle w, \varphi_j \rangle$.

Consider the heat conduction equation on D , with Dirichlet boundary conditions, writing it as an ordinary differential equation in H :

$$\frac{dv}{dt} + Av = 0, \quad v(0) = u. \quad (10.4)$$

We have the following:

Lemma 1. *Let Assumption 1 hold. Then for every $u \in H$ and every $s > 0$, there is a unique solution v of equation (10.4) in the space $C([0, \infty); H) \cap C((0, \infty); \mathcal{H}^s)$. We write $v(t) = \exp(-At)u$.*

To motivate this statement, and in particular the high degree of regularity seen at each fixed t , we argue as follows. Note that, if the initial condition is expanded in the eigenbasis as

$$u = \sum_{j=1}^{\infty} u_j \varphi_j, \quad u_j = \langle u, \varphi_j \rangle,$$

then the solution of (10.4) has the form

$$v(t) = \sum_{j=1}^{\infty} u_j e^{-\alpha_j t} \varphi_j.$$

Thus

$$\begin{aligned} \|v(t)\|_{\mathcal{H}^s}^2 &= \sum_{j=1}^{\infty} j^{2s/d} e^{-2\alpha_j t} |u_j|^2 \asymp \sum_{j=1}^{\infty} \alpha_j^s e^{-2\alpha_j t} |u_j|^2 \\ &= t^{-s} \sum_{j=1}^{\infty} (\alpha_j t)^s e^{-2\alpha_j t} |u_j|^2 \leq C t^{-s} \sum_{j=1}^{\infty} |u_j|^2 \\ &= C t^{-s} \|u\|_H^2. \end{aligned}$$

It follows that $v(t) \in \mathcal{H}^s$ for any $s > 0$, provided $u \in H$.

We are interested in the inverse problem of finding u from y where

$$y = v(1) + \eta = G(u) + \eta = e^{-A}u + \eta.$$

Here $\eta \in H$ is noise and $G(u) := v(1) = e^{-A}u$. Formally this looks like an infinite-dimensional linear version of the inverse problem (10.1), extended from finite dimensions to a Hilbert space setting. However, the infinite-dimensional setting throws up significant new issues. To see this, assume that there is $\beta_c > 0$ such that η has regularity \mathcal{H}^β if and only if $\beta < \beta_c$. Then y is not in the image space of G which is, of course, contained in $\cap_{s>0} \mathcal{H}^s$. Applying the formal inverse of G to y results in an object which is not in H .

To overcome this problem, we will apply a Bayesian approach and hence will need to put probability measures on the Hilbert space H ; in particular we will want to study $\mathbb{P}(u)$, $\mathbb{P}(y|u)$ and $\mathbb{P}(u|y)$, all probability measures on H .

1.3 Elliptic Inverse Problem

One motivation for adopting the Bayesian approach to inverse problems is that prior modeling is a transparent approach to dealing with under-determined inverse

problems; it forms a rational approach to dealing with the second difficulty, labeled 2 in Sect. 1.1. The elliptic inverse problem we now describe is a concrete example of an under-determined inverse problem.

As in Sect. 1.2, $D \subset \mathbb{R}^d$ denotes a bounded open set, with Lipschitz boundary ∂D . We define the Gelfand triple of Hilbert spaces $V \subset H \subset V^*$ by

$$H = (L^2(D), \langle \cdot, \cdot \rangle, \|\cdot\|), \quad V = (H_0^1(D), \langle \nabla \cdot, \nabla \cdot \rangle, \|\cdot\|_V = \|\nabla \cdot\|). \quad (10.5)$$

and V^* the dual of V with respect to the pairing induced by H . Note that $\|\cdot\| \leq C_p \|\cdot\|_V$ for some constant C_p : the Poincaré inequality.

Let $\kappa \in X := L^\infty(D)$ satisfy

$$\operatorname{ess\ inf}_{x \in D} \kappa(x) = \kappa_{\min} > 0. \quad (10.6)$$

Now consider the equation

$$-\nabla \cdot (\kappa \nabla p) = f, \quad x \in D, \quad (10.7a)$$

$$p = 0, \quad x \in \partial D. \quad (10.7b)$$

Lax-Milgram theory yields the following:

Lemma 2. *Assume that $f \in V^*$ and that κ satisfies (10.6). Then (10.7) has a unique weak solution $p \in V$. This solution satisfies*

$$\|p\|_V \leq \|f\|_{V^*}/\kappa_{\min}$$

and, if $f \in H$,

$$\|p\|_V \leq C_p \|f\|/\kappa_{\min}.$$

We will be interested in the inverse problem of finding κ from y where

$$y_j = l_j(p) + \eta_j, \quad j = 1, \dots, J. \quad (10.8)$$

Here $l_j \in V^*$ is a continuous linear functional on V and η_j is a noise.

Notice that the unknown, $\kappa \in X$, is a function (infinite dimensional), whereas the data from which we wish to determine κ is finite dimensional: $y \in \mathbb{R}^J$. The problem is severely under-determined, illustrating point 2 from Sect. 1.1. One way to treat such problems is by adopting the Bayesian framework, using prior modeling to fill in missing information. We will take the unknown function to be u where either $u = \kappa$ or $u = \log \kappa$. In either case, we will define $G_j(u) = l_j(p)$ and, noting that p is then a nonlinear function of u , (10.8) may be written as

$$y = G(u) + \eta \quad (10.9)$$

where $y, \eta \in \mathbb{R}^J$ and $G : X^+ \subseteq X \rightarrow \mathbb{R}^J$. The set X^+ is introduced because G may not be defined on the whole of X . In particular, the positivity constraint (10.6) is only satisfied on

$$X^+ := \left\{ u \in X : \text{ess inf}_{x \in D} u(x) > 0 \right\} \subset X \quad (10.10)$$

in the case where $\kappa = u$. On the other hand, if $\kappa = \exp(u)$, then the positivity constraint (10.6) is satisfied for any $u \in X$ and we may take $X^+ = X$.

Notice that we again need probability measures on function space, here the Banach space $X = L^\infty(D)$. Furthermore, in the case where $u = \kappa$, these probability measures should charge only positive functions, in view of the desired inequality (10.6). Probability on Banach spaces of functions is most naturally developed in the setting of separable spaces, which $L^\infty(D)$ is not. This difficulty can be circumvented in various different ways as we describe in what follows.

1.4 Bibliographic Notes

- Section 1.1. See [11] for a general overview of the Bayesian approach to statistics in the finite-dimensional setting. The Bayesian approach to linear inverse problems with Gaussian noise and prior in finite dimensions is discussed in [92, Chapters 2 and 6] and, with a more algorithmic flavor, in the book [53].
- Section 1.2. For details on the heat equation as an ODE in Hilbert space, and the regularity estimates of Lemma 1, see [70, 80]. The classical approach to linear inverse problems is described in numerous books; see, for example, [32, 51]. The case where the spectrum of the forward map G decays exponentially, as arises for the heat equation, is sometimes termed *severely ill posed*. The Bayesian approach to linear inverse problems was developed systematically in [68, 71], following from the seminal paper [36] in which the approach was first described; for further reading on ill-posed linear problems, see [92, Chapters 3 and 6]. Recovering the truth underlying the data from the Bayesian approach, known as *Bayesian posterior consistency*, is the topic of [3, 55]; generalizations to severely ill-posed problems, such as the heat equation, may be found in [4, 56].
- Section 1.3. See [33] for the Lax-Milgram theory which gives rise to Lemma 2. For classical inversion theory for the elliptic inverse problem – determining the permeability from the pressure in a Darcy model of flow in a porous medium – see [8, 86]; for Bayesian formulations see [24, 25]. For posterior consistency results see [99].

2 Prior Modeling

In this section we show how to construct probability measures on a function space, adopting a constructive approach based on random series. As explained in Sect. A.2.2, the natural setting for probability in a function space is that of a separable Banach space. A countable infinite sequence in the Banach space X will

be used for our random series; in the case where X is not separable, the resulting probability measure will be constructed on a separable subspace X' of X (see the discussion in Sect. 2.1).

Section 2.1 describes this general setting, and Sects. 2.2, 2.3 and 2.4 consider, in turn, three classes of priors termed uniform, Besov and Gaussian. In Sect. 2.5 we link the random series construction to the widely used random field perspective on spatial stochastic processes and we summarize in Sect. 2.6. We denote the prior measures constructed in this section by μ_0 .

2.1 General Setting

We let $\{\phi_j\}_{j=1}^\infty$ denote an infinite sequence in the Banach space X , with norm $\|\cdot\|$, of \mathbb{R} -valued functions defined on a domain D . We will either take $D \subset \mathbb{R}^d$, a bounded, open set with Lipschitz boundary or $D = \mathbb{T}^d$ the d -dimensional torus. We normalize these functions so that $\|\phi_j\| = 1$ for $j = 1, \dots, \infty$. We also introduce another element $m_0 \in X$, not necessarily normalized to 1. Define the function u by

$$u = m_0 + \sum_{j=1}^{\infty} u_j \phi_j. \quad (10.11)$$

By randomizing $\mathbf{u} := \{u_j\}_{j=1}^\infty$, we create real-valued random functions on D . (The extension to \mathbb{R}^n -valued random functions is straightforward, but omitted for brevity.)

We now define the deterministic sequence $\gamma = \{\gamma_j\}_{j=1}^\infty$ and the i.i.d. random sequence $\xi = \{\xi_j\}_{j=1}^\infty$, and set $u_j = \gamma_j \xi_j$. We assume that ξ is centred, i.e., that it has mean zero. Formally we see that the average value of u is then m_0 so that this element of X should be thought of as the *mean function*. We assume that $\gamma \in \ell_w^p$ for some $p \in [1, \infty)$ and some positive weight sequence $\{w_j\}$ (see Sect. A.1.1). We define $\Omega = \mathbb{R}^\infty$ and view ξ as a random element in the probability space $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$ of i.i.d. sequences equipped with the product σ -algebra; we let \mathbb{E} denote expectation. This sigma algebra can be generated by cylinder sets if an appropriate distance d is defined on sequences. However, the distance d captures nothing of the properties of the random function u itself. For this reason we will be interested in the pushforward of the measure \mathbb{P} on the measure space $(\Omega, \mathcal{B}(\Omega))$ into a measure μ on $(X', \mathcal{B}(X'))$, where X' is a separable Banach space and $\mathcal{B}(X')$ denotes its Borel σ -algebra. Sometimes X' will be the same as X but not always: the space X may not be separable; and, although we have stated the normalization of the ϕ_j in X , they may of course live in smaller spaces X' , and u may do so too. For either of these reasons, X' may be a proper subspace of X .

In the next three subsections, we demonstrate how this general setting may be adapted to create a variety of useful prior measures on function space; the fourth subsection, which follows these three, relates the random series construction, in the Gaussian case, to the standard construction of Gaussian random fields. We will express many of our results in terms of the probability measure \mathbb{P} on i.i.d sequences, but all such results will, of course, have direct implications for the induced pushforward measures on the function spaces where the random functions

u live. We discuss this perspective in the summary Sect. 2.6. In dealing with the random series construction, we will also find it useful to consider the truncated random functions

$$u^N = m_0 + \sum_{j=1}^N u_j \phi_j, \quad u_j = \gamma_j \xi_j. \quad (10.12)$$

2.2 Uniform Priors

To construct the random functions (10.11), we take $X = L^\infty(D)$, choose the deterministic sequence $\gamma = \{\gamma_j\}_{j=1}^\infty \in \ell^1$ and specify the i.i.d. sequence $\xi = \{\xi_j\}_{j=1}^\infty$ by $\xi_1 \sim U[-1, 1]$, uniform random variables on $[-1, 1]$. Assume further that there are finite, strictly positive constants m_{\min}, m_{\max} , and δ such that

$$\begin{aligned} \text{ess inf}_{x \in D} m_0(x) &\geq m_{\min}; \\ \text{ess sup}_{x \in D} m_0(x) &\leq m_{\max}; \\ \|\gamma\|_{\ell^1} &= \frac{\delta}{1 + \delta} m_{\min}. \end{aligned}$$

The space X is not separable and so, instead, we work with the space X' found as the closure of the linear span of the functions $(m_0, \{\phi_j\}_{j=1}^\infty)$ with respect to the norm $\|\cdot\|_\infty$ on X . The Banach space $(X', \|\cdot\|_\infty)$ is separable.

Theorem 2. *The following holds \mathbb{P} -almost surely: the sequence of functions $\{u^N\}_{N=1}^\infty$ given by (10.12) is Cauchy in X' , and the limiting function u given by (10.11) satisfies*

$$\frac{1}{1 + \delta} m_{\min} \leq u(x) \leq m_{\max} + \frac{\delta}{1 + \delta} m_{\min} \quad a.e. \quad x \in D.$$

Proof. Let $N > M$. Then, \mathbb{P} -a.s.,

$$\begin{aligned} \|u^N - u^M\|_\infty &= \left\| \sum_{j=M+1}^N u_j \phi_j \right\|_\infty \\ &\leq \left\| \sum_{j=M+1}^N \gamma_j \xi_j \phi_j \right\|_\infty \\ &\leq \sum_{j=M+1}^\infty |\gamma_j| |\xi_j| \|\phi_j\|_\infty \\ &\leq \sum_{j=M+1}^\infty |\gamma_j|. \end{aligned}$$

The right-hand side tends to zero as $M \rightarrow \infty$ by the dominated convergence theorem and hence the sequence is Cauchy in X' .

We have \mathbb{P} -a.s. and for a.e. $x \in D$,

$$\begin{aligned} u(x) &\geq m_0(x) - \sum_{j=1}^{\infty} |u_j| \|\phi_j\|_{\infty} \\ &\geq \text{ess inf}_{x \in D} m_0(x) - \sum_{j=1}^{\infty} |\gamma_j| \\ &\geq m_{\min} - \|\gamma\|_{\ell^1} \\ &= \frac{1}{1+\delta} m_{\min}. \end{aligned}$$

Proof of the upper bound is similar. \square

Example 1. Consider the random function (10.11) as specified in this section. By Theorem 2 we have that, \mathbb{P} -a.s.,

$$u(x) \geq \frac{1}{1+\delta} m_{\min} > 0, \quad \text{a.e. } x \in D. \quad (10.13)$$

Set $\kappa = u$ in the elliptic equation (10.6), so that the coefficient κ in the equation and the solution p are random variables on $(\mathbb{R}^{\infty}, \mathcal{B}(\mathbb{R}^{\infty}), \mathbb{P})$. Since (10.13) holds \mathbb{P} -a.s., Lemma 2 shows that, again \mathbb{P} -a.s.,

$$\|p\|_V \leq (1+\delta) \|f\|_{V^*} / m_{\min}.$$

Since the r.h.s. is nonrandom, we have that for all $r \in \mathbb{Z}^+$ the random variable $p \in L_{\mathbb{P}}^r(\Omega; V)$:

$$\mathbb{E}\|p\|_V^r < \infty.$$

In fact $\mathbb{E} \exp(\alpha \|p\|_V^r) < \infty$ for all $r \in \mathbb{Z}^+$ and $\alpha \in (0, \infty)$. \square

We now consider the situation where the family $\{\phi_j\}_{j=1}^{\infty}$ has a uniform Hölder exponent α and study the implications for Hölder continuity of the random function u . Specifically we assume that there are $C, a > 0$ and $\alpha \in (0, 1]$ such that, for all $j \geq 1$,

$$|\phi_j(x) - \phi_j(y)| \leq C j^a |x - y|^{\alpha}, \quad x, y \in D. \quad (10.14)$$

and

$$|m_0(x) - m_0(y)| \leq C |x - y|^{\alpha}, \quad x, y \in D. \quad (10.15)$$

Theorem 3. Assume that u is given by (10.11) where the collection of functions $(m_0, \{\phi_j\}_{j=1}^\infty)$ satisfy (10.14) and (10.15). Assume further that $\sum_{j=1}^\infty |\gamma_j|^2 j^{a\theta} < \infty$ for some $\theta \in (0, 2)$. Then \mathbb{P} -a.s. we have $u \in C^{0,\beta}(D)$ for all $\beta < \frac{\alpha\theta}{2}$.

Proof. This is an application of Corollary 5 of the Kolmogorov continuity theorem and S_1 and S_2 are as defined there. We use θ in place of the parameter δ appearing in Corollary 5 in order to avoid confusion with δ appearing in Theorem 2 above and in (10.17) below. Note that, since m_0 has assumed Hölder regularity α , which exceeds $\frac{\alpha\theta}{2}$ since $\theta \in (0, 2)$, it suffices to consider the centred case where $m_0 \equiv 0$. We let $f_j = \gamma_j \phi_j$ and complete the proof by noting that

$$S_1 = \sum_{j=1}^\infty |\gamma_j|^2 \leq S_2 \leq \sum_{j=1}^\infty |\gamma_j|^2 j^{a\theta} < \infty.$$

Example 2. Let $\{\phi_j\}$ denote the Fourier basis for $L^2(D)$ with $D = [0, 1]^d$. Then we may take $a = \alpha = 1$. If $\gamma_j = j^{-s}$, then $s > 1$ ensures $\gamma \in \ell^1$. Furthermore

$$\sum_{j=1}^\infty |\gamma_j|^2 j^{a\theta} = \sum_{j=1}^\infty j^{\theta-2s} < \infty$$

for $\theta < 2s - 1$. We thus deduce that $u \in C^{0,\beta}([0, 1]^d)$ for all $\beta < \min\{s - \frac{1}{2}, 1\}$.

2.3 Besov Priors

For this construction of random functions, we take X to be the Hilbert space

$$X := \dot{L}^2(\mathbb{T}^d) = \left\{ u : \mathbb{T}^d \rightarrow \mathbb{R} \mid \int_{\mathbb{T}^d} |u(x)|^2 dx < \infty, \int_{\mathbb{T}^d} u(x) dx = 0 \right\}$$

of real valued periodic functions in dimension $d \leq 3$ with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively. We then set $m_0 = 0$ and let $\{\phi_j\}_{j=1}^\infty$ be an orthonormal basis for X . Consequently, for any $u \in X$, we have for a.e. $x \in \mathbb{T}^d$,

$$u(x) = \sum_{j=1}^\infty u_j \phi_j(x), \quad u_j = \langle u, \phi_j \rangle. \quad (10.16)$$

Given a function $u : \mathbb{T}^d \rightarrow \mathbb{R}$ and the $\{u_j\}$ as defined in (10.16), we define the Banach space $X^{t,q}$ by

$$X^{t,q} = \left\{ u : \mathbb{T}^d \rightarrow \mathbb{R} \mid \|u\|_{X^{t,q}} < \infty, \int_{\mathbb{T}^d} u(x) dx = 0 \right\}$$

where

$$\|u\|_{X^{t,q}} = \left(\sum_{j=1}^{\infty} j^{(\frac{tq}{d} + \frac{q}{2} - 1)} |u_j|^q \right)^{\frac{1}{q}}$$

with $q \in [1, \infty)$ and $t > 0$. If $\{\phi_j\}$ form the Fourier basis and $q = 2$, then $X^{t,2}$ is the Sobolev space $\dot{H}^t(\mathbb{T}^d)$ of mean-zero periodic functions with t (possibly non-integer) square-integrable derivatives; in particular $X^{0,2} = \dot{L}^2(\mathbb{T}^d)$. On the other hand, if the $\{\phi_j\}$ form certain wavelet bases, then $X^{t,q}$ is the Besov space B_{qq}^t .

As described above, we assume that $u_j = \gamma_j \xi_j$ where $\xi = \{\xi_j\}_{j=1}^{\infty}$ is an i.i.d. sequence and $\gamma = \{\gamma_j\}_{j=1}^{\infty}$ is deterministic. Here we assume that ξ_1 is drawn from the centred measure on \mathbb{R} with density proportional to $\exp(-\frac{1}{2}|x|^q)$ for some $1 \leq q < \infty$ – we refer to this as a *q -exponential distribution*, noting that $q = 2$ gives a Gaussian and $q = 1$ a Laplace-distributed random variable. Then for $s > 0$ and $\delta > 0$, we define

$$\gamma_j = j^{-(\frac{s}{d} + \frac{1}{2} - \frac{1}{q})} \left(\frac{1}{\delta} \right)^{\frac{1}{q}}. \quad (10.17)$$

The parameter δ is a key scaling parameter which will appear in the statement of exponential moment bounds below.

We now prove convergence of the series (found from (10.12) with $m_0 = 0$)

$$u^N = \sum_{j=1}^N u_j \phi_j, \quad u_j = \gamma_j \xi_j \quad (10.18)$$

to the limit function

$$u(x) = \sum_{j=1}^{\infty} u_j \phi_j(x), \quad u_j = \gamma_j \xi_j, \quad (10.19)$$

in an appropriate space. To understand the sequence of functions $\{u^N\}$, it is useful to introduce the following function space:

$$L_{\mathbb{P}}^q(\Omega; X^{t,q}) := \left\{ v : D \times \Omega \rightarrow \mathbb{R} \mid \mathbb{E}(\|v\|_{X^{t,q}}^q) < \infty \right\}.$$

This is a Banach space, when equipped with the norm $\left(\mathbb{E}(\|v\|_{X^{t,q}}^q) \right)^{\frac{1}{q}}$. Thus every Cauchy sequence is convergent in this space.

Theorem 4. *For $t < s - \frac{d}{q}$, the sequence of functions $\{u^N\}_{N=1}^{\infty}$, given by (10.18) and (10.17) with ξ_1 drawn from a centred q -exponential distribution, is Cauchy in*

the Banach space $L_{\mathbb{P}}^q(\Omega; X^{t,q})$. Thus the infinite series (10.19) exists as an $L_{\mathbb{P}}^q$ -limit and takes values in $X^{t,q}$ almost surely, for all $t < s - \frac{d}{q}$.

Proof. For $N > M$,

$$\begin{aligned}\mathbb{E}\|u^N - u^M\|_{X^{t,q}}^q &= \delta^{-1} \mathbb{E} \sum_{j=M+1}^N j^{\frac{(t-s)q}{d}} |\xi_j|^q \\ &\asymp \sum_{j=M+1}^N j^{\frac{(t-s)q}{d}} \leq \sum_{j=M+1}^{\infty} j^{\frac{(t-s)q}{d}}.\end{aligned}$$

The sum on the right-hand side tends to 0 as $M \rightarrow \infty$, provided $\frac{(t-s)q}{d} < -1$, by the dominated convergence theorem. This completes the proof. \square

The previous theorem gives a sufficient condition, on t , for existence of the limiting random function. The following theorem refines this to an if and only if statement, in the context of almost sure convergence.

Theorem 5. Assume that u is given by (10.19) and (10.17) with ξ_1 drawn from a centred q -exponential distribution. Then the following are equivalent:

- (i) $\|u\|_{X^{t,q}} < \infty$ \mathbb{P} -a.s.;
- (ii) $\mathbb{E}(\exp(\alpha \|u\|_{X^{t,q}}^q)) < \infty$ for any $\alpha \in [0, \frac{\delta}{2}]$;
- (iii) $t < s - \frac{d}{q}$.

Proof. We first note that, for the random function in question,

$$\|u\|_{X^{t,q}}^q = \sum_{j=1}^{\infty} j^{(\frac{tq}{d} + \frac{q}{2} - 1)} |u_j|^q = \sum_{j=1}^{\infty} \delta^{-1} j^{-\frac{(s-t)q}{d}} |\xi_j|^q.$$

Now, for $\alpha < \frac{1}{2}$,

$$\begin{aligned}\mathbb{E} \exp(\alpha |\xi_1|^q) &= \int_{\mathbb{R}} \exp\left(-\left(\frac{1}{2} - \alpha\right)|x|^q\right) dx / \int_{\mathbb{R}} \exp\left(-\frac{1}{2}|x|^q\right) dx \\ &= (1 - 2\alpha)^{-\frac{1}{q}}.\end{aligned}$$

(iii) \Rightarrow (ii).

$$\begin{aligned}\mathbb{E}(\exp(\alpha \|u\|_{X^{t,q}}^q)) &= \mathbb{E}\left(\exp\left(\alpha \sum_{j=1}^{\infty} \delta^{-1} j^{-\frac{(s-t)q}{d}} |\xi_j|^q\right)\right) \\ &= \prod_{j=1}^{\infty} \left(1 - \frac{2\alpha}{\delta} j^{-\frac{(s-t)q}{d}}\right)^{-\frac{1}{q}}.\end{aligned}$$

For $\alpha < \frac{\delta}{2}$ the product converges if $\frac{(s-t)q}{d} > 1$, i.e., $t < s - \frac{d}{q}$ as required.

(ii) \Rightarrow (i).

If (i) does not hold, $Z := \|u\|_{X^{t,q}}^q$ is positive infinite on a set of positive measure S . Then, since for $\alpha > 0$, $\exp(\alpha Z) = +\infty$ if $Z = +\infty$, and $\mathbb{E} \exp(\alpha Z) \geq \mathbb{E}(\mathbb{1}_S \exp(\alpha Z))$, we get a contradiction.

(i) \Rightarrow (iii).

To show that (i) implies (iii), note that (i) implies that, almost surely,

$$\sum_{j=1}^{\infty} j^{(t-s)q/d} |\xi_j|^q < \infty.$$

This implies that $t < s$. To see this assume for contradiction that $t \geq s$. Then, almost surely,

$$\sum_{j=1}^{\infty} |\xi_j|^q < \infty.$$

Since there is a constant $c > 0$ with $\mathbb{E}|\xi_j|^q = c$ for any $j \in \mathbb{N}$, this contradicts the law of large numbers.

Now define $\zeta_j = j^{(t-s)q/d} |\xi_j|^q$. Using the fact that the ζ_j are nonnegative and independent, we deduce from Lemma 3 (below) that

$$\sum_{j=1}^{\infty} \mathbb{E}(\zeta_j \wedge 1) = \sum_{j=1}^{\infty} \mathbb{E}\left(j^{(t-s)q/d} |\xi_j|^q \wedge 1\right) < \infty.$$

Since $t < s$ we note that then

$$\begin{aligned} \mathbb{E}\zeta_j &= \mathbb{E}\left(j^{-(s-t)q/d} |\xi_j|^q\right) \\ &= \mathbb{E}\left(j^{-(s-t)q/d} |\xi_j|^q \mathbb{1}_{\{|\xi_j| \leq j^{(s-t)/d}\}}\right) + \mathbb{E}\left(j^{-(s-t)q/d} |\xi_j|^q \mathbb{1}_{\{|\xi_j| > j^{(s-t)/d}\}}\right) \\ &\leq \mathbb{E}\left((\zeta_j \wedge 1) \mathbb{1}_{\{|\xi_j| \leq j^{(s-t)/d}\}}\right) + I \\ &\leq \mathbb{E}\left(\zeta_j \wedge 1\right) + I, \end{aligned}$$

where

$$I \propto j^{-(s-t)q/d} \int_{j^{(s-t)/d}}^{\infty} x^q e^{-x^q/2} dx.$$

Noting that, since $q \geq 1$, the function $x \mapsto x^q e^{-x^q/2}$ is bounded, up to a constant of proportionality, by the function $x \mapsto e^{-\alpha x}$ for any $\alpha < \frac{1}{2}$, we see that there is a positive constant K such that

$$\begin{aligned} I &\leq K j^{-(s-t)q/d} \int_{j^{(s-t)/d}}^{\infty} e^{-\alpha x} dx \\ &= \frac{1}{\alpha} K j^{-(s-t)q/d} \exp(-\alpha j^{(s-t)/d}) \\ &:= \iota_j. \end{aligned}$$

Thus we have shown that

$$\sum_{j=1}^{\infty} \mathbb{E}(j^{-(s-t)q/d} |\xi_j|^q) \leq \sum_{j=1}^{\infty} \mathbb{E}(\zeta_j \wedge 1) + \sum_{j=1}^{\infty} \iota_j < \infty.$$

Since the ξ_j are i.i.d. this implies that

$$\sum_{j=1}^{\infty} j^{(t-s)q/d} < \infty,$$

from which it follows that $(s-t)q/d > 1$ and (iii) follows. \square

Lemma 3. *Let $\{I_j\}_{j=1}^{\infty}$ be an independent sequence of \mathbb{R}^+ -valued random variables. Then*

$$\sum_{j=1}^{\infty} I_j < \infty \quad a.s. \Leftrightarrow \sum_{j=1}^{\infty} \mathbb{E}(I_j \wedge 1) < \infty.$$

As in the previous subsection, we now study the situation where the family $\{\phi_j\}$ has a uniform Hölder exponent α and study the implications for Hölder continuity of the random function u . In this case, however, the basis functions are normalized in L^2 and not L^∞ ; thus we must make additional assumptions on the possible growth of the L^∞ norms of $\{\phi_j\}$ with j . We assume that there are $C, a, b > 0$ and $\alpha \in (0, 1]$ such that, for all $j \geq 0$,

$$|\phi_j(x)| = \beta_j \leq C j^b, \quad x \in D. \quad (10.20a)$$

$$|\phi_j(x) - \phi_j(y)| \leq C j^a |x - y|^\alpha, \quad x, y \in D. \quad (10.20b)$$

We also assume that $a > b$ as, since $\|\phi_j\|_{L^2} = 1$, it is natural that the pre-multiplication constant in the Hölder estimate on the $\{\phi_j\}$ grows in j at least as fast as the bound on the functions themselves.

Theorem 6. Assume that u is given by (10.19) and (10.17) with ξ_1 drawn from a centred q -exponential distribution. Suppose also that (10.20) hold and that $s > d(b + q^{-1} + \frac{1}{2}\theta(a - b))$ for some $\theta \in (0, 2)$. Then \mathbb{P} -a.s. we have $u \in C^{0,\beta}(\mathbb{T}^d)$ for all $\beta < \frac{\alpha\theta}{2}$.

Proof. We apply Corollary 5 of the Kolmogorov continuity theorem and S_1 and S_2 are as defined there. We use θ in place of the parameter δ appearing in Corollary 5 in order to avoid confusion with δ appearing in Theorem 2 and (10.17) above. Let $f_j = \gamma_j \phi_j$ and note that

$$\begin{aligned} S_1 &= \sum_{j=1}^{\infty} |\gamma_j|^2 \beta_j^2 \lesssim \sum_{j=1}^{\infty} j^{-c_1} \\ S_2 &= \sum_{j=1}^{\infty} |\gamma_j|^{2-\theta} \beta_j^{2-\theta} \gamma_j^\theta j^{a\theta} \lesssim \sum_{j=1}^{\infty} j^{-c_2}. \end{aligned}$$

Short calculation shows that

$$\begin{aligned} c_1 &= \frac{2s}{d} + 1 - \frac{2}{q} - 2b, \\ c_2 &= \frac{2s}{d} + 1 - \frac{2}{q} - 2b - \theta(a - b). \end{aligned}$$

We require $c_1 > 1$ and $c_2 > 1$ and since $a > b$ satisfaction of the second of these will imply the first. Satisfaction of the second gives the desired lower bound on s .

We note that the result of Theorem 6 holds true when the mean function is nonzero if it satisfies

$$|m_0(x)| \leq C, \quad x \in D.$$

$$|m_0(x) - m_0(y)| \leq C|x - y|^\alpha, \quad x, y \in D.$$

We have the following sharper result if the family $\{\phi_j\}$ is regular enough to be a basis for B'_{qq} instead of satisfying (10.20):

Theorem 7. Assume that u is given by (10.19) and (10.17) with ξ_1 drawn from a centred q -exponential distribution. Suppose also that $\{\phi_j\}_{j \in \mathbb{N}}$ form a basis for B'_{qq} for some $t < s - \frac{d}{q}$. Then $u \in C^{0,t}(\mathbb{T}^d)$ \mathbb{P} -almost surely.

Proof. For any $m \geq 1$, using the definition of $X^{t,q}$ -norm, we can write

$$\|u\|_{B_{mq,mq}^t}^{mq} = (\frac{1}{\delta})^m \sum_{j=1}^{\infty} j^{\frac{mqt}{d} + \frac{mq}{2} - 1} j^{-mq(\frac{s}{d} + \frac{1}{2} - \frac{1}{q})} |\xi_j|^{mq}.$$

For every $m \in \mathbb{N}$, there exists a constant C_m with $\mathbb{E}|\xi_j|^{mq} = C_m$. Since each term of the above series is measurable, we can swap the sum and the integration and write

$$\mathbb{E}\|u\|_{B_{mq,mq}^t}^{mq} = C_m (\frac{1}{\delta})^m \sum_{j=1}^{\infty} j^{\frac{mq}{d}(t-s) + m - 1} \leq \tilde{C}_m,$$

noting that the exponent of j is smaller than -1 (since $t < s - d/q$). Now for a given $t < s - d/q$, one can choose m large enough so that $\frac{d}{mq} < s - d/q - t$. Then the embedding $B_{mq,mq}^{t_1} \subset C^t$ for any t_1 satisfying $t + \frac{d}{mq} < t_1 < s - d/q$ implies that $\mathbb{E}\|u\|_{C^t(\mathbb{T}^d)}^{mq} < \infty$. It follows that $u \in C^t$ \mathbb{P} -almost surely.

If the mean function m_0 is t -Hölder continuous, the result of the above theorem holds for a random series with nonzero mean function as well.

2.4 Gaussian Priors

Let X be a Hilbert space \mathcal{H} of real-valued functions on bounded open $D \subset \mathbb{R}^d$ with Lipschitz boundary and with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively; for example, $\mathcal{H} = L^2(D; \mathbb{R})$. Assume that $\{\phi_j\}_{j=1}^{\infty}$ is an orthonormal basis for \mathcal{H} . We study the Gaussian case where $\xi_1 \sim N(0, 1)$, and then equation (10.11) with $u_j = \gamma_j \xi_j$ generates random draws from the Gaussian measure $N(m_0, C)$ on \mathcal{H} where the covariance operator C depends on the sequence $\gamma = \{\gamma_j\}_{j=1}^{\infty}$. See the Appendix for background on Gaussian measures in a Hilbert space. As in Sect. 2.3, we consider the setting in which $m_0 = 0$ so that the function u is given by (10.16) and has mean zero. We thus focus on identifying C from the random series (10.16) and studying the regularity of random draws from $N(0, C)$.

Define the Hilbert scale of spaces \mathcal{H}^t as in Sect. A.1.3 with, recall, norm

$$\|u\|_{\mathcal{H}^t}^2 = \sum_{j=1}^{\infty} j^{\frac{2t}{d}} |u_j|^2.$$

We choose $\xi_1 \sim N(0, 1)$ and study convergence of the series (10.18) for u^N to a limit function u given by (10.19); the spaces in which this convergence occurs will depend upon the sequence γ . To understand the sequence of functions $\{u^N\}$, it is useful to introduce the following function space:

$$L^2_{\mathbb{P}}(\Omega; \mathcal{H}') := \left\{ v : D \times \Omega \rightarrow \mathbb{R} \mid \mathbb{E}(\|v\|_{\mathcal{H}'})^2 < \infty \right\}.$$

This is in fact a Hilbert space, although we will not use the Hilbert space structure. We will only use the fact that $L^2_{\mathbb{P}}$ is a Banach space when equipped with the norm $(\mathbb{E}(\|v\|_{\mathcal{H}^t}^2))^{\frac{1}{2}}$ and that hence every Cauchy sequence is convergent.

Theorem 8. Assume that $\gamma_j \asymp j^{-\frac{s}{d}}$. Then the sequence of functions $\{u^N\}_{N=1}^\infty$ given by (10.18) is Cauchy in the Hilbert space $L^2_{\mathbb{P}}(\Omega; \mathcal{H}^t)$, $t < s - \frac{d}{2}$. Thus, the infinite series (10.19) exists as an $L^2_{\mathbb{P}}$ limit and takes values in \mathcal{H}^t almost surely, for $t < s - \frac{d}{2}$.

Proof. For $N > M$,

$$\begin{aligned}\mathbb{E}\|u^N - u^M\|_{\mathcal{H}^t}^2 &= \mathbb{E} \sum_{j=M+1}^N j^{\frac{2t}{d}} |u_j|^2 \\ &\asymp \sum_{j=M+1}^N j^{\frac{2(t-s)}{d}} \leq \sum_{j=M+1}^\infty j^{\frac{2(t-s)}{d}}.\end{aligned}$$

The sum on the right-hand side tends to 0 as $M \rightarrow \infty$, provided $\frac{2(t-s)}{d} < -1$, by the dominated convergence theorem. This completes the proof. \square

Remarks 2. We make the following remarks concerning the Gaussian random functions constructed in the preceding theorem.

- The preceding theorem shows that the sum (10.18) has an $L^2_{\mathbb{P}}$ limit in \mathcal{H}^t when $t < s - d/2$, as one can also see from the following direct calculation

$$\begin{aligned}\mathbb{E}\|u\|_{\mathcal{H}^t}^2 &= \sum_{j=1}^\infty j^{\frac{2t}{d}} \mathbb{E}(\gamma_j^2 \xi_j^2) \\ &= \sum_{j=1}^\infty j^{\frac{2t}{d}} \gamma_j^2 \\ &\asymp \sum_{j=1}^\infty j^{\frac{2(t-s)}{d}} < \infty.\end{aligned}$$

Thus $u \in \mathcal{H}^t$ a.s., for $t < s - \frac{d}{2}$.

- From the preceding theorem, we see that, provided $s > \frac{d}{2}$, the random function in (10.19) generates a mean-zero Gaussian measure on \mathcal{H} . The expression (10.19) is known as the *Karhunen-Loève expansion* and the eigenfunctions $\{\phi_j\}_{j=1}^\infty$ as the *Karhunen-Loève basis*.

- The covariance operator \mathcal{C} of a measure μ on \mathcal{H} may then be viewed as a bounded linear operator from \mathcal{H} into itself defined to satisfy

$$\mathcal{C}\ell = \int_{\mathcal{H}} \langle \ell, u \rangle u \mu(du), \quad (10.24)$$

for all $\ell \in \mathcal{H}$. Thus

$$\mathcal{C} = \int_{\mathcal{H}} u \otimes u \mu(du). \quad (10.25)$$

The following formal calculation, which can be made rigorous if \mathcal{C} is trace class on \mathcal{H} , gives an expression for the covariance operator:

$$\begin{aligned} \mathcal{C} &= \mathbb{E} u \otimes u \\ &= \mathbb{E} \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \xi_j \xi_k \phi_j \otimes \phi_k \right) \\ &= \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \mathbb{E}(\xi_j \xi_k) \phi_j \otimes \phi_k \right) \\ &= \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \delta_{jk} \phi_j \otimes \phi_k \right) \\ &= \sum_{j=1}^{\infty} \gamma_j^2 \phi_j \otimes \phi_j. \end{aligned}$$

From this expression for the covariance, we may find eigenpairs explicitly:

$$\begin{aligned} \mathcal{C}\phi_k &= \left(\sum_{j=1}^{\infty} \gamma_j^2 \phi_j \otimes \phi_j \right) \phi_k \\ &= \sum_{j=1}^{\infty} \gamma_j^2 \langle \phi_j, \phi_k \rangle \phi_j = \sum_{j=1}^{\infty} \gamma_j^2 \delta_{jk} \phi_k = \gamma_k^2 \phi_k. \end{aligned}$$

- The Gaussian measure is denoted by $\mu_0 := N(0, \mathcal{C})$, a Gaussian with mean function 0 and covariance operator \mathcal{C} . The eigenfunctions of \mathcal{C} , $\{\phi_j\}_{j=1}^{\infty}$, are known as the *Karhunen-Loëve* basis for measure μ_0 . The γ_j^2 are the eigenvalues associated with this eigenbasis, and thus γ_j is the standard deviation of the Gaussian measure in the direction ϕ_j .

In the case where $\mathcal{H} = \dot{L}^2(\mathbb{T}^d)$, we are in the setting of Sect. 2.3 and we briefly consider this case. We assume that the $\{\phi_j\}_{j=1}^\infty$ constitute the Fourier basis. Let $A = -\Delta$ denote the negative Laplacian equipped with periodic boundary conditions on $[0, 1]^d$ and restricted to functions which integrate to zero over $[0, 1]^d$. This operator is positive self-adjoint and has eigenvalues which grow like $j^{2/d}$, analogously to Assumption 1 made in the case of Dirichlet boundary conditions. It then follows that $\mathcal{H}' = \mathcal{D}(A^{t/2}) = \dot{H}^t(\mathbb{T}^d)$, the Sobolev space of periodic functions on $[0, 1]^d$ with spatial mean equal to zero and t (possibly negative or fractional) square integrable derivatives. Thus, by the preceding Remarks 2, u defined by (10.19) is in the space \dot{H}^t a.s., $t < s - \frac{d}{2}$. In fact we can say more about regularity, using the Kolmogorov continuity test and Corollary 4; this we now do.

Theorem 9. *Consider the Karhunen-Loëve expansion (10.19) so that u is a sample from the measure $N(0, \mathcal{C})$ in the case where $\mathcal{C} = A^{-s}$ with $A = -\Delta$, $\mathcal{D}(A) = \dot{H}^2(\mathbb{T}^d)$ and $s > \frac{d}{2}$. Then, \mathbb{P} -a.s., $u \in \dot{H}^t$, $t < s - \frac{d}{2}$, and $u \in C^{0,t}(\mathbb{T}^d)$ a.s., $t < 1 \wedge (s - \frac{d}{2})$.*

Proof. Because of the stated properties of the eigenvalues of the Laplacian, it follows that the eigenvalues of \mathcal{C} satisfy $\gamma_j^2 \asymp j^{-\frac{2s}{d}}$ and the eigenbasis $\{\phi_j\}$ is the Fourier basis. Thus we may apply the conclusions stated in Remarks 2 to deduce that $u \in \dot{H}^t$, $t < \alpha - \frac{d}{2}$. Furthermore we may apply Corollary 5 to obtain Hölder regularity of u . To do this, we note that the $\{\phi_j\}$ are bounded in $L^\infty(\mathbb{T}^d)$ and are Lipschitz with constants which grow like $j^{1/d}$. We apply that corollary with $\alpha = 1$ and obtain

$$S_1 = \sum_{j=1}^{\infty} \gamma_j^2, \quad S_2 = \sum_{j=1}^{\infty} \gamma_j^2 j^{\delta/d}.$$

The corollary delivers the desired result after noting that any $\delta < 2s - d$ will make S_2 , and hence S_1 , summable.

The previous example illustrates the fact that, although we have constructed Gaussian measures in a Hilbert space setting, and that they are naturally defined on a range of Hilbert (Sobolev-like) spaces defined through fractional powers of the Laplacian, they may also be defined on Banach spaces, such as the space of Hölder continuous functions. We now return to the setting of the general domain D , rather than the d -dimensional torus. In this general context, it is important to highlight the Fernique theorem, here restated from the Appendix because of its importance:

Theorem 10 (Fernique Theorem). *Let μ_0 be a Gaussian measure on the separable Banach space X . Then there exists $\beta_c \in (0, \infty)$ such that, for all $\beta \in (0, \beta_c)$,*

$$\mathbb{E}^{\mu_0} \exp(\beta \|u\|_X^2) < \infty.$$

Remarks 3. We make two remarks concerning the Fernique theorem.

- Theorem 10, when combined with Theorem 9, shows that, with β sufficiently small, $\mathbb{E}^{\mu_0} \exp(\beta \|u\|_X^2) < \infty$ for both $X = \dot{H}^t$ and $X = C^{0,t}(\mathbb{T}^d)$, if $t < s - \frac{d}{2}$.
- Let $\mu_0 = N(0, A^{-s})$ where A is as in Theorem 9. Then Theorem 5 proves the Fernique theorem 10 for $X = X^{t,2} = \dot{H}^t$, if $t < s - \frac{d}{2}$; the proof in the case of the torus is very different from the general proof of the result in the abstract setting of Theorem 10.
- Theorem 5(ii) gives, in the Gaussian case, the Fernique theorem in the case that X is the Hilbert space $X^{t,2}$. Furthermore, the constant β_c is specified explicitly in that setting. More explicit versions of the general Fernique Theorem 10 are possible, but the characterization of β_c is more involved.

Example 3. Consider the random function (10.11) in the case where $\mathcal{H} = \dot{L}^2(\mathbb{T}^d)$ and $\mu_0 = N(0, A^{-s})$, $s > \frac{d}{2}$ as in the preceding example. Then we know that, μ_0 -a.s., $u \in C^{0,t}$, $t < 1 \wedge (s - \frac{d}{2})$. Set $\kappa = e^u$ in the elliptic PDE (10.7) so that the coefficient κ and hence the solution p are random variables on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Then κ_{\min} given in (10.6) satisfies

$$\kappa_{\min} \geq \exp(-\|u\|_\infty).$$

By Lemma 2 we obtain

$$\|p\|_V \leq \exp(\|u\|_\infty) \|f\|_{V^*}.$$

Since $C^{0,t} \subset L^\infty(\mathbb{T}^d)$, $t \in (0, 1)$, we deduce that

$$\|u\|_{L^\infty} \leq K_1 \|u\|_{C^{0,t}}.$$

Furthermore, for any $\epsilon > 0$, there is constant $K_2 = K_2(\epsilon)$ such that $\exp(K_1 r x) \leq K_2 \exp(\epsilon x^2)$ for all $x \geq 0$. Thus

$$\begin{aligned} \|p\|_V^r &\leq \exp(K_1 r \|u\|_{C^{0,t}}) \|f\|_{V^*}^r \\ &\leq K_2 \exp(\epsilon \|u\|_{C^{0,t}}^2) \|f\|_{V^*}^r. \end{aligned}$$

Hence, by Theorem 10, we deduce that

$$\mathbb{E}\|p\|_V^r < \infty, \quad \text{i.e.} \quad p \in L_{\mathbb{P}}^r(\Omega; V) \quad \forall r \in \mathbb{Z}^+.$$

This result holds for any $r \geq 0$. Thus, when the coefficient of the elliptic PDE is *log normal*, that is, κ is the exponential of a Gaussian function, moments of all orders exist for the random variable p . However, unlike the case of the uniform prior, we cannot obtain exponential moments on $\mathbb{E} \exp(\alpha \|p\|_V^r)$ for any $(r, \alpha) \in \mathbb{Z}^+ \times (0, \infty)$.

This is because the coefficient κ , while positive a.s., does not satisfy a uniform positive lower bound across the probability space. \square

2.5 Random Field Perspective

In this subsection we link the preceding constructions of random functions, through randomized series, to the notion of *random fields*. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, with expectation denoted by \mathbb{E} , and $D \subseteq \mathbb{R}^d$ an open set. For the random series constructions developed in the preceding Subsections, $\Omega = \mathbb{R}^\infty$ and $\mathcal{F} = \mathbf{B}(\Omega)$; however, the development of the general theory of random fields does not require this specific choice. A *random field* on D is a measurable mapping $u : D \times \Omega \rightarrow \mathbb{R}^n$. Thus, for any $x \in D$, $u(x; \cdot)$ is an \mathbb{R}^n -valued random variable; on the other hand, for any $\omega \in \Omega$, $u(\cdot; \omega) : D \rightarrow \mathbb{R}^n$ is a vector field. In the construction of random fields, it is commonplace to first construct the *finite-dimensional distributions*. These are found by choosing any integer $K \geq 1$, and any set of points $\{x_k\}_{k=1}^K$ in D , and then considering the random vector $(u(x_1; \cdot)^*, \dots, u(x_K; \cdot)^*)^* \in \mathbb{R}^{nK}$. From the finite-dimensional distributions of this collection of random vectors, we would like to be able to make sense of the probability measure μ on X , a separable Banach space equipped with the Borel σ -algebra $\mathbf{B}(X)$, via the formula

$$\mu(A) = \mathbb{P}(u(\cdot; \omega) \in A), \quad A \in \mathbf{B}(X), \quad (10.26)$$

where ω is taken from a common probability space on which the random element $u \in X$ is defined. It is thus necessary to study the joint distribution of a set of K \mathbb{R}^n -valued random variables, all on a common probability space. Such \mathbb{R}^{nK} -valued random variables are, of course, only defined up to a set of zero measure. It is desirable that all such finite-dimensional distributions are defined on a common subset $\Omega_0 \subset \Omega$ with full measure, so that u may be viewed as a function $u : D \times \Omega_0 \rightarrow \mathbb{R}^n$; such a choice of random field is termed a *modification*. When reinterpreting the previous subsections in terms of random fields, statements about almost sure (regularity) properties should be viewed as statements concerning the existence of a modification possessing of the stated almost sure regularity property.

We may define the space of functions

$$L_{\mathbb{P}}^q(\Omega; X) := \left\{ v : D \times \Omega \rightarrow \mathbb{R}^n \mid \mathbb{E}(\|v\|_X^q) < \infty \right\}.$$

This is a Banach space, when equipped with the norm $(\mathbb{E}(\|v\|_X^q))^{1/q}$. We have used such spaces in the preceding subsections when demonstrating convergence of the randomized series. Note that we often simply write $u(x)$, suppressing the explicit dependence on the probability space.

A *Gaussian random field* is one where, for any integer $K \geq 1$, and any set of points $\{x_k\}_{k=1}^K$ in D , the random vector $(u(x_1; \cdot)^*, \dots, u(x_K; \cdot)^*)^* \in \mathbb{R}^{nK}$ is a Gaussian random vector. The *mean function* of a Gaussian random field is $m(x) = \mathbb{E}u(x)$. The *covariance function* is $c(x, y) = \mathbb{E}(u(x) - m(x))(u(y) - m(y))^*$. For Gaussian random fields, the mean function $m : D \rightarrow \mathbb{R}^n$ and the covariance function $c : D \times D \rightarrow \mathbb{R}^{n \times n}$ together completely specify the joint probability distribution for $(u(x_1; \cdot)^*, \dots, u(x_K)^*)^* \in \mathbb{R}^{nK}$. Furthermore, if we view the Gaussian random field as a Gaussian measure on $L^2(D; \mathbb{R}^n)$, then the covariance operator can be constructed from the covariance function as follows. Without loss of generality, we consider the mean-zero case; the more general case follows by shift of origin. Since the field has mean zero, we have, from (10.24), that for all $h_1, h_2 \in L^2(D; \mathbb{R}^n)$,

$$\begin{aligned}\langle h_1, \mathcal{C}h_2 \rangle &= \mathbb{E}\langle h_1, u \rangle \langle u, h_2 \rangle \\ &= \mathbb{E} \int_D \int_D h_1(x)^*(u(x)u(y)^*)h_2(y)dydx \\ &= \mathbb{E} \int_D h_1(x)^* \left(\int_D (u(x)u(y)^*)h_2(y)dy \right) dx \\ &= \int_D h_1(x)^* \left(\int_D c(x, y)h_2(y)dy \right) dx\end{aligned}$$

and we deduce that, for all $\psi \in L^2(D; \mathbb{R}^n)$,

$$(\mathcal{C}\psi)(x) = \int_D c(x, y)\psi(y)dy. \quad (10.27)$$

Thus the covariance operator of a Gaussian random field is an integral operator with kernel given by the covariance function. As such we may also view the covariance function as the Green's function of the inverse covariance, or *precision*.

A mean-zero Gaussian random field is termed *stationary* if $c(x, y) = s(x - y)$ for some matrix-valued function s , so that shifting the field by a fixed random vector does not change the statistics. It is *isotropic* if it is stationary and, in addition, $s(\cdot) = \iota(|\cdot|)$, for some matrix-valued function ι .

In the previous subsection, we demonstrated how the regularity of random fields maybe established from the properties of the sequences γ (deterministic, with decay) and ξ (i.i.d. random). Here we show similar results but express them in terms of properties of the covariance function and covariance operator.

Theorem 11. *Consider an \mathbb{R}^n -valued Gaussian random field u on $D \subset \mathbb{R}^d$ with mean zero and with isotropic correlation function $c : D \times D \rightarrow \mathbb{R}^{n \times n}$. Assume that D is bounded and that $\text{Tr } c(x, y) = k(|x - y|)$ where $k : \mathbb{R}^+ \rightarrow \mathbb{R}$ is Hölder with any exponent $\alpha \leq 1$. Then u is almost surely Hölder continuous on D with any exponent smaller than $\frac{1}{2}\alpha$.*

Proof. We have

$$\begin{aligned}\mathbb{E}|u(x) - u(y)|^2 &= \mathbb{E}|u(x)|^2 + \mathbb{E}|u(y)|^2 - 2\mathbb{E}\langle u(x), u(y) \rangle \\ &= \text{Tr}\left(c(x, x) + c(y, y) - 2c(x, y)\right) \\ &= 2\left(k(0) - k(|x - y|)\right) \\ &\leq C|x - y|^\alpha.\end{aligned}$$

Since u is Gaussian, it follows that, for any integer $r > 0$,

$$\mathbb{E}|u(x) - u(y)|^{2r} \leq C_r|x - y|^{\alpha r}.$$

Let $p = 2r$ and noting that

$$\alpha r = p\left(\frac{\alpha}{2} - \frac{d}{p}\right) + d$$

we deduce from Corollary 4 that u is Hölder continuous on D with any exponent smaller than

$$\sup_{p \in \mathbb{N}} \min\left\{1, \frac{\alpha}{2} - \frac{d}{p}\right\} = \frac{\alpha}{2},$$

which is precisely what we claimed.

It is often convenient both algorithmically and theoretically to define the covariance operator through fractional inverse powers of a differential operator. Indeed in the previous subsection, we showed that our assumptions on the random series construction we used could be interpreted as having a covariance operator which was an inverse fractional power of the Laplacian on zero spatial average functions with periodic boundary conditions. We now generalize this perspective and consider covariance operators which are a fractional power of an operator A satisfying the following.

Assumption 2. *The operator A , densely defined on the Hilbert space $\mathcal{H} = L^2(D; \mathbb{R}^n)$, satisfies the following properties:*

1. *A is positive definite, self-adjoint and invertible;*
2. *the eigenfunctions $\{\phi_j\}_{j \in \mathbb{N}}$ of A form an orthonormal basis for \mathcal{H} ;*

3. the eigenvalues of A satisfy $\alpha_j \asymp j^{2/d}$;
4. there is $C > 0$ such that

$$\sup_{j \in \mathbb{N}} \left(\|\phi_j\|_{L^\infty} + \frac{1}{j^{1/d}} \text{Lip}(\phi_j) \right) \leq C.$$

These properties are satisfied by the Laplacian on a torus, when applied to functions with spatial mean zero. But they are in fact satisfied for a much wider range of differential operators which are *Laplacian-like*. For example, the Dirichlet Laplacian on a bounded open set D in \mathbb{R}^d , together with various Laplacian operators perturbed by lower order terms, for example, Schrödinger operators. Inspection of the proof of Theorem 9 reveals that it only uses the properties of Assumption 2. Thus we have:

Theorem 12. *Let u be a sample from the measure $N(0, \mathcal{C})$ in the case where $\mathcal{C} = A^{-s}$ with A satisfying Assumptions 2 and $s > \frac{d}{2}$. Then, \mathbb{P} -a.s., $u \in \dot{H}^t$, for $t < s - \frac{d}{2}$, and $u \in C^{0,t}(D)$, for $t < 1 \wedge (s - \frac{d}{2})$.*

Example 4. Consider the case $d = 2, n = 1$ and $D = [0, 1]^2$. Define the Gaussian random field through the measure $\mu = N(0, (-\Delta)^{-\alpha})$ where Δ is the Laplacian with domain $H_0^1(D) \cap H^2(D)$. Then Assumption 2 is satisfied by $-\Delta$. By Theorem 12 it follows that choosing $\alpha > 1$ suffices to ensure that draws from μ are almost surely in $L^2(D)$. It also follows that, in fact, draws from μ are almost surely in $C(D)$.

2.6 Summary

In the preceding four subsections, we have shown how to create random functions by randomizing the coefficients of a series of functions. Using these random series, we have also studied the regularity properties of the resulting functions. Furthermore we have extended our perspective in the Gaussian case to determine regularity properties from the properties of the covariance function or the covariance operator.

For the uniform prior, we have shown that the random functions all live in a subset of $X = L^\infty$ characterized by the upper and lower bounds given in Theorem 2 and found as the closure of the linear span of the set of functions $(m_0, \{\phi_j\}_{j=1}^\infty)$; denote this subset, which is a separable Banach space, by X' . For the Besov priors, we have shown in Theorem 5 that the random functions live in the separable Banach spaces $X^{t,q}$ for all $t < s - d/q$; denote any one of these Banach spaces by X' . And finally for the Gaussian priors, we have shown in Theorem 8 that the random function exists as an L^2 limit in any of the Hilbert spaces \mathcal{H}' for $t < s - d/2$. Furthermore, we have indicated that, by use of the Kolmogorov continuity theorem, we can also show that the Gaussian random functions lie in certain Hölder spaces; these Hölder spaces are not separable but, by the discussion in Sect. A.1.2, we

can embed the spaces $C^{0,\gamma'}$ in the separable uniform Hölder spaces $C_0^{0,\gamma}$ for any $\gamma < \gamma'$; since the upper bound on the range of Hölder exponents established by use of Kolmogorov continuity theorem is open, this means we can work in the same range of Hölder exponents, but restricted to uniform Hölder spaces, thereby regaining separability. In this Gaussian case, we denote any of the separable Hilbert or Banach spaces where the Gaussian random function lives almost surely by X' .

Thus, in all of these examples, we have created a probability measure μ_0 which is the pushforward of the measure \mathbb{P} on the i.i.d. sequence ξ under the map which takes the sequence into the random function. The resulting measure lives on the separable Banach space X' , and we will often write $\mu_0(X') = 1$ to denote this fact. This is shorthand for saying that functions drawn from μ_0 are in X' almost surely. Separability of X' naturally leads to the use of the Borel σ -algebra to define a canonical measurable space and to the development of an integration theory – Bochner integration – which is natural on this space; see Sect. A.2.2.

2.7 Bibliographic Notes

- Section 2.1. For general discussion of the properties of random functions constructed via randomization of coefficients in a series expansion, see [49]. The construction of probability measure on infinite sequences of i.i.d. random variables may be found in [27].
- Section 2.2. These uniform priors have been extensively studied in the context of the field of uncertainty quantification, and the reader is directed to [18, 19] for more details. Uncertainty quantification in this context does not concern inverse problems, but rather studies the effect, on the solution of an equation, of randomizing the input data. Thus, the interest is in the pushforward of a measure on input parameter space onto a measure on solution space, for a differential equation. Recently, however, these priors have been used to study the inverse problem; see [90].
- Section 2.3. Besov priors were introduced in the paper [69] and Theorem 5 is taken from that paper. We notice that the theorem constitutes a special case of the Fernique theorem in the Gaussian case $q = 2$; it is restricted to a specific class of Hilbert space norms, however, whereas the Fernique theorem in full generality applies in all norms on Banach spaces which have full Gaussian measure. See [35, 40] for proof of the Fernique theorem. A more general Fernique-like property of the Besov measures is proved in [24], but it remains open to determine the appropriate complete generalization of the Fernique theorem to Besov measures. For proof of Lemma 3, see [54, Chapter 4]. For properties of families of functions that can form a basis for a Besov space and examples of such families, see [31, 74].
- Section 2.4. The general theory of Gaussian measures on Banach spaces is contained in [14, 67]. The text [28], concerning the theory of stochastic PDEs, also has a useful overview of the subject. The Karhunen-Loëve expansion (10.19) is contained in [1]. The formal calculation concerning the covariance operator

of the Gaussian measure which follows Theorem 8 leads to the answer which may be rigorously justified by using characteristic functions; see, for example, Proposition 2.18 in [28]. All three texts include statement and proof of the Fernique theorem in the generality given here. The Kolmogorov continuity theorem is discussed in [28] and [1]. Proof of Hölder regularity adapted to the case of the periodic setting may be found in [40] and [92, Chapter 6]. For further reading on Gaussian measures, see [27].

- Section 2.5. A key tool in making the random field perspective rigorous is the Kolmogorov Extension Theorem 29.
- Section 2.6. For a discussion of measure theory on general spaces, see [15]. The notion of Bochner integral is introduced in [13]; we discuss it in Sect. A.2.2.

3 Posterior Distribution

In this section we prove a Bayes' theorem appropriate for combining a likelihood with prior measures on separable Banach spaces as constructed in the previous section. In Sect. 3.1, we start with some general remarks about conditioned random variables. Section 3.2 contains our statement and proof of a Bayes' theorem and specifically its application to Bayesian inversion. We note here that, in our setting, the posterior μ^y will always be absolutely continuous with respect to the prior μ_0 , and we use the standard notation $\mu^y \ll \mu_0$ to denote this. It is possible to construct examples, for instance, in the purely Gaussian setting, where the posterior is not absolutely continuous with respect to the prior. Thus, it is certainly not necessary to work in the setting where $\mu^y \ll \mu_0$. However, it is quite natural, from a modeling point of view, to work in this setting: absolute continuity ensures that almost sure properties built into the prior will be inherited by the posterior. For these almost sure properties to be changed by the data would require that the data contains an infinite amount of information, something which is unnatural in most applications.

In Sect. 3.3, we study the example of the heat equation, introduced in Sect. 1.2, from the perspective of Bayesian inversion, and in Sect. 3.4 we do the same for the elliptic inverse problem of Sect. 1.3.

3.1 Conditioned Random Variables

Key to the development of Bayes' Theorem, and the posterior distribution, is the notion of conditional random variables. In this section we state an important theorem concerning conditioning.

Let (X, A) and (Y, B) denote a pair of measurable spaces, and let ν and π be probability measures on $X \times Y$. We assume that $\nu \ll \pi$. Thus there exists π -measurable $\phi : X \times Y \rightarrow \mathbb{R}$ with $\phi \in L_\pi^1$ (see Sect. A.1.4 for definition of L_π^1) and

$$\frac{d\nu}{d\pi}(x, y) = \phi(x, y). \quad (10.28)$$

That is, for $(x, y) \in X \times Y$,

$$\mathbb{E}^v f(x, y) = \mathbb{E}^\pi(\phi(x, y)f(x, y)),$$

or, equivalently,

$$\int_{X \times Y} f(x, y)v(dx, dy) = \int_{X \times Y} \phi(x, y)f(x, y)\pi(dx, dy).$$

Theorem 13. Assume that the conditional random variable $x|y$ exists under π with probability distribution denoted $\pi^y(dx)$. Then the conditional random variable $x|y$ under v exists, with probability distribution denoted by $v^y(dx)$. Furthermore, $v^y \ll \pi^y$ and if $c(y) := \int_X \phi(x, y)d\pi^y(x) > 0$, then

$$\frac{d v^y}{d \pi^y}(x) = \frac{1}{c(y)}\phi(x, y).$$

Example 5. Let $X = C([0, 1]; \mathbb{R})$, $Y = \mathbb{R}$. Let π denote the measure on $X \times Y$ induced by the random variable $(w(\cdot), w(1))$, where w is a draw from standard unit Wiener measure on \mathbb{R} , starting from $w(0) = z$. Let π^y denote measure on X found by conditioning Brownian motion to satisfy $w(1) = y$, thus π^y is a Brownian bridge measure with $w(0) = z, w(1) = y$.

Assume that $v \ll \pi$ with

$$\frac{d v}{d \pi}(x, y) = \exp(-\Phi(x, y)).$$

Assume further that

$$\sup_{x \in X} \Phi(x, y) = \Phi^+(y) < \infty$$

for every $y \in \mathbb{R}$. Then

$$c(y) = \int_{\mathbb{R}} \exp(-\Phi(x, y))d\pi^y(x) > \exp(-\Phi^+(y)) > 0.$$

Thus $v^y(dx)$ exists and

$$\frac{d v^y}{d \pi^y}(x) = \frac{1}{c(y)} \exp(-\Phi(x, y)). \quad \square$$

We will use the preceding theorem to go from a construction of the joint probability distribution on unknown and data to the conditional distribution of the unknown, given data. In constructing the joint probability distribution, we will need to establish measurability of the likelihood, for which the following will be useful:

Lemma 4. Let (Z, B) be a Borel measurable topological space and assume that $G \in C(Z; \mathbb{R})$ and that $\pi(Z) = 1$ for some probability measure π on (Z, B) . Then G is a π -measurable function.

3.2 Bayes' Theorem for Inverse Problems

Let X and Y be separable Banach spaces, equipped with the Borel σ -algebra, and $G : X \rightarrow Y$ a measurable mapping. We wish to solve the inverse problem of finding u from y where

$$y = G(u) + \eta \quad (10.29)$$

and $\eta \in Y$ denotes noise. We employ a Bayesian approach to this problem in which we let $(u, y) \in X \times Y$ be a random variable and compute $u|y$. We specify the random variable (u, y) as follows:

- **Prior:** $u \sim \mu_0$ measure on X .
- **Noise:** $\eta \sim \mathbb{Q}_0$ measure on Y , and (recalling that \perp denotes independence) $\eta \perp u$.

The random variable $y|u$ is then distributed according to the measure \mathbb{Q}_u , the translate of \mathbb{Q}_0 by $G(u)$. We *assume* throughout the following that $\mathbb{Q}_u \ll \mathbb{Q}_0$ for u μ_0 -a.s. Thus, for some **potential** $\Phi : X \times Y \rightarrow \mathbb{R}$,

$$\frac{d\mathbb{Q}_u}{d\mathbb{Q}_0}(y) = \exp(-\Phi(u; y)). \quad (10.30)$$

Thus, for fixed u , $\Phi(u; \cdot) : Y \rightarrow \mathbb{R}$ is measurable and $\mathbb{E}^{\mathbb{Q}_0} \exp(-\Phi(u; y)) = 1$. For given instance of the data y , $-\Phi(\cdot; y)$ is termed the **log likelihood**.

Define v_0 to be the product measure

$$v_0(du, dy) = \mu_0(du)\mathbb{Q}_0(dy). \quad (10.31)$$

We *assume* in what follows that $\Phi(\cdot, \cdot)$ is v_0 measurable. Then the random variable $(u, y) \in X \times Y$ is distributed according to measure $v(du, dy) = \mu_0(du)\mathbb{Q}_u(dy)$. Furthermore, it then follows that $v \ll v_0$ with

$$\frac{d\nu}{dv_0}(u, y) = \exp(-\Phi(u; y)).$$

We have the following infinite-dimensional analogue of Theorem 1.

Theorem 14 (Bayes' Theorem). *Assume that $\Phi : X \times Y \rightarrow \mathbb{R}$ is v_0 measurable and that, for y \mathbb{Q}_0 -a.s.,*

$$Z := \int_X \exp(-\Phi(u; y))\mu_0(du) > 0. \quad (10.32)$$

Then the conditional distribution of $u|y$ exists under ν and is denoted by μ^y . Furthermore $\mu^y \ll \mu_0$ and, for y ν -a.s.,

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u; y)). \quad (10.33)$$

Proof. First note that the positivity of Z holds for y ν_0 -almost surely, and hence by absolute continuity of ν with respect to ν_0 , for y ν -almost surely. The proof is an application of Theorem 13 with π replaced by ν_0 , $\phi(x, y) = \exp(-\Phi(u, y))$ and $(x, y) = (u, y)$. Since $\nu_0(du, dy)$ has product form, the conditional distribution of $u|y$ under ν_0 is simply μ_0 . The result follows. \square

Remarks 4. In order to implement the derivation of Bayes' formula (10.33), four essential steps are required:

- Define a suitable prior measure μ_0 and noise measure \mathbb{Q}_0 whose independent product form the reference measure ν_0 .
- Determine the potential Φ such that formula (10.30) holds.
- Show that Φ is ν_0 measurable.
- Show that the normalization constant Z given by (10.32) is positive almost surely with respect to $y \sim \mathbb{Q}_0$.

We will show how to carry out this program for two examples in the following subsections. The following remark will also be used in studying one of the examples.

Remarks 5. The following comments on the setup above may be useful.

- In formula (10.33) we can shift $\Phi(u, y)$ by any constant $c(y)$, independent of u , provided the constant is finite \mathbb{Q}_0 -a.s. and hence ν -a.s. Such a shift can be absorbed into a redefinition of the normalization constant Z .
- Our Bayes' Theorem only asserts that the posterior is absolutely continuous with respect to the prior μ_0 . In fact equivalence (mutual absolute continuity) will occur when $\Phi(\cdot; y)$ is finite everywhere in X .

3.3 Heat Equation

We apply Bayesian inversion to the heat equation from Sect. 1.2. Recall that for $G(u) = e^{-A}u$, we have the relationship

$$y = G(u) + \eta,$$

which we wish to invert. Let $X = H$ and define

$$\mathcal{H}^t = \mathcal{D}(A^{t/2}) = \left\{ w \mid w = A^{-t/2}w_0, w_0 \in H \right\}.$$

Under Assumption 1, we have $\alpha_j \asymp j^{\frac{2}{d}}$ so that this family of spaces is identical with the Hilbert scale of spaces \mathcal{H}^t as defined in Sects. 1.2 and 2.4.

We choose the prior $\mu_0 = N(0, A^{-\alpha})$, $\alpha > \frac{d}{2}$. Thus $\mu_0(X) = \mu_0(H) = 1$. Indeed the analysis in Sect. 2.4 shows that $\mu_0(\mathcal{H}^t) = 1$, $t < \alpha - \frac{d}{2}$. For the likelihood we assume that $\eta \perp u$ with $\eta \sim \mathbb{Q}_0 = N(0, A^{-\beta})$, and $\beta \in \mathbb{R}$. This measure satisfies $\mathbb{Q}_0(\mathcal{H}^t) = 1$ for $t < \beta - \frac{d}{2}$ and we thus choose $Y = \mathcal{H}^t$ for some $t' < \beta - \frac{d}{2}$. Notice that our analysis includes the case of white observational noise, for which $\beta = 0$. The Cameron-Martin Theorem 32, together with the fact that $e^{-\lambda A}$ commutes with arbitrary fractional powers of A , can be used to show that $y|u \sim \mathbb{Q}_u := N(G(u), A^{-\beta})$ where $\mathbb{Q}_u \ll \mathbb{Q}_0$ with

$$\frac{d\mathbb{Q}_u}{d\mathbb{Q}_0}(y) = \exp(-\Phi(u; y)),$$

and

$$\Phi(u; y) = \frac{1}{2} \|A^{\frac{\beta}{2}} e^{-A} u\|^2 - \langle A^{\frac{\beta}{2}} e^{-\frac{A}{2}} y, A^{\frac{\beta}{2}} e^{-\frac{A}{2}} u \rangle.$$

In the following we repeatedly use the fact that $A^\gamma e^{-\lambda A}$, $\lambda > 0$, is a bounded linear operator from \mathcal{H}^a to \mathcal{H}^b , any $a, b, \gamma \in \mathbb{R}$. Recall that $\nu_0(du, dy) = \mu_0(du)\mathbb{Q}_0(dy)$. Note that $\nu_0(H \times \mathcal{H}^{t'}) = 1$. Using the boundedness of $A^\gamma e^{-\lambda A}$, it may be shown that

$$\Phi : H \times \mathcal{H}^{t'} \rightarrow \mathbb{R}$$

is continuous and hence ν_0 -measurable by Lemma 4.

Theorem 14 shows that the posterior is given by μ^y where

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u; y)),$$

$$Z = \int_H \exp(-\Phi(u; y)) \mu_0(du),$$

provided that $Z > 0$ for y \mathbb{Q}_0 -a.s. We establish this positivity in the remainder of the proof. Since $y \in \mathcal{H}^t$ for any $t < \beta - \frac{d}{2}$, \mathbb{Q}_0 -a.s., we have that $y = A^{-t'/2} w_0$ for some $w_0 \in H$ and $t' < \beta - \frac{d}{2}$. Thus we may write

$$\Phi(u; y) = \frac{1}{2} \|A^{\frac{\beta}{2}} e^{-A} u\|^2 - \langle A^{\frac{\beta-t'}{2}} e^{-\frac{A}{2}} w_0, A^{\frac{\beta}{2}} e^{-\frac{A}{2}} u \rangle. \quad (10.34)$$

Then, using the boundedness of $A^\gamma e^{-\lambda A}$, $\lambda > 0$, together with (10.34), we have

$$\Phi(u; y) \leq C(\|u\|^2 + \|w_0\|^2)$$

where $\|w_0\|$ is finite \mathbb{Q}_0 -a.s. Thus,

$$Z \geq \int_{\|u\|^2 \leq 1} \exp(-C(1 + \|w_0\|^2)) \mu_0(du)$$

and, since $\mu_0(\|u\|^2 \leq 1) > 0$ (by Theorem 33 all balls have positive measure for Gaussians on a separable Banach space), the required positivity follows.

3.4 Elliptic Inverse Problem

We consider the elliptic inverse problem from Sect. 1.3 from the Bayesian perspective. We consider the use of both uniform and Gaussian priors. Before studying the inverse problem, however, it is important to derive some continuity properties of the forward problem. Throughout this section, we consider equation (10.7) under the assumption that $f \in V^*$.

3.4.1 Forward Problem

Recall that in Sect. 1.3, equation (10.10), we defined

$$X^+ = \left\{ v \in L^\infty(D) \mid \text{ess inf}_{x \in D} v(x) > 0 \right\}. \quad (10.35)$$

Then the map $\mathcal{R} : X^+ \rightarrow V$ by $\mathcal{R}(v) = p$. This map is well-defined by Lemma 2 and we have the following result.

Lemma 5. *For $i = 1, 2$, let*

$$\begin{aligned} -\nabla \cdot (\kappa_i \nabla p_i) &= f, \quad x \in D, \\ p_i &= 0, \quad x \in \partial D. \end{aligned}$$

Then

$$\|p_1 - p_2\|_V \leq \frac{1}{\kappa_{\min}^2} \|f\|_{V^*} \|\kappa_1 - \kappa_2\|_{L^\infty}$$

where we assume that

$$\kappa_{\min} := \text{ess inf}_{x \in D} \kappa_1(x) \wedge \text{ess inf}_{x \in D} \kappa_2(x) > 0.$$

Thus the function $\mathcal{R} : X^+ \rightarrow V$ is locally Lipschitz.

Proof. Let $e = \kappa_1 - \kappa_2$, $r = p_1 - p_2$. Then

$$\begin{aligned} -\nabla \cdot (\kappa_1 \nabla r) &= \nabla \cdot ((\kappa_1 - \kappa_2) \nabla p_2), \quad x \in D \\ r &= 0, \quad x \in \partial D. \end{aligned}$$

Multiplying by r and integrating by parts on both sides of the identity gives

$$\kappa_{\min} \int_D |\nabla r|^2 dx \leq \|(\kappa_2 - \kappa_1) \nabla p_2\| \|\nabla r\|.$$

Using the fact that $\|\varphi\|_V = \|\nabla \varphi\|$, and applying Lemma 2 to bound p_2 in V , we find that

$$\begin{aligned} \|r\|_V &\leq \|(\kappa_2 - \kappa_1) \nabla p_2\| / \kappa_{\min} \\ &\leq \|\kappa_2 - \kappa_1\|_{L^\infty} \|p_2\|_V / \kappa_{\min} \\ &\leq \frac{1}{\kappa_{\min}^2} \|f\|_{V^*} \|e\|_{L^\infty}. \end{aligned}$$

□

3.4.2 Uniform Priors

We now study the inverse problem of finding κ from a finite set of continuous linear functionals $\{l_j\}_{j=1}^J$ on V , representing measurements of p ; thus $l_j \in V^*$. To match the notation from Sect. 3.2, we take $\kappa = u$ and we define the separable Banach space X' as in Sect. 2.2. It is straightforward to see that Lemma 5 extends to the case where X^+ given by (10.35) is replaced by

$$X^+ = \left\{ v \in X' \mid \text{ess inf}_{x \in D} v(x) > 0 \right\} \quad (10.36)$$

since $X' \subset L^\infty(D)$. When considering uniform priors for the elliptic problem, we work with this definition of X^+ .

We define $G : X^+ \rightarrow \mathbb{R}^J$ by

$$G_j(u) = l_j(\mathcal{R}(u)), \quad j = 1, \dots, J$$

where, recall, the l_j are elements of V^* : bounded linear functionals on V . Then $G(u) = (G_1(u), \dots, G_J(u))$ and we are interested in the inverse problem of finding $u \in X^+$ from y where

$$y = G(u) + \eta$$

and η is the noise. We assume $\eta \sim N(0, \Gamma)$, for positive symmetric $\Gamma \in \mathbb{R}^{J \times J}$. (Use of other statistical assumptions on η is a straightforward extension of what follows whenever η has a smooth density on \mathbb{R}^J .)

Let μ_0 denote the prior measure constructed in Sect. 2.2. Then μ_0 -almost surely we have, by Theorem 2,

$$u \in X_0^+ := \left\{ v \in X' \mid \frac{1}{1+\delta} m_{\min} \leq v(x) \leq m_{\max} + \frac{\delta}{1+\delta} m_{\min} \quad \text{a.e. } x \in D \right\}. \quad (10.37)$$

Thus $\mu_0(X_0^+) = 1$.

The likelihood is defined as follows. Since $\eta \sim N(0, \Gamma)$, it follows that $\mathbb{Q}_0 = N(0, \Gamma)$, $\mathbb{Q}_u = N(G(u), \Gamma)$ and

$$\begin{aligned} \frac{d\mathbb{Q}_u}{d\mathbb{Q}_0}(y) &= \exp(-\Phi(u; y)), \\ \Phi(u; y) &= \frac{1}{2} |\Gamma^{-\frac{1}{2}}(y - G(u))|^2 - \frac{1}{2} |\Gamma^{-\frac{1}{2}}y|^2. \end{aligned}$$

Recall that $\nu_0(dy, du) = \mathbb{Q}_0(dy)\mu_0(du)$. Since $G : X^+ \rightarrow \mathbb{R}^J$ is locally Lipschitz by Lemma 5, Lemma 4 implies that $\Phi : X^+ \times Y \rightarrow \mathbb{R}$ is ν_0 -measurable. Thus Theorem 14 shows that $u|y \sim \mu^y$ where

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u; y)) \quad (10.38)$$

$$Z = \int_{X^+} \exp(-\Phi(u; y)) \mu_0(du),$$

provided $Z > 0$ for y \mathbb{Q}_0 -almost surely. To see that $Z > 0$, note that

$$Z = \int_{X_0^+} \exp(-\Phi(u; y)) \mu_0(du),$$

since $\mu_0(X_0^+) = 1$. On X_0^+ we have that $\mathcal{R}(\cdot)$ is bounded in V , and hence G is bounded in \mathbb{R}^J . Furthermore y is finite \mathbb{Q}_0 -almost surely. Thus \mathbb{Q}_0 -almost surely with respect to y , $\Phi(\cdot; y)$ is bounded on X_0^+ ; we denote the resulting bound by $M = M(y) < \infty$. Hence

$$Z \geq \int_{X_0^+} \exp(-M) \mu_0(du) = \exp(-M) > 0.$$

and the result is proved.

We may use Remark 5 to shift Φ by $\frac{1}{2}|\Gamma^{-\frac{1}{2}}y|^2$, since this is almost surely finite under \mathbb{Q}_0 and hence under $v(du, dy) = \mathbb{Q}_u(dy)\mu_0(du)$. We then obtain the equivalent form for the posterior distribution μ^y :

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp\left(-\frac{1}{2}|\Gamma^{-\frac{1}{2}}(y - G(u))|^2\right), \quad (10.39a)$$

$$Z = \int_X \exp\left(-\frac{1}{2}|\Gamma^{-\frac{1}{2}}(y - G(u))|^2\right) \mu_0(du). \quad (10.39b)$$

3.4.3 Gaussian Priors

We conclude this subsection by discussing the same inverse problem, but using Gaussian priors from Sect. 2.4. We now set $X = C(\overline{D})$ and $Y = \mathbb{R}^J$ and we note that X embeds continuously into $L^\infty(D)$. We assume that we can find an operator A which satisfies Assumption 2. We now take $\kappa = \exp(u)$ and define $G : X \rightarrow \mathbb{R}^J$ by

$$G_j(u) = l_j(\mathcal{R}(\exp(u))), \quad j = 1, \dots, J.$$

We take as prior on u the measure $N(0, A^{-s})$ with $s > d/2$. Then Theorem 12 shows that $\mu(X) = 1$. The likelihood is unchanged by the prior, since it concerns y given u , and is hence identical to that in the case of the uniform prior, although the mean shift from \mathbb{Q}_0 to \mathbb{Q}_u by $G(u)$ now has a different interpretation since $\kappa = \exp(u)$ rather than $\kappa = u$. Thus we again obtain (10.38) for the posterior distribution (albeit with a different definition of $G(u)$) provided that we can establish that, \mathbb{Q}_0 -a.s.,

$$Z = \int_X \exp\left(\frac{1}{2}|\Gamma^{-\frac{1}{2}}y|^2 - \frac{1}{2}|\Gamma^{-\frac{1}{2}}(y - G(u))|^2\right) \mu_0(du) > 0.$$

To this end, we use the fact that the unit ball in X , denoted B , has positive measure by Theorem 33 and that on this ball $\mathcal{R}(\exp(u))$ is bounded in V by $e^{-a}\|f\|_{V^*}$, by Lemma 2, for some finite positive constant a . This follows from the continuous embedding of X into L^∞ and since the infimum of $\kappa = \exp(u)$ is bounded below by $e^{-\|u\|_{L^\infty}}$. Thus G is bounded on B and, noting that y is \mathbb{Q}_0 -a.s. finite, we have for some $M = M(y) < \infty$,

$$\sup_{u \in B} \left(\frac{1}{2}|\Gamma^{-\frac{1}{2}}(y - G(u))|^2 - \frac{1}{2}|\Gamma^{-\frac{1}{2}}y|^2 \right) < M.$$

Hence

$$Z \geq \int_B \exp(-M) \mu_0(du) = \exp(-M) \mu_0(B) > 0$$

since all balls have positive measure for Gaussian measure on a separable Banach space. Thus we again obtain (10.39) for the posterior measure, now with the new definition of G , and hence Φ .

3.5 Bibliographic Notes

- Section 3.1. Theorem 13 is taken from [43] where it is used to compute expressions for the measure induced by various conditionings applied to SDEs. The existence of regular conditional probability distributions is discussed in [54], Theorem 6.3. Example 5, concerning end-point conditioning of measures defined via a density with respect to Wiener measure, finds application to problems from molecular dynamics in [82, 83]. Further material concerning the equivalence of posterior with respect to the prior may be found in [92, Chapters 3 and 6], [3, 4]. The equivalence of Gaussian measures is studied via the Feldman-Hájek theorem; see [28] and the Appendix. A proof of Lemma 4 can be found in [88, Chapter 1, Theorem 1.12]. See also [54, Lemma 1.5].
- Section 3.2. General development of Bayes' Theorem for inverse problems on function space, along the lines described here, may be found in [17, 92]. The reader is also directed to the papers [61, 62] for earlier related material and to [63–65] for recent developments.
- Section 3.3. The inverse problem for the heat equation was one of the first infinite-dimensional inverse problems to receive Bayesian treatment (see [36]), leading to further developments in [68, 71]. The problem is worked through in detail in [92]. To fully understand the details, the reader will need to study the Cameron-Martin theorem (concerning shifts in the mean of Gaussian measures) and the Feldman-Hájek theorem (concerning equivalence of Gaussian measures); both of these may be found in [14, 28, 67] and are also discussed in [92].
- Section 3.4. The elliptic inverse problem with the uniform prior is studied in [90]. A Gaussian prior is adopted in [25] and a Besov prior in [24].

4 Common Structure

In this section we discuss various common features of the posterior distribution arising from the Bayesian approach to inverse problems. We start, in Sect. 4.1, by studying the continuity properties of the posterior with respect to changes in data, proving a form of well posedness; indeed, we show that the posterior is Lipschitz in the data with respect to the Hellinger metric. In Sect. 4.2 we use similar ideas to study the effect of approximation on the posterior distribution, showing that small changes in the potential Φ lead to small changes in the posterior distribution, again in the Hellinger metric; this work may be used to translate error analysis pertaining to the forward problem into estimates on errors in the posterior distribution. In the final Sect. 4.3, we study an important link between the Bayesian approach to inverse problems and classical regularization techniques for inverse problems;

specifically we link the Bayesian MAP estimator to a Tikhonov-Phillips regularized least squares problem. The first two subsections work with general priors, while the final one is concerned with Gaussians only.

4.1 Well Posedness

In many classical inverse problems, small changes in the data can induce arbitrarily large changes in the solution, and some form of regularization is needed to counteract this ill posedness. We illustrate this effect with the inverse heat equation example. We then proceed to show that the Bayesian approach to inversion has the property that small changes in the data lead to small changes in the posterior distribution. Thus working with probability measures on the solution space, and adopting suitable priors, provides a form of regularization.

Example 6. Consider the heat equation introduced in Sect. 1.2 and both perfect data $y = e^{-A}u$, derived from the forward model with no noise, and noisy data $y' = e^{-A}u + \eta$. Consider the case where $\eta = \epsilon\varphi_j$ with ϵ small and φ_j a normalized eigenfunction of A . Thus $\|\eta\| = \epsilon$. Obviously application of the inverse of e^{-A} to y returns the point u which gave rise to the perfect data. It is natural to apply the inverse of e^{-A} to both y and to y' to understand the effect of the noise. Doing so yields the identity

$$\|e^A y - e^A y'\| = \|e^A(y - y')\| = \|e^A \eta\| = \epsilon \|e^A \varphi_j\| = \epsilon e^{\alpha_j}.$$

Recall Assumption 1 which gives $\alpha_j \asymp j^{2/d}$. Now fix any $a > 0$ and choose j large enough to ensure that $\alpha_j = (a+1)\log(\epsilon^{-1})$. It then follows that $\|y - y'\| = \mathcal{O}(\epsilon)$ while $\|e^A y - e^A y'\| = \mathcal{O}(\epsilon^{-a})$. This is a manifestation of ill posedness. Furthermore, since $a > 0$ is arbitrary, the ill posedness can be made arbitrarily bad by considering $a \rightarrow \infty$. \square

Our aim in this section is to show that this ill-posedness effect does not occur in the Bayesian posterior distribution: small changes in the data y lead to small changes in the measure μ^y . Let X and Y be separable Banach spaces, equipped with the Borel σ -algebra, and μ_0 a measure on X . We will work under assumptions which enable us to make sense of the following measure $\mu^y \ll \mu_0$ defined, for some $\Phi : X \times Y \rightarrow \mathbb{R}$, by

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z(y)} \exp(-\Phi(u; y)), \quad (10.40a)$$

$$Z(y) = \int_X \exp(-\Phi(u; y)) \mu_0(du). \quad (10.40b)$$

We make the following assumptions concerning Φ :

Assumptions 1. Let $X' \subseteq X$ and assume that $\Phi \in C(X' \times Y; \mathbb{R})$. Assume further that there are functions $M_i : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$, $i = 1, 2$, monotonic non-decreasing separately in each argument, and with M_2 strictly positive, such that for all $u \in X'$, $y, y_1, y_2 \in B_Y(0, r)$,

$$\begin{aligned}\Phi(u; y) &\geq -M_1(r, \|u\|_X), \\ |\Phi(u; y_1) - \Phi(u; y_2)| &\leq M_2(r, \|u\|_X) \|y_1 - y_2\|_Y.\end{aligned}\quad \square$$

In order to measure the effect of changes in y on the measure μ^y , we need a metric on measures. We use the Hellinger metric defined in Sect. A.2.4.

Theorem 15. Let Assumptions 1 hold. Assume that $\mu_0(X') = 1$ and that $\mu_0(X' \cap B) > 0$ for some bounded set B in X . Assume additionally that, for every fixed $r > 0$,

$$\exp(M_1(r, \|u\|_X)) \in L_{\mu_0}^1(X; \mathbb{R}).$$

Then, for every $y \in Y$, $Z(y)$ given by (10.40b) is positive and finite and the probability measure μ^y given by (10.40) is well defined.

Proof. The boundedness of $Z(y)$ follows directly from the lower bound on Φ in Assumption 1, together with the assumed integrability condition in the theorem. Since $u \sim \mu_0$ satisfies $u \in X'$ a.s., we have

$$Z(y) = \int_{X'} \exp(-\Phi(u; y)) \mu_0(du).$$

Note that $B' = X' \cap B$ is bounded in X . Define

$$R_1 := \sup_{u \in B'} \|u\|_X < \infty.$$

Since $\Phi : X' \times Y \rightarrow \mathbb{R}$ is continuous, it is finite at every point in $B' \times \{y\}$. Thus, by the continuity of $\Phi(\cdot; \cdot)$ implied by Assumptions 1, we see that

$$\sup_{(u,y) \in B' \times B_Y(0,r)} \Phi(u; y) = R_2 < \infty.$$

Hence

$$Z(y) \geq \int_{B'} \exp(-R_2) \mu_0(du) = \exp(-R_2) \mu_0(B') > 0. \quad (10.41)$$

Since $\mu_0(B')$ is assumed positive and R_2 is finite, we deduce that $Z(y) > 0$. \square

Remarks 6. The following remarks apply to the preceding and following theorem.

- In the preceding theorem, we are not explicitly working in a Bayesian setting: we are showing that, under the stated conditions on Φ , the measure is well defined and normalizable. In Theorem 14, we did not need to check normalizability because μ^y was defined as a regular conditional probability, via Theorem 13, and therefore automatically normalizable.
- The lower bound (10.41) is used repeatedly in what follows, without comment.
- Establishing the integrability conditions for both the preceding and following theorem is often achieved for Gaussian μ_0 by appealing to the Fernique theorem.

Theorem 16. *Let Assumptions 1 hold. Assume that $\mu_0(X') = 1$ and that $\mu_0(X' \cap B) > 0$ for some bounded set B in X . Assume additionally that, for every fixed $r > 0$,*

$$\exp(M_1(r, \|u\|_X)) \left(1 + M_2(r, \|u\|_X)^2\right) \in L_{\mu_0}^1(X; \mathbb{R}).$$

Then there is $C = C(r) > 0$ such that, for all $y, y' \in B_Y(0, r)$

$$d_{\text{Hell}}(\mu^y, \mu^{y'}) \leq C \|y - y'\|_Y.$$

Proof. Throughout this proof, we use C to denote a constant independent of u , but possibly depending on the fixed value of r ; it may change from occurrence to occurrence. We use the fact that, since $M_2(r, \cdot)$ is monotonic non-decreasing and strictly positive on $[0, \infty)$,

$$\exp(M_1(r, \|u\|_X)) M_2(r, \|u\|_X) \leq \exp(M_1(r, \|u\|_X)) \left(1 + M_2(r, \|u\|_X)^2\right), \quad (10.42a)$$

$$\exp(M_1(r, \|u\|_X)) \leq \exp(M_1(r, \|u\|_X)) \left(1 + M_2(r, \|u\|_X)^2\right). \quad (10.42b)$$

Let $Z = Z(y)$ and $Z' = Z(y')$ denote the normalization constants for μ^y and $\mu^{y'}$ so that, by Theorem 15,

$$Z = \int_{X'} \exp(-\Phi(u; y)) \mu_0(du) > 0,$$

$$Z' = \int_{X'} \exp(-\Phi(u; y')) \mu_0(du) > 0.$$

Then, using the local Lipschitz property of the exponential and the assumed Lipschitz continuity of $\Phi(u; \cdot)$, together with (10.42a), we have

$$\begin{aligned} |Z - Z'| &\leq \int_{X'} |\exp(-\Phi(u; y)) - \exp(-\Phi(u; y'))| \mu_0(du) \\ &\leq \int_{X'} \exp(M_1(r, \|u\|_X)) |\Phi(u; y) - \Phi(u; y')| \mu_0(du) \\ &\leq \left(\int_{X'} \exp(M_1(r, \|u\|_X)) M_2(r, \|u\|_X) \mu_0(du) \right) \|y - y'\|_Y \\ &\leq \left(\int_{X'} \exp(M_1(r, \|u\|_X)) (1 + M_2(r, \|u\|_X)^2) \mu_0(du) \right) \|y - y'\|_Y \\ &\leq C \|y - y'\|_Y. \end{aligned}$$

The last line follows because the integrand is in $L^1_{\mu_0}$ by assumption. From the definition of Hellinger distance, we have

$$\left(d_{\text{Hell}}(\mu^y, \mu^{y'}) \right)^2 \leq I_1 + I_2,$$

where

$$\begin{aligned} I_1 &= \frac{1}{Z} \int_{X'} \left(\exp\left(-\frac{1}{2}\Phi(u; y)\right) - \exp\left(-\frac{1}{2}\Phi(u; y')\right) \right)^2 \mu_0(du), \\ I_2 &= |Z^{-\frac{1}{2}} - (Z')^{-\frac{1}{2}}|^2 \int_{X'} \exp(-\Phi(u; y')) \mu_0(du). \end{aligned}$$

Note that, again using similar Lipschitz calculations to those above, using the fact that $Z > 0$ and Assumptions 1,

$$\begin{aligned} I_1 &\leq \frac{1}{4Z} \int_{X'} \exp(M_1(r, \|u\|_X)) |\Phi(u; y) - \Phi(u; y')|^2 \mu_0(du) \\ &\leq \frac{1}{Z} \left(\int_{X'} \exp(M_1(r, \|u\|_X)) M_2(r, \|u\|_X)^2 \mu_0(du) \right) \|y - y'\|_Y^2 \\ &\leq C \|y - y'\|_Y^2. \end{aligned}$$

Also, using Assumptions 1, together with (10.42b),

$$\begin{aligned} \int_{X'} \exp(-\Phi(u; y')) \mu_0(du) &\leq \int_{X'} \exp(M_1(r, \|u\|_X)) \mu_0(du) \\ &< \infty. \end{aligned}$$

Hence

$$I_2 \leq C(Z^{-3} \vee (Z')^{-3})|Z - Z'|^2 \leq C\|y - y'\|_Y^2.$$

The result is complete. \square

Remark 1. The Hellinger metric has the very desirable property that it translates directly into bounds on expectations. For functions f which are in $L_{\mu^y}^2(X; \mathbb{R})$ and $L_{\mu^{y'}}^2(X; \mathbb{R})$, the closeness of the Hellinger metric implies closeness of expectations of f . To be precise, for $y, y' \in B_Y(0, r)$, we have

$$|\mathbb{E}^{\mu^y} f(u) - \mathbb{E}^{\mu^{y'}} f(u)| \leq C d_{\text{Hell}}(\mu^y, \mu^{y'})$$

where constant C depends on r and on the expectations of $|f|^2$ under μ^y and $\mu^{y'}$. It follows that

$$|\mathbb{E}^{\mu^y} f(u) - \mathbb{E}^{\mu^{y'}} f(u)| \leq C \|y - y'\|,$$

for a possibly different constant C which also depends on r and on the expectations of $|f|^2$ under μ^y and $\mu^{y'}$.

4.2 Approximation

In this section we concentrate on continuity properties of the posterior measure with respect to approximation of the potential Φ . The methods used are very similar to those in the previous subsection, and we establish a continuity property of the posterior distribution, in the Hellinger metric, with respect to small changes in the potential Φ .

Because the data y plays no explicit role in this discussion, we drop explicit reference to it. Let X be a Banach space and μ_0 a measure on X . Assume that μ and μ^N are both absolutely continuous with respect to μ_0 and given by

$$\frac{d\mu}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u)), \quad (10.43a)$$

$$Z = \int_X \exp(-\Phi(u)) \mu_0(du) \quad (10.43b)$$

and

$$\frac{d\mu^N}{d\mu_0}(u) = \frac{1}{Z^N} \exp(-\Phi^N(u)), \quad (10.44a)$$

$$Z^N = \int_X \exp(-\Phi^N(u)) \mu_0(du) \quad (10.44b)$$

respectively. The measure μ^N might arise, for example, through an approximation of the forward map G underlying an inverse problem of the form (10.29). It is natural to ask whether closeness of the forward map and its approximation imply closeness of the posterior measures. We now address this question.

Assumptions 2. Let $X' \subseteq X$ and assume that $\Phi \in C(X'; \mathbb{R})$. Assume further that there are functions $M_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, $i = 1, 2$, independent of N and monotonic non-decreasing separately in each argument, and with M_2 strictly positive, such that for all $u \in X'$,

$$\begin{aligned}\Phi(u) &\geq -M_1(\|u\|_X), \\ \Phi^N(u) &\geq -M_1(\|u\|_X), \\ |\Phi(u) - \Phi^N(u)| &\leq M_2(\|u\|_X)\psi(N),\end{aligned}$$

where $\psi(N) \rightarrow 0$ as $N \rightarrow \infty$. \square

The following two theorems are very similar to Theorems 15 and 16, and the proofs are adapted to estimate changes in the posterior caused by changes in the potential Φ , rather than the data y .

Theorem 17. Let Assumptions 2 hold. Assume that $\mu_0(X') = 1$ and that $\mu_0(X' \cap B) > 0$ for some bounded set B in X . Assume additionally that, for every fixed $r > 0$,

$$\exp(M_1(r, \|u\|_X)) \in L_{\mu_0}^1(X; \mathbb{R}).$$

Then Z and Z^N given by (10.43b) and (10.44b) are positive and finite and the probability measures μ and μ^N given by (10.43) and (10.44) are well defined. Furthermore, for sufficiently large N , Z^N given by (10.44b) is bounded below by a positive constant independent of N .

Proof. Finiteness of the normalization constants Z and Z^N follows from the lower bounds on Φ and Φ^N given in Assumptions 2, together with the integrability condition in the theorem. Since $u \sim \mu_0$ satisfies $u \in X'$ a.s., we have

$$Z = \int_{X'} \exp(-\Phi(u)) \mu_0(du).$$

Note that $B' = X' \cap B$ is bounded in X . Thus

$$R_1 := \sup_{u \in B'} \|u\|_X < \infty.$$

Since $\Phi : X' \rightarrow \mathbb{R}$ is continuous, it is finite at every point in B' . Thus, by the properties of $|\Phi(\cdot) - \Phi^N(\cdot)|$ implied by Assumptions 2, we see that

$$\sup_{u \in B'} \Phi(u) = R_2 < \infty.$$

Hence

$$Z \geq \int_{B'} \exp(-R_2) \mu_0(du) = \exp(-R_2) \mu_0(B').$$

Since $\mu_0(B')$ is assumed positive and R_2 is finite, we deduce that $Z > 0$. By Assumption 2, we may choose N large enough so that

$$\sup_{u \in B'} |\Phi(u) - \Phi^N(u)| \leq R_2$$

so that

$$\sup_{u \in B'} \Phi^N(u) \leq 2R_2 < \infty.$$

Hence

$$Z^N \geq \int_{B'} \exp(-2R_2) \mu_0(du) = \exp(-2R_2) \mu_0(B').$$

Since $\mu_0(B')$ is assumed positive and R_2 is finite, we deduce that $Z^N > 0$. Furthermore, the lower bound is independent of N , as required. \square

Theorem 18. *Let Assumptions 2 hold. Assume that $\mu_0(X') = 1$ and that $\mu_0(X' \cap B) > 0$ for some bounded set B in X . Assume additionally that*

$$\exp(M_1(\|u\|_X)) \left(1 + M_2(\|u\|_X)^2\right) \in L^1_{\mu_0}(X; \mathbb{R}).$$

Then there is $C > 0$ such that, for all N sufficiently large,

$$d_{\text{Hell}}(\mu, \mu^N) \leq C \psi(N).$$

Proof. Throughout this proof, we use C to denote a constant independent of u and N ; it may change from occurrence to occurrence. We use the fact that, since $M_2(\cdot)$ is monotonic non-decreasing and since it is strictly positive on $[0, \infty)$,

$$\exp(M_1(\|u\|_X)) M_2(\|u\|_X) \leq \exp(M_1(\|u\|_X)) \left(1 + M_2(\|u\|_X)^2\right), \quad (10.45a)$$

$$\exp(M_1(\|u\|_X)) \leq \exp(M_1(\|u\|_X)) \left(1 + M_2(\|u\|_X)^2\right). \quad (10.45b)$$

Let Z and Z^N denote the normalization constants for μ and μ^N so that for all N sufficiently large, by Theorem 17,

$$Z = \int_{X'} \exp(-\Phi(u)) \mu_0(du) > 0,$$

$$Z^N = \int_{X'} \exp(-\Phi^N(u)) \mu_0(du) > 0,$$

with positive lower bounds independent of N . Then, using the local Lipschitz property of the exponential and the approximation property of $\Phi^N(\cdot)$ from Assumptions 2, together with (10.45a), we have

$$\begin{aligned} |Z - Z^N| &\leq \int_{X'} |\exp(-\Phi(u)) - \exp(-\Phi^N(u))| \mu_0(du) \\ &\leq \int_{X'} \exp(M_1(\|u\|_X)) |\Phi(u) - \Phi^N(u)| \mu_0(du) \\ &\leq \left(\int_{X'} \exp(M_1(\|u\|_X)) M_2(\|u\|_X) \mu_0(du) \right) \psi(N) \\ &\leq \left(\int_{X'} \exp(M_1(\|u\|_X)) (1 + M_2(\|u\|_X)^2) \mu_0(du) \right) \psi(N) \\ &\leq C \psi(N). \end{aligned}$$

The last line follows because the integrand is in $L^1_{\mu_0}$ by assumption. From the definition of Hellinger distance, we have

$$\left(d_{\text{Hell}}(\mu^y, \mu^{y'}) \right)^2 \leq I_1 + I_2,$$

where

$$I_1 = \frac{1}{Z} \int_{X'} \left(\exp\left(-\frac{1}{2}\Phi(u)\right) - \exp\left(-\frac{1}{2}\Phi^N(u)\right) \right)^2 \mu_0(du),$$

$$I_2 = |Z^{-\frac{1}{2}} - (Z^N)^{-\frac{1}{2}}|^2 \int_{X'} \exp(-\Phi^N(u)) \mu_0(du).$$

Note that, again by means of similar Lipschitz calculations to those above, using the fact that $Z, Z^N > 0$ uniformly for N sufficiently large by Theorem 17, and Assumptions 2,

$$\begin{aligned} I_1 &\leq \frac{1}{4Z} \int_{X'} \exp(M_1(\|u\|_X)) |\Phi(u) - \Phi^N(u)|^2 \mu_0(du) \\ &\leq \frac{1}{Z} \left(\int_{X'} \exp(M_1(\|u\|_X)) M_2(\|u\|_X)^2 \mu_0(du) \right) \psi(N)^2 \\ &\leq C \psi(N)^2. \end{aligned}$$

Also, using Assumptions 2, together with (10.45b),

$$\begin{aligned} \int_{X'} \exp(-\Phi^N(u)) \mu_0(du) &\leq \int_{X'} \exp(M_1(\|u\|_X)) \mu_0(du) \\ &< \infty, \end{aligned}$$

and the upper bound is independent of N . Hence

$$I_2 \leq C(Z^{-3} \vee (Z^N)^{-3}) |Z - Z^N|^2 \leq C \psi(N)^2.$$

The result is complete. \square

Remarks 7. The following two remarks are relevant to establishing the conditions of the preceding two theorems and to applying them.

- As mentioned in the previous subsection concerning well posedness, the Fernique theorem can frequently be used to establish integrability conditions, such as those in the two preceding theorems when μ_0 is Gaussian.
- Using the ideas underlying Remark 1, the preceding theorem enables us to translate errors arising from approximation of the forward problem into errors in the Bayesian solution of the inverse problem. Furthermore, the errors in the forward and inverse problems scale the same way with respect to N . For functions f which are in L_μ^2 and $L_{\mu^N}^2$, uniformly with respect to N , the closeness of the Hellinger metric implies closeness of expectations of f :

$$|\mathbb{E}^\mu f(u) - \mathbb{E}^{\mu^N} f(u)| \leq C \psi(N).$$

4.3 MAP Estimators and Tikhonov Regularization

The aim of this section is to connect the probabilistic approach to inverse problems with the classical method of Tikhonov regularization. We consider the setting in which the prior measure μ_0 is a Gaussian measure. We then show that MAP estimators, points of maximal probability, coincide with minimizers of a Tikhonov-Phillips regularized least squares function, with regularization being with respect to the Cameron-Martin norm of the Gaussian prior. The data y plays no explicit role in our developments here, and so we work in the setting of equation (10.43). Recall, however, that in the context of inverse problems, a classical methodology is to simply try and minimize (subject to some regularization) $\Phi(u)$. Indeed for finite data and Gaussian observational noise with Gaussian distribution $N(0, \Gamma)$, we have

$$\Phi(u) = \frac{1}{2} |\Gamma^{-\frac{1}{2}}(y - G(u))|^2.$$

Thus Φ is simply a covariance weighted model-data misfit least squares function.

In this section we show that maximizing probability under μ (in a sense that we will make precise in what follows) is equivalent to minimizing

$$I(u) = \begin{cases} \Phi(u) + \frac{1}{2}\|u\|_E^2 & \text{if } u \in E, \text{ and} \\ +\infty & \text{else.} \end{cases} \quad (10.46)$$

Here $(E, \|\cdot\|_E)$ denotes the Cameron-Martin space associated to the Gaussian prior μ_0 . We view μ_0 as a Gaussian probability measure on a separable Banach space $(X, \|\cdot\|_X)$ so that $\mu_0(X) = 1$. We make the following assumptions about the function Φ :

Assumption 3. *The function $\Phi: X \rightarrow \mathbb{R}$ satisfies the following conditions:*

(i) *For every $\epsilon > 0$, there is an $M = M(\epsilon) \in \mathbb{R}$, such that for all $u \in X$,*

$$\Phi(u) \geq M - \epsilon\|u\|_X^2.$$

(ii) *Φ is locally bounded from above, i.e., for every $r > 0$ there exists $K = K(r) > 0$ such that, for all $u \in X$ with $\|u\|_X < r$, we have*

$$\Phi(u) \leq K.$$

(iii) *Φ is locally Lipschitz continuous, i.e., for every $r > 0$ there exists $L = L(r) > 0$ such that for all $u_1, u_2 \in X$ with $\|u_1\|_X, \|u_2\|_X < r$, we have*

$$|\Phi(u_1) - \Phi(u_2)| \leq L\|u_1 - u_2\|_X.$$

In finite dimensions, for measures which have a continuous density with respect to Lebesgue measure, there is an obvious notion of most likely point(s): simply the point(s) at which the Lebesgue density is maximized. This way of thinking does not translate into the infinite-dimensional context, but there is a way of restating it which does. Fix a small radius $\delta > 0$ and identify centres of balls of radius δ which have maximal probability. Letting $\delta \rightarrow 0$ then recovers the preceding definition, when there is a continuous Lebesgue density. We adopt this small ball approach in the infinite-dimensional setting.

For $z \in E$, let $B^\delta(z) \subset X$ be the open ball centred at $z \in X$ with radius δ in X . Let

$$J^\delta(z) = \mu(B^\delta(z))$$

be the mass of the ball $B^\delta(z)$ under the measure μ . Similarly we define

$$J_0^\delta(z) = \mu_0(B^\delta(z))$$

the mass of the ball $B^\delta(z)$ under the Gaussian prior. Recall that all balls in a separable Banach space have positive Gaussian measure, by Theorem 33; it thus follows that $J_0^\delta(z)$ is finite and positive for any $z \in E$. By Assumptions 3(i) and (ii) together with the Fernique Theorem 10, the same is true for $J^\delta(z)$. Our first theorem encapsulates the idea that probability is maximized where I is minimized. To see this, fix any point z_2 in the Cameron-Martin space E and notice that the probability of the small ball at z_1 is maximized, asymptotically as the radius of the ball tends to zero, at minimizers of I .

Theorem 19. *Let Assumptions 3 hold and assume that $\mu_0(X) = 1$. Then the function I defined by (10.46) satisfies, for any $z_1, z_2 \in E$,*

$$\lim_{\delta \rightarrow 0} \frac{J^\delta(z_1)}{J^\delta(z_2)} = \exp(I(z_2) - I(z_1)).$$

Proof. Since $J^\delta(z)$ is finite and positive for any $z \in E$, the ratio of interest is finite and positive. The key estimate in the proof is given in Theorem 35:

$$\lim_{\delta \rightarrow 0} \frac{J_0^\delta(z_1)}{J_0^\delta(z_2)} = \exp\left(\frac{1}{2}\|z_2\|_E^2 - \frac{1}{2}\|z_1\|_E^2\right). \quad (10.47)$$

This estimate transfers questions about probability, naturally asked on the space X of full measure under μ_0 , into statements concerning the Cameron-Martin norm of μ_0 ; note that under this norm, a random variable distributed as μ_0 is almost surely infinite so the result is nontrivial.

We have

$$\begin{aligned} \frac{J^\delta(z_1)}{J^\delta(z_2)} &= \frac{\int_{B^\delta(z_1)} \exp(-\Phi(u)) \mu_0(du)}{\int_{B^\delta(z_2)} \exp(-\Phi(v)) \mu_0(dv)} \\ &= \frac{\int_{B^\delta(z_1)} \exp(-\Phi(u) + \Phi(z_1)) \exp(-\Phi(z_1)) \mu_0(du)}{\int_{B^\delta(z_2)} \exp(-\Phi(v) + \Phi(z_2)) \exp(-\Phi(z_2)) \mu_0(dv)}. \end{aligned}$$

By Assumption 3 (iii), there is $L = L(r)$ such that, for all $u, v \in X$ with $\max\{\|u\|_X, \|v\|_X\} < r$,

$$-L\|u - v\|_X \leq \Phi(u) - \Phi(v) \leq L\|u - v\|_X.$$

If we define $L_1 = L(\|z_1\|_X + \delta)$ and $L_2 = L(\|z_2\|_X + \delta)$, then we have

$$\begin{aligned} \frac{J^\delta(z_1)}{J^\delta(z_2)} &\leq e^{\delta(L_1 + L_2)} \frac{\int_{B^\delta(z_1)} \exp(-\Phi(z_1)) \mu_0(du)}{\int_{B^\delta(z_2)} \exp(-\Phi(z_2)) \mu_0(dv)} \\ &= e^{\delta(L_1 + L_2)} e^{-\Phi(z_1) + \Phi(z_2)} \frac{\int_{B^\delta(z_1)} \mu_0(du)}{\int_{B^\delta(z_2)} \mu_0(dv)}. \end{aligned}$$

Now, by (10.47), we have

$$\frac{J^\delta(z_1)}{J^\delta(z_2)} \leq r_1(\delta) e^{\delta(L_2 + L_1)} e^{-I(z_1) + I(z_2)}$$

with $r_1(\delta) \rightarrow 1$ as $\delta \rightarrow 0$. Thus

$$\limsup_{\delta \rightarrow 0} \frac{J^\delta(z_1)}{J^\delta(z_2)} \leq e^{-I(z_1) + I(z_2)}. \quad (10.48)$$

Similarly we obtain

$$\frac{J^\delta(z_1)}{J^\delta(z_2)} \geq \frac{1}{r_2(\delta)} e^{-\delta(L_2 + L_1)} e^{-I(z_1) + I(z_2)}$$

with $r_2(\delta) \rightarrow 1$ as $\delta \rightarrow 0$ and deduce that

$$\liminf_{\delta \rightarrow 0} \frac{J^\delta(z_1)}{J^\delta(z_2)} \geq e^{-I(z_1) + I(z_2)} \quad (10.49)$$

Inequalities (10.48) and (10.49) give the desired result.

We have thus linked the Bayesian approach to inverse problems with a classical regularization technique. We conclude the subsection by showing that, under the prevailing Assumption 3, the minimization problem for I is well defined. We first recall a basic definition and lemma from the calculus of variations.

Definition 1. The function $I : E \rightarrow \mathbb{R}$ is *weakly lower semicontinuous* if

$$\liminf_{n \rightarrow \infty} I(u_n) \geq I(u)$$

whenever $u_n \rightharpoonup u$ in E . The function $I : E \rightarrow \mathbb{R}$ is *weakly continuous* if

$$\lim_{n \rightarrow \infty} I(u_n) = I(u)$$

whenever $u_n \rightharpoonup u$ in E . \square

Clearly weak continuity implies weak lower semicontinuity.

Lemma 6. If $(E, \langle \cdot, \cdot \rangle_E)$ is a Hilbert space with induced norm $\|\cdot\|_E$, then the quadratic form $J(u) := \frac{1}{2}\|u\|_E^2$ is weakly lower semicontinuous.

Proof. The result follows from the fact that

$$\begin{aligned} J(u_n) - J(u) &= \frac{1}{2} \|u_n\|_E^2 - \frac{1}{2} \|u\|_E^2 \\ &= \frac{1}{2} \langle u_n - u, u_n + u \rangle_E \\ &= \frac{1}{2} \langle u_n - u, 2u \rangle_E + \frac{1}{2} \|u_n - u\|_E^2 \\ &\geq \frac{1}{2} \langle u_n - u, 2u \rangle_E. \end{aligned}$$

But the right-hand side tends to zero since $u_n \rightarrow u$ in E . Hence, the result follows.

Theorem 20. Suppose that Assumption 3 hold and let E be a Hilbert space compactly embedded in X . Then there exists $\bar{u} \in E$ such that

$$I(\bar{u}) = \bar{I} := \inf\{I(u) : u \in E\}.$$

Furthermore, if $\{u_n\}$ is a minimizing sequence satisfying $I(u_n) \rightarrow I(\bar{u})$, then there is a subsequence $\{u_{n'}\}$ that converges strongly to \bar{u} in E .

Proof. Compactness of E in X implies that, for some universal constant C ,

$$\|u\|_X^2 \leq C \|u\|_E^2.$$

Hence, by Assumption 3(i), it follows that, for any $\epsilon > 0$, there is $M(\epsilon) \in \mathbb{R}$ such that

$$\left(\frac{1}{2} - C\epsilon\right) \|u\|_E^2 + M(\epsilon) \leq I(u).$$

By choosing ϵ sufficiently small, we deduce that there is $M \in \mathbb{R}$ such that, for all $u \in E$,

$$\frac{1}{4} \|u\|_E^2 + M \leq I(u). \quad (10.50)$$

Let u_n be an infimizing sequence satisfying $I(u_n) \rightarrow \bar{I}$ as $n \rightarrow \infty$. For any $\delta > 0$ there is $N = N_1(\delta)$:

$$\bar{I} \leq I(u_n) \leq \bar{I} + \delta, \quad \forall n \geq N_1. \quad (10.51)$$

Using (10.50) we deduce that the sequence $\{u_n\}$ is bounded in E and, since E is a Hilbert space, there exists $\bar{u} \in E$ such that $u_n \rightharpoonup \bar{u}$ in E . By the compact embedding

of E in X we deduce that $u_n \rightarrow \bar{u}$, strongly in X . By the Lipschitz continuity of Φ in X (Assumption 3(iii)), we deduce that $\Phi(u_n) \rightarrow \Phi(\bar{u})$. Thus, Φ is weakly continuous on E . The functional $J(u) := \frac{1}{2}\|u\|_E^2$ is weakly lower semicontinuous on E by Lemma 6. Hence $I(u) = J(u) + \Phi(u)$ is weakly lower semicontinuous on E . Using this fact in (10.51), it follows that, for any $\delta > 0$,

$$\bar{I} \leq I(\bar{u}) \leq \bar{I} + \delta.$$

Since δ is arbitrary, the first result follows.

By passing to a further subsequence, and for $n, \ell \geq N_2(\delta)$,

$$\begin{aligned} \frac{1}{4}\|u_n - u_\ell\|_E^2 &= \frac{1}{2}\|u_n\|_E^2 + \frac{1}{2}\|u_\ell\|_E^2 - \frac{1}{4}\|u_n + u_\ell\|_E^2 \\ &= I(u_n) + I(u_\ell) - 2I\left(\frac{1}{2}(u_n + u_\ell)\right) - \Phi(u_n) - \Phi(u_\ell) + 2\Phi\left(\frac{1}{2}(u_n + u_\ell)\right) \\ &\leq 2(\bar{I} + \delta) - 2\bar{I} - \Phi(u_n) - \Phi(u_\ell) + 2\Phi\left(\frac{1}{2}(u_n + u_\ell)\right) \\ &\leq 2\delta - \Phi(u_n) - \Phi(u_\ell) + 2\Phi\left(\frac{1}{2}(u_n + u_\ell)\right). \end{aligned}$$

But u_n, u_ℓ and $\frac{1}{2}(u_n + u_\ell)$ all converge strongly to \bar{u} in X . Thus, by continuity of Φ , we deduce that for all $n, \ell \geq N_3(\delta)$,

$$\frac{1}{4}\|u_n - u_\ell\|_E^2 \leq 3\delta.$$

Hence the sequence is Cauchy in E and converges strongly and the proof is complete.

Corollary 1. *Suppose that Assumptions 3 hold and the Gaussian measure μ_0 with Cameron-Martin space E satisfies $\mu_0(X) = 1$. Then there exists $\bar{u} \in E$ such that*

$$I(\bar{u}) = \bar{I} := \inf\{I(u) : u \in E\}.$$

Furthermore, if $\{u_n\}$ is a minimizing sequence satisfying $I(u_n) \rightarrow I(\bar{u})$, then there is a subsequence $\{u_{n'}\}$ that converges strongly to \bar{u} in E .

Proof. By Theorem 34, E is compactly embedded in X . Hence the result follows by Theorem 20.

4.4 Bibliographic Notes

- Section 4.1. The well-posedness theory described here was introduced in the papers [17] and [92]. Relationships between the Hellinger distance on probability measures and the total variation distance and Kullback-Leibler divergence may be found in [38] and Pollard (Distances and affinities between measures. Unpublished manuscript, <http://www.stat.yale.edu/~pollard/Books/Asymptopia/Metrics.pdf>), as well as in [92].
- Section 4.2. Generalization of the well-posedness theory to study the effect of numerical approximation of the forward model on the inverse problem may be found in [20]. The relationship between expectations and Hellinger distance, as used in Remark 7, is demonstrated in [92].
- Section 4.3. The connection between Tikhonov-Phillips regularization and MAP estimators is widely appreciated in computational Bayesian inverse problems; see [53]. Making the connection rigorous in the separable Banach space setting is the subject of the paper [30]; further references to the historical development of the subject may be found therein. Related to Lemma 6, see also [23, Chapter 3].

5 Measure Preserving Dynamics

The aim of this section is to study Markov processes, in continuous time, and Markov chains, in discrete time, which preserve the measure μ given by (10.43). The overall setting is described in Sect. 5.1 and introduces the role of detailed balance and reversibility in constructing measure-preserving Markov chains/processes. Section 5.2 concerns Markov chain Monte Carlo (MCMC) methods; these are Markov chains which are invariant with respect to μ . Metropolis-Hastings methods are introduced and the role of detailed balance in their construction is explained. The benefit of conceiving MCMC methods which are defined on the infinite-dimensional space is emphasized. In particular, the idea of using proposals which preserve the prior, more specifically which are prior reversible, is introduced as an example. In Sect. 5.3 we show how sequential Monte Carlo (SMC) methods can be used to construct approximate samples from the measure μ given by (10.43). Again our perspective is to construct algorithms which are probably well defined on the infinite-dimensional space, and in fact we find an upper bound for the approximation error of the SMC method which proves its convergence on an infinite-dimensional space. The MCMC methods from the previous section play an important role in the construction of these SMC methods. Sections 5.4–5.6 concern continuous time μ -reversible processes. In particular they concern derivation and study of a Langevin equation which is invariant with respect to the measure μ . (Note that this is called the overdamped Langevin equation for a physicist and the plain Langevin equation for a statistician.) In continuous time we work entirely in the case of Gaussian prior measure $\mu_0 = N(0, \mathcal{C})$ on Hilbert space \mathcal{H} with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively; however, in discrete time our analysis is more general, applying on a separable Banach space $(X, \| \cdot \|)$ and for quite general prior measure.

5.1 General Setting

This section is devoted to Banach space valued Markov chains or processes which are invariant with respect to the posterior measure μ^y constructed in Sect. 3.2. Within this section, the data y arising in the inverse problems plays no explicit role; indeed the theory applies to a wide range of measures μ on separable Banach space X . Thus the discussion in this chapter includes, but is not limited to, Bayesian inverse problems. All of the Markov chains we construct will exploit structure in a reference measure μ_0 with respect to which the measure μ is absolutely continuous; thus μ has a density with respect to μ_0 . In continuous time we will explicitly use the Gaussianity of μ_0 , but in discrete time we will be more general.

Let μ_0 be a reference measure on the separable Banach space X equipped with the Borel σ -algebra $\mathcal{B}(X)$. We assume that $\mu \ll \mu_0$ is given by

$$\frac{d\mu}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi(u)), \quad (10.52a)$$

$$Z = \int_X \exp(-\Phi(u)) \mu_0(du), \quad (10.52b)$$

where $Z \in (0, \infty)$. In the following we let $P(u, dv)$ denote a Markov transition kernel so that $P(u, \cdot)$ is a probability measure on $(X, \mathcal{B}(X))$ for each $u \in X$. Our interest is in probability kernels which preserve μ .

Definition 2. The Markov chain with transition kernel P is *invariant* with respect to μ if

$$\int_X \mu(du) P(u, \cdot) = \mu(\cdot)$$

as measures on $(X, \mathcal{B}(X))$. The Markov kernel is said to satisfy *detailed balance* with respect to μ if

$$\mu(du) P(u, dv) = \mu(dv) P(v, du)$$

as measures on $(X \times X, \mathcal{B}(X) \otimes \mathcal{B}(X))$. The resulting Markov chain is then said to be *reversible* with respect to μ . \square

It is straightforward to see, by integrating the detailed balance condition with respect to u and using the fact that $P(v, du)$ is a Markov kernel, the following:

Lemma 7. A Markov chain which is reversible with respect to μ is also invariant with respect to μ .

Reversible Markov chains and processes arise naturally in many physical systems which are in statistical equilibrium. They are also important, however, as a means of

constructing Markov chains which are invariant with respect to a given probability measure. We demonstrate this in Sect. 5.2 where we consider the Metropolis-Hastings variant of MCMC methods. Then, in Sects. 5.4, 5.5 and 5.6, we move to continuous time Markov processes. In particular we show that the equation

$$\frac{du}{dt} = -u - \mathcal{C}D\Phi(u) + \sqrt{2}\frac{dW}{dt}, \quad u(0) = u_0, \quad (10.53)$$

preserves the measure μ , where W is a \mathcal{C} -Wiener process, defined below in Sect. A.4. Precisely we show that if $u_0 \sim \mu$, independently of the driving Wiener process, then $\mathbb{E}\varphi(u(t)) = \mathbb{E}\varphi(u_0)$ for all $t > 0$ for continuous bounded φ defined on an appropriately chosen subspaces, under boundedness conditions on Φ and its derivatives.

Example 7. Consider the (measurable) Hilbert space $(\mathcal{H}, \mathbf{B}(\mathcal{H}))$ equipped, as usual, with the Borel σ -algebra. Let μ denote the Gaussian measure $N(0, \mathcal{C})$ on \mathcal{H} and, for fixed u , let $P(u, dv)$ denote the Gaussian measure $N((1 - \beta^2)^{\frac{1}{2}}u, \beta^2\mathcal{C})$, also viewed as a probability measure on \mathcal{H} . Thus $v \sim P(u, dv)$ can be expressed as $v = (1 - \beta^2)^{\frac{1}{2}}u + \beta\xi$ where $\xi \sim N(0, \mathcal{C})$ is independent of u . We show that P is reversible, and hence invariant, with respect to μ . To see this we note that $\mu(du)P(u, dv)$ is a centred Gaussian measure on $\mathcal{H} \times \mathcal{H}$, equipped with the σ -algebra $\mathbf{B}(\mathcal{H}) \otimes \mathbf{B}(\mathcal{H})$. The covariance of the jointly varying random variable is characterized by the identities

$$\mathbb{E}u \otimes u = \mathcal{C}, \quad \mathbb{E}v \otimes v = \mathcal{C}, \quad \mathbb{E}u \otimes v = (1 - \beta^2)^{\frac{1}{2}}\mathcal{C}. \quad (10.54)$$

Indeed, letting $v(du, dv) := \mu(du)P(u, dv)$, and with $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, the inner product and norm on \mathcal{H} , respectively, we can write, using (10.115),

$$\begin{aligned} \hat{v}(d\xi, d\eta) &= \int_{\mathcal{H} \times \mathcal{H}} e^{i\langle u, \xi \rangle + i\langle v, \eta \rangle} \mu(du)P(u, dv) \\ &= \int_{\mathcal{H}} e^{i\langle u, \xi \rangle} \int_{\mathcal{H}} e^{i\langle v, \eta \rangle} P(u, dv) \mu(du) \\ &= \int_{\mathcal{H}} e^{i\langle u, \xi \rangle} e^{i\sqrt{1-\beta^2}\langle u, \eta \rangle - \frac{1}{2}\|\beta\mathcal{C}^{\frac{1}{2}}\eta\|^2} \mu(du) \\ &= e^{-\frac{\beta^2}{2}\|\mathcal{C}^{\frac{1}{2}}\eta\|^2} \int_{\mathcal{H}} e^{i\langle u, \sqrt{1-\beta^2}\eta + \xi \rangle} \mu(du) \\ &= e^{-\frac{\beta^2}{2}\|\mathcal{C}^{\frac{1}{2}}\eta\|^2} e^{-\frac{1}{2}\|\mathcal{C}^{\frac{1}{2}}(\sqrt{1-\beta^2}\eta + \xi)\|^2} \\ &= \exp\left(-\frac{1}{2}\|\mathcal{C}^{\frac{1}{2}}\eta\|^2 - \frac{1}{2}\|\mathcal{C}^{\frac{1}{2}}\xi\|^2 - (1 - \beta^2)^{\frac{1}{2}}\langle \mathcal{C}^{\frac{1}{2}}\xi, \mathcal{C}^{\frac{1}{2}}\eta \rangle\right). \end{aligned}$$

Hence, by Lemma 19 and equation (10.115), $\mu(du)P(u, dv)$ is a centred Gaussian measure with the covariance operator given by (10.54). Since the expression in the last line of the above equation is symmetric in ξ and η , $\mu(dv)P(v, du)$ is a centred Gaussian measure with the same covariance as $\mu(du)P(u, dv)$ and so the reversibility is proved. \square

Example 8. Consider the equation

$$\frac{du}{dt} = -u + \sqrt{2} \frac{dW}{dt}, \quad u(0) = u_0, \quad (10.55)$$

where W is a \mathcal{C} -Wiener process (defined in Sect. A.4 below). Then

$$u(t) = e^{-t}u_0 + \sqrt{2} \int_0^t e^{-(t-s)} dW(s).$$

Use of the Itô isometry demonstrates that $u(t)$ is distributed according to the Gaussian $N(e^{-t}u_0, (1-e^{-2t})\mathcal{C})$. Setting $\beta^2 = 1 - e^{-2t}$ and employing the previous example shows that the Markov process is reversible since, for every $t > 0$, the transition kernel of the process is reversible. \square

5.2 Metropolis-Hastings Methods

In this section we study Metropolis-Hastings methods designed to sample from the probability measure μ given by (10.52). The perspective that we have described on inverse problems, specifically the formulation of Bayesian inversion on function space, leads to new sampling methods which are specifically tailored to the high-dimensional problems which arise from discretization of the infinite-dimensional setting. In particular it leads naturally to the philosophy that it is advantageous to design algorithms which, in principle, make sense in infinite dimensions; it is these methods which will perform well under refinement of finite-dimensional approximations. Most Metropolis-Hastings methods which are defined in finite dimensions will not make sense in the infinite-dimensional limit. This is because the acceptance probability for Metropolis-Hastings methods is defined as the Radon-Nikodym derivative between two measures describing the behavior of the Markov chain in stationarity. Since measures in infinite dimensions have a tendency to be mutually singular, only carefully designed methods will have interpretations in infinite dimensions. To simplify the presentation, we work with the following assumptions throughout:

Assumptions 3. *The function $\Phi : X \rightarrow \mathbb{R}$ is bounded on bounded subsets of X .*

\square

We now consider the following prototype Metropolis-Hastings method which accepts and rejects proposals from a Markov kernel Q to produce a Markov chain with kernel P which is reversible with respect to μ .

Algorithm 1

Given $a : X \times X \rightarrow [0, 1]$ generate $\{u^{(k)}\}_{k \geq 0}$ as follows:

- 1 Set $k = 0$ and pick $u^{(0)} \in X$.
 - 2 Propose $v^{(k)} \sim Q(u^{(k)}, dv)$.
 - 3 Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$, independently of $(u^{(k)}, v^{(k)})$.
 - 4 Set $u^{(k+1)} = u^{(k)}$ otherwise.
 - 5 $k \rightarrow k + 1$ and return to 2. □
-

Given a proposal kernel Q , a key question in the design of MCMC methods is the question of how to choose $a(u, v)$ to ensure that $P(u, dv)$ satisfies detailed balance with respect to μ . If the resulting Markov chain is ergodic, this then yields an algorithm which, asymptotically, samples from μ and can be used to estimate expectations against μ .

To determine conditions on a which are necessary and sufficient for detailed balance, we first note that the Markov kernel which arises from accepting/rejecting proposals from Q is given by

$$P(u, dv) = Q(u, dv)a(u, v) + \delta_u(dv) \int_X (1 - a(u, w))Q(u, dw). \quad (10.56)$$

Notice that

$$\int_X P(u, dv) = 1$$

as required. Substituting the expression for P into the detailed balance condition from Definition 2, we obtain

$$\begin{aligned} & \mu(du)Q(u, dv)a(u, v) + \mu(du)\delta_u(dv) \int_X (1 - a(u, w))Q(u, dw) \\ &= \\ & \mu(dv)Q(v, du)a(v, u) + \mu(dv)\delta_v(du) \int_X (1 - a(v, w))Q(v, dw). \end{aligned}$$

We now note that the measure $\mu(du)\delta_u(dv)$ is in fact symmetric in the pair (u, v) and that $u = v$ almost surely under it. As a consequence the identity reduces to

$$\mu(du)Q(u, dv)a(u, v) = \mu(dv)Q(v, du)a(v, u). \quad (10.57)$$

Our aim now is to identify choices of a which ensure that (10.57) is satisfied. This will then ensure that the prototype algorithm does indeed lead to a Markov chain for which μ is invariant. To this end we define the measures

$$\nu(du, dv) = \mu(du)Q(u, dv)$$

and

$$\nu^T(du, dv) = \mu(dv)Q(v, du)$$

on $(X \times X, \mathcal{B}(X) \otimes \mathcal{B}(X))$. The following theorem determines a necessary and sufficient condition for the choice of a to make the algorithm μ reversible and identifies the canonical Metropolis-Hastings choice.

Theorem 21. *Assume that ν and ν^T are equivalent as measures on $X \times X$, equipped with the σ -algebra $\mathcal{B}(X) \otimes \mathcal{B}(X)$, and that $\nu(du, dv) = r(u, v)\nu^T(du, dv)$. Then the probability kernel (10.56) satisfies detailed balance if and only if*

$$r(u, v)a(u, v) = a(v, u), \quad \nu\text{-a.s.} \quad (10.58)$$

In particular the choice $\alpha_{\text{mh}}(u, v) = \min\{1, r(v, u)\}$ will imply detailed balance.

Proof. Since ν and ν^T are equivalent (10.57) holds if and only if

$$\frac{d\nu}{d\nu^T}(u, v)a(u, v) = a(v, u).$$

This is precisely (10.58). Now note that $\nu(du, dv) = r(u, v)\nu^T(du, dv)$ and $\nu^T(du, dv) = r(v, u)\nu(du, dv)$ since ν and ν^T are equivalent. Thus $r(u, v)r(v, u) = 1$. It follows that

$$\begin{aligned} r(u, v)\alpha_{\text{mh}}(u, v) &= \min\{r(u, v), r(u, v)r(v, u)\} \\ &= \min\{r(u, v), 1\} \\ &= \alpha_{\text{mh}}(v, u) \end{aligned}$$

as required.

A good example of the resulting methodology arises in the case where $Q(u, dv)$ is reversible with respect to μ_0 :

Theorem 22. *Let Assumption 3 hold. Consider Algorithm 5 applied to (10.52) in the case where the proposal kernel Q is reversible with respect to the prior μ_0 . Then the resulting Markov kernel P given by (10.56) is reversible with respect to μ if $a(u, v) = \min\{1, \exp(\Phi(u) - \Phi(v))\}$.*

Proof. Prior reversibility implies that

$$\mu_0(d u) Q(u, d v) = \mu_0(d v) Q(v, d u).$$

Multiplying both sides by $\exp(-\Phi(u))$ gives

$$\mu(d u) Q(u, d v) = \exp(-\Phi(u)) \mu_0(d v) Q(v, d u)$$

and then multiplication by $\exp(-\Phi(v))$ gives

$$\exp(-\Phi(v)) \mu(d u) Q(u, d v) = \exp(-\Phi(u)) \mu(d v) Q(v, d u).$$

This is the statement that

$$\exp(-\Phi(v)) v(d u, d v) = \exp(-\Phi(u)) v^\top(d u, d v).$$

Since Φ is bounded on bounded sets by Assumption 3, we deduce that

$$\frac{d v}{d v^\top}(u, v) = r(u, v) = \exp(\Phi(v) - \Phi(u)).$$

Theorem 21 gives the desired result.

We provide two examples of prior reversible proposals, the first applying in the general Banach space setting and the second when the prior is a Gaussian measure.

Algorithm 2 Independence Sampler

The independence sampler arises when $Q(u, d v) = \mu_0(d v)$ so that proposals are independent draws from the prior. Clearly prior reversibility is satisfied. The following algorithm results. Define

$$a(u, v) = \min\{1, \exp(\Phi(u) - \Phi(v))\}$$

and generate $\{u^{(k)}\}_{k \geq 0}$ as follows:

1. Set $k = 0$ and pick $u^{(0)} \in X$.
2. Propose $v^{(k)} \sim \mu_0$ independently of $u^{(k)}$.
3. Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$, independently of $(u^{(k)}, v^{(k)})$.
4. Set $u^{(k+1)} = u^{(k)}$ otherwise.
5. $k \rightarrow k + 1$ and return to 2.

□

The preceding algorithm works well when the likelihood is not *too* informative; however, when the information in the likelihood is substantial, and $\Phi(\cdot)$ varies significantly depending on where it is evaluated, the independence sampler will

not work well. In such a situation, it is typically the case that *local* proposals are needed, with a parameter controlling the degree of locality; this parameter can then be optimized by choosing it as large as possible, consistent with achieving a reasonable acceptance probability. The following algorithm is an example of this concept, with parameter β playing the role of the locality parameter. The algorithm may be viewed as the natural generalization of the random walk Metropolis method, for targets defined by density with respect to Lebesgue measure, to the situation where the targets are defined by density with respect to Gaussian measure. The name pCN is used because of the original derivation of the algorithm via a Crank-Nicolson discretization of the Hilbert space valued SDE (10.55).

Algorithm 3 pCN Method

Assume that X is a Hilbert space $(\mathcal{H}, \mathcal{B}(\mathcal{H}))$ and that $\mu_0 = N(0, \mathcal{C})$ is a Gaussian prior on \mathcal{H} .

Now define $Q(u, dv)$ to be the Gaussian measure $N((1 - \beta^2)^{\frac{1}{2}}u, \beta^2\mathcal{C})$, also on \mathcal{H} . Example 7 shows that Q is μ_0 reversible. The following algorithm results.

Define

$$a(u, v) = \min\{1, \exp(\Phi(u) - \Phi(v))\}$$

and generate $\{u^{(k)}\}_{k \geq 0}$ as follows:

1. Set $k = 0$ and pick $u^{(0)} \in X$.
2. Propose $v^{(k)} = \sqrt{(1 - \beta^2)}u^{(k)} + \beta\xi^{(k)}$, $\xi^{(k)} \sim N(0, \mathcal{C})$.
3. Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$, independently of $(u^{(k)}, \xi^{(k)})$.
4. Set $u^{(k+1)} = u^{(k)}$ otherwise.
5. $k \rightarrow k + 1$ and return to 2.

□

Example 9. Example 8 shows that using the proposal from Example 7 within a Metropolis-Hastings context may be viewed as using a proposal based on the μ -measure-preserving equation (10.53), but with the $D\Phi$ term dropped. The accept-reject mechanism of Algorithm 3, which is based on differences of Φ , then compensates for the missing $D\Phi$ term.

5.3 Sequential Monte Carlo Methods

In this section we introduce sequential Monte Carlo methods and show how these may be viewed as a generic tool for sampling the posterior distribution arising in Bayesian inverse problems. These methods have their origin in filtering of dynamical systems but, as we will demonstrate, have the potential as algorithms for probing a very wide class of probability measures. The key idea is to introduce

a sequence of measures which evolve the prior distribution into the posterior distribution. Particle filtering methods are then applied to this sequence of measures in order to evolve a set of particles that are prior distributed into a set of particles that are approximately posterior distributed. From a practical perspective, a key step in the construction of these methods is the use of MCMC methods which preserve the measure of interest and other measures closely related to it; furthermore, our interest is in designing SMC methods which, in principle, are well defined on the infinite-dimensional space; for these two reasons, the MCMC methods from the previous subsection play a central role in what follows.

Given integer J , let $h = J^{-1}$, and for nonnegative integer $j \leq J$, define the sequence of measures $\mu_j \ll \mu_0$ by

$$\frac{d\mu_j}{d\mu_0}(u) = \frac{1}{Z_j} \exp(-jh\Phi(u)), \quad (10.59a)$$

$$Z_j = \int_X \exp(-jh\Phi(u)) \mu_0(du). \quad (10.59b)$$

Then $\mu_J = \mu$ given by (10.52); thus, our interest is in approximating μ_J , and we will achieve this by approximating the sequence of measures $\{\mu_j\}_{j=0}^J$, using information about μ_j to inform approximation of μ_{j+1} . To simplify the analysis, we assume that Φ is bounded above and below on X so that there is $\phi^\pm \in \mathbb{R}$ such that

$$\phi^- \leq \Phi(u) \leq \phi^+ \quad \forall u \in X. \quad (10.60)$$

Without loss of generality, we assume that $\phi^- \leq 0$ and that $\phi^+ \geq 0$, which may be achieved by normalization. Note that then the family of measures $\{\mu_j\}_{j=0}^J$ are mutually absolutely continuous and, furthermore,

$$\frac{d\mu_{j+1}}{d\mu_j}(u) = \frac{Z_j}{Z_{j+1}} \exp(-h\Phi(u)). \quad (10.61)$$

An important idea here is that, while μ_0 and μ may be quite far apart as measures, the pair of measures μ_j, μ_{j+1} can be quite close, for sufficiently small h . This fact can be used to incrementally evolve samples from μ_0 into approximate samples of μ_J .

Let L denote the operator on probability measures which corresponds to application of Bayes' theorem with likelihood proportional to $\exp(-h\Phi(u))$, and let P_j denote any Markov kernel which preserves the measure μ_j ; such kernels arise, for example, from the MCMC methods of the previous subsection. These considerations imply that

$$\mu_{j+1} = \mathsf{L}P_j\mu_j. \quad (10.62)$$

Sequential Monte Carlo methods proceed by approximating the sequence $\{\mu_j\}$ by a set of Dirac measures, as we now describe. It is useful to break up the iteration (10.62) and write it as

$$\hat{\mu}_{j+1} = P_j \mu_j, \quad (10.63a)$$

$$\mu_{j+1} = \mathcal{L}\hat{\mu}_{j+1}. \quad (10.63b)$$

We approximate each of the two steps in (10.63) separately. To this end it helps to note that, since P_j preserves μ_j ,

$$\frac{d\mu_{j+1}}{d\hat{\mu}_{j+1}}(u) = \frac{Z_j}{Z_{j+1}} \exp(-h\Phi(u)). \quad (10.64)$$

To define the method, we write an N -particle Dirac measure approximation of the form

$$\mu_j \approx \mu_j^N := \sum_{n=1}^N w_j^{(n)} \delta(v_j - v_j^{(n)}). \quad (10.65)$$

The approximate distribution is completely defined by particle positions $v_j^{(n)}$ and weights $w_j^{(n)}$, respectively. Thus the objective of the method is to find an update rule for $\{v_j^{(n)}, w_j^{(n)}\}_{n=1}^N \mapsto \{v_{j+1}^{(n)}, w_{j+1}^{(n)}\}_{n=1}^N$. The weights must sum to one. To do this we proceed as follows. First each particle $v_j^{(n)}$ is updated by proposing a new candidate particle $\hat{v}_{j+1}^{(n)}$ according to the Markov kernel P_j ; this corresponds to (10.63a) and creates an approximation to $\hat{\mu}_{j+1}$. (See the last two parts of Remark 8 for a discussion on the role of P_j in the algorithm.) We can think of this approximation as a prior distribution for application of Bayes' rule in the form (10.63b), or equivalently (10.64). Secondly, each new particle is re-weighted according to the desired distribution μ_{j+1} given by (10.64). The required calculations are very straightforward because of the assumed form of the measures as sums of Dirac's, as we now explain.

The first step of the algorithm has made the approximation

$$\hat{\mu}_{j+1} \approx \hat{\mu}_{j+1}^N = \sum_{n=1}^N w_{j+1}^{(n)} \delta(v_{j+1} - \hat{v}_{j+1}^{(n)}). \quad (10.66)$$

We now apply Bayes' formula in the form (10.64). Using an approximation proportional to (10.66) for $\hat{\mu}_{j+1}$, we obtain

$$\mu_{j+1} \approx \mu_{j+1}^N := \sum_{n=1}^N w_{j+1}^{(n)} \delta(v_{j+1} - \hat{v}_{j+1}^{(n)}). \quad (10.67)$$

where

$$\hat{w}_{j+1}^{(n)} = \exp(-h\Phi(\hat{v}_{j+1}^{(n)}))w_j^{(n)} \quad (10.68)$$

and normalization requires

$$w_{j+1}^{(n)} = \hat{w}_{j+1}^{(n)} / \left(\sum_{n=1}^N \hat{w}_{j+1}^{(n)} \right). \quad (10.69)$$

Practical experience shows that some weights become very small, and for this reason it is desirable to add a *resampling step* to determine the $\{v_{j+1}^{(n)}\}$ by drawing from (10.67); this has the effect of removing particles with very low weights and replacing them with multiple copies of the particles with higher weights. Because the initial measure $\mathbb{P}(v_0)$ is not in Dirac form, it is convenient to place this resampling step at the *start* of each iteration, rather than at the end as we have presented here, as this naturally introduces a particle approximation of the initial measure. This reordering makes no difference to the iteration we have described and results in the following algorithm.

Algorithm 4

1. Let $\mu_0^N = \mu_0$ and set $j = 0$.
2. Draw $v_j^{(n)} \sim \mu_j^N, n = 1, \dots, N$.
3. Set $w_j^{(n)} = 1/N, n = 1, \dots, N$ and define μ_j^N by (10.65).
4. Draw $\hat{v}_{j+1}^{(n)} \sim P_j(v_j^{(n)}, \cdot)$.
5. Define $w_{j+1}^{(n)}$ by (10.68), (10.69) and μ_{j+1}^N by (10.67).
6. $j \rightarrow j + 1$ and return to 2.

□

We define S^N to be the mapping between probability measures defined by sampling N i.i.d. points from a measure and approximating that measure by an equally weighted sum of Dirac's at the sample points. Then the preceding algorithm may be written as

$$\mu_{j+1}^N = \mathsf{L} S^N P_j \mu_j^N. \quad (10.70)$$

Although we have written the sampling step S^N *after* application of P_j , some reflection shows that this is well justified: applying P_j followed by S^N can be shown, by first conditioning on the initial point and sampling with respect to P_j and then sampling over the distribution of the initial point, to be the algorithm as defined. The sequence of distributions that we wish to approximate simply satisfies

the iteration (10.62). Thus, analyzing the particle filter requires estimation of the error induced by application of S^N (the *resampling error*) together with estimation of the rate of accumulation of this error in time.

The operators L , P_j and S^N map the space $\mathsf{P}(X)$ of probability measures on X into itself according to the following:

$$(\mathsf{L}\mu)(dv) = \frac{\exp(-h\Phi(v))\mu(dv)}{\int_X \exp(-h\Phi(v))\mu(dv)},$$

$$(P_j\mu)(dv) = \int_X P_j(v', dv)\mu(dv'),$$

$$(S^N\mu)(dv) = \frac{1}{N} \sum_{n=1}^N \delta(v - v^{(n)})dv, \quad v^{(n)} \sim \mu \text{ i.i.d.}$$

where P_j is the kernel associated with the μ_j -invariant Markov chain.

Let $\mu = \mu^{(\omega)}$ denote, for each ω , an element of $\mathsf{P}(X)$. If we assume that ω is a random variable and let \mathbb{E}^ω denote expectation over ω , then we may define a distance $d(\cdot, \cdot)$ between two random probability measures $\mu^{(\omega)}$ and $\nu^{(\omega)}$, as follows:

$$d(\mu, \nu) = \sup_{|f|_\infty \leq 1} \sqrt{\mathbb{E}^\omega |\mu(f) - \nu(f)|^2},$$

with $|f|_\infty := \sup_{v \in X} |f(v)|$, and where we have used the convention that $\mu(f) = \int_X f(v)\mu(dv)$ for measurable $f : X \rightarrow \mathbb{R}$, and similar for ν . This distance does indeed generate a metric and, in particular, satisfies the triangle inequality. In fact it is simply the total variation distance in the case of measures which are not random.

With respect to this distance between random probability measures, we may prove that the SMC method generates a good approximation of the true measure μ , in the limit $N \rightarrow \infty$. We use the fact that, under (10.60), we have

$$\exp(-h\phi^+) < \exp(-h\Phi(v)) < \exp(-h\phi^-).$$

Since $\phi^- \leq 0$ and $\phi^+ \geq 0$, we deduce that there exists $\kappa \in (0, 1)$ such that for all $v \in X$,

$$\kappa < \exp(-h\Phi(v)) < \kappa^{-1}.$$

This constant κ appears in the following.

Theorem 23. *We assume in the following that (10.60) holds. Then*

$$d(\mu_J^N, \mu_J) \leq \sum_{j=1}^J (2\kappa^{-2})^j \frac{1}{\sqrt{N}}.$$

Proof. The desired result is a consequence of the following three facts, whose proof we postpone to three lemmas at the end of the subsection:

$$\begin{aligned} \sup_{\mu \in \mathcal{P}(X)} d(S^N \mu, \mu) &\leq \frac{1}{\sqrt{N}}, \\ d(P_j v, P_j \mu) &\leq d(v, \mu), \\ d(\mathsf{L}v, \mathsf{L}\mu) &\leq 2\kappa^{-2} d(v, \mu). \end{aligned}$$

By the triangle inequality, we have, for $v_j^N = P_j \mu_j^N$,

$$\begin{aligned} d(\mu_{j+1}^N, \mu_{j+1}) &= d(\mathsf{L}S^N P_j \mu_j^N, \mathsf{L}P_j \mu_j) \\ &\leq d(\mathsf{L}P_j \mu_j^N, \mathsf{L}P_j \mu_j) + d(\mathsf{L}S^N P_j \mu_j^N, \mathsf{L}P_j \mu_j^N) \\ &\leq 2\kappa^{-2} \left(d(\mu_j^N, \mu_j) + d(S^N v_j^N, v_j^N) \right) \\ &\leq 2\kappa^{-2} \left(d(\mu_j^N, \mu_j) + \frac{1}{\sqrt{N}} \right). \end{aligned}$$

Iterating, after noting that $\mu_0^N = \mu_0$, gives the desired result.

Remarks 8. This theorem shows that the sequential particle filter actually reproduces the true posterior distribution $\mu = \mu_J$, in the limit $N \rightarrow \infty$. We make some comments about this.

- The measure $\mu = \mu_J$ is well approximated by μ_j^N in the sense that, as the number of particles $N \rightarrow \infty$, the approximating measure converges to the true measure. The result holds in the infinite-dimensional setting. As a consequence the algorithm as stated is robust to finite-dimensional approximation.
- Note that $\kappa = \kappa(J)$ and that $\kappa \rightarrow 1$ as $J \rightarrow \infty$. Using this fact shows that the error constant in Theorem 23 behaves as $\sum_{j=1}^J (2\kappa^{-2})^j \asymp J 2^J$. Optimizing this upper bound does not give a useful rule of thumb for choosing J and in fact suggests choosing $J = 1$. In any case in applications Φ is not bounded from above, or even below in general, and a more refined analysis is then required.
- In principle the theory applies even if the Markov kernel P_j is simply the identity mapping on probability measures. However, moving the particles according to a nontrivial μ_j -invariant measure is absolutely essential for the methodology to work in practice. This can be seen by noting that if P_j is indeed taken to be the identity map on measures, then the particle positions will be unchanged as j changes, meaning that the measure $\mu = \mu_J$ is approximated by weighted samples from the prior, clearly undesirable in general.
- In fact, if the Markov kernel P_j is ergodic, then it is sometimes possible to obtain bounds which are *uniform* in J .

We now prove the three lemmas which underlie the convergence proof.

Lemma 8. *The sampling operator satisfies*

$$\sup_{\mu \in \mathbb{P}(X)} d(S^N \mu, \mu) \leq \frac{1}{\sqrt{N}}.$$

Proof. Let ν be an element of $\mathbb{P}(X)$ and $\{v^{(k)}\}_{k=1}^N$ a set of i.i.d. samples with $v^{(1)} \sim \nu$; the randomness entering the probability measures is through these samples, expectation with respect to which we denote by \mathbb{E}^ω in what follows. Then

$$S^N \nu(f) = \frac{1}{N} \sum_{k=1}^N f(v^{(k)})$$

and, defining $\bar{f} = f - \nu(f)$, we deduce that

$$S^N \nu(f) - \nu(f) = \frac{1}{N} \sum_{k=1}^N \bar{f}(v^{(k)}).$$

It is straightforward to see that

$$\mathbb{E}^\omega \bar{f}(v^{(k)}) \bar{f}(v^{(l)}) = \delta_{kl} \mathbb{E}^\omega |\bar{f}(v^{(k)})|^2.$$

Furthermore, for $|f|_\infty \leq 1$,

$$\mathbb{E}^\omega |\bar{f}(v^{(1)})|^2 = \mathbb{E}^\omega |f(v^{(1)})|^2 - |\mathbb{E}^\omega f(v^{(1)})|^2 \leq 1.$$

It follows that, for $|f|_\infty \leq 1$,

$$\mathbb{E}^\omega |\nu(f) - S^N \nu(f)|^2 = \frac{1}{N^2} \sum_{k=1}^N \mathbb{E}^\omega |\bar{f}(v^{(k)})|^2 \leq \frac{1}{N}.$$

Since the result is independent of ν , we may take the supremum over all probability measures and obtain the desired result.

Lemma 9. *Since P_j is a Markov kernel, we have*

$$d(P_j \nu, P_j \nu') \leq d(\nu, \nu').$$

Proof. The result is generic for any Markov kernel P , so we drop the index j on P_j for the duration of the proof. Define

$$q(v') = \int_X P(v', dv) f(v),$$

that is the expected value of f under one step of the Markov chain started from v' . Clearly, since

$$|q(v')| \leq \left(\int_X P(v', dv) \right) \sup_v |f(v)| = \sup_v |f(v)|$$

it follows that

$$\sup_v |q(v)| \leq \sup_v |f(v)|.$$

Also, since

$$Pv(f) = \int_X f(v) \left(\int_X P(v', dv) v(dv') \right),$$

exchanging the order of integration shows that

$$|Pv(f) - Pv'(f)| = |v(q) - v'(q)|.$$

Thus

$$\begin{aligned} d(Pv, Pv') &= \sup_{\|f\|_\infty \leq 1} \left(\mathbb{E}^\omega |Pv(f) - Pv'(f)|^2 \right)^{\frac{1}{2}} \\ &\leq \sup_{\|q\|_\infty \leq 1} \left(\mathbb{E}^\omega |v(q) - v'(q)|^2 \right)^{\frac{1}{2}} \\ &= d(v, v') \end{aligned}$$

as required.

Lemma 10. *Under the Assumptions of Theorem 23, we have*

$$d(\mathsf{L}v, \mathsf{L}\mu) \leq 2\kappa^{-2} d(v, \mu).$$

Proof. Define $g(v) = \exp(-h\Phi(v))$. Notice that for $|f|_\infty < \infty$, we can rewrite

$$\begin{aligned} (\mathbb{L}\nu)(f) - (\mathbb{L}\mu)(f) &= \frac{\nu(fg)}{\nu(g)} - \frac{\mu(fg)}{\mu(g)} \\ &= \frac{\nu(fg)}{\nu(g)} - \frac{\mu(fg)}{\nu(g)} + \frac{\mu(fg)}{\nu(g)} - \frac{\mu(fg)}{\mu(g)} \\ &= \frac{\kappa^{-1}}{\nu(g)} [\nu(\kappa fg) - \mu(\kappa fg)] + \frac{\mu(fg)}{\mu(g)} \frac{\kappa^{-1}}{\nu(g)} [\mu(\kappa g) - \nu(\kappa g)]. \end{aligned}$$

Now notice that $\nu(g)^{-1} \leq \kappa^{-1}$ and that, for $|f|_\infty \leq 1$, $\mu(fg)/\mu(g) \leq 1$ since the expression corresponds to an expectation with respect to measure found from μ by reweighting with likelihood proportional to g . Thus

$$|(\mathbb{L}\nu)(f) - (\mathbb{L}\mu)(f)| \leq \kappa^{-2} |\nu(\kappa fg) - \mu(\kappa fg)| + \kappa^{-2} |\nu(\kappa g) - \mu(\kappa g)|.$$

Since $|\kappa g| \leq 1$, it follows that

$$\mathbb{E}^\omega |(\mathbb{L}\nu)(f) - (\mathbb{L}\mu)(f)|^2 \leq 4\kappa^{-4} \sup_{|f|_\infty \leq 1} \mathbb{E}^\omega |\nu(f) - \mu(f)|^2$$

and the desired result follows.

5.4 Continuous Time Markov Processes

In the remainder of this section, we shift our attention to continuous time processes which preserve μ ; these are important in the construction of proposals for MCMC methods and also as diffusion limits for MCMC. Our main goal is to show that the equation (10.53) preserves μ . Our setting is to work in the separable Hilbert space \mathcal{H} with Inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. We assume that the prior μ_0 is a Gaussian on \mathcal{H} and, furthermore, we specify the space $X \subset \mathcal{H}$ that will play a central role in this continuous time setting. This choice of space X will link the properties of the reference measure μ_0 and the potential Φ . We assume that \mathcal{C} has eigendecomposition

$$\mathcal{C}\phi_j = \gamma_j^2 \phi_j \tag{10.74}$$

where $\{\phi_j\}_{j=1}^\infty$ forms an orthonormal basis for \mathcal{H} and where $\gamma_j \asymp j^{-s}$. Necessarily $s > \frac{1}{2}$ since \mathcal{C} must be trace class to be a covariance on \mathcal{H} . We define the following scale of Hilbert subspaces, defined for $r > 0$, by

$$\mathcal{X}^r = \left\{ u \in \mathcal{H} \mid \sum_{j=1}^{\infty} j^{2r} |\langle u, \phi_j \rangle|^2 < \infty \right\}$$

and then extend to superspaces $r < 0$ by duality. We use $\|\cdot\|_r$ to denote the norm induced by the inner product

$$\langle u, v \rangle_r = \sum_{j=1}^{\infty} j^{2r} u_j v_j$$

for $u_j = \langle u, \phi_j \rangle$ and $v_j = \langle v, \phi_j \rangle$. Application of Theorem 5 with $d = 1$ and $q = 2$ shows that $\mu_0(\mathcal{X}^r) = 1$ for all $r \in [0, s - \frac{1}{2}]$. In what follows we will take $X = \mathcal{X}^t$ for some fixed $t \in [0, s - \frac{1}{2}]$.

Notice that we have not assumed that the underlying Hilbert space is comprised of L^2 functions mapping $D \subset \mathbb{R}^d$ into \mathbb{R} , and hence we have not introduced the dimension d of an underlying physical space \mathbb{R}^d into either the decay assumptions on the γ_j or the spaces \mathcal{X}^r . However, note that the spaces \mathcal{H}^t introduced in Sect. 2.4 are, in the case where $\mathcal{H} = L^2(D; \mathbb{R})$, the same as the spaces $\mathcal{X}^{t/d}$.

We now break our developments into introductory discussion of the finite-dimensional setting, in Sect. 5.5, and into the Hilbert space setting in Sect. 5.6. In Sect. 5.5.1, we introduce a family of Langevin equations which are invariant with respect to a given measure with smooth Lebesgue density. Using this, in Sect. 5.5.2, we motivate equation (10.53) showing that, in finite dimensions, it corresponds to a particular choice of Langevin equation. In Sect. 5.6.1, for the infinite-dimensional setting, we describe the precise assumptions under which we will prove invariance of measure μ under the dynamics (10.53). Section 5.6.2 describes the elements of the finite-dimensional approximation of (10.53) which will underlie our proof of invariance. Finally, Sect. 5.6.3 contains statement of the measure invariance result as Theorem 27, together with its proof; this is preceded by Theorem 26 which establishes existence and uniqueness of a solution to (10.53), as well as continuous dependence of the solution on the initial condition and Brownian forcing. Theorems 25 and 24 are the finite-dimensional analogues of Theorems 27 and 26, respectively, and play a useful role in motivating the infinite-dimensional theory.

5.5 Finite-Dimensional Langevin Equation

5.5.1 Background Theory

Before setting up the (rather involved) technical assumptions required for our proof of measure invariance, we give some finite-dimensional intuition. Recall that $|\cdot|$

denotes the Euclidean norm on \mathbb{R}^n , and we also use this notation for the induced matrix norm on \mathbb{R}^n . We assume that

$$I \in C^2(\mathbb{R}^n, \mathbb{R}^+), \quad \int_{\mathbb{R}^n} e^{-I(u)} du = 1.$$

Thus $\rho(u) = e^{-I(u)}$ is the Lebesgue density corresponding to a random variable on \mathbb{R}^n . Let μ be the corresponding measure.

Let \mathbb{W} denote standard Wiener measure on \mathbb{R}^n . Thus $B \sim \mathbb{W}$ is a standard Brownian motion in $C([0, \infty); \mathbb{R}^n)$. Let $u \in C([0, \infty); \mathbb{R}^n)$ satisfy the SDE

$$\frac{du}{dt} = -A DI(u) + \sqrt{2A} \frac{dB}{dt}, \quad u(0) = u_0 \quad (10.75)$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric and strictly positive definite and $DI \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ is the gradient of I . Assume that $\exists M > 0 : \forall u \in \mathbb{R}^n$, the Hessian of I satisfies

$$|D^2 I(u)| \leq M.$$

We refer to equations of the form (10.75) as *Langevin equations* (as mentioned earlier, they correspond to overdamped Langevin equations in the physics literature and to Langevin equations in the statistics literature) and the matrix A as a *preconditioner*.

Theorem 24. *For every $u_0 \in \mathbb{R}^n$ and \mathbb{W} -a.s., equation (10.75) has a unique global in time solution $u \in C([0, \infty); \mathbb{R}^n)$.*

Proof. A solution of the SDE is a solution of the integral equation

$$u(t) = u_0 - \int_0^t A DI(u(s)) ds + \sqrt{2A} B(t). \quad (10.76)$$

Define $X = C([0, T]; \mathbb{R}^n)$ and $\mathcal{F} : X \rightarrow X$ by

$$(\mathcal{F}v)(t) = u_0 - \int_0^t A DI(v(s)) ds + \sqrt{2A} B(t). \quad (10.77)$$

Thus $u \in X$ solving (10.76) is a fixed point of \mathcal{F} . We show that \mathcal{F} has a unique fixed point, for T sufficiently small. To this end we study a contraction property of \mathcal{F} :

$$\begin{aligned}
\|(\mathcal{F}v_1) - (\mathcal{F}v_2)\|_X &= \sup_{0 \leq t \leq T} \left| \int_0^t \left(A DI(v_1(s)) - A DI(v_2(s)) \right) ds \right| \\
&\leq \int_0^T \left| A DI(v_1(s)) - A DI(v_2(s)) \right| ds \\
&\leq \int_0^T |A|M|v_1(s) - v_2(s)| ds \\
&\leq T|A|M\|v_1 - v_2\|_X.
\end{aligned}$$

Choosing $T : T|A|M < 1$ shows that \mathcal{F} is a contraction on X . This argument may be repeated on successive intervals $[T, 2T], [2T, 3T], \dots$ to obtain a unique global solution in $C([0, \infty); \mathbb{R}^n)$.

Remark 2. Note that, since A is positive-definite symmetric, its eigenvectors e_j form an orthonormal basis for \mathbb{R}^n . We write $Ae_j = \alpha_j^2 e_j$. Thus

$$B(t) = \sum_{j=1}^n \beta_j(t) e_j$$

where the $\{\beta_j\}_{j=1}^n$ are an i.i.d. collection of standard unit Brownian motions on \mathbb{R} . Thus we obtain

$$\sqrt{A}B(t) = \sum_{j=1}^n \alpha_j \beta_j e_j =: W(t).$$

We refer to W as an A -Wiener process. Such a process is Gaussian with mean zero and covariance structure

$$\mathbb{E}W(t) \otimes W(s) = A(t \wedge s).$$

The equation (10.75) may be written as

$$\frac{du}{dt} = -ADI(u) + \sqrt{2} \frac{dW}{dt}, \quad u(0) = u_0. \quad (10.78)$$

Theorem 25. Let $u(t)$ solve (10.75). If $u_0 \sim \mu$, then $u(t) \sim \mu$ for all $t > 0$. More precisely, for all $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^+$ bounded and continuous, $u_0 \sim \mu$ implies

$$\mathbb{E}\varphi(u(t)) = \mathbb{E}\varphi(u_0), \quad \forall t > 0.$$

Proof. Consider the additive noise SDE, for additive noise with strictly positive-definite diffusion matrix Σ ,

$$\frac{du}{dt} = f(u) + \sqrt{2\Sigma} \frac{dB}{dt}, \quad u(0) = u_0 \sim \nu_0.$$

If ν_0 has pdf ρ_0 , then the Fokker-Planck equation for this SDE is

$$\begin{aligned} \frac{\partial \bar{\rho}}{\partial t} &= \nabla \cdot (-f \bar{\rho} + \Sigma \nabla \bar{\rho}), \quad (u, t) \in \mathbb{R}^n \times \mathbb{R}^+, \\ \bar{\rho}|_{t=0} &= \rho_0. \end{aligned}$$

At time $t > 0$ the solution of the SDE is distributed according to measure $\nu(t)$ with density $\bar{\rho}(u, t)$ solving the Fokker-Planck equation. Thus the initial measure ν_0 is preserved if

$$\nabla \cdot (-f \rho_0 + \Sigma \nabla \rho_0) = 0$$

and then $\bar{\rho}(\cdot, t) = \rho_0$, $\forall t \geq 0$.

We apply this Fokker-Planck equation to show that μ is invariant for equation (10.76). We need to show that

$$\nabla \cdot (ADI(u)\rho + A \nabla \rho) = 0$$

if $\rho = e^{-I(u)}$. With this choice of ρ we have

$$\nabla \rho = -DI(u)e^{-I(u)} = -DI(u)\rho.$$

Thus

$$A DI(u)\rho + A \nabla \rho = A DI(u)\rho - A DI(u)\rho = 0,$$

so that

$$\nabla \cdot (A DI(u)\rho + A \nabla \rho) = \nabla \cdot (0) = 0.$$

Hence the proof is complete.

5.5.2 Motivation for Equation (10.53)

Using the preceding finite-dimensional development, we now motivate the form of equation (10.53). For (10.52) we have, if \mathcal{H} is \mathbb{R}^n ,

$$\mu(du) = \exp(-I(u)) du, \quad I(u) = \frac{1}{2}|\mathcal{C}^{-\frac{1}{2}}u|^2 + \Phi(u) + \ln Z.$$

Thus

$$DI(u) = \mathcal{C}^{-1}u + D\Phi(u)$$

and equation (10.75), which preserves μ , is

$$\frac{du}{dt} = -A(\mathcal{C}^{-1}u + D\Phi(u)) + \sqrt{2A}\frac{dB}{dt}.$$

Choosing the preconditioner $A = \mathcal{C}$ gives

$$\frac{du}{dt} = -u - \mathcal{C}D\Phi(u) + \sqrt{2\mathcal{C}}\frac{dB}{dt}.$$

This is exactly (10.53) provided $W = \sqrt{\mathcal{C}}B$, where B is a Brownian motion with covariance \mathcal{I} . Then W is a Brownian motion with covariance \mathcal{C} . This is the finite-dimensional analogue of the construction of a \mathcal{C} -Wiener process in the Appendix. We are now in a position to prove Theorems 26 and 27 which are the infinite-dimensional analogues of Theorems 24 and 25.

5.6 Infinite-Dimensional Langevin Equation

5.6.1 Assumptions on Change of Measure

Recall that $\mu_0(\mathcal{X}^r) = 1$ for all $r \in [0, s - \frac{1}{2}]$. The functional $\Phi(\cdot)$ is assumed to be defined on \mathcal{X}^t for some $t \in [0, s - \frac{1}{2})$, and indeed we will assume appropriate bounds on the first and second derivatives, building on this assumption. (Thus, in this Sect. 5.6.1, t does not denote time; instead we use τ to denote the generic time argument.) These regularity assumptions on $\Phi(\cdot)$ ensure that the probability distribution μ is not too different from μ_0 , when projected into directions associated with ϕ_j for j large.

For each $u \in \mathcal{X}^t$, the derivative $D\Phi(u)$ is an element of the dual $(\mathcal{X}^t)^*$ of \mathcal{X}^t comprising continuous linear functionals on \mathcal{X}^t . However, we may identify $(\mathcal{X}^t)^*$ with \mathcal{X}^{-t} and view $D\Phi(u)$ as an element of \mathcal{X}^{-t} for each $u \in \mathcal{X}^t$. With this identification, the following identity holds

$$\|D\Phi(u)\|_{\mathcal{L}(\mathcal{X}^t, \mathbb{R})} = \|D\Phi(u)\|_{-t}$$

and the second derivative $D^2\Phi(u)$ can be identified as an element of $\mathcal{L}(\mathcal{X}^t, \mathcal{X}^{-t})$. To avoid technicalities, we assume that $\Phi(\cdot)$ is quadratically bounded, with first derivative linearly bounded and second derivative globally bounded. Weaker assumptions could be dealt with by use of stopping time arguments.

Assumptions 4. There exist constants $M_i \in \mathbb{R}^+, i \leq 4$ and $t \in [0, s - 1/2)$ such that, for all $u \in \mathcal{X}^t$, the functional $\Phi : \mathcal{X}^t \rightarrow \mathbb{R}$ satisfies

$$\begin{aligned} -M_1 &\leq \Phi(u) \leq M_2 \left(1 + \|u\|_t^2\right); \\ \|D\Phi(u)\|_{-t} &\leq M_3 \left(1 + \|u\|_t\right); \\ \|D^2\Phi(u)\|_{\mathcal{L}(\mathcal{X}^t, \mathcal{X}^{-t})} &\leq M_4. \end{aligned}$$

□

Example 10. The functional $\Phi(u) = \frac{1}{2}\|u\|_t^2$ satisfies Assumptions 4. To see this note that we may write $\Phi(u) = \frac{1}{2}\langle u, \mathcal{K}u \rangle$ where

$$\mathcal{K} = \frac{1}{2} \sum_{j=1}^{\infty} j^{2t} \phi_j \phi_j^*.$$

The functional $\Phi : \mathcal{X}^t \rightarrow \mathbb{R}^+$ is clearly well defined by definition. Its derivative at $u \in \mathcal{X}^t$ is given by $\mathcal{K}u = D\Phi(u) = \sum_{j \geq 1} j^{2t} u_j \phi_j$, where $u_j = \langle \phi_j, u \rangle$. Furthermore $D\Phi(u) \in \mathcal{X}^{-t}$ with $\|D\Phi(u)\|_{-t} = \|u\|_t$. The second derivative $D^2\Phi(u) \in \mathcal{L}(\mathcal{X}^t, \mathcal{X}^{-t})$ is the linear operator \mathcal{K} that is the operator that maps $u \in \mathcal{X}^t$ to $\sum_{j \geq 1} j^{2t} \langle u, \phi_j \rangle \phi_j \in \mathcal{X}^t$: its norm satisfies $\|D^2\Phi(u)\|_{\mathcal{L}(\mathcal{X}^t, \mathcal{X}^{-t})} = 1$ for any $u \in \mathcal{X}^t$. □

Since the eigenvalues γ_j^2 of \mathcal{C} decrease as $\gamma_j \asymp j^{-s}$, the operator \mathcal{C} has a smoothing effect: $\mathcal{C}^\alpha h$ gains $2\alpha s$ orders of regularity in the sense that the \mathcal{X}^β -norm of $\mathcal{C}^\alpha h$ is controlled by the $\mathcal{X}^{\beta-2\alpha s}$ -norm of $h \in \mathcal{H}$. Indeed it is straightforward to show the following:

Lemma 11. Under Assumption 4, the following estimates hold:

1. The operator \mathcal{C} satisfies

$$\|\mathcal{C}^\alpha h\|_\beta \asymp \|h\|_{\beta-2\alpha s}.$$

2. The function $\mathcal{CD}\Phi : \mathcal{X}^t \rightarrow \mathcal{X}^t$ is globally Lipschitz on \mathcal{X}^t : there exists a constant $M_5 > 0$ such that

$$\|\mathcal{CD}\Phi(u) - \mathcal{CD}\Phi(v)\|_t \leq M_5 \|u - v\|_t \quad \forall u, v \in \mathcal{X}^t. \quad (10.79)$$

3. The function $F : \mathcal{X}^t \rightarrow \mathcal{X}^t$ defined by

$$F(u) = -u - \mathcal{CD}\Phi(u) \quad (10.80)$$

is globally Lipschitz on \mathcal{X}^t .

4. The functional $\Phi(\cdot) : \mathcal{X}^t \rightarrow \mathbb{R}$ satisfies a second-order Taylor formula (for which we extend $\langle \cdot, \cdot \rangle$ from an inner product on \mathcal{X} to the dual pairing between \mathcal{X}^{-t} and \mathcal{X}^t). There exists a constant $M_6 > 0$ such that

$$\Phi(v) - \left(\Phi(u) + \langle D\Phi(u), v - u \rangle \right) \leq M_6 \|u - v\|_t^2 \quad \forall u, v \in \mathcal{X}^t. \quad (10.81)$$

5.6.2 Finite-Dimensional Approximation

Our analysis now proceeds as follows. First we introduce an approximation of the measure μ , denoted by μ^N . To this end we let P^N denote orthogonal projection in \mathcal{H} onto $X^N := \text{span}\{\phi_1, \dots, \phi_N\}$ and denote by Q^N orthogonal projection in \mathcal{H} onto $X^\perp := \text{span}\{\phi_{N+1}, \phi_{N+2}, \dots\}$. Thus $Q^N = I - P^N$. Then define the measure μ^N by

$$\frac{d\mu^N}{d\mu_0}(u) = \frac{1}{Z^N} \exp(-\Phi(P^N u)), \quad (10.82a)$$

$$Z^N = \int_{X'} \exp(-\Phi(P^N u)) \mu_0(du). \quad (10.82b)$$

This is a specific example of the approximating family in (10.44) if we define

$$\Phi^N := \Phi \circ P^N. \quad (10.83)$$

Indeed if we take $X = \mathcal{X}^\tau$ for any $\tau \in (t, s - \frac{1}{2})$, we see that $\|P^N\|_{\mathcal{L}(X, X)} = 1$ and that, for any $u \in X$,

$$\begin{aligned} \|\Phi(u) - \Phi^N(u)\| &= \|\Phi(u) - \Phi(P^N u)\| \\ &\leq M_3(1 + \|u\|_t) \|(I - P^N)u\|_t \\ &\leq CM_3(1 + \|u\|_\tau) \|u\|_\tau N^{-(\tau-t)}. \end{aligned}$$

Since Φ and hence Φ^N are bounded below by $-M_1$, and since the function $1 + \|u\|_\tau^2$ is integrable by the Fernique theorem 10, the approximation Theorem 18 applies. We deduce that the Hellinger distance between μ and μ^N is bounded above by $\mathcal{O}(N^{-r})$ for any $r < s - \frac{1}{2} - t$ since $\tau - t \in (0, s - \frac{1}{2} - t)$.

We will not use this explicit convergence rate in what follows, but we will use the idea that μ^N converges to μ in order to prove invariance of the measure μ under the SDE (10.53). The measure μ^N has a product structure that we will exploit in the following. We note that any element $u \in \mathcal{H}$ is uniquely decomposed as $u = p + q$ where $p \in X^N$ and $q \in X^\perp$. Thus we will write $\mu^N(du) = \mu^N(dp, dq)$, and similar expressions for μ_0 and so forth, in what follows.

Lemma 12. Define $\mathcal{C}^N = P^N \mathcal{C} P^N$ and $\mathcal{C}^\perp = Q^N \mathcal{C} Q^N$. Then μ_0 factors as the product of measures $\mu_{0,P} = N(0, \mathcal{C}^N)$ and $\mu_{0,Q} = N(0, \mathcal{C}^\perp)$ on X^N and X^\perp ,

respectively. Furthermore μ^N itself also factors as a product measure on $X^N \oplus X^\perp$: $\mu^N(dp, dq) = \mu_P(dp)\mu_Q(dq)$ with $\mu_Q = \mu_{0,Q}$ and

$$\frac{d\mu_P}{d\mu_{0,P}}(u) \propto \exp(-\Phi(p)).$$

Proof. Because P^N and Q^N commute with \mathcal{C} , and because $P^N Q^N = Q^N P^N = 0$, the factorization of the reference measure μ_0 follows automatically. The factorization of the measure μ follows from the fact that $\Phi^N(u) = \Phi(p)$ and hence does not depend on q .

To facilitate the proof of the desired measure preservation property, we introduce the equation

$$\frac{du^N}{dt} = -u^N - \mathcal{C}P^N D\Phi^N(u^N) + \sqrt{2}\frac{dW}{dt}. \quad (10.84)$$

By using well-known properties of finite-dimensional SDEs, we will show that if $u^N(0) \sim \mu^N$, then $u^N(t) \sim \mu^N$ for any $t > 0$. By passing to the limit $N = \infty$, we will deduce that for (10.53), if $u(0) \sim \mu$, then $u(t) \sim \mu$ for any $t > 0$.

The next lemma gathers various regularity estimates on the functional $\Phi^N(\cdot)$ that are repeatedly used in the sequel; they follow from the analogous properties of Φ by using the structure $\Phi^N = \Phi \circ P^N$.

Lemma 13. *Under Assumption 4, the following estimates hold with all constants uniform in N :*

1. *The estimates of Assumption 4 hold with Φ replaced by Φ^N .*
2. *The function $\mathcal{CD}\Phi^N : \mathcal{X}^t \rightarrow \mathcal{X}^t$ is globally Lipschitz on \mathcal{X}^t : there exists a constant $M_5 > 0$ such that*

$$\|\mathcal{CD}\Phi^N(u) - \mathcal{CD}\Phi^N(v)\|_t \leq M_5 \|u - v\|_t \quad \forall u, v \in \mathcal{X}^t.$$

3. *The function $F^N : \mathcal{X}^t \rightarrow \mathcal{X}^t$ defined by*

$$F^N(u) = -u - \mathcal{C}P^N D\Phi^N(u) \quad (10.85)$$

is globally Lipschitz on \mathcal{X}^t .

4. *The functional $\Phi^N(\cdot) : \mathcal{X}^t \rightarrow \mathbb{R}$ satisfies a second-order Taylor formula (for which we extend $\langle \cdot, \cdot \rangle$ from an inner product on \mathcal{X} to the dual pairing between \mathcal{X}^{-t} and \mathcal{X}^t). There exists a constant $M_6 > 0$ such that*

$$\Phi^N(v) - \left(\Phi^N(u) + \langle D\Phi^N(u), v - u \rangle \right) \leq M_6 \|u - v\|_t^2 \quad \forall u, v \in \mathcal{X}^t. \quad (10.86)$$

5.6.3 Main Theorem and Proof

Fix a function $W \in C([0, T]; \mathcal{X}^t)$. Recalling F defined by (10.80), we define a solution of (10.53) to be a function $u \in C([0, T]; \mathcal{X}^t)$ satisfying the integral equation

$$u(\tau) = u_0 + \int_0^\tau F(u(s)) ds + \sqrt{2} W(\tau) \quad \forall \tau \in [0, T]. \quad (10.87)$$

The solution is said to be *global* if $T > 0$ is arbitrary. For us, W will be a \mathcal{C} -Wiener process and hence random; we look for existence of a global solution, almost surely with respect to the Wiener measure. Similarly a solution of (10.84) is a function $u^N \in C([0, T]; \mathcal{X}^t)$ satisfying the integral equation

$$u^N(\tau) = u_0 + \int_0^\tau F^N(u^N(s)) ds + \sqrt{2} W(\tau) \quad \forall t \in [0, T]. \quad (10.88)$$

Again, the solution is random because W is a \mathcal{C} -Wiener process. Note that the solution to this equation is not confined to X^N , because both u_0 and W have nontrivial components in X^\perp . However, within X^\perp , the behavior is purely Gaussian and within X^N , it is finite dimensional. We will exploit these two facts.

The following establishes basic existence, uniqueness, continuity and approximation properties of the solutions of (10.87) and (10.88).

Theorem 26. *For every $u_0 \in \mathcal{X}^t$ and for almost every \mathcal{C} -Wiener process W , equation (10.87) (respectively, (10.88)) has a unique global solution. For any pair $(u_0, W) \in \mathcal{X}^t \times C([0, T]; \mathcal{X}^t)$, we define the Itô map*

$$\Theta: \mathcal{X}^t \times C([0, T]; \mathcal{X}^t) \rightarrow C([0, T]; \mathcal{X}^t)$$

which maps (u_0, W) to the unique solution u (resp. u^N for (10.88)) of the integral equation (10.87) (resp. Θ^N for (10.88)). The map Θ (respectively, Θ^N) is globally Lipschitz continuous. Finally we have that $\Theta^N(u_0, W) \rightarrow \Theta(u_0, W)$ strongly in $C([0, T]; \mathcal{X}^t)$ for every pair $(u_0, W) \in \mathcal{X}^t \times C([0, T]; \mathcal{X}^t)$.

Proof. The existence and uniqueness of local solutions to the integral equation (10.87) is a simple application of the contraction mapping principle, following arguments similar to those employed in the proof of Theorem 24. Extension to a global solution may be achieved by repeating the local argument on successive intervals.

Now let $u^{(i)}$ solve

$$u^{(i)} = u_0^{(i)} + \int_0^\tau F(u^{(i)})(s) ds + \sqrt{2} W^{(i)}(\tau), \quad \tau \in [0, T],$$

for $i = 1, 2$. Subtracting and using the Lipschitz property of F shows that $e = u^{(1)} - u^{(2)}$ satisfies

$$\begin{aligned}\|e(\tau)\|_t &\leq \|u_0^{(1)} - u_0^{(2)}\|_t + L \int_0^\tau \|e(s)\|_t ds + \sqrt{2} \|W^{(1)}(\tau) - W^{(2)}(\tau)\|_t \\ &\leq \|u_0^{(1)} - u_0^{(2)}\|_t + L \int_0^\tau \|e(s)\|_t ds + \sqrt{2} \sup_{0 \leq s \leq T} \|W^{(1)}(s) - W^{(2)}(s)\|_t.\end{aligned}$$

By application of the Gronwall inequality, we find that

$$\sup_{0 \leq \tau \leq T} \|e(\tau)\|_t \leq C(T) (\|u_0^{(1)} - u_0^{(2)}\|_t + \sup_{0 \leq s \leq T} \|W^{(1)}(s) - W^{(2)}(s)\|_t)$$

and the desired continuity is established.

Now we prove pointwise convergence of Θ^N to Θ . Let $e = u - u^N$ where u and u^N solve (10.87) and (10.88), respectively. The pointwise convergence of Θ^N to Θ is established by proving that $e \rightarrow 0$ in $C([0, T]; \mathcal{X}')$. Note that

$$F(u) - F^N(u^N) = (F^N(u) - F^N(u^N)) + (F(u) - F^N(u)).$$

Also, by Lemma 13, $\|F^N(u) - F^N(u^N)\|_t \leq L \|e\|_t$. Thus we have

$$\|e\|_t \leq L \int_0^\tau \|e(s)\|_t ds + \int_0^\tau \|F(u(s)) - F^N(u(s))\|_t ds.$$

Thus, by Gronwall, it suffices to show that

$$\delta^N := \sup_{0 \leq s \leq T} \|F(u(s)) - F^N(u(s))\|_t$$

tends to zero as $N \rightarrow \infty$. Note that

$$\begin{aligned}F(u) - F^N(u) &= \mathcal{C}D\Phi(u) - \mathcal{C}P^N D\Phi(P^N u) \\ &= (I - P^N)\mathcal{C}D\Phi(u) + P^N(\mathcal{C}D\Phi(u) - \mathcal{C}D\Phi(P^N u)).\end{aligned}$$

Thus, since $\mathcal{C}D\Phi$ is globally Lipschitz on \mathcal{X}' , by Lemma 11, and P^N has norm one as a mapping from \mathcal{X}' into itself,

$$\|F(u) - F^N(u)\|_t \leq \|(I - P^N)\mathcal{C}D\Phi(u)\|_t + C \|(I - P^N)u\|_t.$$

By dominated convergence $\|(I - P_N)a\|_t \rightarrow 0$ for any fixed element $a \in \mathcal{X}'$. Thus, because $\mathcal{C}D\Phi$ is globally Lipschitz, by Lemma 11, and as $u \in C([0, T]; \mathcal{X}')$, we deduce that it suffices to bound $\sup_{0 \leq s \leq T} \|u(s)\|_t$. But such a bound is a consequence of the existence theory outlined at the start of the proof, based on the proof of Theorem 24. \square

The following is a straightforward corollary of the preceding theorem:

Corollary 2. *For any pair $(u_0, W) \in \mathcal{X}^t \times C([0, T]; \mathcal{X}^t)$, we define the point Itô map*

$$\Theta_\tau : \mathcal{X}^t \times C([0, T]; \mathcal{X}^t) \rightarrow \mathcal{X}^t$$

(respectively, Θ_τ^N for (10.88)) which maps (u_0, W) to the unique solution $u(\tau)$ of the integral equation (10.87) (respectively, $u^N(\tau)$ for (10.88)) at time τ . The map Θ_τ (respectively, Θ_τ^N) is globally Lipschitz continuous. Finally we have that $\Theta_\tau^N(u_0, W) \rightarrow \Theta_\tau(u_0, W)$ for every pair $(u_0, W) \in \mathcal{X}^t \times C([0, T]; \mathcal{X}^t)$.

Theorem 27. *Let Assumption 4 hold. Then the measure μ given by (10.43) is invariant for (10.53); for all continuous bounded functions $\varphi : \mathcal{X}^t \rightarrow \mathbb{R}$, it follows that if \mathbb{E} denotes expectation with respect to the product measure found from initial condition $u_0 \sim \mu$ and $W \sim \mathbb{W}$, the C -Wiener measure on \mathcal{X}^t , then $\mathbb{E}\varphi(u(\tau)) = \mathbb{E}\varphi(u_0)$.*

Proof. We have that

$$\mathbb{E}\varphi(u(\tau)) = \int \varphi(\Theta_\tau(u_0, W))\mu(d u_0)\mathbb{W}(d W), \quad (10.89)$$

$$\mathbb{E}\varphi(u_0) = \int \varphi(u_0)\mu(d u_0). \quad (10.90)$$

If we solve equation (10.84) with $u_0 \sim \mu^N$, then, using \mathbb{E}^N with the obvious notation,

$$\mathbb{E}^N\varphi(u^N(\tau)) = \int \varphi(\Theta_\tau^N(u_0, W))\mu^N(d u_0)\mathbb{W}(d W), \quad (10.91)$$

$$\mathbb{E}^N\varphi(u_0) = \int \varphi(u_0)\mu^N(d u_0). \quad (10.92)$$

Lemma 14 below shows that, in fact,

$$\mathbb{E}^N\varphi(u^N(\tau)) = \mathbb{E}^N\varphi(u_0).$$

Thus it suffices to show that

$$\mathbb{E}^N\varphi(u^N(\tau)) \rightarrow \mathbb{E}\varphi(u(\tau)) \quad (10.93)$$

and

$$\mathbb{E}^N\varphi(u_0) \rightarrow \mathbb{E}\varphi(u_0). \quad (10.94)$$

Both of these facts follow from the dominated convergence theorem as we now show. First note that

$$\mathbb{E}^N \varphi(u_0) = \int \varphi(u_0) e^{-\Phi(P^N u_0)} \mu_0(du_0).$$

Since $\varphi(\cdot) e^{-\Phi \circ P^N}$ is bounded independently of N , by $(\sup \varphi) e^{M_1}$, and since $(\Phi \circ P^N)(u)$ converges pointwise to $\Phi(u)$ on \mathcal{X}' , we deduce that

$$\mathbb{E}^N \varphi(u_0) \rightarrow \int \varphi(u_0) e^{-\Phi(u_0)} \mu_0(du_0) = \mathbb{E} \varphi(u_0)$$

so that (10.94) holds. The convergence in (10.93) holds by a similar argument. From (10.91) we have

$$\mathbb{E}^N \varphi(u^N(\tau)) = \int \varphi(\Theta_\tau^N(u_0, W)) e^{-\Phi(P^N u_0)} \mu_0(du_0) \mathbb{W}(dW). \quad (10.95)$$

The integrand is again dominated by $(\sup \varphi) e^{M_1}$. Using the pointwise convergence of Θ_τ^N to Θ_τ on $\mathcal{X}' \times C([0, T]; \mathcal{X}')$, as proved in Corollary 2, as well as the pointwise convergence of $(\Phi \circ P^N)(u)$ to $\Phi(u)$, the desired result follows from dominated convergence: we find that

$$\mathbb{E}^N \varphi(u^N(\tau)) \rightarrow \int \varphi(\Theta_\tau(u_0, W)) e^{-\Phi(u_0)} \mu_0(du_0) \mathbb{W}(dW) = \mathbb{E} \varphi(u(\tau)).$$

The desired result follows. \square

Lemma 14. *Let Assumptions 4 hold. Then the measure μ^N given by (10.82) is invariant for (10.84); for all continuous bounded functions $\varphi : \mathcal{X}' \rightarrow \mathbb{R}$, it follows that if \mathbb{E}^N denotes expectation with respect to the product measure found from initial condition $u_0 \sim \mu^N$ and $W \sim \mathbb{W}$, the C-Wiener measure on \mathcal{X}' , then $\mathbb{E}^N \varphi(u^N(\tau)) = \mathbb{E}^N \varphi(u_0)$.*

Proof. Recall from Lemma 12 that measure μ^N given by (10.82) factors as the independent product of two measures on μ_P on X^N and μ_Q on X^\perp . On X^\perp the measure is simply the Gaussian $\mu_Q = \mathcal{N}(0, \mathcal{C}^\perp)$, while on X^N the measure μ_P is finite dimensional with density proportional to

$$\exp \left(-\Phi(p) - \frac{1}{2} \|(\mathcal{C}^N)^{-\frac{1}{2}} p\|^2 \right). \quad (10.96)$$

The equation (10.84) also decouples on the spaces X^N and X^\perp . On X^\perp it gives the integral equation

$$q(\tau) = - \int_0^\tau q(s) + \sqrt{2} Q^N W(\tau) \quad (10.97)$$

while on X^N it gives the integral equation

$$p(\tau) = - \int_0^\tau \left(p(s) + C^N D\Phi(p(s)) \right) ds + \sqrt{2} P^N W(\tau). \quad (10.98)$$

Measure μ_Q is preserved by (10.97), because (10.97) simply gives an integral equation formulation of the Ornstein-Uhlenbeck process with desired Gaussian invariant measure. On the other hand, equation (10.98) is simply an integral equation formulation of the Langevin equation for measure on \mathbb{R}^N with density (10.96), and a calculation with the Fokker-Planck equation, as in Theorem 25, demonstrates the required invariance of μ_P . \square

5.7 Bibliographic Notes

- Section 5.1 describes general background on Markov processes and invariant measures. The book [78] is a good starting point in this area. The book [75] provides a good overview of this subject area, from an applied and computational statistics perspective. For continuous time Markov chains, see [101].
- Section 5.2 concerns MCMC methods. The standard RWM was introduced in [73] and led, via the paper [46], to the development of the more general class of Metropolis-Hastings methods. The paper [94] is a key reference which provides a framework for the study of Metropolis-Hastings methods on general state spaces. The subject of MCMC methods which are invariant with respect to the target measure μ on infinite-dimensional spaces is overviewed in the paper [21]. The specific idea behind the Algorithm 3 is contained in [76, equation (15)], in the finite-dimensional setting. It is possible to show that, in the limit $\beta \rightarrow 0$, suitably interpolated output of Algorithm 3 converges to solution of the equation (10.53): see [82]. Furthermore it is also possible to compute a spectral gap for the Algorithm 3 in the infinite-dimensional setting [44]. This implies the existence of a dimension-independent spectral gap when finite-dimensional approximation is used; in contrast standard Metropolis-Hastings methods, such as random walk Metropolis, have a dimension-dependent spectral gap which shrinks with increasing dimension [99].
- Section 5.3 concerns SMC methods and the foundational work in this area is overviewed in the book [26]. The application of those ideas to the solution of PDE inverse problems was first demonstrated in [50], where the inverse problem is to determine the initial condition of the Navier-Stokes equations from observations. The method is applied to the elliptic inverse problem, with uniform

priors, in [10]. The proof of Theorem 23 follows the very clear exposition given in [84] in the context of filtering for hidden Markov models.

- Sections 5.4–5.6 concern measure preserving continuous time dynamics. The finite-dimensional aspects of this subsection, which we introduce for motivation, are covered in the texts [79] and [37]; the first of these books is an excellent introduction to the basic existence and uniqueness theory, outlined in a simple case in Theorem 24, while the second provides an in-depth treatment of the subject from the viewpoint of the Fokker-Planck equation, as used in Theorem 25. This subject has a long history which is overviewed in the paper [41] where the idea is applied to finding SPDEs which are invariant with respect to the measure generated by a conditioned diffusion process. This idea is generalized to certain conditioned hypoelliptic diffusions in [42]. It is also possible to study deterministic Hamiltonian dynamics which preserves the same measure. This idea is described in [9] in the same setup as employed here; that paper also contains references to the wider literature. Lemma 11 is proved in [72] and Lemma 13 in [82] Lemma 14 requires knowledge of the invariance of Ornstein-Uhlenbeck processes together with invariance of finite-dimensional first order Langevin equations with the form of gradient dynamics subject to additive noise. The invariance of the Ornstein-Uhlenbeck process is covered in [29] and invariance of finite-dimensional SDEs using the Fokker-Planck equation is discussed in [37]. The \mathcal{C} -Wiener process and its properties are described in [28].
- The primary focus of this section has been on the theory of measure-preserving dynamics and its relations to algorithms. The SPDEs are of interest in their own right as a theoretical object, but have particular importance in the construction of MCMC methods and in understanding the limiting behavior of MCMC methods. It is also important to appreciate that MCMC and SMC methods are by no means the only tools available to study the Bayesian inverse problem. In this context we note that computing the expectation with respect to the posterior can be reformulated as computing the ratio of two expectations with respect to the prior, the denominator being the normalization constant. Effectively in some such high-dimensional integration problems, [59] and [77] are general references on the QMC methodology. The paper [57] is a survey on the theory of QMC for bounded integration domains and is relevant for uniform priors. The paper [60] contains theoretical results for unbounded integration domains and is relevant to, for example, Gaussian priors. The use of QMC in plain uncertainty quantification (calculating the pushforward of a measure through a map) is studied for elliptic PDEs with random coefficients in [58] (uniform) and [39] (Gaussian). More sophisticated integration tools can be employed, using polynomial chaos representations of the prior measure, and computing posterior expectations in a manner which exploits sparsity in the map from unknown random coefficients to measured data; see [89, 90]. Much of this work, viewing uncertainty quantification from the point of high-dimensional integration, has its roots in early papers concerning plain uncertainty quantification in elliptic PDEs with random coefficients; the paper [7] was foundational in this area.

6 Conclusions

We have highlighted a theoretical treatment for Bayesian inversion over infinite-dimensional spaces. The resulting framework is appropriate for the mathematical analysis of inverse problems, as well as the development of algorithms. For example, on the analysis side, the idea of MAP estimators, which links the Bayesian approach with classical regularization, developed for Gaussian priors in [30], has recently been extended to other prior models in [47]; the study of contraction of the posterior distribution to a Dirac measure on the truth underlying the data is undertaken in [3, 4, 99]. On the algorithmic side, algorithms for Bayesian inversion in geophysical applications are formulated in [16, 81], and on the computational statistics side, methods for optimal experimental design are formulated in [5, 6]. All of these cited papers build on the framework developed in detail here and first outlined in [92]. It is thus anticipated that the framework herein will form the bedrock of other, related, developments of both the theory and computational practice of Bayesian inverse problems.

A Appendix

A.1 Function Spaces

In this subsection we briefly define the Hilbert and Banach spaces that will be important in our developments of probability and integration in infinite-dimensional spaces. As a consequence we pay particular attention to the issue of separability (the existence of a countable dense subset) which we require in that context. We primarily restrict our discussion to \mathbb{R} - or \mathbb{C} -valued functions, but the reader will easily be able to extend to \mathbb{R}^n -valued or $\mathbb{R}^{n \times n}$ -valued situations, and we discuss Banach space-valued functions at the end of the subsection.

A.1.1 ℓ^p and L^p Spaces

Consider real-valued sequences $u = \{u_j\}_{j=1}^\infty \in \mathbb{R}^\infty$. Let $w \in \mathbb{R}^\infty$ denote a positive sequence so that $w_j > 0$ for each $j \in \mathbb{N}$. For every $p \in [1, \infty)$, we define

$$\ell_w^p = \ell_w^p(\mathbb{N}; \mathbb{R}) = \left\{ u \in \mathbb{R}^\infty \mid \sum_{j=1}^{\infty} w_j |u_j|^p < \infty \right\}.$$

Then ℓ_w^p is a Banach space when equipped with the norm

$$\|u\|_{\ell_w^p} = \left(\sum_{j=1}^{\infty} w_j |u_j|^p \right)^{\frac{1}{p}}.$$

In the case $p = 2$, the resulting spaces are Hilbert spaces when equipped with the inner product

$$\langle u, v \rangle = \sum_{j=1}^{\infty} w_j u_j v_j.$$

These ℓ^p spaces, with $p \in [1, \infty)$, are separable. Throughout we simply write ℓ^p for the spaces ℓ_w^p with $w_j \equiv 1$. In the case $w_j \equiv 1$, we extend the definition of Banach spaces to the case $p = \infty$ by defining

$$\ell^\infty = \ell^\infty(\mathbb{N}; \mathbb{R}) = \left\{ u \in \mathbb{R} \mid \sup_{j \in \mathbb{N}} (|u_j|) < \infty \right\}$$

and

$$\|u\|_{\ell^\infty} = \sup_{j \in \mathbb{N}} (|u_j|).$$

The space ℓ^∞ of bounded sequences is *not* separable. Each element of the sequence u_j is real valued, but the definitions may be readily extended to complex-valued, \mathbb{R}^n -valued, and $\mathbb{R}^{n \times n}$ -valued sequences, replacing $|\cdot|$ by the complex modulus, the vector ℓ^p norm, and the operator ℓ^p norm on matrices, respectively.

We now extend the idea of p -summability to functions and to p -integrability. Let D be a bounded open set in \mathbb{R}^d with Lipschitz boundary and define the space $L^p = L^p(D; \mathbb{R})$ of Lebesgue measurable functions $f : D \rightarrow \mathbb{R}$ with norm $\|\cdot\|_{L^p(D)}$ defined by

$$\|f\|_{L^p(D)} := \begin{cases} \left(\int_D |f|^p dx \right)^{\frac{1}{p}} & \text{for } 1 \leq p < \infty \\ \text{ess sup}_D |f| & \text{for } p = \infty. \end{cases}$$

In the above definition we have used the notation

$$\text{ess sup}_D |f| = \inf \{C : |f| \leq C \text{ a.e. on } D\}.$$

Here *a.e.* is with respect to Lebesgue measure and the integral is, of course, the Lebesgue integral. Sometimes we drop explicit reference to the set D in the norm and simply write $\|\cdot\|_{L^p}$. For Lebesgue measurable functions $f : D \rightarrow \mathbb{R}^n$, the norm is readily extended replacing $|f|$ under the integral by the vector p -norm on \mathbb{R}^n . Likewise we may consider Lebesgue measurable $f : D \rightarrow \mathbb{R}^{n \times n}$, using the operator p -norm on $\mathbb{R}^{n \times n}$. In all these cases, we write $L^p(D)$ as shorthand for $L^p(D; X)$ where $X = \mathbb{R}$, \mathbb{R}^n or $\mathbb{R}^{n \times n}$. Then $L^p(D)$ is the vector space of all (equivalence classes of) measurable functions $f : D \rightarrow \mathbb{R}$ for which $\|f\|_{L^p(D)} < \infty$. The space $L^p(D)$ is separable for $p \in [1, \infty)$, while $L^\infty(D)$ is not separable. We define periodic versions of $L^p(D)$, denoted by $L_{\text{per}}^p(D)$, in the case where D is a unit

cube; these spaces are defined as the completion of C^∞ periodic functions on the unit cube, with respect to the L^p -norm. If we define \mathbb{T}^d to be the d -dimensional unit torus, then we write $L_{\text{per}}^p([0, 1]^d) = L^p(\mathbb{T}^d)$. Again these spaces are separable for $1 \leq p < \infty$, but not for $p = \infty$.

A.1.2 Continuous and Hölder Continuous Functions

Let D be an open and bounded set in \mathbb{R}^d with Lipschitz boundary. We will denote by $C(\overline{D}, \mathbb{R})$, or simply $C(\overline{D})$, the space of continuous functions $f : \overline{D} \rightarrow \mathbb{R}$. When equipped with the supremum norm,

$$\|f\|_{C(\overline{D})} = \sup_{x \in \overline{D}} |f(x)|,$$

$C(\overline{D})$ is a Banach space. Building on this we define the space $C^{0,\gamma}(\overline{D})$ to be the space of functions in $C(\overline{D})$ which are Hölder with any exponent $\gamma \in (0, 1]$ with norm

$$\|f\|_{C^{0,\gamma}(\overline{D})} = \sup_{x \in \overline{D}} |f(x)| + \sup_{x, y \in \overline{D}} \left(\frac{|f(x) - f(y)|}{|x - y|^\gamma} \right). \quad (10.99)$$

The case $\gamma = 1$ corresponds to Lipschitz functions.

We remark that $C(\overline{D})$ is separable since $\overline{D} \subset \mathbb{R}^d$ is compact here. The space of Hölder functions $C^{0,\gamma}(\overline{D}; \mathbb{R})$ is, however, *not* separable. Separability can be recovered by working in the subset of $C^{0,\gamma}(\overline{D}; \mathbb{R})$ where, in addition to (10.99) being finite,

$$\lim_{y \rightarrow x} \frac{|f(x) - f(y)|}{|x - y|^\gamma} = 0,$$

uniformly in x ; we denote the resulting separable space by $C_0^{0,\gamma}(\overline{D}, \mathbb{R})$. This is analogous to the fact that the space of bounded measurable functions is not separable, while the space of continuous functions on a compact domain is. Furthermore it may be shown that $C^{0,\gamma'} \subset C_0^{0,\gamma}$ for every $\gamma' > \gamma$. All of the preceding spaces can be generalized to functions $C^{0,\gamma}(\overline{D}, \mathbb{R}^n)$ and $C_0^{0,\gamma}(\overline{D}, \mathbb{R}^n)$; they may also be extended to periodic functions on the unit torus \mathbb{T}^d found by identifying opposite faces of the unit cube $[0, 1]^d$. The same separability issues arise for these generalizations.

A.1.3 Sobolev Spaces

We define Sobolev spaces of functions with integer number of derivatives, extend to fractional and negative derivatives, and make the connection with Hilbert scales. Here D is a bounded open set in \mathbb{R}^d with Lipschitz boundary. In the context of a function $u \in L^2(D)$, we will use the notation $\frac{\partial u}{\partial x_i}$ to denote the weak derivative with respect to x_i and the notation ∇u for the weak gradient.

The *Sobolev space* $W^{r,p}(D)$ consists of all L^p -integrable functions $u : D \rightarrow \mathbb{R}$ whose α^{th} order weak derivatives exist and are L^p -integrable for all $|\alpha| \leq r$:

$$W^{r,p}(D) = \left\{ u \mid D^\alpha u \in L^p(D) \text{ for } |\alpha| \leq r \right\} \quad (10.100)$$

with norm

$$\|u\|_{W^{r,p}(D)} = \begin{cases} \left(\sum_{|\alpha| \leq r} \|D^\alpha u\|_{L^p(D)}^p \right)^{\frac{1}{p}} & \text{for } 1 \leq p < \infty, \\ \sum_{|\alpha| \leq r} \|D^\alpha u\|_{L^\infty(D)} & \text{for } p = \infty. \end{cases} \quad (10.101)$$

We denote $W^{r,2}(D)$ by $H^r(D)$. We define periodic versions of $H^s(D)$, denoted by $H_{\text{per}}^s(D)$, in the case where D is a unit cube $[0, 1]^d$; these spaces are defined as the completion of C^∞ periodic functions on the unit cube, with respect to the H^s -norm. If we define \mathbb{T}^d to be d -dimensional unit torus, we then write $H^s(\mathbb{T}^d) = H_{\text{per}}^s([0, 1]^d)$.

The spaces $H^s(D)$ with D a bounded open set in \mathbb{R}^d , and $H_{\text{per}}^s([0, 1]^d)$, are separable Hilbert spaces. In particular if we define the inner-product $(\cdot, \cdot)_{L^2(D)}$ on $L^2(D)$ by

$$(u, v)_{L^2(D)} := \int_D u(x)v(x)dx$$

and define the resulting norm $\|\cdot\|_{L^2(D)}$ by the identity

$$\|u\|_{L^2(D)}^2 = (u, u)_{L^2(D)}$$

then the space $H^1(D)$ is a separable Hilbert space with inner product

$$\langle u, v \rangle_{H^1(D)} = (u, v)_{L^2(D)} + (\nabla u, \nabla v)_{L^2(D)}$$

and norm (10.101) with $p = 2$. Likewise the space $H_0^1(D)$ is a separable Hilbert space with inner product

$$\langle u, v \rangle_{H_0^1(D)} = (\nabla u, \nabla v)_{L^2(D)}$$

and norm

$$\|u\|_{H_0^1(D)} = \|\nabla u\|_{L^2(D)}. \quad (10.102)$$

As defined above, Sobolev spaces concern integer numbers of derivatives. However the concept can be extended to fractional derivatives, and there is then a natural connection to Hilbert scales of functions. To explain this we start our

development in the periodic setting. Recall that, given an element u in $L^2(\mathbb{T}^d)$, we can decompose it as a Fourier series:

$$u(x) = \sum_{k \in \mathbb{Z}^d} u_k e^{2\pi i \langle k, x \rangle},$$

where the identity holds for (Lebesgue) almost every $x \in \mathbb{T}^d$. Furthermore, the L^2 norm of u is given by Parseval's identity $\|u\|_{L^2}^2 = \sum |u_k|^2$. The fractional Sobolev space $H^s(\mathbb{T}^d)$ for $s \geq 0$ is given by the subspace of functions $u \in L^2(\mathbb{T}^d)$ such that

$$\|u\|_{H^s}^2 := \sum_{k \in \mathbb{Z}^d} (1 + 4\pi^2 |k|^2)^s |u_k|^2 < \infty. \quad (10.103)$$

Note that this is a separable Hilbert space by virtue of ℓ_w^2 being separable. Note also that $H^0(\mathbb{T}^d) = L^2(\mathbb{T}^d)$ and that, for positive integer s , the definition agrees with the definition $H^s(\mathbb{T}^d) = W^{s,2}(\mathbb{T}^d)$ obtained from (10.100) with the obvious generalization from D to \mathbb{T}^d . For $s < 0$, we define $H^s(\mathbb{T}^d)$ as the closure of L^2 under the norm (10.103). The spaces $H^s(\mathbb{T}^d)$ for $s < 0$ may also be defined via duality. The resulting spaces H^s are separable for all $s \in \mathbb{R}$.

We now link the spaces $H^s(\mathbb{T}^d)$ to a specific Hilbert scale of spaces. Hilbert scales are families of spaces defined by $\mathcal{D}(A^{s/2})$ for A a positive, unbounded, self-adjoint operator on a Hilbert space. To view the fractional Sobolev spaces from this perspective, let $A = I - \Delta$ with domain $H^2(\mathbb{T}^d)$, noting that the eigenvalues of A are simply $1 + 4\pi^2 |k|^2$ for $k \in \mathbb{Z}^d$. We thus see that, by the spectral decomposition theorem, $H^s = \mathcal{D}(A^{s/2})$, and we have $\|u\|_{H^s} = \|A^{s/2}u\|_{L^2}$. Note that we may work in the space of real-valued functions where the eigenfunctions of A , $\{\varphi_j\}_{j=1}^\infty$, comprise sine and cosine functions; the eigenvalues of A , when ordered on a one-dimensional lattice, then satisfy $\alpha_j \asymp j^{2/d}$. This is relevant to the more general perspective of Hilbert scales that we now introduce.

We can now generalize the previous construction of fractional Sobolev spaces to more general domains than the torus. The resulting spaces do not, in general, coincide with Sobolev spaces, because of the effect of the boundary conditions of the operator A used in the construction. On an arbitrary bounded open set $D \subset \mathbb{R}^d$ with Lipschitz boundary, we consider a positive self-adjoint operator A satisfying Assumption 1 so that its eigenvalues satisfy $\alpha_j \asymp j^{2/d}$; then we define the spaces $\mathcal{H}^s = \mathcal{D}(A^{s/2})$ for $s > 0$. Given a Hilbert space $(H, \langle \cdot, \cdot \rangle, \|\cdot\|)$ of real-valued functions on a bounded open set D in \mathbb{R}^d , we recall from Assumption 1 the orthonormal basis for H denoted by $\{\varphi_j\}_{j=1}^\infty$. Any $u \in H$ can be written as

$$u = \sum_{j=1}^{\infty} \langle u, \varphi_j \rangle \varphi_j.$$

Thus

$$\mathcal{H}^s = \left\{ u : D \rightarrow \mathbb{R} \mid \|w\|_{\mathcal{H}^s}^2 < \infty \right\} \quad (10.104)$$

where, for $u_j = \langle u, \varphi_j \rangle$,

$$\|u\|_{\mathcal{H}^s}^2 = \sum_{j=1}^{\infty} j^{\frac{2s}{d}} |u_j|^2.$$

In fact \mathcal{H}^s is a Hilbert space: for $v_j = \langle v, \varphi_j \rangle$ we may define the inner product

$$\langle u, v \rangle_{\mathcal{H}^s} = \sum_{j=1}^{\infty} j^{\frac{2s}{d}} u_j v_j.$$

For any $s > 0$, the Hilbert space $(\mathcal{H}^s, \langle \cdot, \cdot \rangle_{\mathcal{H}^s}, \|\cdot\|_{\mathcal{H}^s})$ is a subset of the original Hilbert space H ; for $s < 0$ the spaces are defined by duality and are supersets of H . Note also that we have Parseval-like identities showing that the \mathcal{H}^s norm on a function u is equivalent to the ℓ_w^2 norm on the sequence $\{u_j\}_{j=1}^{\infty}$ with the choice $w_j = j^{2s/d}$. The spaces \mathcal{H}^s are separable Hilbert spaces for any $s \in \mathbb{R}$.

A.1.4 Other Useful Function Spaces

As mentioned in passing, all of the preceding function spaces can be extended to functions taking values in $\mathbb{R}^n, \mathbb{R}^{n \times n}$; thus, we may then write $C(D; \mathbb{R}^n), L^p(D; \mathbb{R}^n)$, and $H^s(D; \mathbb{R}^n)$, for example. More generally we may wish to consider functions taking values in a separable Banach space E . For example, when we are interested in solutions of time-dependent PDEs, then these may be formulated as ordinary differential equations taking values in a separable Banach space E , with norm $\|\cdot\|_E$. It is then natural to consider Banach spaces such as $L^2((0, T); E)$ and $C([0, T]; E)$ with norms

$$\|u\|_{L^2((0, T); E)} = \sqrt{\left(\int_0^T \|u(\cdot, t)\|_E^2 dt \right)}, \quad \|u\|_{C([0, T]; E)} = \sup_{t \in [0, T]} \|u(\cdot, t)\|_E.$$

These norms can be generalized in a variety of ways, by generalizing the norm on the time variable.

The preceding idea of defining Banach space-valued L^p spaces defined on an interval $(0, T)$ can be taken further to define Banach space-valued L^p spaces defined on a measure space. Let (\mathcal{M}, ν) any countably generated measure space, like, for example, any Polish space (a separable completely metrizable topological space) equipped with a positive Radon measure ν . Again let E denote a separable Banach space. Then $L_v^p(\mathcal{M}; E)$ is the space of functions $u : \mathcal{M} \rightarrow E$ with norm (in this definition of norm we use Bochner integration, defined in the next subsection)

$$\|u\|_{L_v^p(\mathcal{M}; E)} = \left(\int_{\mathcal{M}} \|u(x)\|_E^p \nu(dx) \right)^{\frac{1}{p}}.$$

For $p \in (1, \infty)$ these spaces are separable. However, separability fails to hold for $p = \infty$. We will use these Banach spaces in the case where ν is a probability measure \mathbb{P} , with corresponding expectation \mathbb{E} , and we then have

$$\|u\|_{L_\nu^p(\mathcal{M}; E)} = \left(\mathbb{E}(\|u\|_E^p) \right)^{\frac{1}{p}}.$$

A.1.5 Interpolation Inequalities and Sobolev Embeddings

Here we state some useful interpolation inequalities and use them to prove a Sobolev embedding result, all in the context of fractional Sobolev spaces, in the generalized sense defined through a Hilbert scale of functions.

Let $p, q \in [1, \infty]$ be a pair of conjugate exponents so that $p^{-1} + q^{-1} = 1$. Then for any positive real a, b , we have the Young inequality

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

As a corollary of this elementary bound, we obtain the following Hölder inequality. Let (\mathcal{M}, μ) be a measure space and denote the norm $\|\cdot\|_{L_\nu^p(\mathcal{M}; \mathbb{R})}$ by $\|\cdot\|_p$. For $p, q \in [1, \infty]$ as above and $u, v: \mathcal{M} \rightarrow \mathbb{R}$ a pair of measurable functions, we have

$$\int_{\mathcal{M}} |u(x)v(x)| \mu(dx) \leq \|u\|_p \|v\|_q. \quad (10.105)$$

From this Hölder-like inequality, the following interpolation bound results: let $\alpha \in [0, 1]$ and let L denote a (possibly unbounded) self-adjoint operator on the Hilbert space $(H, \langle \cdot, \cdot \rangle, \|\cdot\|)$. Then, the bound

$$\|L^\alpha u\| \leq \|Lu\|^\alpha \|u\|^{1-\alpha} \quad (10.106)$$

holds for every $u \in \mathcal{D}(L) \subset H$.

Now assume that A is a self-adjoint unbounded operator on $L^2(D)$ with $D \subset \mathbb{R}^d$ a bounded open set with Lipschitz boundary. Assume further that A has eigenvalues $\alpha_j \asymp j^{\frac{2}{d}}$ and define the Hilbert scale of spaces $\mathcal{H}' = \mathcal{D}(A^{\frac{1}{2}})$. An immediate corollary of the bound (10.106), obtained by choosing $H = \mathcal{H}^s$, $L = A^{\frac{t-s}{2}}$, and $\alpha = (r-s)/(t-s)$, is:

Lemma 15. *Let Assumption 1 hold. Then for any $t > s$, any $r \in [s, t]$ and any $u \in \mathcal{H}'$, it follows that*

$$\|u\|_{\mathcal{H}'}^{t-s} \leq \|u\|_{\mathcal{H}'}^{r-s} \|u\|_{\mathcal{H}^s}^{t-r}.$$

It is of interest to bound the L^p norm of a function in terms of one of the fractional Sobolev norms, or more generally in terms of norms from a Hilbert scale. To do this we need to not only make assumptions on the eigenvalues of the operator

A which defines the Hilbert scale, but also on the behavior of the corresponding orthonormal basis of eigenfunctions in L^∞ . To this end we let Assumption 2 hold. It then turns out that bounding the L^∞ norm is rather straightforward and we start with this case.

Lemma 16. *Let Assumption 2 hold and define the resulting Hilbert scale of spaces \mathcal{H}^s by (10.104). Then for every $s > \frac{d}{2}$, the space \mathcal{H}^s is contained in the space $L^\infty(D)$ and there exists a constant K_1 such that $\|u\|_{L^\infty} \leq K_1 \|u\|_{\mathcal{H}^s}$.*

Proof. It follows from Cauchy-Schwarz that

$$\frac{1}{C} \|u\|_{L^\infty} \leq \sum_{k \in \mathbb{Z}^d} |u_k| \leq \left(\sum_{k \in \mathbb{Z}^d} (1 + |k|^2)^s |u_k|^2 \right)^{1/2} \left(\sum_{k \in \mathbb{Z}^d} (1 + |k|^2)^{-s} \right)^{1/2}.$$

Since the sum in the second factor converges if and only if $s > \frac{d}{2}$, the claim follows.

As a consequence of Lemma 16, we are able to obtain a more general Sobolev embedding for all L^p spaces:

Theorem 28 (Sobolev Embeddings). *Let Assumption 2 hold, define the resulting Hilbert scale of spaces \mathcal{H}^s by (10.104) and assume that $p \in [2, \infty]$. Then, for every $s > \frac{d}{2} - \frac{d}{p}$, the space \mathcal{H}^s is contained in the space $L^p(D)$, and there exists a constant K_2 such that $\|u\|_{L^p} \leq K_2 \|u\|_{\mathcal{H}^s}$.*

Proof. The case $p = 2$ is obvious and the case $p = \infty$ has already been shown, so it remains to show the claim for $p \in (2, \infty)$. The idea is to divide the space of eigenfunctions into ‘‘blocks’’ and to estimate separately the L^p norm of every block. More precisely, we define a sequence of functions $u^{(n)}$ by

$$u^{(-1)} = u_0 \varphi_0, \quad u^{(n)} = \sum_{2^n \leq j < 2^{n+1}} u_j \varphi_j,$$

where the φ_j are an orthonormal basis of eigenfunctions for A , so that $u = \sum_{n \geq -1} u^{(n)}$. For $n \geq 0$ the Hölder inequality gives

$$\|u^{(n)}\|_{L^p}^p \leq \|u^{(n)}\|_{L^2}^2 \|u^{(n)}\|_{L^\infty}^{p-2}. \quad (10.107)$$

Now set $s' = \frac{d}{2} + \epsilon$ for some $\epsilon > 0$ and note that the construction of $u^{(n)}$, together with Lemma 16, gives the bounds

$$\|u^{(n)}\|_{L^2} \leq K 2^{-ns/d} \|u^{(n)}\|_{\mathcal{H}^s}, \quad \|u^{(n)}\|_{L^\infty} \leq K_1 \|u^{(n)}\|_{\mathcal{H}^{s'}} \leq K 2^{n(s'-s)/d} \|u^{(n)}\|_{\mathcal{H}^s}. \quad (10.108)$$

Inserting this into (10.107), we obtain (possibly for an enlarged K)

$$\begin{aligned}\|u^{(n)}\|_{L^p} &\leq K \|u^{(n)}\|_{\mathcal{H}^s} 2^{n(\frac{(s'-s)\frac{p-2}{p}-\frac{2s}{p}}{d})/d} = K \|u^{(n)}\|_{\mathcal{H}^s} 2^{n(\epsilon \frac{p-2}{p} + \frac{d}{2} - \frac{d}{p} - s)/d} \\ &\leq K \|u\|_{\mathcal{H}^s} 2^{n(\epsilon + \frac{d}{2} - \frac{d}{p} - s)/d}.\end{aligned}$$

It follows that $\|u\|_{L^p} \leq |u_0| + \sum_{n \geq 0} \|u^{(n)}\|_{L^p} \leq K_2 \|u\|_{\mathcal{H}^s}$, provided that the exponent appearing in this expression is negative which, since ϵ can be chosen arbitrarily small, is precisely the case whenever $s > \frac{d}{2} - \frac{d}{p}$.

A.2 Probability and Integration In Infinite Dimensions

A.2.1 Product Measure for i.i.d. Sequences

Perhaps the most straightforward setting in which probability measures in infinite dimensions are encountered is when studying i.i.d. sequences of real-valued random variables. Furthermore, this is our basic building block for the construction of random functions – see Sect. 2.1 – so we briefly overview the subject. Let \mathbb{P}_0 be a probability measure on \mathbb{R} so that $(\mathbb{R}, \mathcal{B}(\mathbb{R}), \mathbb{P}_0)$ is a probability space and consider the i.i.d. sequence $\xi := \{\xi_j\}_{j=1}^\infty$ with $\xi_1 \sim \mathbb{P}_0$.

The construction of such a sequence can be formalised as follows. We consider ξ as a random variable taking values in the space \mathbb{R}^∞ endowed with the product topology, i.e. the smallest topology for which the projection maps $\ell_n : \xi \mapsto \xi_n$ are continuous for every n . This is a complete metric space; an example of a distance generating the product topology is given by

$$d(x, y) = \sum_{n=1}^{\infty} 2^{-n} \frac{|x_n - y_n|}{1 + |x_n - y_n|}.$$

Since we are considering a *countable* product, the resulting σ -algebra $\mathcal{B}(\mathbb{R}^\infty)$ coincides with the product σ -algebra, which is the smallest σ -algebra for which all ℓ_n 's are measurable.

In what follows we need the notion of the *pushforward* of a probability measure under a measurable map. If $f : B_1 \rightarrow B_2$ is a measurable map between two measurable spaces $(B_i, \mathcal{B}(B_i))$ $i = 1, 2$ and μ_1 is a probability measure on B_1 , then $\mu_2 = f^\# \mu_1$ denotes the pushforward probability measure on B_2 defined by $\mu_2(A) = \mu_1(f^{-1}(A))$ for all $A \in \mathcal{B}(B_2)$. (The notation $f^* \mu$ is sometimes used in place of $f^\# \mu$, but we reserve this notation for adjoints.) Recall that in Sect. 2, we construct random functions via the random series (10.11) whose coefficients are constructed from an i.i.d sequence. Our interest is in studying the pushforward measure $\mathcal{F}^\# \mathbb{P}_0$ where $\mathcal{F} : \mathbb{R}^\infty \rightarrow X'$ is defined by

$$\mathcal{F}\xi = m_0 + \sum_{j=1}^{\infty} \gamma_j \xi_j \phi_j. \quad (10.109)$$

In particular Sect. 2 is devoted to determining suitable separable Banach spaces X' on which to define the pushforward measure.

With the pushforward notation at hand, we may also describe Kolmogorov's extension theorem which can be stated as follows.

Theorem 29 (Kolmogorov Extension). *Let X be a Polish space and let I be an arbitrary set. Assume that, for any finite subset $A \subset I$, we are given a probability measure \mathbb{P}_A on the finite product space X^A . Assume furthermore that the family of measures $\{\mathbb{P}_A\}$ is consistent in the sense that if $B \subset A$ and $\Pi_{A,B}: X^A \rightarrow X^B$ denotes the natural projection map, then $\Pi_{A,B}^\sharp \mathbb{P}_A = \mathbb{P}_B$. Then, there exists a unique probability measure \mathbb{P} on X^I endowed with the product σ -algebra with the property that $\Pi_{I,A}^\sharp \mathbb{P} = \mathbb{P}_A$ for every finite subset $A \subset I$.*

Loosely speaking, one can interpret this theorem as stating that if one knows the law of any *finite* number of components of a random vector or function, then this determines the law of the *whole* random vector or function; in particular, in the case of the random function, this comprises *uncountably* many components. This statement is thus highly nontrivial as soon as the set I is infinite since we have a priori defined \mathbb{P}_A only for finite subsets $A \subset I$, and the theorem allows us to extend this uniquely also to infinite subsets.

As a simple application, we can use this theorem to define the infinite product measure $\mathbb{P} = \bigotimes_{k=1}^\infty \mathbb{P}_0$ as the measure given by Kolmogorov's Extension Theorem 29 if we take as our family of specifications $\mathbb{P}_A = \bigotimes_{k \in A} \mathbb{P}_0$. Our i.i.d. sequence ξ is then naturally defined as a random sample taken from the probability space $(\mathbb{R}^\infty, \mathcal{B}(\mathbb{R}^\infty), \mathbb{P})$. A more complicated example follows from making sense of the random field perspective on random functions as explained in Sect. 2.5.

A.2.2 Probability and Integration on Separable Banach Spaces

We now study probability and integration on separable Banach spaces B ; we let B^* denote the dual space of bounded linear functionals on B . The assumption of separability rules out some important function spaces like $L^\infty(D; \mathbb{R})$, but is required in order for the basic results of integration theory to hold. This is because, when considering a non-separable Banach space B , it is not clear what the “natural” σ -algebra on B is. One natural candidate is the Borel σ -algebra, denoted $\mathcal{B}(B)$, namely, the smallest σ -algebra containing all open sets; another is the cylindrical σ -algebra, namely, the smallest σ -algebra for which all bounded linear functionals on B are measurable. For i.i.d. sequences, the analogues of these two σ -algebras can be identified, whereas, in the general setting, the cylindrical σ -algebra can be strictly smaller than the Borel σ -algebra. In the case of separable Banach spaces, however, both σ -algebras agree:

Lemma 17. *Let B be a separable Banach space and let μ and ν be two Borel probability measures on B . If $\ell^\sharp \mu = \ell^\sharp \nu$ for every $\ell \in B^*$, then $\mu = \nu$.*

Thus, as for i.i.d. sequences, there is therefore a canonical notion of measurability. Whenever we refer to (probability) measures on a separable Banach space B in the sequel, we really mean (probability) measures on $(B, \mathcal{B}(B))$.

We now turn to the definition of integration with respect to probability measures on B . Given a (Borel) measurable function $f: \Omega \rightarrow B$ where $(\Omega, \mathcal{F}, \mathbb{P})$ is a standard probability space, we say that f is integrable with respect to \mathbb{P} if the map $\omega \mapsto \|f(\omega)\|$ belongs to $L^1_{\mathbb{P}}(\Omega; \mathbb{R})$. (Note that this map is certainly Borel measurable since the norm $\|\cdot\|: B \rightarrow \mathbb{R}$ is a continuous, and therefore also Borel measurable, function.) Given such an integrable function f , we *define* its Bochner integral by

$$\int f(\omega) \mathbb{P}(d\omega) = \lim_{n \rightarrow \infty} \int f_n(\omega) \mathbb{P}(d\omega),$$

where f_n is a sequence of simple functions, for which the integral on the right-hand side may be defined in the usual way, chosen such that

$$\lim_{n \rightarrow \infty} \int \|f_n(\omega) - f(\omega)\| \mathbb{P}(d\omega) = 0.$$

With this definition the value of the integral does not depend on the approximating sequence, it is linear in f , and

$$\int \ell(f(\omega)) \mathbb{P}(d\omega) = \ell\left(\int f(\omega) \mathbb{P}(d\omega)\right), \quad (10.110)$$

for every element ℓ in the dual space B^* .

Given a probability measure μ on a separable Banach space B , we now say that μ has *finite expectation* if the identity function $x \mapsto x$ is integrable with respect to μ . If this is the case, we define the expectation of μ as

$$\int_B x \mu(dx),$$

where the integral is interpreted as a Bochner integral.

Similarly, it is natural to say that μ has *finite variance* if the map $x \mapsto \|x\|^2$ is integrable with respect to μ . Regarding the covariance C_μ of μ itself, it is natural to define it as a bounded linear operator $C_\mu: B^* \rightarrow B$ with the property that

$$C_\mu \ell = \int_B x \ell(x) \mu(dx), \quad (10.111)$$

for every $\ell \in B^*$. At this stage, however, it is not clear whether such an operator C_μ always exists solely under the assumption that μ has finite variance. For any $x \in B$, we define the projection operator $P_x: B^* \rightarrow B$ by

$$P_x \ell = x \ell(x), \quad (10.112)$$

suggesting that we define

$$C_\mu := \int_B P_x \mu(dx). \quad (10.113)$$

The problem with this definition is that if we view the map $x \mapsto P_x$ as a map taking values in the space $\mathcal{L}(B^*, B)$ of bounded linear operators from $B^* \rightarrow B$, then, since this space is not separable in general, it is not clear a priori whether (10.113) makes sense as a Bochner integral. This suggests to define the subspace $B_*(B) \subset \mathcal{L}(B^*, B)$ given by the closure (in the usual operator norm) of the linear span of operators of the type P_x given in (10.112) for $x \in B$. We then have:

Lemma 18. *If B is separable, then $B_*(B)$ is also separable. Furthermore, $B_*(B)$ consists of compact operators.*

This leads to the following corollary:

Corollary 3. *Assume that μ has finite variance so that the map $x \mapsto \|x\|^2$ is integrable with respect to μ . Then the covariance operator C_μ defined by (10.113) exists as a Bochner integral in $B_*(B)$.*

Remark 3. Once the covariance is defined, the fact that (10.111) holds is then an immediate consequence of (10.110). In general, not every element $C \in B_*(B)$ can be realised as the covariance of some probability measure. This is the case even if we impose the positivity condition $\ell(C\ell) \geq 0$, which by (10.111) is a condition satisfied by every covariance operator. For further insight into this issue, see Lemma 23 which characterizes precisely the covariance operators of a Gaussian measure in separable Hilbert space. \square

Given any probability measure μ on B , we can define its *Fourier transform* $\hat{\mu}: B^* \rightarrow \mathbb{C}$ by

$$\hat{\mu}(\ell) := \int_B e^{i\ell(x)} \mu(dx). \quad (10.114)$$

For a Gaussian measure μ_0 on B with mean a and covariance operator C , it may be shown that, for any $\ell \in B^*$, the characteristic function is given by

$$\hat{\mu}_0(\ell) = e^{i\ell(a) - \frac{1}{2}\ell(C\ell)}. \quad (10.115)$$

As a consequence of Lemma 17, it is almost immediate that a measure is uniquely determined by its Fourier transform, and this is the content of the following result.

Lemma 19. *Let μ and ν be any two probability measures on a separable Banach space B . If $\hat{\mu}(\ell) = \hat{\nu}(\ell)$ for every $\ell \in B^*$, then $\mu = \nu$.*

A.2.3 Probability and Integration on Separable Hilbert Spaces

We will frequently be interested in the case where $B = \mathcal{H}$ for $(\mathcal{H}, \langle \cdot, \cdot \rangle, \|\cdot\|)$ some separable Hilbert space. Bochner integration can then, of course, be defined as a special case of the preceding development on separable Banach spaces. We make use of the Riesz representation theorem to identify \mathcal{H} with its dual and $\mathcal{H} \otimes \mathcal{H}$ with a subspace of the space of linear operators on \mathcal{H} . The covariance operator of a measure μ on \mathcal{H} may then be viewed as a bounded linear operator from \mathcal{H} into itself. The definition (10.111) of C_μ becomes

$$C_\mu \ell = \int_{\mathcal{H}} \langle \ell, x \rangle x \mu(dx), \quad (10.116)$$

for all $\ell \in \mathcal{H}$ and (10.113) becomes

$$C_\mu = \int_{\mathcal{H}} x \otimes x \mu(dx). \quad (10.117)$$

Corollary 3 shows that we can indeed make sense of the second formulation as a Bochner integral, provided that μ has finite variance in \mathcal{H} .

A.2.4 Metrics on Probability Measures

When discussing well posedness and approximation theory for the posterior distribution, it is of interest to estimate the distance between two probability measures, and thus we will be interested in metrics between probability measures. In this subsection we introduce two useful metrics on measures: the *total variation distance* and the *Hellinger distance*. We discuss the relationships between the metrics and indicate how they may be used to estimate differences between expectations of random variables under two different measures. We also discuss the *Kullback-Leibler divergence*, a useful distance measure which does not satisfy the axioms of a metric, but which may be used to bound both the Hellinger and total variation distances, and which is also useful in defining algorithms for finding the best approximation to a given measure from within some restricted class of measures, such as Gaussians.

Assume that we have two probability measures μ and μ' on a separable Banach space denoted by B (actually the considerations here apply on a Polish space but we do not need this level of generality). Assume that μ and μ' are both absolutely continuous with respect to a common reference measure ν , also defined on the same measure space. Such a measure always exists – take $\nu = \frac{1}{2}(\mu + \mu')$, for example. In the following, all integrals of real-valued functions over B are simply denoted by \int . The following define two concepts of distance between μ and μ' . The resulting metrics that we define are independent of the choice of this common reference measure.

Definition 3. The *total variation distance* between μ and μ' is

$$d_{\text{TV}}(\mu, \mu') = \frac{1}{2} \int \left| \frac{d\mu}{dv} - \frac{d\mu'}{dv} \right| dv. \quad \square$$

In particular, if μ' is absolutely continuous with respect to μ , then

$$d_{\text{TV}}(\mu, \mu') = \frac{1}{2} \int \left| 1 - \frac{d\mu'}{d\mu} \right| d\mu. \quad (10.118)$$

Definition 4. The *Hellinger distance* between μ and μ' is

$$d_{\text{Hell}}(\mu, \mu') = \sqrt{\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{dv}} - \sqrt{\frac{d\mu'}{dv}} \right)^2 dv}. \quad \square$$

In particular, if μ' is absolutely continuous with respect to μ , then

$$d_{\text{Hell}}(\mu, \mu') = \sqrt{\frac{1}{2} \int \left(1 - \sqrt{\frac{d\mu'}{d\mu}} \right)^2 d\mu}. \quad (10.119)$$

Note that the numerical constant $\frac{1}{2}$ appearing in both definitions is chosen in such a way as to ensure the bounds

$$0 \leq d_{\text{TV}}(\mu, \mu') \leq 1, \quad 0 \leq d_{\text{Hell}}(\mu, \mu') \leq 1.$$

In the case of the total variation inequality, this is an immediate consequence of the triangle inequality, combined with the fact that both μ and μ' are probability measures, so that $\int \frac{d\mu}{dv} dv = 1$ and similarly for μ' . In the case of the Hellinger distance, it follows by expanding the square and applying similar considerations.

The Hellinger and total variation distances are related as follows, which shows in particular that they both generate the same topology:

Lemma 20. *The total variation and Hellinger metrics are related by the inequalities*

$$\frac{1}{\sqrt{2}} d_{\text{TV}}(\mu, \mu') \leq d_{\text{Hell}}(\mu, \mu') \leq d_{\text{TV}}(\mu, \mu')^{\frac{1}{2}}.$$

Proof. We have

$$\begin{aligned}
d_{\text{TV}}(\mu, \mu') &= \frac{1}{2} \int \left| \sqrt{\frac{d\mu}{dv}} - \sqrt{\frac{d\mu'}{dv}} \right| \left| \sqrt{\frac{d\mu}{dv}} + \sqrt{\frac{d\mu'}{dv}} \right| dv \\
&\leq \sqrt{\left(\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{dv}} - \sqrt{\frac{d\mu'}{dv}} \right)^2 dv \right)} \sqrt{\left(\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{dv}} + \sqrt{\frac{d\mu'}{dv}} \right)^2 dv \right)} \\
&\leq \sqrt{\left(\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{dv}} - \sqrt{\frac{d\mu'}{dv}} \right)^2 dv \right)} \sqrt{\left(\int \left(\frac{d\mu}{dv} + \frac{d\mu'}{dv} \right) dv \right)} \\
&= \sqrt{2} d_{\text{Hell}}(\mu, \mu')
\end{aligned}$$

as required for the first bound.

For the second bound note that, for any positive a and b , one has the bound $|\sqrt{a} - \sqrt{b}| \leq \sqrt{a} + \sqrt{b}$. As a consequence, we have the bound

$$\begin{aligned}
d_{\text{Hell}}(\mu, \mu')^2 &\leq \frac{1}{2} \int \left| \sqrt{\frac{d\mu}{dv}} - \sqrt{\frac{d\mu'}{dv}} \right| \left| \sqrt{\frac{d\mu}{dv}} + \sqrt{\frac{d\mu'}{dv}} \right| dv \\
&= \frac{1}{2} \int \left| \frac{d\mu}{dv} - \frac{d\mu'}{dv} \right| dv \\
&= d_{\text{TV}}(\mu, \mu'),
\end{aligned}$$

as required.

Example 11. Consider two Gaussian densities on \mathbb{R} : $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$. The Hellinger distance between them is given by

$$d_{\text{Hell}}(\mu, \mu')^2 = 1 - \sqrt{\exp\left(-\frac{(m_1 - m_2)^2}{2(\sigma_1^2 + \sigma_2^2)}\right) \frac{2\sigma_1\sigma_2}{(\sigma_1^2 + \sigma_2^2)}}.$$

To see this note that

$$d_{\text{Hell}}(\mu, \mu')^2 = 1 - \frac{1}{(2\pi\sigma_1\sigma_2)^{\frac{1}{2}}} \int_{\mathbb{R}} \exp(-Q) dx$$

where

$$Q = \frac{1}{4\sigma_1^2}(x - m_1)^2 + \frac{1}{4\sigma_2^2}(x - m_2)^2.$$

Define σ^2 by

$$\frac{1}{2\sigma^2} = \frac{1}{4\sigma_1^2} + \frac{1}{4\sigma_2^2}.$$

We change variable under the integral to y given by

$$y = x - \frac{m_1 + m_2}{2}$$

and note that then, by completing the square,

$$Q = \frac{1}{2\sigma^2}(y - m)^2 + \frac{1}{4(\sigma_1^2 + \sigma_2^2)}(m_2 - m_1)^2$$

where m does not appear in what follows and so we do not detail it. Noting that the integral is then a multiple of a standard Gaussian $N(m, \sigma^2)$ gives the desired result. In particular this calculation shows that the Hellinger distance between two Gaussians on \mathbb{R} tends to zero if and only if the means and variances of the two Gaussians approach one another. Furthermore, by the previous lemma, the same is true for the total variation distance. \square

The preceding example generalizes to higher dimension and shows that, for example, the total variation and Hellinger metrics cannot metrize weak convergence of probability measures (as one can also show that convergence in total variation metric implies strong convergence). They are nonetheless useful distance measures, for example, between families of measures which are mutually absolutely continuous. Furthermore, the Hellinger distance is particularly useful for estimating the difference between expectation values of functions of random variables under different measures. This is encapsulated in the following lemma:

Lemma 21. *Let μ and μ' be two probability measures on a separable Banach space X . Assume also that $f : X \rightarrow E$, where $(E, \|\cdot\|)$ is a separable Banach space, is measurable and has second moments with respect to both μ and μ' . Then*

$$\|\mathbb{E}^\mu f - \mathbb{E}^{\mu'} f\| \leq 2\left(\mathbb{E}^\mu \|f\|^2 + \mathbb{E}^{\mu'} \|f\|^2\right)^{\frac{1}{2}} d_{\text{Hell}}(\mu, \mu').$$

Furthermore, if E is a separable Hilbert space and $f : X \rightarrow E$ as before has fourth moments, then

$$\|\mathbb{E}^\mu(f \otimes f) - \mathbb{E}^{\mu'}(f \otimes f)\| \leq 2\left(\mathbb{E}^\mu \|f\|^4 + \mathbb{E}^{\mu'} \|f\|^4\right)^{\frac{1}{2}} d_{\text{Hell}}(\mu, \mu').$$

Proof. Let ν be a reference probability measure as above. We then have the bound

$$\begin{aligned} \|\mathbb{E}^\mu f - \mathbb{E}^{\mu'} f\| &\leq \int \|f\| \left| \frac{d\mu}{d\nu} - \frac{d\mu'}{d\nu} \right| d\nu \\ &= \int \left(\frac{1}{\sqrt{2}} \left| \sqrt{\frac{d\mu}{d\nu}} - \sqrt{\frac{d\mu'}{d\nu}} \right| \right) \left(\sqrt{2} \|f\| \left| \sqrt{\frac{d\mu}{d\nu}} + \sqrt{\frac{d\mu'}{d\nu}} \right| \right) d\nu \\ &\leq \sqrt{\left(\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{d\nu}} - \sqrt{\frac{d\mu'}{d\nu}} \right)^2 d\nu \right)} \sqrt{\left(2 \int \|f\|^2 \left(\sqrt{\frac{d\mu}{d\nu}} + \sqrt{\frac{d\mu'}{d\nu}} \right)^2 d\nu \right)} \\ &\leq \sqrt{\left(\frac{1}{2} \int \left(\sqrt{\frac{d\mu}{d\nu}} - \sqrt{\frac{d\mu'}{d\nu}} \right)^2 d\nu \right)} \sqrt{\left(4 \int \|f\|^2 \left(\frac{d\mu}{d\nu} + \frac{d\mu'}{d\nu} \right) d\nu \right)} \\ &= 2 \left(\mathbb{E}^\mu \|f\|^2 + \mathbb{E}^{\mu'} \|f\|^2 \right)^{\frac{1}{2}} d_{\text{Hell}}(\mu, \mu') \end{aligned}$$

as required.

The proof for $f \otimes f$ follows from the bound

$$\begin{aligned} \|\mathbb{E}^\mu(f \otimes f) - \mathbb{E}^{\mu'}(f \otimes f)\| &= \sup_{\|h\|=1} \|\mathbb{E}^\mu \langle f, h \rangle f - \mathbb{E}^{\mu'} \langle f, h \rangle f\| \\ &\leq \int \|f\|^2 \left| \frac{d\mu}{d\nu} - \frac{d\mu'}{d\nu} \right| d\nu, \end{aligned}$$

and then arguing similarly to the first case but with $\|f\|$ replaced by $\|f\|^2$.

Remark 4. Note, in particular, that choosing $X = E$, and with f chosen to be the identity mapping, we deduce that the differences between the mean (respectively, covariance operator) of two measures are bounded above by their Hellinger distance, provided that one has some a priori control on the second (respectively, fourth) moments. \square

We now define a third widely used distance concept for comparing two probability measures. Note, however, that it does not give rise to a metric in the strict sense, because it violates both symmetry and the triangle inequality.

Definition 5. The Kullback-Leibler divergence between two measures μ' and μ , with μ' absolutely continuous with respect to μ , is

$$D_{\text{KL}}(\mu' \parallel \mu) = \int \frac{d\mu'}{d\mu} \log \left(\frac{d\mu'}{d\mu} \right) d\mu. \quad \square$$

If μ is also absolutely continuous with respect to μ' , so that the two measures are equivalent, then

$$D_{\text{KL}}(\mu' \parallel \mu) = - \int \log\left(\frac{d\mu}{d\mu'}\right) d\mu'$$

and the two definitions coincide.

Example 12. Consider two Gaussian densities on \mathbb{R} : $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$. The Kullback-Leibler divergence between them is given by

$$D_{\text{KL}}(\mu_1 \parallel \mu_2) = \ln\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - 1\right) + \frac{(m_2 - m_1)^2}{2\sigma_2^2}.$$

To see this note that

$$\begin{aligned} D_{\text{KL}}(\mu_1 \parallel \mu_2) &= \mathbb{E}^{\mu_1} \left(\ln \sqrt{\frac{\sigma_2^2}{\sigma_1^2}} + \frac{1}{2\sigma_2^2} |x - m_2|^2 - \frac{1}{2\sigma_1^2} |x - m_1|^2 \right) \\ &= \ln \frac{\sigma_2}{\sigma_1} + \mathbb{E}^{\mu_1} \left(\left(\frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2} \right) |x - m_1|^2 \right) \\ &\quad + \mathbb{E}^{\mu_1} \frac{1}{2\sigma_2^2} \left(|x - m_2|^2 - |x - m_1|^2 \right) \\ &= \ln \frac{\sigma_2}{\sigma_1} + \frac{1}{2} \left(\frac{\sigma_1^2}{\sigma_2^2} - 1 \right) + \frac{1}{2\sigma_2^2} \mathbb{E}^{\mu_1} \left(m_2^2 - m_1^2 + 2x(m_1 - m_2) \right) \\ &= \ln \frac{\sigma_2}{\sigma_1} + \frac{1}{2} \left(\frac{\sigma_1^2}{\sigma_2^2} - 1 \right) + \frac{1}{2\sigma_2^2} (m_2 - m_1)^2 \end{aligned}$$

as required. \square

As for Hellinger distance, this example shows that two Gaussians on \mathbb{R} approach one another in the Kullback-Leibler divergence if and only if their means and variances approach one another. This generalizes to higher dimensions. The Kullback-Leibler divergence provides an upper bound for the square of the Hellinger distance and for the square of the total variation distance.

Lemma 22. *Assume that two measures μ and μ' are equivalent. Then the bounds*

$$d_{\text{Hell}}(\mu, \mu')^2 \leq \frac{1}{2} D_{\text{KL}}(\mu \parallel \mu'), \quad d_{\text{TV}}(\mu, \mu')^2 \leq D_{\text{KL}}(\mu \parallel \mu'),$$

hold.

Proof. The second bound follows from the first by using Lemma 20, thus it suffices to proof the first. In the following we use the fact that

$$x - 1 \geq \log(x) \quad \forall x \geq 0,$$

so that

$$\sqrt{x} - 1 \geq \frac{1}{2} \log(x) \quad \forall x \geq 0.$$

This yields the bound

$$\begin{aligned} d_{\text{Hell}}(\mu, \mu')^2 &= \frac{1}{2} \int \left(\sqrt{\frac{d\mu'}{d\mu}} - 1 \right)^2 d\mu = \frac{1}{2} \int \left(\frac{d\mu'}{d\mu} + 1 - 2\sqrt{\frac{d\mu'}{d\mu}} \right) d\mu \\ &= \int \left(1 - \sqrt{\frac{d\mu'}{d\mu}} \right) d\mu \leq \frac{1}{2} \int \left(-\log \frac{d\mu'}{d\mu} \right) d\mu \\ &= \frac{1}{2} D_{\text{KL}}(\mu || \mu'), \end{aligned}$$

as required.

A.2.5 Kolmogorov Continuity Test

The setting of Kolmogorov's continuity test is the following. We assume that we are given a compact domain $D \subset \mathbb{R}^d$, a complete separable metric space X , as well as a collection of X -valued random variables $u : x \in D \mapsto X$. At this stage we assume no regularity whatsoever on the parameter x : the distribution of this collection of random variables is a measure μ_0 on the space X^D of all functions from D to X endowed with the product σ -algebra. Any consistent family of marginal distributions does yield such a measure by Kolmogorov's extension Theorem 29. With these notations at hand, Kolmogorov's continuity test can be formulated as follows and enables the extraction of regularity with respect to variation of $u(x)$ with respect to x .

Theorem 30 (Kolmogorov Continuity Test). *Let D and u be as above and assume that there exist $p > 1$, $\alpha > 0$ and $K > 0$ such that*

$$\mathbb{E} d(u(x), u(y))^p \leq K|x - y|^{p\alpha+d}, \quad \forall x, y \in D, \quad (10.120)$$

where d denotes the distance function on X and d the dimension of the compact domain D . Then, for every $\beta < \alpha$, there exists a unique measure μ on $C^{0,\beta}(D, X)$ such that the canonical process under μ has the same law as u .

We have here generalized the notion of Hölder spaces from Sect. A.1.2 to functions taking values in a Polish space; such generalizations are discussed in Sect. A.1.4. The notion of *canonical process* is defined in Sect. A.4.

We will frequently use Kolmogorov's continuity test in the following setting: we again assume that we are given a compact domain $D \subset \mathbb{R}^d$, and now a collection $u(x)$ of \mathbb{R}^n -valued random variables indexed by $x \in D$. We have the following:

Corollary 4. *Assume that there exist $p > 1$, $\alpha > 0$ and $K > 0$ such that*

$$\mathbb{E}|u(x) - u(y)|^p \leq K|x - y|^{p\alpha+d}, \quad \forall x, y \in D.$$

Then, for every $\beta < \alpha$, there exists a unique measure μ on $C^{0,\beta}(D)$ such that the canonical process under μ has the same law as u .

Remark 5. Recall that $C^{0,\gamma'}(D) \subset C_0^{0,\gamma}(D)$ for all $\gamma' > \gamma$ so that, since the interval $\beta < \alpha$ for this theorem is open, we may interpret the result as giving an equivalent measure defined on a separable Banach space.

A very useful consequence of Kolmogorov's continuity criterion is the following result. The setting is to consider a random function f given by the random series

$$u = \sum_{k \geq 0} \xi_k \psi_k \tag{10.121}$$

where $\{\xi_k\}_{k \geq 0}$ is an i.i.d. sequence and the ψ_k are real- or complex-valued Hölder functions on bounded open $D \subset \mathbb{R}^d$ satisfying, for some $\alpha \in (0, 1]$,

$$|\psi_k(x) - \psi_k(y)| \leq h(\alpha, \psi_k)|x - y|^\alpha \quad x, y \in D; \tag{10.122}$$

of course if $\alpha = 1$ the functions are Lipschitz.

Corollary 5. *Let $\{\xi_k\}_{k \geq 0}$ be countably many centred i.i.d. random variables (real or complex) with bounded moments of all orders. Moreover let $\{\psi_k\}_{k \geq 0}$ satisfy (10.122). Suppose there is some $\delta \in (0, 2)$ such that*

$$S_1 := \sum_{k \geq 0} \|\psi_k\|_{L^\infty}^2 < \infty \quad \text{and} \quad S_2 := \sum_{k \geq 0} \|\psi_k\|_{L^\infty}^{2-\delta} h(\alpha, \psi_k)^\delta < \infty. \tag{10.123}$$

Then u defined by (10.121) is almost surely finite for every $x \in D$, and u is Hölder continuous for every Hölder exponent smaller than $\alpha\delta/2$.

Proof. Let us denote by $\kappa_n(X)$ the n th cumulant of a random variable X . The odd cumulants of centred random variables are zero. Furthermore, using the fact that the

cumulants of independent random variables simply add up and that the cumulants of ξ_k are all finite by assumption, we obtain for $p \geq 1$ the bound

$$\begin{aligned} |\kappa_{2p}(u(x) - u(y))| &= \left| \sum_{k \geq 0} \kappa_{2p}(\xi_k) (\psi_k(x) - \psi_k(y))^{2p} \right| \\ &\lesssim C_p \sum_{k \geq 0} \min\{2^{2p} \|\psi_k\|_{L^\infty}^{2p}, h(\alpha, \psi_k)^{2p} |x - y|^{2p\alpha}\} \\ &\lesssim C_p \sum_{k \geq 0} \|\psi_k\|_{L^\infty}^{(1-\frac{\delta}{2})2p} h(\alpha, \psi_k)^{2p \cdot \frac{\delta}{2}} |x - y|^{2p\alpha \cdot \frac{\delta}{2}} \\ &\lesssim C_p |x - y|^{p\alpha\delta}, \end{aligned}$$

with C_p denoting positive constants depending on p which can change from occurrence to occurrence and where we used that $\min\{a, bx^2\} \leq a^{1-\delta/2} b^{\delta/2} |x|^\delta$ for any $a, b \geq 0$ and the finiteness of S_2 . In a similar way, we obtain $|\kappa_{2p}u(x)| < \infty$ for every $p \geq 1$. Since the random variables $u(x)$ are centred, all moments of even order $2p$, $p \geq 1$, can be expressed in terms of homogeneous polynomials of the even cumulants of order upto $2p$, so that

$$\mathbb{E}|u(x) - u(y)|^{2p} \lesssim C_p |x - y|^{p\alpha\delta}, \quad \mathbb{E}|u(x)|^{2p} < \infty,$$

uniformly over $x, y \in D$. The almost sure boundedness on L^∞ follows from the second bound. The Hölder continuity claim follows from Kolmogorov's continuity test in the form of Corollary 4, after noting that $p\alpha\delta = 2p\left(\frac{1}{2}\alpha\delta - \frac{d}{2p}\right) + d$ and choosing p arbitrarily large.

Remark 6. Note that (10.121) is simply a rewrite of (10.11), with $\psi_0 = m_0$, $\xi_0 = 1$ and $\psi_k = \gamma_k \phi_k$. In the case where the ξ_k are standard normal, then the ψ_k 's in Corollary 5 form an orthonormal basis of the Cameron-Martin space (see Definition 7) of a Gaussian measure. The criterion (10.123) then provides an effective way of showing that the measure in question can be realised on a space of Hölder continuous functions. \square

A.3 Gaussian Measures

A.3.1 Separable Banach Space Setting

We start with the definition of a Gaussian measure on a separable Banach space B . There is no equivalent to Lebesgue measure in infinite dimensions (as it could not be σ -additive), and so we cannot define a Gaussian measure by prescribing the form of its density. However, note that Gaussian measures on \mathbb{R}^n can be characterised by prescribing that the projections of the measure onto any one-dimensional subspace

of \mathbb{R}^n are all Gaussian. This is a property that can readily be generalised to infinite-dimensional spaces:

Definition 6. A *Gaussian probability measure* μ on a separable Banach space B is a Borel measure such that $\ell^\sharp\mu$ is a Gaussian probability measure on \mathbb{R} for every continuous linear functional $\ell: B \rightarrow \mathbb{R}$. (Here, Dirac measures are considered to be Gaussian measures with zero variance.) The measure is said to be *centred* if $\ell^\sharp\mu$ has mean zero for every ℓ . \square

This is a reasonable definition since, provided that B is separable, the one-dimensional projections of any probability measure carry sufficient information to characterise it – see Lemma 17. We now state an important result which controls the tails of Gaussian distributions:

Theorem 31 (Fernique). Let μ be a Gaussian probability measure on a separable Banach space B . Then, there exists $\alpha > 0$ such that $\int_B \exp(\alpha \|x\|^2) \mu(dx) < \infty$.

As a consequence of the Fernique theorem and the Corollary 3, every Gaussian measure μ admits a compact covariance operator C_μ given by (10.113), because the second moment is bounded. In fact the techniques used to prove the Fernique theorem show that, if $M = \int_B \|x\| \mu(dx)$, then there is a global constant $K > 0$ such that

$$\int_B \|x\|^{2n} \mu(dx) \leq n! K \alpha^{-n} M^{2n}. \quad (10.124)$$

Since the covariance operator, and hence the mean, exist for a Gaussian measure, and since they may be shown to characterize the measure completely, we write $N(m, C_\mu)$ for a Gaussian with mean m and covariance operator C_μ .

Measures in infinite-dimensional spaces are typically mutually singular. Furthermore, two Gaussian measures are either mutually singular or equivalent (mutually absolutely continuous). The Cameron-Martin space plays a key role in characterizing whether or not two Gaussians are equivalent.

Definition 7. The *Cameron-Martin space* \mathcal{H}_μ of measure μ on a separable Banach space B is the completion of the linear subspace $\mathring{\mathcal{H}}_\mu \subset B$ defined by

$$\mathring{\mathcal{H}}_\mu = \{h \in B : \exists h^* \in B^* \text{ with } h = C_\mu h^*\}, \quad (10.125)$$

under the norm $\|h\|_\mu^2 = \langle h, h \rangle_\mu = h^*(C_\mu h^*)$. It is a Hilbert space when endowed with the scalar product $\langle h, k \rangle_\mu = h^*(C_\mu k^*) = h^*(k) = k^*(h)$.

The Cameron-Martin space is actually independent of the space B in the sense that, although we may view the measure as living on a range of separable Hilbert or Banach spaces, the Cameron-Martin space will be the same in all cases. The space

characterizes exactly the directions in which a centred Gaussian measure may be shifted to obtain an equivalent Gaussian measure:

Theorem 32 (Cameron-Martin). *For $h \in B$, define the map $T_h: B \rightarrow B$ by $T_h(x) = x + h$. Then, the measure $T_h^\sharp\mu$ is absolutely continuous with respect to μ if and only if $h \in \mathcal{H}_\mu$. Furthermore, in the latter case, its Radon-Nikodym derivative is given by*

$$\frac{dT_h^\sharp\mu}{d\mu}(u) = \exp(h^*(u) - \frac{1}{2}\|h\|_\mu^2)$$

where $h = C_\mu h^*$.

Thus, this theorem characterizes the Radon-Nikodym derivative of the measure $N(h, C_\mu)$ with respect to the measure $N(0, C_\mu)$. Below, in the Hilbert space setting, we also consider changes in the covariance operator which lead to equivalent Gaussian measures. However, before moving to the Hilbert space setting, we conclude this subsection with several useful observations concerning Gaussians on separable Banach spaces. The *topological support* of measure μ on the separable Banach space B is the set of all $u \in B$ such that any neighborhood of u has a positive measure.

Theorem 33. *The topological support of a centred Gaussian measure μ on B is the closure of the Cameron-Martin space in B . Furthermore the Cameron-Martin space is dense in X . Therefore all balls in B have positive μ -measure.*

Since the Cameron-Martin space of Gaussian measure μ is independent of the space on which we view the measure as living, this following useful theorem shows that the unit ball in the Cameron-Martin space is compact in any separable Banach space X for which $\mu(X) = 1$:

Theorem 34. *The closed unit ball in the Cameron-Martin space \mathcal{H}_μ is compactly embedded into the separable Banach space B .*

In the setting of Gaussian measures on a separable Banach space, all balls have positive probability. The Cameron-Martin norm is useful in the characterization of small-ball properties of Gaussians. Let $B^\delta(z)$ denote a ball of radius δ in B centred at a point $z \in \mathcal{H}_\mu$.

Theorem 35. *The ratio of small ball probabilities under Gaussian measure μ satisfy*

$$\lim_{\delta \rightarrow 0} \frac{\mu(B^\delta(z_1))}{\mu(B^\delta(z_2))} = \exp\left(\frac{1}{2}\|z_2\|_\mu^2 - \frac{1}{2}\|z_1\|_\mu^2\right).$$

Example 13. Let μ denote the Gaussian measure $N(0, K)$ on \mathbb{R}^n with K positive definite. Then Theorem 35 is the statement that

$$\lim_{\delta \rightarrow 0} \frac{\mu(B^\delta(z_1))}{\mu(B^\delta(z_2))} = \exp\left(\frac{1}{2}|K^{-\frac{1}{2}}z_2|^2 - \frac{1}{2}|K^{-\frac{1}{2}}z_1|^2\right)$$

which follows directly from the fact that the Gaussian measure at point $z \in \mathbb{R}^n$ has Lebesgue density proportional to $\exp\left(-\frac{1}{2}|K^{-\frac{1}{2}}z|^2\right)$ and the fact that the Lebesgue density is a continuous function. \square

A.3.2 Separable Hilbert Space Setting

In these notes our approach is primarily based on defining Gaussian measures on Hilbert space; the Banach spaces which are of full measure under the Gaussian are then determined via the Kolmogorov continuity theorem. In this subsection we develop the theory of Gaussian measures in greater detail within the Hilbert space setting. Throughout $(\mathcal{H}, \langle \cdot, \cdot \rangle, \|\cdot\|)$ denotes the separable Hilbert space on which the Gaussian is constructed. Actually, in this Hilbert space setting, the covariance operator C_μ has considerably more structure than just the boundedness implied by (10.124): it is trace class and hence necessarily compact on \mathcal{H} :

Lemma 23. *A Gaussian measure μ on a separable Hilbert space \mathcal{H} has covariance operator $C_\mu: \mathcal{H} \rightarrow \mathcal{H}$ which is trace class and satisfies*

$$\int_{\mathcal{H}} \|x\|^2 \mu(dx) = \text{Tr } C_\mu. \quad (10.126)$$

Conversely, for every positive trace class symmetric operator $K: \mathcal{H} \rightarrow \mathcal{H}$, there exists a Gaussian measure μ on \mathcal{H} such that $C_\mu = K$.

Since the covariance operator $C_\mu: \mathcal{H} \rightarrow \mathcal{H}$ of a Gaussian on \mathcal{H} is a compact operator, it follows that if operator $C_\mu: \mathcal{H} \rightarrow \mathcal{H}$ has an inverse, then it will be a densely defined unbounded operator on \mathcal{H} ; we call this the *precision operator*. Both the covariance and the precision operators are self-adjoint on appropriate domains, and fractional powers of them may be defined via the spectral theorem.

Theorem 36 (Cameron-Martin Space on Hilbert Space). *Let μ be a Gaussian measure on a Hilbert space \mathcal{H} with strictly positive covariance operator K . Then the Cameron-Martin space \mathcal{H}_μ consists of the image of \mathcal{H} under $K^{1/2}$ and the Cameron-Martin norm is given by $\|h\|_\mu^2 = \|K^{-\frac{1}{2}}h\|^2$.*

Example 14. Consider two Gaussian measures μ_i on $\mathcal{H} = L^2(J)$, $J = (0, 1)$ both with precision operator $L = -\frac{d^2}{dx^2}$ where $\mathcal{D}(L) = H_0^1(J) \cap H^2(J)$. (Informally $-L$ is the Laplacian on J with homogeneous Dirichlet boundary conditions.) Let

\mathcal{C} denote the inverse of L on \mathcal{H} . Assume that $\mu_1 \sim N(m, \mathcal{C})$ and $\mu_2 \sim N(0, \mathcal{C})$. Then \mathcal{H}_{μ_i} is the image of \mathcal{H} under $\mathcal{C}^{\frac{1}{2}}$ which is the space $= H_0^1(J)$. It follows that the measures are equivalent if and only if $m \in H_0^1(J)$. If this condition is satisfied then, from Theorem 36, the Radon-Nikodym derivative between the two measures is given by

$$\frac{d\mu_1}{d\mu_2}(x) = \exp\left(\langle m, x \rangle_{H_0^1} - \frac{1}{2}\|m\|_{H_0^1}^2\right). \quad \square$$

We now turn to the Feldman-Hájek theorem in the Hilbert Space setting. Let $\{\varphi_j\}_{j=1}^\infty$ denote an orthonormal basis for \mathcal{H} . Then the *Hilbert-Schmidt norm* of a linear operator $L : \mathcal{H} \rightarrow \mathcal{H}$ is defined by

$$\|L\|_{\text{HS}}^2 := \sum_{j=1}^{\infty} \|L\varphi_j\|^2.$$

The value of the norm is, in fact, independent of the choice of orthonormal basis. In the finite-dimensional setting, the norm is known as the *Frobenius norm*.

Theorem 37 (Feldman-Hájek on Hilbert Space). *Let μ_i with $i = 1, 2$ be two centred Gaussian measures on some fixed Hilbert space \mathcal{H} with means m_i and strictly positive covariance operators \mathcal{C}_i . Then the following hold:*

1. μ_1 and μ_2 are either singular or equivalent.
2. The measures μ_1 and μ_2 are equivalent Gaussian measures if and only if:
 - a) The images of \mathcal{H} under $\mathcal{C}_i^{\frac{1}{2}}$ coincide for $i = 1, 2$, and we denote this common image space by E ;
 - b) $m_1 - m_2 \in E$;
 - c) $\|(\mathcal{C}_1^{-1/2}\mathcal{C}_2^{1/2})(\mathcal{C}_1^{-1/2}\mathcal{C}_2^{1/2})^* - I\|_{\text{HS}} < \infty$.

Remark 7. The final condition may be replaced by the condition that

$$\|(\mathcal{C}_1^{1/2}\mathcal{C}_2^{-1/2})(\mathcal{C}_1^{1/2}\mathcal{C}_2^{-1/2})^* - I\|_{\text{HS}} < \infty$$

and the theorem remains true; this formulation is sometimes useful. \square

Example 15. Consider two mean-zero Gaussian measures μ_i on $\mathcal{H} = L^2(J)$, $J = (0, 1)$ with precision operators $L_1 = -\frac{d^2}{dx^2} + I$ and $L_2 = -\frac{d^2}{ds^2}$, respectively, both with domain $H_0^1(J) \cap H^2(J)$. The operators L_1, L_2 share the same eigenfunctions

$$\phi_k(x) = \sqrt{2} \sin(k\pi x)$$

and have eigenvalues

$$\lambda_k(1) = \lambda_k(2) + 1, \quad \lambda_k(2) = k^2\pi^2,$$

respectively. Thus $\mu_1 \sim N(0, \mathcal{C}_1)$ and $\mu_2 \sim N(0, \mathcal{C}_2)$ where, in the basis of eigenfunctions, \mathcal{C}_1 and \mathcal{C}_2 are diagonal with eigenvalues

$$\frac{1}{k^2\pi^2 + 1}, \quad \frac{1}{k^2\pi^2}$$

respectively. We have, for $h_k = \langle h, \phi_k \rangle$,

$$\frac{\pi^2}{\pi^2 + 1} \leq \frac{\langle h, \mathcal{C}_1 h \rangle}{\langle h, \mathcal{C}_2 h \rangle} = \frac{\sum_{k \in \mathbb{Z}^+} (1 + k^2\pi^2)^{-1} h_k^2}{\sum_{k \in \mathbb{Z}^+} (k\pi)^{-2} h_k^2} \leq 1.$$

From this it follows that the Cameron-Martin spaces of the two measures coincide and are equal to $H_0^1(J)$. Notice that

$$T = \mathcal{C}_1^{-\frac{1}{2}} \mathcal{C}_2 \mathcal{C}_1^{-\frac{1}{2}} - I$$

is diagonalized in the same basis as the \mathcal{C}_i and has eigenvalues

$$\frac{1}{k^2\pi^2}.$$

These are square summable and so by Theorem 37 the two measures are absolutely continuous with respect to one another. \square

A.4 Wiener Processes in Infinite-Dimensional Spaces

Central to the theory of stochastic PDEs is the notion of a *cylindrical Wiener process*, which can be thought of as an infinite-dimensional generalization of a standard n -dimensional Wiener process. This leads to the notion of the A -Wiener process (A -) for certain classes of operators A . Before we proceed to the definition and construction of such Wiener processes in separable Hilbert spaces, let us recall a few basic facts about stochastic processes in general.

In general, a stochastic process u over a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and taking values in a separable Hilbert space \mathcal{H} is nothing but a collection $\{u(t)\}$ of \mathcal{H} -valued random variables indexed by time $t \in \mathbb{R}$ (or taking values in some subset of \mathbb{R}). By Kolmogorov's Extension Theorem 29, we can also view this as a map $u: \Omega \rightarrow \mathcal{H}^\mathbb{R}$, where $\mathcal{H}^\mathbb{R}$ is endowed with the product sigma-algebra. A notable special case which will be of interest here is the case where the probability space is taken to be $\Omega = C([0, T], \mathcal{H})$ (or some other space of \mathcal{H} -valued continuous functions) endowed with some Gaussian measure \mathbb{P} and where the process X is given by

$$u(t)(\omega) = \omega(t), \quad \omega \in \Omega.$$

In this case, u is called the canonical process on Ω .

The usual (one-dimensional) Wiener process is a real-valued centred Gaussian process $B(t)$ such that $B(0) = 0$ and $\mathbb{E}|B(t) - B(s)|^2 = |t - s|$ for any pair of times s, t . From our point of view, the Wiener process on any finite time interval I can always be realised as the canonical process for the Gaussian measure on $C(I, \mathbb{R})$ with covariance function $c(s, t) = s \wedge t = \min\{s, t\}$. Note that such a measure exists by the Kolmogorov continuity test, and Corollary 4 in particular.

The standard n -dimensional Wiener process $B(t)$ is simply given by n independent copies of a standard one-dimensional Wiener process $\{\beta_j\}_{j=1}^n$, so that its covariance is given by

$$\mathbb{E}\beta_i(s)\beta_j(t) = (s \wedge t)\delta_{i,j}.$$

In other words, if u and v are any two elements in \mathbb{R}^n , we have

$$\mathbb{E}\langle u, B(s) \rangle \langle B(t), v \rangle = (s \wedge t)\langle u, v \rangle.$$

This is the characterization that we will now extend to an arbitrary separable Hilbert space \mathcal{H} . One natural way of constructing such an extension is to fix an orthonormal basis $\{e_n\}_{n \geq 1}$ of \mathcal{H} and a countable collection $\{\beta_j\}_{j=1}^\infty$ of independent one-dimensional Wiener processes, and to set

$$B(t) := \sum_{n=1}^{\infty} \beta_n(t) e_n. \quad (10.127)$$

If we define

$$B^N(t) := \sum_{n=1}^N \beta_n(t) e_n$$

then clearly $\mathbb{E}\|B^N(t)\|_{\mathcal{H}}^2 = tN$ and so the series will not converge in \mathcal{H} for fixed $t > 0$. However the expression (10.127) is nonetheless the right way to think of a cylindrical Wiener process on \mathcal{H} ; indeed for fixed $t > 0$ the truncated series for B^N will converge in a larger space containing \mathcal{H} . We define the following scale of Hilbert subspaces, for $r > 0$, by

$$\mathcal{X}^r = \{u \in \mathcal{H} \mid \sum_{j=1}^{\infty} j^{2r} |\langle u, \phi_j \rangle|^2 < \infty\}$$

and then extend to superspaces $r < 0$ by duality. We use $\|\cdot\|_r$ to denote the norm induced by the inner-product

$$\langle u, v \rangle_r = \sum_{j=1}^{\infty} j^{2r} u_j v_j$$

for $u_j = \langle u, \phi_j \rangle$ and $v_j = \langle v, \phi_j \rangle$. A simple argument, similar to that used to prove Theorem 8, shows that $\{B^N(t)\}$ is, for fixed $t > 0$, Cauchy in \mathcal{X}^r for any $r < -\frac{1}{2}$. In fact it is possible to construct a stochastic process as the limit of the truncated series, living on the space $C([0, \infty), \mathcal{X}^r)$ for any $r < -\frac{1}{2}$, by the Kolmogorov Continuity Theorem 30 in the setting where $D = [0, T]$ and $X = \mathcal{X}^r$. We give details in the more general setting that follows.

Building on the preceding we now discuss construction of a \mathcal{C} -Wiener process W , using the finite-dimensional case described in Remark 2 to guide us. Here $\mathcal{C} : \mathcal{H} \rightarrow \mathcal{H}$ is assumed to be trace-class with eigenvalues γ_j^2 . Consider the cylindrical Wiener process given by

$$B(t) = \sum_{j=1}^{\infty} \beta_j e_j,$$

where $\{\beta_j\}_{j=1}^{\infty}$ is an i.i.d. family of unit Brownian motions on \mathbb{R} with $\beta_j \in C([0, \infty); \mathbb{R})$. We note that

$$\mathbb{E}|\beta_j(t) - \beta_j(s)|^2 = |t - s|. \quad (10.128)$$

Since $\sqrt{\mathcal{C}}e_j = \gamma_j e_j$, the \mathcal{C} -Wiener process $W = \sqrt{\mathcal{C}}B$ is then

$$W(t) = \sum_{j=1}^{\infty} \gamma_j \beta_j(t) e_j. \quad (10.129)$$

The following formal calculation gives insight into the properties of W :

$$\begin{aligned} \mathbb{E} W(t) \otimes W(s) &= \mathbb{E} \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \beta_j(t) \beta_k(s) e_j \otimes e_k \right) \\ &= \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \mathbb{E}(\beta_j(t) \beta_k(s)) e_j \otimes e_k \right) \\ &= \left(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_j \gamma_k \delta_{jk}(t \wedge s) e_j \otimes e_k \right) \\ &= \sum_{j=1}^{\infty} (\gamma_j^2 \phi_j \otimes \phi_j) t \wedge s \\ &= \mathcal{C}(t \wedge s). \end{aligned}$$

Thus the process has the covariance structure of Brownian motion in time, and covariance operator \mathcal{C} in space. Hence the name \mathcal{C} -Wiener process.

Assume now that the sequence $\gamma = \{\gamma_j\}_{j=1}^\infty$ is such that $\sum_{j=1}^\infty j^{2r} \gamma_j^2 = M < \infty$ for some $r \in \mathbb{R}$. For fixed t it is then possible to construct a stochastic process as the limit of the truncated series

$$W^N(t) = \sum_{j=1}^N \gamma_j \beta_j(t) e_j,$$

by means of a Cauchy sequence argument in $L^2_{\mathbb{P}}(\Omega; \mathcal{X}^r)$. Similarly $W(t) - W(s)$ may be defined for any t, s . We may then also discuss the regularity of this process in time. Together equations (10.128), (10.129) give $\mathbb{E}\|W(t) - W(s)\|_r^2 = M^2|t - s|$. It follows that $\mathbb{E}\|W(t) - W(s)\|_r \leq M|t - s|^{\frac{1}{2}}$. Furthermore, since $W(t) - W(s)$ is Gaussian, we have by (10.124) that $\mathbb{E}\|W(t) - W(s)\|_r^{2q} \leq K_q|t - s|^q$. Applying the Kolmogorov continuity test of Theorem 30 then demonstrates that the process given by (10.129) may be viewed as an element of the space $C^{0,\alpha}([0, T]; \mathcal{X}^r)$ for any $\alpha < \frac{1}{2}$. Similar arguments may be used to study the cylindrical Wiener process, showing that it lives in $C^{0,\alpha}([0, T]; \mathcal{X}^r)$ for $\alpha < \frac{1}{2}$ and $r < -\frac{1}{2}$.

A.5 Bibliographical Notes

- Section A.1 introduces various Banach and Hilbert spaces, as well as the notion of separability; see [100]. In the context of PDEs, see [33] and [87], for all of the function spaces defined in Sects. A.1.1–A.1.3; Sobolev spaces are developed in detail in [2]. The nonseparability of the Hölder spaces $C^{0,\beta}$ and the separability of $C_0^{0,\beta}$ is discussed in [40]. For asymptotics of the eigenvalues of the Laplacian operator see [91, Chapter 11]. For discussion of the more general spaces of E -valued functions over a measure space (\mathcal{M}, ν) we refer the reader to [100]. Section A.1.5 concerns Sobolev embedding theorems, building rather explicitly on the case of periodic functions. The corresponding embedding results in domains with more general boundary conditions or even on more general manifolds or unbounded domains, we refer to the comprehensive series of monographs [95–97]. The interpolation inequality of (10.106) and Lemma 15 may be found in [87]; see also Proposition 6.10 and Corollary 6.11 of [40]. The proof of Theorem 28 closely follows that given in [40, Theorem 6.16], and is a slight generalization to the Hilbert scale setting used here.
- Section A.2 briefly introduces the theory of probability measures on infinite-dimensional spaces. We refer to the extensive treatise by Bogachev [15], and to the much shorter but more readily accessible book by Billingsley [12], for more details. The subject of independent sequences of random variables, as overviewed in Sect. A.2.1 in the i.i.d. case, is discussed in detail in [27, section 1.5.1]. The Kolmogorov Extension Theorem 29 is proved in numerous texts in the setting where $X = \mathbb{R}$ [79]; since any Polish space is isomorphic to \mathbb{R} it may be stated as it is here. Proofs of Lemmas 17 and 19 may be found in

[40], where they appear as Proposition 3.6 and Proposition 3.9 respectively. For (10.115) see [28, Chapter 2]. In Sect. A.2.2 we introduce the Bochner integral; see [13, 48] for further details. Lemma 18 and the resulting Corollary 3 are stated and proved in [14]. The topic of metrics on probability measures, introduced in Sect. A.2.4 is overviewed in [38], where detailed references to the literature on the subject may also be found; the second inequality in Lemma 22 is often termed the *Pinsker inequality* and can be found in [22]. Note that the choice of normalization constants in the definitions of the total variation and Hellinger metrics differs in the literature. For a more detailed account of material on weak convergence of probability measures we refer, for example, to [12, 15, 98]. A proof of the Kolmogorov continuity test as stated in Theorem 30 can be found in [85, p. 26] for simple case of D an interval and X a separable Banach space; the generalization given here may be found in a forthcoming up-to-date version of [40].

- The subject of Gaussian measures, as introduced in Sect. A.3, is comprehensively studied in [14] in the setting of locally convex topological spaces, including separable Banach spaces as a special case. See also [67] which is concerned with Gaussian random functions. The Fernique theorem 31 is proved in [35] and the reader is directed to [40] for a very clear exposition. In Theorem 31 it is possible to take for α any value smaller than $1/(2\|C_\mu\|)$ and this value is sharp: see [66, Thm 4.1]. See [14, 67] for more details on the Cameron-Martin space, and proof of Theorem 32. Theorem 33 follows from Theorem 3.6.1 and Corollary 3.5.8 of [14]: Theorem 3.6.1 shows that the topological support is the closure of the Cameron-Martin space in B and Corollary 3.5.8 shows that the Cameron-Martin space is dense in B . The *reproducing kernel Hilbert space* for μ (or just *reproducing kernel* for short) appears widely in the literature and is isomorphic to the Cameron-Martin space in a natural way. There is considerable confusion between the two as a result. We retain in these notes the terminology from [14], but the reader should keep in mind that there are authors who use a slightly different terminology. Theorem 35 as stated is a consequence of Proposition 3 in section 18 in [67]. Turning now to the Hilbert space setting we note that Lemma 23 is proved as Proposition 3.15, and Theorem 36 appears as Exercise 3.34, in [40]. See [14, 28, 52] for alternative developments of the Cameron-Martin and Feldman-Hájek theorems. The original statement of the Feldman-Hájek Theorem 37 can be found in [34, 45]. Our statement of Theorem 37 mirrors Theorem 2.23 of [28] and Remark 7 is Lemma 6.3.1(ii) of [14]. Note that we have not stated a result analogous to Theorem 32 in the case where of two equivalent Gaussian measures with differing covariances. Such a result can be stated, but is technically complicated in general because the ratio of normalization constants of approximating finite-dimensional measures can blow up as the limiting infinite-dimensional Radon-Nikodym derivative is attained; see Corollary 6.4.11 in [14].
- Section A.4 contains a discussion of cylindrical and \mathcal{C} -Wiener processes. The development is given in more detail in section 3.4 of [40], and in section 4.3 of [28].

Acknowledgements The authors are indebted to Martin Hairer for help in the development of these notes, and in particular for considerable help in structuring the Appendix, for the proof of Theorem 28 (which is a slight generalization to Hilbert scales of Theorem 6.16 in [40]) and for the proof of Corollary 5 (which is a generalization of Corollary 3.22 in [40] to the non-Gaussian setting and to Hölder, rather than Lipschitz, functions $\{\psi_k\}$). They are also grateful to Joris Bierkens, Patrick Conrad, Matthew Dunlop, Shiwei Lan, Yulong Lu, Daniel Sanz-Alonso, Claudia Schillings and Aretha Teckentrup for careful proof-reading of the notes and related comments. AMS is grateful for various hosts who gave him the opportunity to teach this material in short course form at TIFR-Bangalore (Amit Apté), Göttingen (Axel Munk), PKU-Beijing (Teijun Li), ETH-Zurich (Christoph Schwab) and Cambridge CCA (Arieh Iserles), a process which led to refinements of the material; the authors are also grateful to the students on those courses, who provided useful feedback. The authors would also like to thank Sergios Agapiou and Yuan-Xiang Zhang for help in the preparation of these lecture notes, including type-setting, proof-reading, providing the proof of Lemma 3 and delivering problems classes related to the short courses. AMS is also pleased to acknowledge the financial support of EPSRC, ERC and ONR over the last decade, while the research that underpins this work has been developed.

References

1. Adler, R.: *The Geometry of Random Fields*. SIAM, Philadelphia (1981)
2. Adams, R.A., Fournier, J.J.: *Sobolev Spaces*. Pure and Applied Mathematics. Elsevier, Oxford (2003)
3. Agapiou, S., Larsson, S., Stuart, A.M.: Posterior contraction rates for the Bayesian approach to linear ill-posed inverse problems. *Stoch. Process. Appl.* **123**, 3828–3860 (2013)
4. Agapiou, S., Stuart, A.M., Zhang, Y.X.: Bayesian posterior consistency for linear severely ill-posed inverse problems. *J. Inverse Ill-posed Probl.* **22**(3), 297–321 (2014)
5. Alexanderian, A., Petra, N., Stadler, G., Ghattas, O.: A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. *SIAM J. Sci. Comput.* **38**(1), A243–A272 (2016)
6. Alexanderian, A., Gloor, P., Ghattas, O.: On Bayesian A- and D-optimal experimental designs in infinite dimensions. <http://arxiv.org/abs/1408.6323> (2016)
7. Babuska, I., Tempone, R., Zouraris, G.: Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.* **42**, 800–825 (2004)
8. Banks, H.T., Kunisch, H.: *Estimation Techniques for Distributed Parameter Systems*. Birkhäuser, Boston (1989)
9. Beskos, A., Pinski, F.J., Sanz-Serna, J.-M., Stuart, A.M.: Hybrid Monte-Carlo on Hilbert spaces. *Stoch. Process. Appl.* **121**, 2201–2230 (2011)
10. Beskos, A., Jasra, A., Muzaffer, E.A., Stuart, A.M.: Sequential Monte Carlo methods for Bayesian elliptic inverse problems. *Stat. Comput.* (2015)
11. Bernardo, J., Smith, A.: *Bayesian Theory*. Wiley, Chichester (1994)
12. Billingsley, P.: *Convergence of Probability Measures*. Wiley, New York (1968)
13. Bochner, S.: Integration von Funktionen, deren Werte die Elemente eines Vektorraumes sind. *Fund. Math.* **20**, 262–276 (1933)
14. Bogachev, V.I.: *Gaussian Measures*. Mathematical Surveys and Monographs, vol. 62. American Mathematical Society, Providence (1998)
15. Bogachev, V.I.: *Measure Theory*, vol. I, II. Springer, Berlin (2007)
16. Bui-Thanh, T., Ghattas, O., Martin, J., Stadler, G.: A computational framework for infinite-dimensional Bayesian inverse problems. Part I: the linearized case, with application to global seismic inversion. *SIAM J. Sci. Comput.* **35**(6), A2494–A2523 (2013)
17. Cotter, S., Dashti, M., Robinson, J., Stuart, A.: Bayesian inverse problems for functions and applications to fluid mechanics. *Inverse Probl.* **25**. doi:10.1088/0266-5611/25/11/115008 (2009)

18. Cohen, A., DeVore, R., Schwab, C.: Convergence rates of best n -term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.* **10**, 615–646 (2010)
19. Cohen, A., DeVore, R., Schwab, Ch.: Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. *Anal. Appl.* **9**(1), 11–47 (2011)
20. Cotter, S., Dashti, M., Stuart, A.: Approximation of Bayesian inverse problems. *SIAM J. Numer. Anal.* **48**, 322–345 (2010)
21. Cotter, S., Roberts, G., Stuart, A., White, D.: MCMC methods for functions: modifying old algorithms to make them faster. *Stat. Sci.* **28**, 424–446 (2013)
22. Csiszar, I., Körner, J.: *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, Cambridge (2011)
23. Dacorogna, B.: *Introduction to the Calculus of Variations*. Translated from the 1992 French original, 2nd edn. Imperial College Press, London (2009)
24. Dashti, M., Harris, S., Stuart, A.: Besov priors for Bayesian inverse problems. *Inverse Probl. Imaging* **6**, 183–200 (2012)
25. Dashti, M., Stuart, A.: Uncertainty quantification and weak approximation of an elliptic inverse problem. *SIAM J. Numer. Anal.* **49**, 2524–2542 (2011)
26. Del Moral, P.: *Feynman-Kac Formulae*. Springer, New York (2004)
27. Da Prato, G.: An introduction to infinite-dimensional analysis. Universitext. Springer, Berlin (2006). Revised and extended from the 2001 original by Da Prato
28. DaPrato, G., Zabczyk, J.: *Stochastic Equations in Infinite Dimensions*. Encyclopedia of Mathematics and its Applications, vol. 44. Cambridge University Press, Cambridge (1992)
29. DaPrato, G., Zabczyk, J.: *Ergodicity for Infinite Dimensional Systems*. Cambridge University Press, Cambridge (1996)
30. Dashti, M., Law, K.J.H., Stuart, A.M., Voss, J.: MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Probl.* **29**, 095017 (2013)
31. Daubechies, I.: Ten lectures on wavelets. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 61. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1992)
32. Engl, H., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*. Kluwer, Dordrecht/Boston (1996)
33. Evans, L.: *Partial Differential Equations*. AMS, Providence (1998)
34. Feldman, J.: Equivalence and perpendicularity of Gaussian processes. *Pac. J. Math.* **8**, 699–708 (1958)
35. Fernique, X.: Intégrabilité des vecteurs Gaussiens. *C. R. Acad. Sci. Paris Sér. A-B* **270**, A1698–A1699 (1970)
36. Franklin, J.: Well-posed stochastic extensions of ill-posed linear problems. *J. Math. Anal. Appl.* **31**, 682–716 (1970)
37. Gardiner, C.W.: *Handbook of stochastic methods*. Springer, Berlin, 2nd edn. (1985). For Physics, Chemistry and the Natural Sciences
38. Gibbs, A., Su, F.: On choosing and bounding probability metrics. *Int. Stat. Rev.* **70**, 419–435 (2002)
39. Graham, I.G., Kuo, F.Y., Nicholls, J.A., Scheichl, R., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo Finite Element methods for Elliptic PDEs with Log-normal Random Coefficients, Seminar for Applied Mathematics, ETH, SAM Report 2013–14 (2013)
40. Hairer, M.: Introduction to Stochastic PDEs. Lecture Notes <http://arxiv.org/abs/0907.4178> (2009)
41. Hairer, M., Stuart, A.M., Voss, J.: Analysis of SPDEs arising in path sampling, part II: the nonlinear case. *Ann. Appl. Probab.* **17**, 1657–1706 (2007)
42. Hairer, M., Stuart, A., Voss, J.: Sampling conditioned hypoelliptic diffusions. *Ann. Appl. Probab.* **21**(2), 669–698 (2011)
43. Hairer, M., Stuart, A., Voss, J., Wiberg, P.: Analysis of SPDEs arising in path sampling. Part I: the Gaussian case. *Comm. Math. Sci.* **3**, 587–603 (2005)

44. Hairer, M., Stuart, A., Vollmer, S.: Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions. *Ann. Appl. Probab.* **24**(6), 2455–2490 (2014)
45. Hájek, Y.: On a property of normal distribution of any stochastic process. *Czechoslov. Math. J.* **8**(83), 610–618 (1958)
46. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970)
47. Helin, T., Burger, M.: Maximum a posteriori probability estimates in infinite-dimensional Bayesian inverse problems. *Inverse Probl.* **31**(8), 085009 (2015)
48. Hildebrandt, T.H.: Integration in abstract spaces. *Bull. Am. Math. Soc.* **59**, 111–139 (1953)
49. Kahane, J.-P.: Some Random Series of Functions. Cambridge Studies in Advanced Mathematics, vol. 5. Cambridge University Press, Cambridge (1985)
50. Kantas, N., Beskos, A., Jasra, A.: Sequential Monte Carlo methods for high-dimensional inverse problems: a case study for the Navier-Stokes equations. arXiv preprint, arXiv:1307.6127
51. Kirsch, A.: An Introduction to the Mathematical Theory of Inverse Problems. Springer, New York (1996)
52. Kühn, T., Liese, F.: A short proof of the Hájek-Feldman theorem. *Teor. Verojatnost. i Primenen.* **23**(2), 448–450 (1978)
53. Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems. Applied Mathematical Sciences, vol. 160. Springer, New York (2005)
54. Kallenberg, O.: Foundations of Modern Probability, 2nd edn. Probability and its Applications. Springer, New York, (2002)
55. Knapik, B., van Der Vaart, A., van Zanten, J.: Bayesian inverse problems with Gaussian priors. *Ann. Stat.* **39**(5), 2626–2657 (2011)
56. Knapik, B., van der Vaart, A., van Zanten, J.H.: Bayesian recovery of the initial condition for the heat equation. *Commun. Stat. Theory Methods* **42**, 1294–1313 (2013)
57. Kuo, F.Y., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo methods for very high dimensional integration: the standard (weighted Hilbert space) setting and beyond. *ANZIAM J.* **53**, 1–37 (2011)
58. Kuo, F.Y., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo Finite Element Methods for a Class of Elliptic Partial Differential Equations with Random Coefficients. *SIAM J. Numer. Anal.* **50**(6), 3351–3374 (2012)
59. Kuo, F.Y., Sloan, I.H.: Lifting the curse of dimensionality. *Not. AMS* **52**(11), 1320–1328 (2005)
60. Kuo, F.Y., Sloan, I.H., Wasilkowski, G.W., Waterhouse, B.J.: Randomly shifted lattice rules with the optimal rate of convergence for unbounded integrands. *J. Complex.* **26**, 135–160 (2010)
61. Lasanen, S.: Discretizations of generalized random variables with applications to inverse problems. *Ann. Acad. Sci. Fenn. Math. Dissertation*, University of Oulu, 130 (2002)
62. Lasanen, S.: Measurements and infinite-dimensional statistical inverse theory. *PAMM* **7**, 1080101–1080102 (2007)
63. Lasanen, S.: Non-Gaussian statistical inverse problems part II: posterior convergence for approximated unknowns. *Inverse Probl. Imaging* **6**(2), 267 (2012)
64. Lasanen, S.: Non-Gaussian statistical inverse problems. Part I: posterior distributions. *Inverse Probl. Imaging* **6**(2), 215–266 (2012)
65. Lasanen, S.: Non-Gaussian statistical inverse problems. Part II: posterior distributions. *Inverse Probl. Imaging* **6**(2), 267–287 (2012)
66. Ledoux, M.: Isoperimetry and Gaussian analysis. In: Lectures on Probability Theory and Statistics (Saint-Flour, 1994). Lecture Notes in Mathematics, vol. 1648, pp. 165–294. Springer, Berlin (1996)
67. Lifshits, M.: Gaussian Random Functions. Mathematics and its Applications, vol. 322. Kluwer, Dordrecht (1995)
68. Lehtinen, M.S., Pääväranta, L., Somersalo, E.: Linear inverse problems for generalised random variables. *Inverse Probl.* **5**(4), 599–612 (1989). <http://stacks.iop.org/0266-5611/5/599>

69. Lassas, M., Saksman, E., Siltnen, S.: Discretization-invariant Bayesian inversion and Besov space priors. *Inverse Probl. Imaging* **3**, 87–122 (2009)
70. Lunardi, A.: Analytic Semigroups and Optimal Regularity in Parabolic Problems. Progress in Nonlinear Differential Equations and their Applications, vol. 16. Birkhäuser Verlag, Basel (1995)
71. Mandelbaum, A.: Linear estimators and measurable linear transformations on a Hilbert space. *Z. Wahrsch. Verw. Gebiete* **65**(3), 385–397 (1984). <http://dx.doi.org/10.1007/BF00533743>
72. Mattingly, J., Pillai, N., Stuart, A.: Diffusion limits of the random walk Metropolis algorithm in high dimensions. *Ann. Appl. Probl.* **22**, 881–930 (2012)
73. Metropolis, N., Rosenbluth, R., Teller, M., Teller, E.: Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953)
74. Meyer, Y.: Wavelets and operators. Translated from the 1990 French original by D.H. Salinger. Cambridge Studies in Advanced Mathematics, vol. 37. Cambridge University Press, Cambridge (1992)
75. Meyn, S.P., Tweedie, R.L.: Markov Chains and Stochastic Stability. Communications and Control Engineering Series. Springer, London (1993)
76. Neal, R.: Regression and classification using Gaussian process priors. <http://www.cs.toronto.edu/~radford/valencia.abstract.html> (1998)
77. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1994)
78. Norris, J.: Markov Chains. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge (1998)
79. Oksendal, B.: Stochastic Differential Equations. An Introduction with Applications. Universitext, 6th edn. Springer, Berlin (2003)
80. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations. Springer, New York (1983)
81. Petra, N., Martin, J., Stadler, G., Ghattas, O.: A computational framework for infinite-dimensional Bayesian inverse problems. Part II: stochastic Newton MCMC with application to ice sheet flow inverse problems. *SIAM J. Sci. Comput.* **36**(4), A1525–A1555 (2014)
82. Pillai, N.S., Stuart, A.M., Thiery, A.H.: Noisy gradient flow from a random walk in Hilbert space. *Stoch. PDEs: Anal. Comput.* **2**, 196–232 (2014)
83. Pinski, F., Stuart, A.: Transition paths in molecules at finite temperature. *J. Chem. Phys.* **132**, 184104 (2010)
84. Rebeschini, P., van Handel, R.: Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.* **25**(5), 2809–2866 (2015)
85. Revuz, D., Yor, M.: Continuous Martingales and Brownian Motion. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 293, 2nd edn. Springer, Berlin (1994)
86. Richter, G.: An inverse problem for the steady state diffusion equation. *SIAM J. Appl. Math.* **41**(2), 210–221 (1981)
87. Robinson, J.C.: Infinite-Dimensional Dynamical Systems. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (2001)
88. Rudin, W.: Real and Complex Analysis, 3rd edn. McGraw-Hill, New York (1987)
89. Schillings, C., Schwab, C.: Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. *Inverse Probl.* **29**, 065011 (2013)
90. Schwab, C., Stuart, A.: Sparse deterministic approximation of Bayesian inverse problems. *Inverse Probl.* **28**, 045003 (2012)
91. Strauss, W.A.: Partial differential equations. An introduction, 2nd edn. Wiley, Chichester (2008)
92. Stuart, A.M.: Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559 (2010)
93. Stuart, A.M.: Uncertainty quantification in Bayesian inversion. ICM2014. Invited Lecture (2014)
94. Tierney, L.: A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.* **8**(1), 1–9 (1998)

95. Triebel, H.: Theory of function spaces. Mathematik und ihre Anwendungen in Physik und Technik [Mathematics and its Applications in Physics and Technology], vol. 38. Akademische Verlagsgesellschaft Geest & Portig K.-G., Leipzig (1983)
96. Triebel, H.: Theory of Function Spaces. II. Monographs in Mathematics, vol. 84. Birkhäuser Verlag, Basel (1992)
97. Triebel, H.: Theory of Function Spaces. III. Monographs in Mathematics, vol. 100. Birkhäuser Verlag, Basel (2006)
98. Villani, C.: Topics in Optimal Transportation. Graduate Studies in Mathematics, vol. 58. American Mathematical Society, Providence (2003)
99. Vollmer, S.: Posterior consistency for Bayesian inverse problems through stability and regression results. Inverse Probl. **29**, 125011 (2013)
100. Yosida, K.: Functional Analysis. Classics in Mathematics. Springer, Berlin (1995). Reprint of the sixth (1980) edition.
101. Yin, G., Zhang, Q.: Continuous-Time Markov Chains and Applications. Applications of Mathematics (New York), vol. 37. Springer, New York (1998)

Sankaran Mahadevan, Shankar Sankararaman, and Chenzhao Li

Abstract

This chapter discusses a Bayesian methodology to integrate model verification, validation, and calibration activities for the purpose of overall uncertainty quantification in model-based prediction. The methodology is first developed for single-level models and then extended to systems that are studied using multilevel models that interact with each other. Two types of interactions among multilevel models are considered: (1) Type-I, where the output of a lower-level model (component and/or subsystem) becomes an input to a higher-level system model, and (2) Type-II, where parameters of the system model are inferred using lower-level models and tests (that describe simplified components and/or isolated physics). The various models; their inputs, parameters, and outputs; experimental data; and various sources of model error are connected through a Bayesian network. The results of calibration, verification, and validation with respect to each individual model are integrated using the principles of conditional probability and total probability and propagated through the Bayesian network in order to quantify the overall system-level prediction uncertainty. For Type-II model, the relevance of each lower-level output to the system-level quantity of interest is quantified by comparing Sobol indices, thus measuring the extent to which a lower-level test represents the characteristics of the system so that the calibration results can be reliably used in the system level. The proposed methodology is illustrated with numerical examples that deal with heat conduction and structural dynamics.

S. Mahadevan (✉) • C. Li

Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN, USA
e-mail: sankaran.mahadevan@vanderbilt.edu

S. Sankararaman

NASA Ames Research Center, SGT Inc., Moffett Field, CA, USA

Keywords

Multilevel system • Uncertainty quantification • Bayesian network • Calibration • Validation • Verification • Relevance • Sobol indices • Bayes factor • Model reliability metric

Contents

1	Introduction	430
1.1	Motivation	430
1.2	Multilevel System Models	432
1.3	Organization of the Chapter	435
2	Integration of Verification, Validation, and Calibration (Single Level)	436
2.1	Verification	436
2.2	Calibration	440
2.3	Validation	441
2.4	Integration for Overall Uncertainty Quantification	447
3	Numerical Example: Single-Level Model	447
3.1	Description of the Problem	447
3.2	Verification, Validation, and Calibration	448
3.3	Integration and Overall Uncertainty Quantification	449
4	Multilevel Models with Type-I Interaction	450
4.1	Verification, Calibration, and Validation	451
4.2	Integration for Overall Uncertainty Quantification	452
4.3	Extension to Multiple Models	453
5	Numerical Example: Two Models with Type-I Interaction	455
6	Multilevel Models with Type-II Interaction	457
6.1	Verification, Calibration, and Validation	458
6.2	Relevance Analysis	459
6.3	Integration for Overall Uncertainty Quantification	462
7	Numerical Example: Models with Type-II Interaction	464
7.1	Problem Description	464
7.2	Results and Analysis	466
8	Conclusion	470
	References	471

1 Introduction

1.1 Motivation

Individual activities such as calibration, verification, and validation address different aspects of model uncertainty (i.e., model parameters, solution approximations, and model form), and corresponding methods have been discussed in previous chapters. A pertinent question is how to integrate the results of these activities for the purpose of overall uncertainty quantification in the model prediction. This is not trivial because of several reasons. First, the solution approximation errors calculated as a result of the verification process should be considered during calibration, validation, and prediction. Second, the result of validation may lead to a binary result, i.e., the model is accepted or rejected; however, even when the model is accepted, it is not completely correct. Hence, it is necessary to account for the degree of correctness

of the model during prediction uncertainty quantification. Third, if calibration and validation are performed using independent data sets, it is not straightforward to compute their combined effect on the overall uncertainty in the response.

The issue gets further complicated when the behavior of complex engineering systems is studied using multiple component-level and subsystem-level models that integrate to form an overall multilevel system model. In each level, there is a computational model with inputs, parameters, and outputs, experimental data (hopefully available for calibration and validation separately), and several sources of uncertainty physical variability, data uncertainty (sparse or imprecise data, measurement errors), and model uncertainty (parameter uncertainty, solution approximation errors, and model form error). In such a multilevel system, the first task would be to connect all the available models and associated sources of uncertainty.

Recent studies [1, 2] have demonstrated that the Bayesian network methodology provides an efficient and powerful tool to integrate multiple levels of models, associated sources of uncertainty and error, and available data at multiple levels. While the Bayesian approach can be used to perform calibration and validation individually for each model in the multilevel system, it is not straightforward to integrate the information from these activities in order to compute the overall uncertainty in the system-level prediction. This chapter extends the Bayesian approach to integrate and propagate information from verification, calibration, and validation activities in order to quantify the margins and uncertainties in the overall system-level prediction. In Bayesian calibration, the goal is to estimate the probability distributions of the underlying model parameters, using the data available for calibration. Once the model is calibrated, it is validated using an independent set of input-output data. There are two advantages in using a Bayesian methodology for both calibration and validation:

1. Both calibration and validation involve comparing model prediction against experimental data; the Bayesian approach not only allows the comparison of entire distributions of model prediction and experimental data, but also provides a systematic approach to include various types of uncertainty – physical variability, data uncertainty, and model uncertainty/errors – through a Bayesian network.
2. The Bayesian approach can systematically handle epistemic uncertainty due to sparse, imprecise, and unpaired input-output data, as demonstrated by the authors for calibration [3] as well as validation [4–6].

While the Bayesian approach offers several advantages, there are some practical challenges in implementing Bayesian methods. The need to assume prior probability distributions might be challenging in some situations, whereas it might be attractive in other situations to incorporate prior knowledge (if available) and use non-informative prior probability distributions [7] when no prior knowledge is available. Bayesian techniques involve the computation of high-dimensional integrals that are often solved through Markov chain Monte Carlo (MCMC) sampling techniques that require extensive computational effort. With the advent of the high-performance

computing techniques, it has become easier to implement such computationally intensive methods for practical applications, and therefore, Bayesian methods are being increasingly studied in engineering disciplines in recent years.

Methods for calibration and validation have been demonstrated only for individual models with calibration and validation data. What happens when there is flow of information across multiple levels of models that are used to study a multilevel system? Since the Bayesian approach represents the various types of uncertainty at multiple levels through probability distributions, the problem reduces to propagating these probability distributions through component and subsystems, in order to compute the uncertainty in the system-level prediction. Both solution approximation errors and model form errors can be included in a Bayesian network representation as additional nodes [2, 8, 9]. The Bayesian network can be used for both the forward problem of uncertainty propagation [8] and inverse problem of calibration [9, 10]. The results of calibration and validation activities are expressed in terms of probability distributions for the model parameters and the probability that each model is valid, respectively. The Bayesian approach is thus able to provide a unified framework for integrating information from verification, calibration, and validation at multiple levels to calculate the overall system-level prediction uncertainty. Using the principles of conditional probability and total probability, this chapter introduces a computational approach for such integration. This approach is first developed for single-level models and then extended to multilevel system models.

1.2 Multilevel System Models

Typically, a multilevel system is studied using different types of models; each model may represent a particular component, a particular subsystem, or an isolated set of features/physics of the original system. The interaction between any two models depends on what features of the original system they represent, and it is necessary to translate the dependency between the features into mathematical relationship between the models. In order to facilitate such translation, and hence, the objectives of this chapter, two types of interactions (designated as Type-I and Type-II in this chapter) are considered in detail. In both of these types, the quantity of interest is an overall system-level response, but there is a significant difference in how this quantity is calculated using information from lower-level models, verification, validation, and calibration.

The interaction between two models (denoted by G and H in Fig. 11.1a) is considered to be “Type-I,” when the lower-level model G represents the behavior of a particular component/subsystem whose output (Y) becomes an input to a higher-level subsystem/component whose behavior is represented by the higher-level model H . Each model has its own set of model parameters (not indicated in Fig. 11.1a); there may or may not be any model parameter common between two models. For example, the rise in the temperature (Y , computed using G) of a conducting wire leads to a change in its resistance and, hence, in its current carrying capacity (Z , which is computed using H).

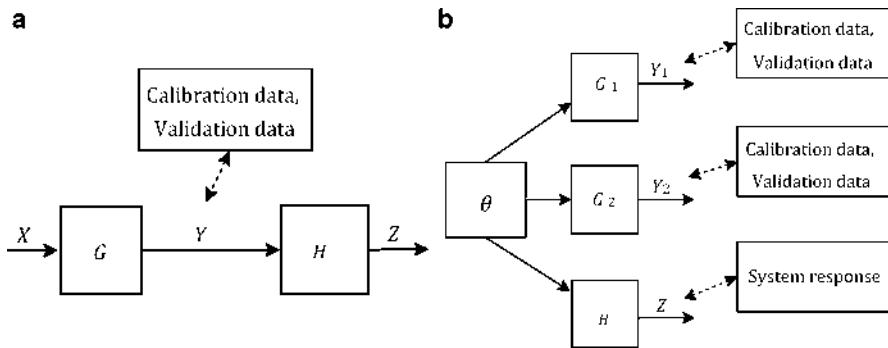


Fig. 11.1 Two types of interactions between models. (a) Type-I interaction. (b) Type-II interaction

The interaction between two models is said to be “Type-II,” when the lower-level model represents a simplification of the overall system. Typically, it may not be possible to study certain features of the system (i.e., the system parameters) since extensive testing of the overall system may be prohibitory. Therefore, a simplified configuration that consists of an isolated set of features and/or an isolated set of physics of the multilevel system is often considered for testing. The response of the simplified subsystem is not directly related to the response of the overall system; however, there are some parameters (θ) of the system model (H) that can be inferred using lower-level models and experiments. Further, multiple test options may be available (two in Fig. 11.1b), where the Level 2 model G_2 may describe more features than the Level 1 model G_1 , in terms of complexity. The model of the highest complexity (H) represents the system of interest and the system-level response (Z) needs to be calculated. The model parameters (θ) are calibrated using models and experiments of reduced complexity (e.g., isolated components or physics) and then propagated through the system model to compute the desired response.

For example, consider a coupon subjected to axial testing (say, the axial deflection is modeled using G_1), a cantilever beam subjected to point loading (say, the deflection is modeled using G_2), and a plate subjected to bending (say, the bending stress is modeled using H), where the coupon, the beam, and the plate are made of the same material. While both the axial deflection of the coupon (Y_1) and the deflection of the beam (Y_2) are not directly related to the bending stress in the plate (Z), some material properties (like the elastic modulus, denoted by θ) of the plate can be studied using tests performed on the coupon and the beam. Note that there is no interaction between models G_1 and G_2 ; there is “Type-II” interaction between models G_1 and H and between models G_2 and H . Urbina et al. [1], Sankararaman et al. [11], Mullins et al. [12], and Li et al. [13] discuss practical multilevel systems with Type-II interaction; these studies consider lower-level models of increasing complexity and physics, i.e., only a few aspects of physics are captured at the lowest level model and more aspects are increasingly captured in subsequent higher levels.

In some cases, an extrapolation problem can be described using two models with “Type-II” interaction. When the application conditions are physically different from validation conditions (e.g., validating using a beam but extrapolating to a plate), model G_2 may correspond to the validation conditions and model H may correspond to the extrapolation conditions.

For Type-II interaction, a reasonable route is to quantify the model parameters using lower-level data and propagate the results through the computational model at the system level. However, model calibration can be conducted using the data from a single level or multiple levels. For the problem in Fig. 11.1b with two lower levels, three calibration options are possible: (1) calibration using the data and model from Level 1 alone, (2) calibration using the data and model from Level 2 alone, and (3) calibration using the data and models from both Level 1 and Level 2. Generally, if data are available at n_m different levels, $2^{n_m} - 1$ model calibration options are possible to quantify the uncertainty of model parameters [14].

This chapter uses Bayesian inference for model calibration, thus the result of model calibration is a joint posterior distribution of model parameters. As Kennedy and O’Hagan [15] pointed out, the posterior distribution is the “best-fitting” results in the sense of representing the calibration data faithfully, not necessarily representing the true physical values. One possibility is to use all the lower-level data in model calibration and propagate the resultant posterior distribution to predict the system-level output. However, this result is conditioned on the event that both the models at Level 1 and Level 2 are valid, which may or may not be true [16]. This chapter answers this question by assigning a “confidence” measure to each posterior distribution. Note that this chapter is not using the term “confidence” in the same sense as is used in statistics (as in confidence interval). This “confidence” measure constitutes of two components: (1) the model validity at the corresponding lower level (one can think of this as local confidence regarding each lower level) and (2) the relationship between the lower level and the system level, i.e., the relevance of the posterior distribution obtained at the lower level to the system-level prediction problem (one can think of this as inter-level confidence).

The reason to use model validation to quantify the local confidence is explained here. In model validation, the assessed model validity of the prediction model $G(\mathbf{x}; \boldsymbol{\theta})$ at a lower level is a combined effect of (1) $G(\mathbf{x}; \boldsymbol{\theta})$ and (2) the posterior distribution of $\boldsymbol{\theta}$. The second aspect corresponds to the “local confidence” (not to be confused with confidence intervals used in statistics); thus this chapter takes the model validity as one factor affecting our confidence in extrapolating the posterior distribution of the model parameter from the lower level to the system level. This is reasonable since the model parameter has been calibrated with a model corresponding to the lower-level experiment, and it is important to know whether the model was calibrated accurately; the calibration result is obviously affected by how accurately the lower-level model represents the physics in the lower-level experiment.

The inter-level confidence to extrapolate a lower-level posterior distribution to the system level is about the relationship between the lower level and the system level. In this chapter, the relationship between the lower level and the system level

is quantified by a relevance analysis [17, 18]. The necessity of relevance analysis is explained here. An inherent assumption in the relevance analysis is that if the physical configuration of a lower-level experiment (say Level 2 in Fig. 11.1b) is more similar to the system level than another lower-level experiment (say Level 1 in Fig. 11.1b), it is reasonable to assign higher confidence to the calibration result at this level (i.e., Level 2). Thus the relevance of the lower level to the system level is the degree to which the experimental configuration at a lower level reflects the physical characteristics of the system so that the calibration results can be reliably used in the system-level prediction. The relevance decides the inter-level confidence on the calibration at lower levels and influences the uncertainty integration. This chapter discusses a method to quantify the relevance using Sobol indices and the cosine similarity of sensitivity vectors.

Note that there is a third type of interaction between two models that is commonly observed in multidisciplinary systems. The two models represent different physics, but the output of each model is the input to the other. This is feedback coupling, and it is necessary to perform iterative analysis between these two models in order to compute the system-level response. The authors developed a likelihood-based method [19] to mathematically transform two-way coupling to one-way coupling; as a result, the method in this chapter for Type-I interaction can also be applied to models with feedback coupling. Further, there may be other types of interactions in multilevel models, but this chapter studies only the two types of interactions (Type-I and Type-II) in detail. The primary goal of this chapter is to develop a framework for the integration of verification, validation, and calibration activities in order to facilitate system-level uncertainty quantification, by considering multilevel models that exhibit Type-I or Type-II interaction. This is accomplished by computing the probability density function (PDF) or the cumulative distribution function (CDF) of the system-level response quantity of interest, and this PDF or CDF needs to incorporate the results of verification, validation, and calibration.

The integration methodology is different for system models with Type-I and Type-II interactions. In the former case, the linking variables between two models are the outputs of the lower-level models that become inputs to the higher-level models, whereas in the latter case, the linking variables are the common model parameters. With the focus on the linking variables and using the principles of conditional probability and total probability, this chapter develops a Bayesian network-based methodology to integrate the results of verification, validation, and calibration activities and to compute the uncertainty in the overall system-level prediction.

1.3 Organization of the Chapter

The rest of the chapter is organized as follows. Section 2 develops the methodology for the integration of calibration, verification, and validation in a single-level model, followed by a numerical example in Sect. 3. Section 4 develops the integration

methodology for multilevel models with Type-I interaction, followed by a numerical example in Sect. 5. Section 6 develops the integration methodology for multilevel models with Type-II interaction, followed by a numerical example in Sect. 7. Though the methodology is individually developed for each type of interaction, it is straightforward to extend it to multilevel models that may contain both types of interactions.

2 Integration of Verification, Validation, and Calibration (Single Level)

Consider a single-level model as shown in Fig. 11.2. The inputs are x , the model parameters are θ , the true solution of the mathematical equation is y , and the code output is y_c . Both y_c and y are deterministic functions of inputs (x) and model parameters (θ).

This section discusses methods to integrate the results of calibration, verification, and validation of the model. Since the process of verification is not related to experimental data, it needs to be performed first; both calibration and validation must include the results of verification analysis (i.e., solution error quantification). Then, the results of verification, calibration, and validation are integrated to compute the overall uncertainty in the response quantity.

2.1 Verification

The process of verification checks how close the code output is to the true solution of the mathematical equation. Verification includes both code verification (identification of programming errors and debugging) and solution verification (convergence studies, identifying and computing solution approximation errors) [20,21]. Methods for code verification [22–27] and estimation of solution approximation error [25, 27–34] have been investigated by several researchers. It is desirable to perform verification before calibration and validation so that the solution approximation errors are accounted for during calibration and validation. Solution approximation errors in finite element analysis have been estimated using a variety

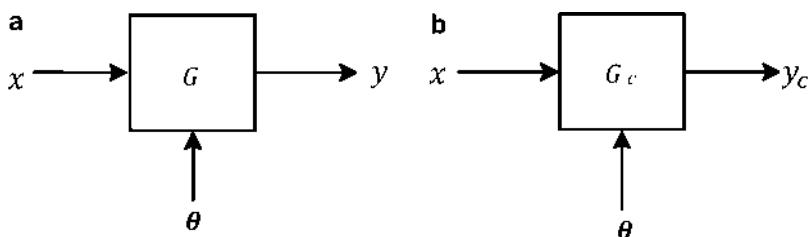


Fig. 11.2 A single-level model. (a) Mathematical equation. (b) Computer code

of techniques, such as convergence analysis [35] a posteriori error estimation [36] Richardson extrapolation [34, 37, 38] etc. Another type of solution approximation error arises when the underlying model is replaced with a surrogate model for fast uncertainty propagation and/or model calibration. Many surrogate modeling techniques have been developed, such as regression models [38] polynomial chaos expansions [39], radial basis functions [40] or Gaussian processes [41]. The quantification of this surrogate model error is different for different types of surrogate models, and the methods are well established in the literature. Once the solution approximation error is computed, the true solution of mathematical equation can be computed as a function of the model inputs and parameters as $y(\mathbf{x}; \boldsymbol{\theta}) = y_c(\mathbf{x}; \boldsymbol{\theta}) + G_{se}(\mathbf{x}; \boldsymbol{\theta})$, where $G_{se}(\mathbf{x}; \boldsymbol{\theta})$ denotes the solution approximation error. Two types of solution approximation errors – discretization error and surrogate modeling error – are considered in this chapter. However, the integration methodology is general enough to accommodate other types of solution approximation errors, once such errors can be quantified.

In general, solution approximation errors ($G_{se}(\mathbf{x}; \boldsymbol{\theta})$) are deterministic quantities; however, sometimes, when probabilistic approaches are used to quantify them, it becomes necessary to represent solution approximation errors using probability distributions. For example, the discretization error in finite element analysis is a deterministic quantity. When Richardson extrapolation [42] is used to quantify this error, then $G_{se}(\mathbf{x}; \boldsymbol{\theta})$ can be calculated deterministically. When a probabilistic method, such as Gaussian process modeling [34, 43], is used, both the mean and variance of $G_{se}(\mathbf{x}; \boldsymbol{\theta})$ are predicted; this variance is simply dependent on the training points used to train the Gaussian process model and on how far it is necessary to extrapolate, and it does not imply that $G_{se}(\mathbf{x}; \boldsymbol{\theta})$ is a physically random quantity. Similarly, when a high fidelity simulation (such as a computationally intensive finite element analysis) is replaced by an inexpensive surrogate model, the surrogate model error (i.e., the difference between the prediction of the original high fidelity simulation and the surrogate model) is a deterministic quantity. However, the solution of the underlying high fidelity simulation is only available at a few input settings. The surrogate model prediction and the error (at untrained input locations) are therefore expressed using a probability distribution; this probability distribution only indicates the analyst's uncertainty regarding the surrogate model error and does not imply that the surrogate model error is random. This uncertainty is epistemic and reduces as the number of training points increases. The Bayesian framework represents such epistemic uncertainty through probability distributions.

In this chapter, Richardson extrapolation is used to compute discretization error, and Gaussian process surrogate models are used to replace high fidelity simulations. While the former leads to deterministic error estimates ($G_{se}(\mathbf{x}; \boldsymbol{\theta})$ is point-valued), the latter leads to stochastic error representation ($G_{se}(\mathbf{x}; \boldsymbol{\theta})$) is a probability distribution). In practical engineering problems, both discretization and surrogate modeling may be necessary, and therefore, the overall solution approximation error is composed of both deterministic and stochastic terms. In the context of uncertainty propagation, deterministic errors are addressed by correcting the bias, whenever they occur, and the corrected solutions are used to train the surrogate model; the

stochastic errors of the surrogate model are accounted for through sampling based on their estimated distributions. As a result, the overall solution approximation error $G_{se}(\mathbf{x}; \boldsymbol{\theta})$ is also stochastic, i.e., y is stochastic even for given values of \mathbf{x} and $\boldsymbol{\theta}$, and this stochasticity, without physical randomness, also needs to be interpreted subjectively. The remainder of this subsection briefly reviews the estimation of discretization error and surrogate model uncertainty.

Recall that verification is performed before calibration and validation in the approach of this chapter. Once the solution approximation errors are quantified, the model predictions are corrected as explained above; the corrected solution (y) and not the code output (y_c) is used for calibration and validation. Such an approach integrates the result of verification into calibration, validation, and all subsequent uncertainty quantification.

2.1.1 Discretization Error

Several methods are available in the literature [36, 44, 45] to estimate discretization error in finite element analysis, but many of them only quantify a surrogate measure of error to facilitate adaptive mesh refinement. The Richardson extrapolation (RE) method has been found to come closest to quantifying the actual discretization error [32, 42]. This technique has been commonly applied to quantifying discretization error in finite element analysis by several researchers [2, 24, 25, 28].

Consider a polynomial model $y = y_c + Ah^p$, where y_c is the solution corresponding to mesh size h , and y corresponds to the “true” solution of the mathematical equation which is obtained as h tends to zero. Three different mesh sizes ($h_1 < h_2 < h_3$) are considered, and the corresponding finite element solutions ($y_c(h_1) = \Psi_1$, $y_c(h_2) = \Psi_2$, $y_c(h_3) = \Psi_3$) are calculated. Using the aforementioned polynomial model, y can be estimated by solving three simultaneous equations in three variables. Closed-form solutions are available in some special cases; for example, if $r = \frac{h_3}{h_2} = \frac{h_2}{h_1}$, then the discretization error (ϵ_h) and the true solution can be calculated as:

$$\begin{aligned} y &= \Psi_1 - \epsilon_h \\ \Psi_2 - \Psi_1 &= \epsilon(r^p - 1) \\ p \log(r) &= \log\left(\frac{\Psi_3 - \Psi_2}{\Psi_2 - \Psi_1}\right) \end{aligned} \tag{11.1}$$

The solutions Ψ_1 , Ψ_2 , and Ψ_3 are dependent on both \mathbf{x} and $\boldsymbol{\theta}$, and hence the error estimate ϵ_h and the true solution y are also functions of both \mathbf{x} and $\boldsymbol{\theta}$. Since the discretization error is a deterministic quantity, it needs to be corrected for, in the context of uncertainty propagation.

Recently, Rangavajhala et al. [34] extended the Richardson extrapolation methodology from a polynomial relation to a more flexible Gaussian process extrapolation. This approach expresses the discretization error as a probability distribution, and therefore, the training points (in particular, the output values)

for the surrogate model are themselves stochastic. Rasmussen [46–48] discusses constructing GP models when the training point values are themselves stochastic.

2.1.2 Surrogate Model Uncertainty

This section considers the case where the original computer code is replaced with a Gaussian process surrogate model. The basic idea of the GP model is that the response values Y evaluated at different values of the input variables \mathbf{x} , are modeled as a Gaussian random field, with a mean and covariance function. Suppose that there are m training points, $x_1, x_2, x_3, \dots, x_m$ of a d -dimensional input variable vector, yielding the output values $Y(x_1), Y(x_2), Y(x_3), \dots, Y(x_m)$. The training points can be compactly written as x_T vs. y_T where the former is a $m \times d$ matrix and the latter is a $m \times 1$ vector. Suppose that it is desired to predict the response (output values y_P) corresponding to the input x_P , where x_P is $p \times d$ matrix; in other words, it is desired to predict the output at p input combinations simultaneously. Then, the joint density of the output values y_P can be calculated as

$$p(y_P | x_P, x_T, y_T; \Theta) \sim N(m, s) \quad (11.2)$$

where Θ refers to the hyper-parameters of the Gaussian process, which need to be estimated based on the training data. The prediction mean and covariance matrix (m and S respectively) can be calculated as

$$\begin{aligned} m &= K_{PT}(K_{TT} + \sigma_n^2 I)^{-1}y_T \\ S &= K_{PP} - K_{PT}(K_{TT} + \sigma_n^2 I)^{-1}K_{TP} \end{aligned} \quad (11.3)$$

In Eq. (11.3), K_{TT} is the covariance function matrix (size $m \times m$) among the input training points (x_T), and K_{PT} is the covariance function matrix (size $p \times m$) between the input prediction point (x_P) and the input training points (x_T). These covariance matrices are constructed using the chosen covariance function (squared exponential is chosen, in this chapter), which are functions of the training points terms and the hyper-parameters (Θ): a multiplicative term (θ), the length scale in all dimensions (l_q , $q = 1$ to d), and the noise standard deviation (σ_n). These hyper-parameters are estimated based on the training data by maximizing the following log-likelihood function:

$$\log p(y_T | x_T; \Theta) = -\frac{y_T^T}{2}(K_{TT} + \sigma_n^2 I)^{-1}y_T - \frac{1}{2} \log |(K_{TT} + \sigma_n^2 I)| + \frac{d}{2} \log(2\pi) \quad (11.4)$$

Once the hyper-parameters are estimated, then the Gaussian process model can be used for predictions using Eq. (11.3). For details of this method, refer to [46, 48–54].

Note that the variance in Eq. (11.3) is representative of the uncertainty due to the use of the GP surrogate model, and this uncertainty needs to be accounted for, while computing the overall prediction uncertainty.

2.2 Calibration

The purpose of model calibration is to adjust a set of parameters associated with a computational model so that the agreement between model prediction and experimental observation is maximized [55]. Least squares [56], likelihood-based [57, 58], and Bayesian [11, 15, 59–63] methods are available for model parameter estimation. In classical statistics, the fundamental assumption is that the parameter is a deterministic unknown quantity, and it is not meaningful to discuss the probability distribution of the parameter; therefore, the uncertainty about the value of the parameter is expressed in terms of confidence intervals. On the other hand, the Bayesian approach attributes a probability distribution (prior and posterior) to the model parameters, and this distribution represents the analyst’s uncertainty about the model parameter.

The model parameters ($\boldsymbol{\theta}$) may be calibrated with input-output (x versus y) data collected for calibration (D^C), using Bayes’ theorem as

$$f_{\Theta}(\boldsymbol{\theta}|G, D^c) = \frac{L(\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})d\boldsymbol{\theta}} \quad (11.5)$$

In Eq. (11.5), $f_{\Theta}(\boldsymbol{\theta})$ is the prior PDF and $f_{\Theta}(\boldsymbol{\theta})$ is the prior PDF and $f_{\Theta}(\boldsymbol{\theta}|G, D^c)$ is the posterior PDF; the calibration procedure uses the model form G and hence the posterior is conditioned on G . The function $L(\boldsymbol{\theta})$ is the likelihood of $\boldsymbol{\theta}$ defined as being proportional to the probability of observing the data D^C (given as x_i vs. y_i ; $i = 1$ to n) conditioned on the parameters $\boldsymbol{\theta}$. This likelihood function is evaluated as

$$L(\boldsymbol{\theta}) \propto \prod_{i=1}^n \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(y_i - G(\mathbf{x}_i, \boldsymbol{\theta}))^2}{2\sigma^2}\right) \quad (11.6)$$

where σ is the standard deviation of $\epsilon_{obs} = y - G(\mathbf{x}, \boldsymbol{\theta})$. Note that the likelihood is constructed using the actual solution of the mathematical equation (y), i.e., after correcting the raw code prediction (y_c) with solution approximation errors (ϵ_{soln}). Sometimes, σ is also inferred along with $\boldsymbol{\theta}$, by constructing the joint likelihood $L(\boldsymbol{\theta}, \sigma)$. Note that Eq. (11.6) assumes that the experimental observations are unbiased. If the predictions are biased due to modeling errors, then the output can be modeled as $y = G(\mathbf{x}, \boldsymbol{\theta}) + \delta + \epsilon_{obs}$, where δ represents the modeling error and is inferred along with $\boldsymbol{\theta}$. This approach was further extended by Kennedy and O’Hagan [30] by modeling the output as $= G(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon_{obs}$, where $\delta(\mathbf{x})$ (sometimes, referred to as the model inadequacy function) is represented using a Gaussian process (GP) whose “hyper-parameters” are also inferred along with $\boldsymbol{\theta}$. A challenge with the Kennedy O’Hagan (KOH) framework is that it is necessary to (1) simultaneously calibrate both the model parameters and the GP hyper-parameters and (2) possess good prior knowledge regarding both the model parameters and the GP hyper-parameters. This is a challenging issue and still being addressed by several researchers [64, 65]. Using the KOH framework, all the parameters to

calibrate include (1) model parameters θ , (2) hyper-parameters in the surrogate model for $G(\theta; \mathbf{x})$ if it is replaced by a surrogate model, (3) hyper-parameters θ_δ of the model discrepancy $\delta(\mathbf{x})$, and (4) standard deviation σ of ϵ_{obs} . The presence of so many calibration parameters is challenging if calibration data are sparse. All the available data have been used in [15] to simultaneously estimate both the model parameters and the model discrepancy in the calibration step. An alternative approach is to consider separate sets of data for calibration and validation; first, the model parameters are estimated (with or without including the model discrepancy term) in the calibration step, and then, the effect of model form error is inferred in the validation step by calculating a validation metric. The latter route is pursued in this chapter, in an effort to distinctly separate the two steps of calibration and validation and use different sets of data for these two activities. In addition, this chapter ignores the hyper-parameter uncertainty in the GP model of $G(\theta; \mathbf{x})$ for three reasons: (1) enough training points are used to build an accurate GP model with small variance in the GP prediction, thus the hyper-parameter uncertainty is expected to be small; (2) considering this hyper-parameter uncertainty will bring enormous computational effort [64] in model calibration and validation, whereas this hyper-parameter uncertainty is not the focus of this chapter; and (3) the uncertainty in the hyper-parameters is typically negligible compared to actual model parameters [66]. Thus we first estimate the hyper-parameters of the GP model and then fix them as deterministic values in the subsequent calibration of model parameters θ .

Note that the model “ G ” is used for calibration in Eq. (11.5); recall that “ G ” refers to the model corrected for solution approximation errors. Hence, the results of verification are included in the calibration procedure. Deterministic discretization errors are corrected for before training the surrogate model, and the surrogate model uncertainty is included in the likelihood function as demonstrated by Kennedy and O’Hagan [15] and McFarland [51]. In addition, the construction of the likelihood function can also include additional uncertainty in inputs and parameters and account for imprecise and unpaired data. Refer to [2, 9, 57, 67] for further details on the construction of the likelihood function in different types of situations.

The posterior PDFs of the model parameters can be calculated using direct integration of the denominator in Eq. (11.5), if the number of calibration parameters is small. Alternatively, Markov chain Monte Carlo sampling [68] methods such as Metropolis algorithm [69], Gibbs algorithm [70], or slice sampling [71] can be used to generate samples of the posterior distributions of the parameters.

2.3 Validation

Model validation refers to the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended use of the model [22, 72]. Generally model validation is realized by comparing the model prediction against experimental data. Different types of validation metrics have been studied in the literature to express the accuracy of a computational model through comparison of its prediction against observed data and to determine whether the

model is adequate for its intended use (sometimes referred to as qualification [25]). Coleman and Stern [73] and Oberkampf and Trucano [25] discussed conceptual and practical aspects of model validation and provided guidelines for conducting validation experiments and developing validation metrics. Available approaches for quantitative model validation are based on statistical confidence intervals [74], computing distance between the model prediction and experimental data by computing the area metric [9, 75], normalizing residuals [76], classical statistics-based hypothesis testing [77], Bayesian hypothesis testing [4, 78–81], and reliability analysis-based techniques [82–84]. Liu et al. [85] and Ling and Mahadevan [5] investigated several of these validation approaches in detail and discussed their practical implications in engineering.

The term “validation” has been used to imply different approaches in different studies. While some studies compute validation metrics, some other approaches focus on estimating the model discrepancy [15, 32] as the difference between the model prediction and the underlying physical phenomenon the model seeks to represent. Comprehensive reviews on model validation can be found in [25, 74, 86, 87]. The present chapter only focuses on computing validation metrics with validation data and does not estimate the model form error with validation data. The model discrepancy is computed through the use of a discrepancy function in the Kennedy O’Hagan framework during model calibration, a previous task. Separate sets of data are considered for calibration and validation.

In this chapter, model calibration and model validation are distinct activities. Model validation can be conducted exclusive of any model calibration [74] if the model parameters are assumed to be known. However, the model parameters θ are often unknown. Therefore, prior to model validation, model calibration can be conducted to quantify the values of θ or reduce the uncertainty about their values. Thus validation is a subsequent and distinct activity after validation. Both model calibration and model validation are conducted in this chapter, but they use different sets of experimental data (no calibration data is used in model validation), as suggested by [88]. The model parameters θ do not change as a result of model validation.

Among the validation methods mentioned above, this chapter uses Bayesian hypothesis testing and the model reliability metric for the purpose of uncertainty integration, since both methods give the validation result as a probability value. The Bayes factor metric in Bayesian hypothesis testing is the ratio of the likelihoods that the model is valid and that the model is invalid. The Bayes factor can be used to directly calculate the posterior probability that the model is valid. Further, the threshold Bayes factor for model acceptance can be derived based on a risk vs. cost trade-off, thereby aiding in robust, meaningful decision-making, as shown by Jiang and Mahadevan [89]. Alternatively, the model reliability metric [83] can also compute the probability that the model is valid.

2.3.1 Validation by Bayesian Hypothesis Testing

Assume that an independent set of validation data (D^V) is available. The prediction of the verified and calibrated model prediction is compared against the validation data. The model prediction can be computed as a function of input as

$$f_Y(y|\mathbf{x}, G, D^C) = \int f_Y(y|\mathbf{x}, \boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta}|G, D^C) d\boldsymbol{\theta} \quad (11.7)$$

In the case of partially characterized validation data (e.g., field data), the input \mathbf{x} may not be measured, in which case the model prediction must include the uncertainty in the input as

$$f_Y(y|G, D^C) = \int f_Y(y|\mathbf{x}, \boldsymbol{\theta}) f_X(\mathbf{x}) f_{\Theta}(\boldsymbol{\theta}|G, D^C) d\boldsymbol{\theta} d\mathbf{x} \quad (11.8)$$

The above equations simply refer to uncertainty propagation through the model, and hence the model prediction is conditioned on the event that the mathematical model is correct and written as $f_y(y|G, D^c)$. The results of verification are included while computing y , and the results of calibration are included by using the posterior PDF of the model parameter ($f_{\Theta}(\boldsymbol{\theta}|G, D^c)$) in the prediction.

The model prediction is then compared with the validation data using Bayesian hypothesis testing in this section. Let $P(G)$ and $P(G')$ denote the probabilities that the model is valid (null hypothesis) and that the model is invalid (alternate hypothesis), respectively. Prior to validation, if no information is available, $P(G) = P(G') = 0.5$. Using Bayesian hypothesis testing, these probabilities can be updated using the validation data (D^V), and the likelihood ratio, referred to as Bayes factor, is defined as

$$B = \frac{P(D^V|G)}{P(D^V|G')} \quad (11.9)$$

The likelihoods $P(D^V|G)$ and $P(D^V|G')$ are denoted as $L(G)$ and $L(G')$, respectively. The numerator $P(D^V|G)$ can be calculated using $f_y(y|G, D^c)$ as

$$L(G) \propto P(D^V|G) \propto \int f(D^V|y) f_Y(y|G, D^C) dy \quad (11.10)$$

In Eq. (11.10), the term $f(D^V|y)$ is calculated based on the measurement error $\epsilon_{obs} \sim N(0, \sigma^2)$, as

$$f(D^V|y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(D^V - y)^2}{2\sigma^2}\right) \quad (11.11)$$

In order to compute $P(D^V|G')$, it is necessary to assume the alternate PDF $f_Y(y|G')$, i.e., the PDF of Y when the model is wrong. Expert opinion may be used to construct this PDF, or a uniform PDF may be used if no additional information is available. (This issue is a limitation of this method.) Then $L(G')$ is calculated similar to Eq. (11.10) by replacing $f_Y(y|G, D^C)$ with $f_Y(y|G')$. Using Bayes' theorem and assuming that G and G' are equally likely before collecting

data, the probability that the model is correct, i.e., $P(G|D^V)$, can be calculated as $B/B + 1$ [32].

2.3.2 Model Validation by Model Reliability Metric

Model Reliability Metric

The Bayes factor gives a probabilistic measure of model validity. While the Bayesian hypothesis testing is one approach to calculate the probability that the model is correct, the model reliability metric [82, 83] provides an alternative methodology.

In the model reliability metric, the model is defined to be valid if the difference between the model prediction y and the corresponding validation measurement is less than a predefined tolerance λ . The value of λ is chosen by the user based on the accuracy requirement for the specific application. Due to the measurement error ($\epsilon_{obs} \sim N(0, \sigma^2)$), the measurement is actually a random variable. For a single observed value D , this random variable is denoted by d with mean value D and standard deviation σ , i.e., $d \sim N(D, \sigma^2)$. Let G denote the event that the model is valid and y denotes the prediction, then the model reliability is defined as the probability of event G :

$$P(G|D) = P(|y - d| < \lambda) \quad (11.12)$$

The probability in Eq. (11.12) is used as a metric to measure model validity; thus this metric is named as “model reliability metric.” If y and σ are deterministic, Eq. (11.13) computes the model reliability where ε is a dummy variable for integration:

$$P(G|D) = \int_{-\lambda}^{\lambda} \frac{1}{\sigma_m \sqrt{2\pi}} \exp \left[-\frac{(\varepsilon - (y - D))^2}{2\sigma_m^2} \right] d\varepsilon \quad (11.13)$$

Note here that the model prediction y refers to the computational model output $y = G(\mathbf{x}; \boldsymbol{\theta})$; this prediction may either include a model discrepancy term explicitly or not, depending on the situation. The discussion below is valid in both cases, and the numerical examples in this chapter illustrate both types of situations. Although model input \mathbf{x} is known, the model prediction y is still stochastic due to the uncertainty of $\boldsymbol{\theta}$. In this case, the model reliability is

$$P(G|D) = \int P(G|\boldsymbol{\theta}, D) f''(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (11.14)$$

where $P(G|\boldsymbol{\theta}, D)$ is given by the right side of Eq. (11.13) and $f''(\boldsymbol{\theta})$ is the joint posterior distribution of $\boldsymbol{\theta}$.

Equations (11.13) and (11.14) are only suitable for a single observed value D for a scalar output quantity. The concept of the model reliability metric can be extended to deal with multiple data points and multivariate output, as follows.

Stochastic Model Reliability Metric

The first extension is the stochastic model reliability metric. As shown in Eqs. (11.13) and (11.14), the value of model reliability $P(G)$ is deterministic at a single data point D , but changes over different data points. If model inputs \mathbf{x} of these data points are known, a mathematical function $P(G|\mathbf{x}) = s(\mathbf{x})$ can be established where $P(G|\mathbf{x})$ is the model reliability at model input \mathbf{x} . However, this function may be not accurate when the validation data is sparse. Thus constructing a mathematical function for model reliability (as a function of \mathbf{x}) is not considered in this chapter. Instead, this chapter uses a probability distribution to represent the variability in $P(G)$, and this distribution is constructed using the model reliability values computed at different validation data points (the first option could be considered if a large number of validation experiments are conducted).

In this chapter, model reliability $P(G)$ is assumed to have a beta distribution since $P(G) \in [0,1]$ and the sample space of beta distribution is also the interval $[0,1]$. Another option is the t -distribution [14]. If a data set $\mathbf{D} = \{D_1, D_2, \dots, D_n\}$ of one output quantity is observed for model validation from n experiments with different inputs, the corresponding model reliability values computed by Eq. (11.14) at each experiment are $\mathbf{D}_R = \{D_{R_1}, D_{R_2}, \dots, D_{R_n}\}$. Using \mathbf{D}_R , several methods can be used to construct the PDF of model reliability, such as the method of maximum likelihood, method of moments, or Bayesian inference. This chapter uses the method of moments to construct the PDF of $P(G)$. In summary, this approach gives a stochastic representation of model reliability, i.e., $P(G)$ is not a single value but represented by a probabilistic distribution.

Multivariate Output

In the case of multivariate output, if K output quantities are observed in a validation experiment, we have a set of K models sharing the same model input and model parameters:

$$\mathbf{y} = \mathbf{G}(\mathbf{x}; \boldsymbol{\theta}) \Leftrightarrow \begin{cases} y_1 = G_1(\mathbf{x}; \boldsymbol{\theta}) \\ y_2 = G_2(\mathbf{x}; \boldsymbol{\theta}) \\ \dots \\ y_K = G_K(\mathbf{x}; \boldsymbol{\theta}) \end{cases} \quad (11.15)$$

where $G_j(\mathbf{x}; \boldsymbol{\theta})$ ($j = 1$ to K) is the computational model of the j th quantity (as mentioned earlier in Sect. 2.3, the prediction may explicitly or implicitly include the effect of model discrepancy). Each quantity also has a measurement error $\epsilon_{obs_j} \sim N(0, \sigma_j^2)$ and the corresponding variable $Z_j = y_j + N(0, \sigma_j^2)$ representing the measurement. We denote $\mathbf{Z} = \{Z_1, \dots, Z_j, \dots, Z_k\}^T$. Assume that n experiments are conducted. In the i th experiment ($i = 1$ to n), data points for k quantities form

a data set $\mathbf{D}_i = \{D_{i1}, \dots, D_{ij}, \dots, D_{ik}\}^T$. In addition, the predefined tolerance for each quantity is included in a vector $\boldsymbol{\lambda} = \{\lambda_1, \dots, \lambda_j, \dots, \lambda_k\}^T$.

The distance between \mathbf{Z} and \mathbf{D}_i can be measured by multiple distance functions such as the Euclidean distance, Chebyshev distance, Manhattan distance, and Minkowski distance [90]. This chapter uses the Mahalanobis distance [91]. The Mahalanobis distance between \mathbf{Z} and \mathbf{D}_i is defined as $M = \sqrt{(\mathbf{Z} - \mathbf{D}_i)^T \boldsymbol{\Sigma}_Z^{-1} (\mathbf{Z} - \mathbf{D}_i)}$ where $\boldsymbol{\Sigma}_Z$ is the covariance matrix of \mathbf{Z} . The Mahalanobis distance transfers \mathbf{Z} and \mathbf{D}_i into the normalized principal component (PC) space [91] by using $\boldsymbol{\Sigma}_Z^{-1}$. Compared to other distance functions, the Mahalanobis distance brings two advantages: (1) the correlations between output quantities are considered, and (2) the output quantities are normalized to the same scale to prevent any quantity from dominating the metric simply due to large numerical values. Using the Mahalanobis distance, the model reliability for multivariate output is defined as

$$P(G|\mathbf{D}_i) = P(M < \lambda_M) = P\left(\sqrt{(\mathbf{Z} - \mathbf{D}_i)^T \boldsymbol{\Sigma}_Z^{-1} (\mathbf{Z} - \mathbf{D}_i)} < \sqrt{\lambda^T \boldsymbol{\Sigma}_Z^{-1} \lambda}\right) \quad (11.16)$$

where $\lambda_M = \sqrt{\lambda^T \boldsymbol{\Sigma}_Z^{-1} \lambda}$ is the normalized tolerance.

Generally the posterior distributions obtained in model calibration are numerical samples generated by MCMC, so the subsequent model reliability in Eqs. (11.13) and (11.14) is also computed numerically. Numerical computation also facilitates the realization of the extended model reliability in Eq. (11.16). Here the model reliability is expressed as

$$\begin{aligned} P(G|\mathbf{D}_i) &= P(M < \lambda_M|\mathbf{D}_i) = \int_0^{\lambda_M} f(M|\mathbf{D}_i) dM \\ &= \int_0^{\lambda_M} (f(M|\mathbf{D}_i, \theta) f''(\theta) d\theta) dm \end{aligned} \quad (11.17)$$

Eq. (11.17) indicates a numerical algorithm to compute the model reliability:

1. Generate a random sample of $\boldsymbol{\theta}$ from its posterior distribution $f''(\boldsymbol{\theta})$;
2. Generate a sample of M conditioned on $\boldsymbol{\theta}$ by generating a sample of and computing its Mahalanobis distance from \mathbf{D}_i ;
3. Repeat steps 1 and 2 to obtain N samples of M ; these samples can be used to construct the distribution $f(M|\mathbf{D}_i)$, which is not conditioned on $\boldsymbol{\theta}$;
4. If N' out of N samples in step 3 satisfy $M < \lambda_M$, the model reliability is $P(G|\mathbf{D}_i) = N'/N$

The model reliability $P(G|\mathbf{D}_i)$ by Eq. (11.17) is regarding a single experiment and $P(G|\mathbf{D}_i)$ is a deterministic value. Thus n experiments will give n different

model reliability values $P(G|D_1), \dots, P(G|D_n)$. Using the stochastic model reliability metric, these values can be used to build a probability distribution for the model reliability $P(G)$, by treating $P(G)$ as a random variable instead of a deterministic value.

2.4 Integration for Overall Uncertainty Quantification

The calibration procedure in Sect. 2.2 assumed that the model form G is valid and estimated the model parameters θ . In contrast, the validation procedure in Sect. 2.3.1 calculated the probability that the model G is valid by assuming the uncertainty in the model parameters θ . The two results can be combined to calculate the overall uncertainty in the model prediction, using the theorem of total probability as

$$f_Y(y|D^C, D^V) = P(G|D^V) f_Y(y|G, D^C) + P(G') f_Y(y|G') \quad (11.18)$$

In Eq. (11.18), $P(G|D^V)$ can also be calculated using the model reliability metric instead of Bayesian hypothesis testing. Note that the result of verification, i.e., solution approximation error, was already included in both calibration and validation. Thus, the PDF $f_Y(y|D^C, D^V)$ includes the results of verification, calibration, and validation activities.

3 Numerical Example: Single-Level Model

This section discusses a numerical example, where a single-level model is subject to verification, validation, and calibration. The results of these activities are integrated to calculate the overall uncertainty in the response quantity.

3.1 Description of the Problem

Consider steady state heat transfer in a thin wire of length L , with thermal conductivity k and convective heat coefficient β . Assume that the heat source is $Q(x) = 25(2x - L)^2$, where x is measured along the length of the wire. For the sake of illustration, it is assumed that this problem is essentially one dimensional and that the solution can be obtained from the following boundary value problem [32]:

$$\begin{aligned} -k \frac{\partial^2 T}{\partial x^2} + \beta T &= Q(x) \\ T(0) &= T_0 \\ T(L) &= T_L \end{aligned} \quad (11.19)$$

The length of the wire is assumed to be deterministic ($L = 4\text{ m}$). The boundary conditions, i.e., the temperatures at the ends of the wire ($T(0)$ and $T(L)$), are assumed to be normally distributed with statistics $N(0, 1)$. The thermal conductivity of the wire (k) is assumed to be normally distributed $N(5, 0.2)$ with units $\text{Wm}^{-1}/^\circ\text{C}$. The convective heat coefficient (β) is an unknown parameter which needs to be estimated using calibration data (D^C); this quantity is assumed to have a normally distributed prior as $N(0.5, 0.05)$. The goal of the model is to predict the temperature (Y) at the midpoint of the wire.

3.2 Verification, Validation, and Calibration

First, the differential equation in Eq. (11.19) is solved using a finite difference code. Three different discretization sizes are considered, and Richardson extrapolation [42] is used to calculate the solution approximation error which is used to correct the model prediction every time this differential equation is solved. Since there are four uncertain quantities (T_0 , T_L , k , and the model parameter being updated, i.e., β), it is necessary to quantify the solution approximation error as a function of these four quantities. Every time the model prediction Y needs to be computed as a function of these four quantities, a mesh refinement study is conducted, i.e., three different mesh sizes are considered (with sizes 0.01, 0.005, and 0.0025), and Eq. 11.1 is used to compute the solution approximation error and, hence, the corrected prediction.

It may be noted that, for this particular numerical example, linear diffusion dominates and therefore, the solution approximation error is not very sensitive to k . However, this behavior is not explicitly considered during verification. This is because of two reasons. First, it may be recalled that, in order to integrate the results of verification into calibration and validation, the solution approximation errors are computed and corrected whenever needed while performing calibration (update β using data $D^C = \{22; 23; 25; 261; 254\}$, in $^\circ\text{C}$) and validation. In other words, during calibration, samples of the aforementioned four uncertain quantities are generated through slice sampling; for each generated sample, the solution approximation error is quantified through mesh refinement. Since it is necessary to repeat this procedure for every generated sample, it does not matter whether the nominal value of k or the actually sampled value of k is used, from a computational point of view. The second reason is that the underlying physics behavior and the choice of numerical values are not exploited during the implementation of the methodology. The code used to solve the differential equation in Eq. (11.19) is treated as if it were a black box model, in an effort to keep the illustration as general as possible. In specific problems, special features may be exploited to reduce computational effort.

The prior ($f_\Theta(\theta)$) and posterior ($f_\Theta(\theta|G, D^C)$) PDFs of β are shown in Fig. 11.3. Additional validation data ($D^V = \{24; 245; 246; 238\}$, in $^\circ\text{C}$) is used to compute the probability that the temperature prediction model is correct, i.e., $P(G) = 0.84$.

Fig. 11.3 PDF of convective heat coefficient (β)

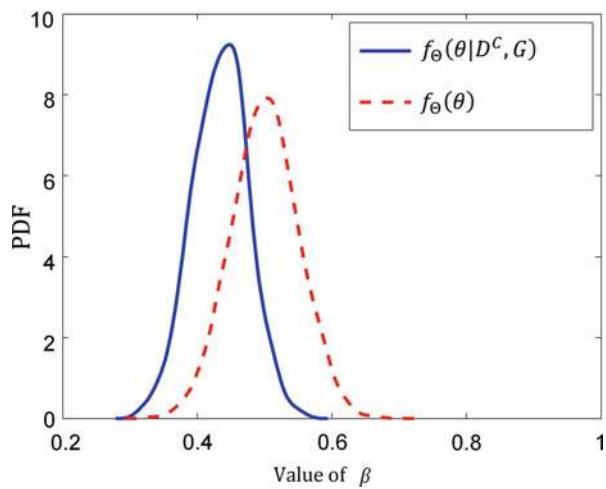
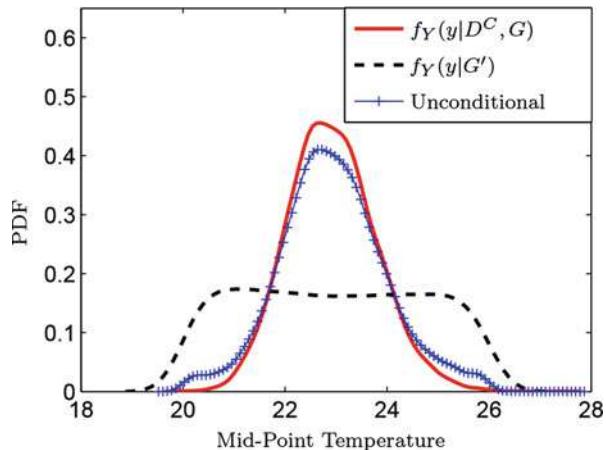


Fig. 11.4 PDF of midpoint temperature



3.3 Integration and Overall Uncertainty Quantification

The method developed in Sect. 2.4 is used to calculate the unconditional PDF of temperature using the principle of total probability, as shown in Fig. 11.4. This PDF integrates the results of verification, validation, and calibration to compute the overall uncertainty in the temperature at the midpoint of the wire.

Figure 11.4 indicates three PDFs: (i) $f_Y(y|G, D^C)$ denotes the model prediction, (ii) $f_Y(y|G')$ denotes the prediction under the alternate hypothesis (assumed uniform; due to sampling errors and use of kernel density estimation for plotting, the PDF is not perfectly horizontal in Fig. 11.4), and (iii) $f_Y(y|G, D^C, D^V)$ which represents the PDF that integrates the validation result with the previous calibration and verification activities. The third PDF is referred to as the unconditional PDF of the temperature response, since it is not conditioned on the model form.

Conventionally, the model prediction is used for performance prediction, failure analysis, and reliability analysis. Since failures are, generally, events with low probabilities of occurrence, it is important to be able to accurately capture tail probabilities in order to predict failures. For example, if the component is assumed to fail when the temperature is greater than 25 °C, then the model prediction PDF gives the failure probability as 0.0135, whereas the unconditional PDF gives the failure probability as 0.0390. Thus, it is clear that, using the raw model prediction (i.e., by simply considering the calibrated model, without accounting for the result of validation) underestimates the failure probability, whereas the approach outlined in Sect. 2 and followed in this example systematically includes the effect of model uncertainty in reliability analysis by integrating verification, validation, and calibration during system-level uncertainty quantification.

4 Multilevel Models with Type-I Interaction

Consider a system that is studied using multiple levels of models with Type-I interaction, i.e., the output of a lower-level model becomes an input to the higher-level model and hence is the linking variable between the two models. While the methods of verification, validation, and calibration can be applied to each of the individual models, the challenge is to integrate the results from these activities performed at multiple levels. This section discusses a methodology for the integration of verification, validation, and calibration across multiple levels of modeling with Type-I interaction. The methodology in this section is illustrated for two levels of models, as shown in Fig. 11.5 and Eq. (11.20), but can be extended to any number of levels of modeling without loss of generality.

$$\begin{aligned} Y &= G(X; \theta) \\ Z &= H(Y, W; \alpha) \end{aligned} \quad (11.20)$$

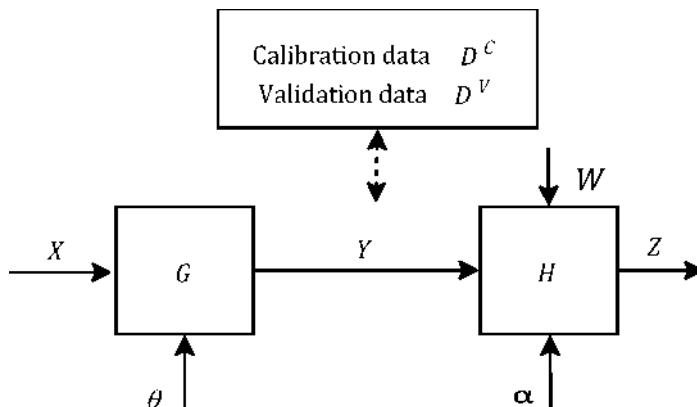


Fig. 11.5 Type-I interaction: two levels of models

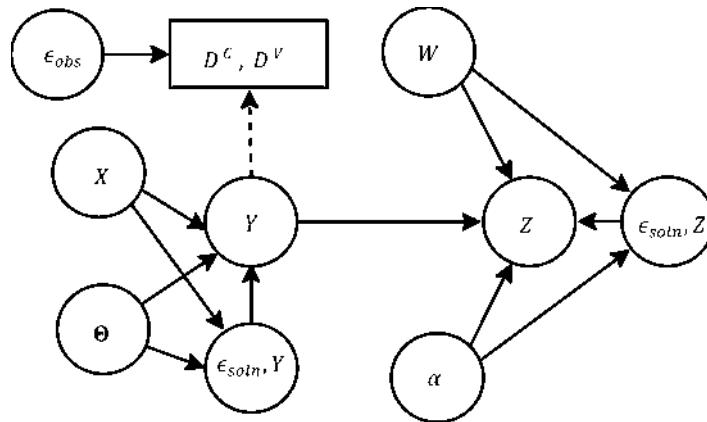


Fig. 11.6 Bayesian network: Type-I interaction between two models

Assume that data are not available at the system level, i.e., it is not possible to validate/calibrate model H . Let D^C and D^V denote the data available on Y for calibration (of θ) and validation (of G), respectively. Let $\epsilon_{obs} \sim N(0, \sigma_{obs}^2)$ denote the measurement errors in the data.

The first step is to connect the various sources of uncertainty using a Bayesian network, as shown in Fig. 11.6.

This Bayesian network indicates that two sets of data are available for calibration and validation; the Bayesian methods for calibration and validation can be applied to these sets. (If the KOH framework is pursued for calibration, then both the parameters and the model inadequacy function can be included in the Bayesian network.) While Bayesian updating is used for model calibration, either Bayesian hypothesis testing or the model reliability metric can be pursued for model validation.

The task is to compute the overall uncertainty in Z by using lower-level data; this uncertainty must include the effect of verification, calibration, and validation activities.

4.1 Verification, Calibration, and Validation

Both the models G and H can be verified since experimental data are not required for verification. During the process of verification, the solution approximation error (ϵ_{soln}) is quantified for both the models G and H . Note that the solution approximation error is a function of the inputs and the model parameters. Note that these solution approximation errors (ϵ_{soln} for both G and H) account for the combined effect of both deterministic and stochastic errors, as discussed in Sect. 2.1.

Now the Bayesian network includes quantification of solution approximation error, and it can now be used for calibration, validation, and system-level prediction.

The next step is to calibrate the model parameters. Suppose that the PDFs of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are assumed to be $f_{\Theta}(\boldsymbol{\theta})$ and $f_{\alpha}(\boldsymbol{\alpha})$ before any testing; these are the prior PDFs. Since no data are available on Z , it is not possible to update the PDF of $\boldsymbol{\alpha}$. The data on Y , i.e., D^C , is used to calibrate the parameters $\boldsymbol{\theta}$, using Bayesian inference, as in Sect. 2.2. The calibration procedure uses the data and assumes that the model is correct, and hence the posterior PDF of $\boldsymbol{\theta}$ is denoted by $f_{\Theta}(\boldsymbol{\theta}|G, D^C)$. During the calibration procedure, for every realization of $(\boldsymbol{\theta})$, the corresponding solution approximation error is estimated, and therefore, calibration is based on comparing y against experimental data, rather than y_c , thereby accounting for the results of verification during calibration.

Additional independent data (D^V) is assumed to be available for the purpose of validating the model G . The alternate hypothesis PDF $f_Y(y|G')$ is assumed, and the posterior probability of model being correct, i.e., $P(G|D^V)$, is calculated as explained in Sect. 2.3.1; alternatively, the model reliability metric $P(M)$ can also be used instead of $P(G|D^V)$.

4.2 Integration for Overall Uncertainty Quantification

The Bayesian network can be used for forward propagation of uncertainty using the principles of conditional probability and total probability. Prior to the collection of any data, the uncertainty in x , $\boldsymbol{\theta}$, and $\boldsymbol{\alpha}$ can be propagated through the models as

$$\begin{aligned} f_Z(Z|H) &= \int f_Z(Z|\mathbf{w}, \boldsymbol{\alpha}, y, H) f_W(\mathbf{w}) f_{\alpha}(\boldsymbol{\alpha}) f_Y(y|G) d\mathbf{w} d\boldsymbol{\alpha} dy \\ f_Y(y|G) &= \int f_Y(y|x, \boldsymbol{\theta}, G) f_X(x) f_{\Theta}(\boldsymbol{\theta}) dx d\boldsymbol{\theta} \end{aligned} \quad (11.21)$$

However, this procedure assumes that (1) the PDFs of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are $f_{\Theta}(\boldsymbol{\theta})$ and $f_{\alpha}(\boldsymbol{\alpha})$ and (2) the models G and H are correct. These two issues were addressed in calibration and validation, respectively. While the PDF of $\boldsymbol{\alpha}$ did not change, the PDF of $f_{\Theta}(\boldsymbol{\theta})$ was updated to $f_{\Theta}(\boldsymbol{\theta}|G, D^C)$. Further, the probability that G is correct, i.e., $P(G|D^V)$, was evaluated. These two quantities can now be used to calculate the overall uncertainty in Z . First, if the calibration data alone was used, then the PDFs of Y and Z are given by

$$\begin{aligned} f_Z(Z|G, H, D^C) &= \int f_Z(Z|\mathbf{w}, \boldsymbol{\alpha}, y, H) f_W(\mathbf{w}) f_{\alpha}(\boldsymbol{\alpha}) f_Y(y|G, D^C) d\mathbf{w} d\boldsymbol{\alpha} dy \\ f_Y(y|G, D^C) &= \int f_Y(y|x, \boldsymbol{\theta}, G) f_X(x) f_{\Theta}(\boldsymbol{\theta}|G, D^C) dx d\boldsymbol{\theta} \end{aligned} \quad (11.22)$$

The theorem of total probability can then be used to include the result of validation. The PDF of Y is modified as

$$f_Y(y|D^C, D^V) = P(G|D^V) f_Y(y|G, D^C) + P(G'|D^V) f_Y(y|G') \quad (11.23)$$

The overall uncertainty in Z , which includes the results of verification, calibration, and validation, can be calculated as

$$\begin{aligned} f_Z(Z|H, D^C, D^V) &= P(G|D^v) f_Z(Z|G, H, D^C) + P(G'|D^V) f_Z(Z|G', H) \\ f_Z(Z|G', H) &= \int f_Z(Z|\boldsymbol{w}, \boldsymbol{\alpha}, y, H) f_{\boldsymbol{w}}(\boldsymbol{w}) f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) f_Y(y|G') d\boldsymbol{w} d\boldsymbol{\alpha} dy \end{aligned} \quad (11.24)$$

The PDF of Z is still conditioned on H because it is assumed that the model H is correct and it is not possible to calibrate/validate this model. In fact, Eq. (11.24) is equivalent to simply propagating the PDF $f_Y(y|D^C, D^V)$ (in Eq. (11.23)) through the model H . Note that the model H has been verified; therefore, during uncertainty propagation, it is necessary to estimate and account for the solution approximation error, thereby including the result of verification of H during calibration, validation, and response computation. Thus, the PDF of the linking variable can be used to compute the uncertainty in the system-level response, thereby integrating the results of verification, validation, and calibration activities at a lower level.

The principles of conditional probability and total probability can also be extended to multiple models that exhibit Type-I interaction, as explained below.

4.3 Extension to Multiple Models

Until now, only the first model G was considered for verification, validation, and calibration. However, the methodology is general and can be extended to multiple models. For example, consider the case where there are two models whose individual outputs become inputs for the system model. For example, consider the equations:

$$\begin{aligned} Y_1 &= G_1(X_1, \theta_1) \\ Y_2 &= G_2(X_2, \theta_2) \\ Z &= H(Y_1, Y_2) \end{aligned} \quad (11.25)$$

The inputs to the models G_1 and G_2 are X_1 and X_2 , respectively; the corresponding parameters are θ_1 and θ_2 , respectively. The Bayesian network for this multilevel system is shown in Fig. 11.7.

Assume that there is no data at the system level Z , but data are available for calibration and validation of lower-level models G_1 and G_2 , as shown in the Bayesian network in Fig. 11.7. Using the calibration data, the PDFs $f(\theta_1|G_1, D_1^c)$, $f(\theta_2|G_2, D_2^c)$, $f(y_1|G_1, D_1^c)$, and $f(y_2|G_2, D_2^c)$ are calculated. Using the validation data, the probability $P(G_1|D_1^v)$ and $P(G_2|D_2^v)$ are calculated; further $P(G'_1|D_1^v) = 1 - P(G_1|D_1^v)$ and $P(G'_2|D_2^v) = 1 - P(G_2|D_2^v)$. As explained earlier, the probabilities that G_1 and G_2 are correct can also be calculated using the reliability-based metric.

The unconditional PDF of Z needs to be calculated by considering four quantities:

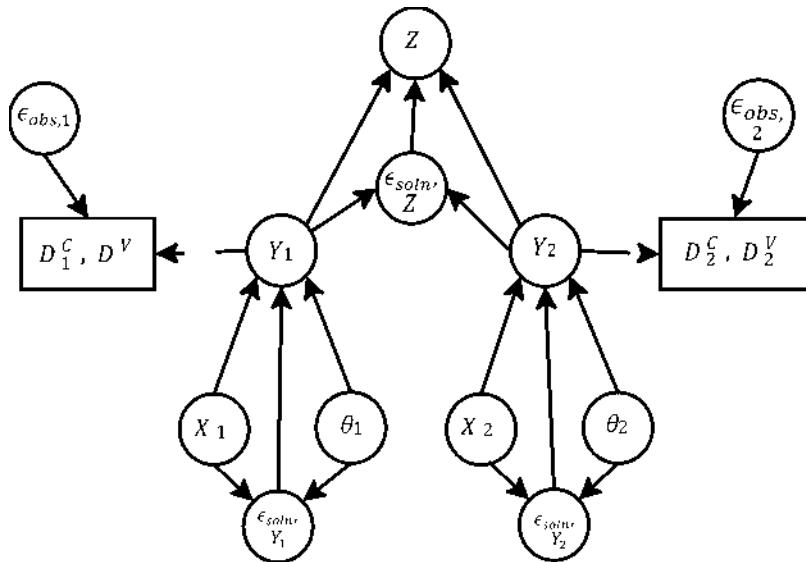


Fig. 11.7 Bayesian network: Type-I interaction between multiple models

1. $P(G_1 \cap G_2 | D_1^V, D_2^V) = P(G_1 | D_1^V) P(G_2 | D_2^V)$
2. $P(G_1 \cap G'_2 | D_1^V, D_2^V) = P(G_1 | D_1^V) P(G'_2 | D_2^V)$
3. $P(G'_1 \cap G_2 | D_1^V, D_2^V) = P(G'_1 | D_1^V) P(G_2 | D_2^V)$
4. $P(G'_1 \cap G'_2 | D_1^V, D_2^V) = P(G'_1 | D_1^V) P(G'_2 | D_2^V)$

Note the assumption that the two models G_1 and G_2 are independent. If the dependence is known, then it can be included in the calculation of the joint probabilities. Then, the unconditional PDF of Z is written as

$$\begin{aligned}
 f_Z(Z | D_1^C, D_1^V, D_2^C, D_2^V, H) \\
 &= P(G_1 | D_1^V) P(G_2 | D_2^V) f_Z(Z | G_1, G_2, H) \\
 &\quad + P(G'_1 | D_1^V) P(G_2 | D_2^V) f_Z(Z | G'_1, G_2, H) \\
 &\quad + P(G_1 | D_1^V) P(G'_2 | D_2^V) f_Z(Z | G_1, G'_2, H) \\
 &\quad + P(G'_1 | D_1^V) P(G'_2 | D_2^V) f_Z(Z | G'_1, G'_2, H)
 \end{aligned} \tag{11.26}$$

In Eq. (11.26), $f_Z(Z | G_1, G_2, H)$ is calculated by propagating the posteriors of Y_1 and Y_2 through H , since both the models are correct; $f_Z(Z | G'_1, G_2, H)$ is calculated by propagating the alternate PDF of Y_1 and the posterior of Y_2 through H , since only G_2 is correct; similarly, $f_Z(Z | G_1, G'_2, H)$ is calculated by propagating the posterior of Y_1 and alternate PDF of Y_2 , and $f_Z(Z | G'_1, G'_2, H)$ is calculated by propagating the alternate PDFs of Y_1 and Y_2 . If there are more than two lower-

level models (G_1 , G_2 , G_3 and so on), the number of terms on the right hand side of Eq. (11.26) increases exponentially, since the number of terms will be equal to 2^{n_m} where n_m is the number of models. Each of these terms indicates which subset of the models is correct. For example, in the case of three models, the event “ $G_1 \cap G'_2 \cap G'_3$ ” indicates that the model G_1 is correct, whereas the models G_2 and G_3 are not. Similarly, the event “ $G_1 \cap G_2 \cap G'_3$ ” indicates that the model G_2 is correct, whereas the models G_1 and G_3 are not. Though Eq. (11.26) clearly distinguishes between all such possibilities, the computation of the right hand side of this equation may be cumbersome due to the exponential number of terms. Such computational complexity can be easily avoided by first computing the unconditional PDFs of the lower-level outputs ($f(y_1|D_1^c, D_1^v)$ and $f(y_2|D_2^c, D_2^v)$ in this case) similar to Eq. (11.23) and then propagating these PDFs through the model H . Both the approaches will yield the same resultant PDF of Z , which accounts for the results of verification, validation, and calibration activities in both the models G_1 and G_2 .

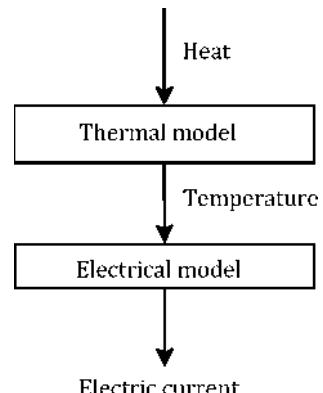
5 Numerical Example: Two Models with Type-I Interaction

This section illustrates two models that represent thermal and electrical analyses, with Type-I interaction. This example is an extension of the heat conduction problem in Sect. 3; the temperature rise in the wire causes change in the electrical resistance. The goal is to predict the system response, which is the electric current in the wire. Hence, the output of the lower-level model (temperature predictor in Eq. (11.19)), i.e., temperature, becomes an input to a higher-level model (current predictor), as shown in Fig. 11.8.

Consider the same wire as in Sect. 3. Before application of the heat, the resistance of the wire is given in terms of the resistivity (ρ), the cross-section area (A), and length (L) as

$$R_{old} = \rho \frac{L}{A} \quad (11.27)$$

Fig. 11.8 Thermal electrical analysis



After steady state is reached, the midpoint temperature (Y) computed in Eq. (11.19) causes an increase in the resistance of the wire; this increase is evaluated using the coefficient of resistivity (α). The current through the wire when a 10V voltage is applied is calculated as

$$I = \frac{10}{R_{old} (1 + \alpha Y)} \quad (11.28)$$

Assume that there is no electrical performance test data for the wire, and it is required to predict the uncertainty in the electrical current, by including the results of verification (using Richardson extrapolation), validation, and calibration in the lower-level model. The two models and the associated sources of uncertainty are connected through a Bayesian network as shown in Fig. 11.9. The four uncertain quantities (T_0 , T_L , k , β) are used to predict the midpoint temperature (Y) which is then used to compute the resistance (R) as a function of higher-level model parameters (A , α , I , ρ). Notice that the lower-level solution approximation error is actually a function of the aforementioned four uncertain quantities (T_0 , T_L , k , β). Data (D^C , D^V) is available for comparison against Y , and ϵ_m is the measurement error associated with such data.

Since the thermal model used for temperature prediction has already been verified, calibrated, and validated, the unconditional PDF of the temperature is simply propagated through the current-predictor model to calculate the current in the wire. For the purpose of illustration and to see the effect of uncertainty in Y on the uncertainty in electrical current (I), the other parameters of the current-predictor model (α , A , ρ) are chosen to be deterministic. The PDF of the current of the wire is shown in Fig. 11.10, for three cases.

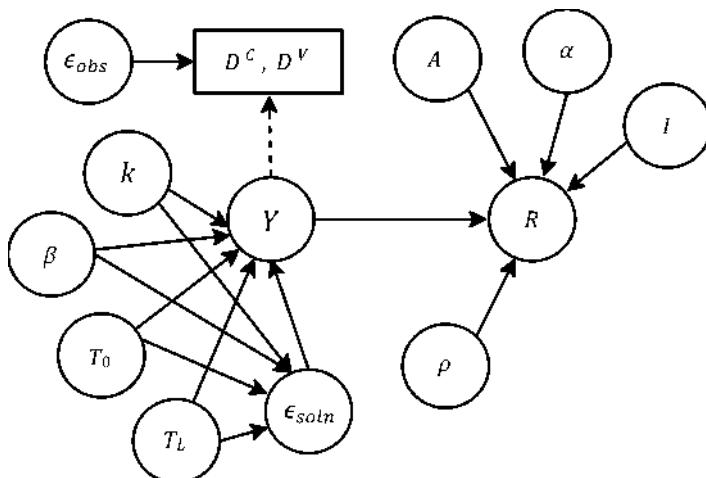
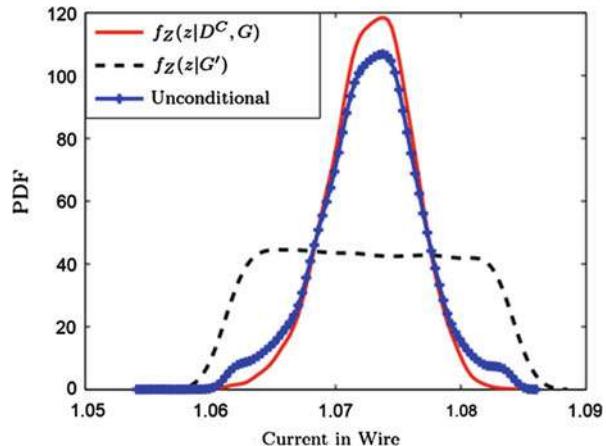


Fig. 11.9 Bayesian network: thermal electrical analysis

Fig. 11.10 PDF of electric current: system response



The PDF $f_Z(Z|G, D^C)$ is obtained by propagating the model prediction of thermal model through the electrical model, and the PDF $f_Z(Z|G')$ is obtained by propagating the alternate PDF of temperature ($f_Y(y|G')$) through the electrical model. The unconditional PDF ($f_Z(Z|D^C, D^V)$) represents the current response by integrating verification, validation, and calibration activities for the lower-level heat conduction model. Similar to the previous example, the difference between $f_Z(Z|G, D^C)$ and the unconditional $f_Z(Z|D^C, D^V)$ can be quantified; for example, $1 - F_Z(Z = 1.08|G) = 0.0086$, whereas $1 - F_Z(Z = 1.08) = 0.0400$.

6 Multilevel Models with Type-II Interaction

Sometimes, a system model is developed using progressively complex models and corresponding experiments (isolated features, isolated physics, simplified geometry, scaled models, etc.). The experiments of lowest complexity (simplest geometry or single physics) have been referred to as unit-level experiments [76]. A higher-level experiment could include an assembly of units or combined physics.

A typical example of such a system is discussed in [1], where material-level tests (lowermost level), performance of a single joint, and performance of three joints are used to calibrate underlying material and model parameters that are used in the overall system-level model. Usually, in such a system, the complexity increases going from the lower level to the higher level (more physics, features, components, etc.). The response of a lower-level experiment may not be directly related to the system-level response. However, there are some system-level parameters that can be inferred using lower-level experiments.

Assume that a generic system-level model is given by

$$Z = H(\theta, X, \Psi) \quad (11.29)$$

In Eq. (11.29), Z is the system-level prediction, θ is the set of model parameters which are calibrated based on lower-level models and tests, Ψ is the set of additional model parameters at the system level, and X are the inputs. Consider two lower-level models G_1 and G_2 . Both these models have common model parameters θ , but they have their own inputs (X_1 and X_2) and outputs (Y_1 and Y_2); in addition, they may have additional lower-level model parameters (Ψ_1 and Ψ_2).

$$\begin{aligned} Y_1 &= G_1(\theta, X_1, \Psi_1) \\ Y_2 &= G_2(\theta, X_2, \Psi_2) \end{aligned} \quad (11.30)$$

Assume that separate sets of data are available for calibration (D_1^c and D_2^c for levels 1 and 2, respectively) and validation (D_1^v and D_2^v for levels 1 and 2, respectively). Full system testing is not possible, i.e., no test data are available at the system level (Z), and it is required to quantify the uncertainty in the system-level prediction using the data at the lower levels (Y_1 and Y_2). The inputs, model parameters, outputs, and data at all levels are connected through a Bayesian network, as shown in Fig. 11.11.

6.1 Verification, Calibration, and Validation

The steps of verification, calibration, and validation in each model are similar to the previous sections; however the procedure for integration of these activities is different.

If θ is estimated using each individual model (G_1 or G_2) and the corresponding calibration data (D_1^c or D_2^c), then the corresponding PDFs of the model parameter θ are $f(\theta|D_1^c, G_1)$ or $(\theta|D_2^c, G_2)$, respectively.

The Bayesian network facilitates the simultaneous use of both models and the corresponding data to calibrate θ and obtain the PDF $f(\theta|D_1^c, D_2^c, G_1, G_2)$. This step of simultaneous calibration using multiple data sets from experiments of differing complexity is different from the calibration considered in Sects. 2 and 4, where only one model and the corresponding calibration data were used to estimate θ . In order to integrate the results of verification, validation, and calibration in Sect. 6.3 below, all the PDFs, i.e., those calibrated using individual data sets ($f(\theta|D_1^c, G_1)$ and $f(\theta|D_2^c, G_2)$) as well as those calibrated using multiple data sets ($f(\theta|D_1^c, D_2^c, G_1, G_2)$), are necessary.

The use of validation data is identical to the procedure in Sects. 2 and 4. The quantities $P(G_1|D_1^c)$ and $P(G_2|D_2^v)$ are calculated using the Bayes factor metric; further $P(G'_1|D_1^v) = 1 - P(G_1|D_1^v)$ and $P(G'_2|D_2^v) = 1 - P(G_2|D_2^v)$. Alternatively, the reliability-based method can also be used to calculate this probability. Since the two models are assumed independent, $P(G_1 \cap G_2|D_1^v, D_2^v) = P(G_1|D_1^v)P(G_2|D_2^v)$.

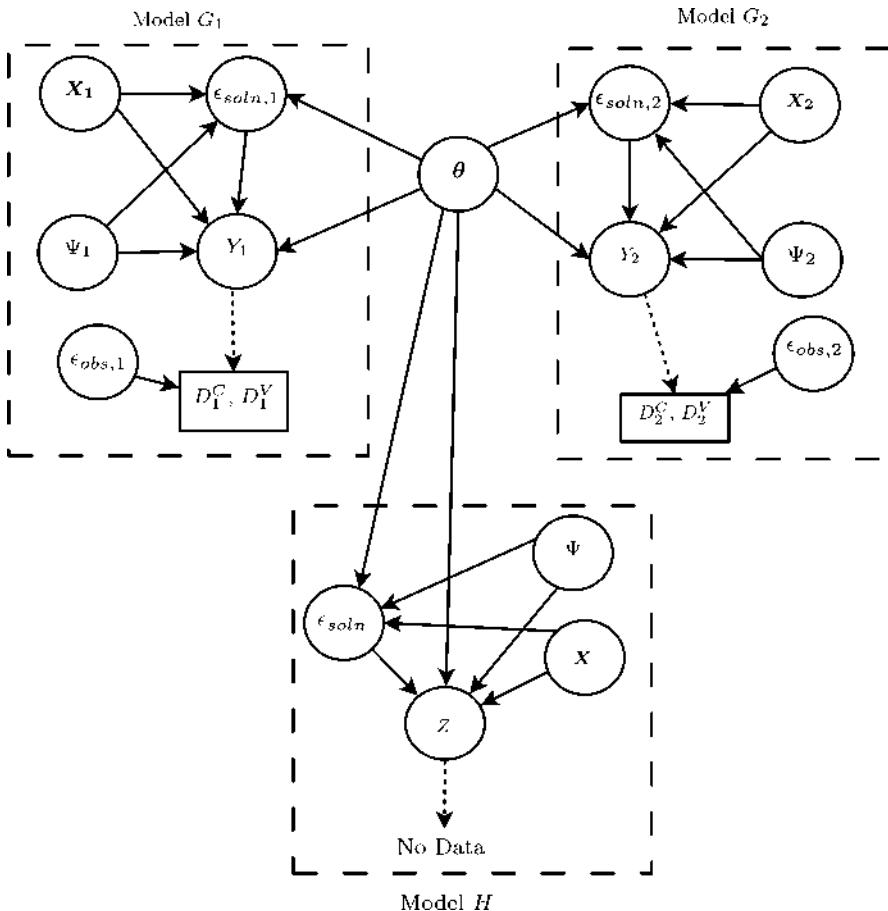


Fig. 11.11 Bayesian network: Type-II interaction between models

6.2 Relevance Analysis

Section 1.2 mentioned the necessity to assign larger weight to the level physically “closer” or more relevant to the system level than the other in the case of Type-II interaction. Hence this section develops a method for relevance analysis, which measures the degree to which the experimental configuration at a lower level has similar physical characteristics as the system of interest. Currently such measure is only intuitive and qualitative; an objective quantitative measure of relevance is needed for uncertainty integration.

The methodology to measure relevance should have two desired features. First, the defined methodology needs no mathematical details of the model in each level, since the model in each level could be a black box. Second, the resultant relevance

measure can be used conveniently as a weighting term in uncertainty integration. To fulfill these two criteria, a relevance analysis using Sobol indices is discussed in this section.

Consider a model $Y = F(X)$ where $X = \{X^1, \dots, X^N\}$ is a vector containing all the inputs. Sensitivity analysis measures the contribution of each input to the uncertainty of Y [92]. Compared to local sensitivity analysis, global sensitivity analysis (GSA) considers the entire probability distribution of the input, not just the contribution at a local point. The Sobol indices for GSA have been developed in the literature based on the variance decomposition theorem [93], including first-order index and total effects index. For a particular input X^i , its first-order index is $S_1^i = V(E(Y|X^i))/V(Y)$, and its total effects index is $S_T^i = 1 - V(E(Y|X^{-i}))/V(Y)$ where X^{-i} means all the inputs other than X^i . The first-order index S_1^i measures the contribution of X^i by itself, and the sum of first-order indices of all inputs is always less than or equal to unity. The difference between this sum and unity is the contribution of the interaction among inputs. In contrast, the total effects index S_T^i contains not only the contribution of X^i , but also the interaction effect of X^i with other inputs. The interaction between variables will be ignored if the first-order index is used, thus this chapter uses the total effects index to develop a method to quantify the relevance. In the following discussion, the term sensitivity index indicates the total effects index.

Without loss of generality, this chapter takes the multilevel problem in Fig. 11.1b for the illustration of relevance analysis. To predict the system output y_s , the same quantity is also measured at lower levels (in the numerical example, the maximum acceleration at the top mass is also measured at Level 1 and Level 2). The three prediction models for this quantity at different levels are $y_{L_1} = G_{L_1}(\mathbf{x}_{L_1}; \boldsymbol{\theta})$, $Y_{L_2} = G_{L_2}(\mathbf{x}_{L_2}; \boldsymbol{\theta})$, and $y_{L_1} = G_s(\mathbf{x}_s; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are model parameters and \mathbf{x}_{L_1} , \mathbf{x}_{L_2} , and \mathbf{x}_s are the model inputs at each level. Note that (1) the computational models are replaced by the GP models to improve computational efficiency, (2) model errors are considered in Level 1 and Level 2, and (3) model error at the system level is not considered since no information on it is available. These prediction models are stochastic, i.e., the output is stochastic even at fixed values of model inputs and model parameters. However, the Sobol indices computation requires a deterministic model, i.e., deterministic output at given values of model inputs and model parameters. This chapter applies the auxiliary variable methodology based on the probability integral transform, as developed in Ref. [94, 95], to obtain a deterministic value of the output for a given realization of inputs and model parameters; thus the Sobol indices can be computed.

Assume model parameters, model inputs, and auxiliary variables constitute N_{L_1} elements in total at Level 1; since each element has a corresponding sensitivity index, a N_{L_1} -dimensional vector V_{L_1} of sensitivity indices will be obtained at Level 1. Similarly, a N_{L_2} -dimensional sensitivity vector N_{L_2} will be obtained at Level 2, and a N_s -dimensional sensitivity vector V_s will be obtained at the system level.

Rigorously, measuring the relevance requires comparing the mathematical model of the lower level and the mathematical model of the system level. However, this

comparison is not easy if the models at different levels have distinct formats and are addressing different physical configurations. Further, the model sometimes may be a black box; thus we cannot access its mathematical details and a direct comparison would be difficult. The obtained sensitivity vectors quantify the contribution of each model input/parameter toward the uncertainty in the model output. In other words, the sensitivity vector indicates which model input/parameter is more important in affecting the model output uncertainty. Whether a model input/parameter is important is determined by the physics of the model, thus the sensitivity vector represents the physics to the extent that the model represents the physics accurately. Therefore, this chapter considers the sensitivity vector as an indicator of the physics captured in the model. (Of course, how well the physics is captured in the model is already indicated by the model reliability metric.) Thus the comparison of the vectors from two different levels is used to quantify the relevance between these two levels.

One issue in the comparison of V_{L_i} ($i = 1, 2$) and V_s is that they may have different sizes (N_{L_1}, N_{L_2}, N_s may not be equal to each other) and some elements in one vector may not be present in the other vector. The shared dimensions of V_{L_i} and V_s are model parameters θ , and the unshared dimensions are the different model inputs and auxiliary variables at each level. To solve this problem, we add the unshared dimension in V_{L_i} or V_s to the other vectors but set the corresponding sensitivity indices as zero since the added dimensions have no effect in the computation of the original sensitivity vector. Thus all the vectors V_{L_i} or V_s are brought to the same size.

Several methods are available to compare two vectors, such as Euclidean distance [90], Manhattan distance [90], Chebyshev distance [90], and cosine similarity [90, 96]. To include the relevance in the subsequent uncertainty integration conveniently, we define the relevance index R as the square of cosine similarity of the sensitivity vectors, where the cosine similarity is the standardized dot product of two vectors:

$$R = \left(\frac{V_{L_i} \cdot V_s}{\|V_{L_i}\| \|V_s\|} \right)^2 \quad (11.31)$$

In other words, the above relevance index is the square of the cosine value of the angle between two sensitivity vectors, the elements in which are all positive. If the angle is zero, the relevance among the two levels is 1; if the two vectors are perpendicular, the relevance is 0.

In addition, this definition of relevance generates a value on the interval $[0, 1]$, and its complement, the square of the sine value, indicates physical non-relevance; hence the sum of “relevance” and “non-relevance” is the unity. Here the relevance index is a plausibility model for the proposition “The lower level model reflects the physical characteristics of the system Level,” and the plausibility of this proposition is the relevance index. Based on Cox’s theorem [97], this plausibility model is isomorphic to probability since (1) the relevance index is a real value depending on the information of sensitivity vectors we obtained and (2) the relevance index changes sensibly as the sensitivity vectors change. Thus the relevance index can

be converted to probability by scaling, which has been done since the relevance index defined in Eq. (11.31) is already on the interval [0, 1]. Therefore in the roll-up methodology Sect. 6.3, we treat the relevance index as a probability and conveniently include it as a weighting term in the uncertainty integration.

A further question arises in the computation of relevance index. Sobol indices consider the entire distribution of the influencing variable, but the posterior distribution of θ_m (to be used in system-level prediction) is unknown before the uncertainty integration. In order to solve this problem, a straightforward iterative algorithm to compute the relevance index R is as below:

1. Set an initial value of R
2. Obtain the integrated distribution of each model parameter using the current relevance and the roll-up method in Sect. 6.3 below.
3. Use the integrated distributions from step 2 to compute the sensitivity indices, and recompute the updated relevance index R .
4. Repeat steps 2 and 3 until the relevance index R converges.

Thus, the results of calibration and validation at each lower level and relevance indices between the lower levels and the system level have been obtained. The next task is to construct the integrated distribution of the system-level model parameters and predict the system output.

6.3 Integration for Overall Uncertainty Quantification

The method for the integration of the verification, calibration, validation, and relevance analysis results for Type-II interaction is different from Sect. 4 because the linking variables in this case are the model parameters, whereas the linking variables in Type-I interaction were the outputs of lower-level models. While the unconditional PDF of the lower-level output was calculated in Sect. 4, it is now necessary to calculate the integrated distribution of the model parameters that also accounts for validation and relevance results.

For a multilevel problem of Type-II interaction, the purpose of uncertainty integration is to combine all the available information (from calibration, validation, and relevance analysis) from the lower levels and predict the response at the system level. In this chapter the information from the lower level includes (1) the posterior distributions from model calibration by considering data at each individual lower level, as well as data from multiple lower levels, (2) the model reliability distributions from model validation at each lower level, and (3) the relevance indices between each lower level and the system level. A roll-up methodology has been proposed in [16, 98] for uncertainty integration. This methodology computes the unconditional PDF of the model parameters as a weighted average of each posterior distributions, where the weight term of each posterior is purely decided by model validation, and the model validity is a fixed value. This section extends this method to include two additional concepts:

1. *Stochastic model reliability*: The model reliability $P(G_i)$ is a random variable with PDF $f(P(G_i))$ as explained in Sect. 2.3.2;
2. *Relevance index*: This has been defined in Sect. 6.2 as the square of the cosine value of the angle between the sensitivity vectors at lower level and system level. We treat the relevance index similar to probability in the roll-up methodology, based on Cox's theorem. If S_i denotes the event that Level i is relevant to the system level, then the probability $P(S_i|G_i)$ is equal to the value of the relevance index R ; this probability is conditioned on G_i since the computation of the relevance index uses the model at Level i ; in contrast $p(S'_i|G_i)$ denotes the probability of non-relevance and is equal to $1 - R$.

Take the multilevel problem in Fig. 11.1b as an example. The integrated distribution of model parameters θ conditioned on the calibration and validation data and model reliability $P(G_i)(i = 1, 2)$ is [99]:

$$\begin{aligned} f_{\Theta}(\theta | D_1^C, D_1^V, D_2^C, D_2^V, P(G_1), P(G_2)) \\ = P(G_1 G_2 S_1 S_2) f(\theta | D_1^C, D_2^C) + P(G_1 S_1 \cap (G_2' \cup S_2')) f(\theta | D_1^C) \\ + P(G_2 S_2 \cap (G_1' \cup S_1')) f(\theta | D_2^C) + P((G_1' \cup S_1') \cap (G_2' \cup S_2')) f(\theta) \end{aligned} \quad (11.32)$$

From the view of generating samples, Eq. (11.32) indicates two criteria: (1) whether a level is relevant to the system level and (2) whether a level has a valid model. A sample of θ is generated from $f_{\Theta}(\theta | D_1^c, D_2^c)$ only when both levels satisfy both criteria, a sample of θ is generated from $f_{\Theta}(\theta | D_i^c)$ if level i satisfies both criteria but the other level does not, and a sample of θ is generated from the prior distribution $f_{\Theta}(\theta)$ if neither level satisfies both criteria. By assuming independence of model validity and relevance between different lower levels, the weight terms in Eq. (11.32) are computed by using the values of $P(G_i)$, $P(S_i|G_i)$ and two fundamental probability relationships: $P(G_i S_i) = P(G_i)P(S_i|G_i)$, $P(G'_i \cup S'_i) = 1 - P(G_i S_i)$. Eq. (11.11) also implies the option of “using only data from one level.” If both the model validity and relevance are 1 for Level 1 and either model validity or relevance is 0 for Level 2, Eq. (11.11) reduces to $f_{\Theta}(\theta | D_1^{c,v}, D_2^{c,v}) = f_{\Theta}(\theta | D_1^c)$, i.e., only Level 1 data is used.

The integrated distribution of θ , which is conditioned on both calibration and validation data, can now be computed as

$$\begin{aligned} f_{\Theta}(\theta | D_1^C, D_1^V, D_2^C, D_2^V) \\ = \iint f_{\Theta}(\theta | D_1^C, D_1^V, D_2^C, D_2^V, P(G_1), P(G_2)) f(P(G_1)) \\ f(P(G_2)) dP(G_1) dP(G_2) \end{aligned} \quad (11.33)$$

Eqs. (11.32) and (11.33) express the approach to integrate calibration, validation, and relevance results at lower levels. Note that Eq. (11.33) accounts for stochastic

model reliability. The analytical expression of $f_{\Theta}(\boldsymbol{\theta}|D_1^C, D_1^V, D_2^C, D_2^V)$ is difficult to derive since the results we collect in model calibration and validation are all numerical. A single loop sampling approach is given here to construct $f_{\Theta}(\boldsymbol{\theta}|D_1^C, D_1^V, D_2^C, D_2^V)$ numerically, as follows:

1. Generate a sample of $P(G_1)$ and $P(G_2)$ from their distributions.
2. Compute the weight terms in Eq. (11.32). Divide the interval $[0, 1]$ into four ranges; the length of the k th range is equal to the value of the k th weight in Eq. (11.32)
3. Generate a random number from the uniform distribution $U(0, 1)$
4. Generate a sample of $\boldsymbol{\theta}$ using stratified sampling, i.e., from $f_{\Theta}(\boldsymbol{\theta}|D_1^C, D_2^C)$ if the random number in step 3 is located in the first range, from $f_{\Theta}(\boldsymbol{\theta}|D_1^V)$ if located in the second range, from $f_{\Theta}(\boldsymbol{\theta}|D_2^V)$ if located in the third domain, and from $f_{\Theta}(\boldsymbol{\theta})$ if located in the fourth domain.
5. Repeat steps 1 to 4 to obtain multiple samples of $\boldsymbol{\theta}$; then construct the PDF $f_{\Theta}(\boldsymbol{\theta}|D_1^C, D_1^V, D_2^C, D_2^V)$ by any method such as kernel density estimation [100]

(In general, if there are n_m models, then the right hand side of Eq. (11.32) has 2^{n_m} terms. Each of these terms, except the one corresponding to the case where all the models are not correct, needs to be computed through a Bayesian inference procedure. Hence, this may be a computational challenge in larger applications. In such scenarios, the use of an inexpensive surrogate model is recommended for such repeated Bayesian inference calculations, as performed in this chapter.)

The PDF $f_{\Theta}(\boldsymbol{\theta}|D_1^C, D_1^V, D_2^C, D_2^V)$ calculated in Eq. (11.33) accounts for the verification, calibration, validation, and relevance analysis activities with respect to each of the lower-level models. This unconditional PDF is propagated through the system model $H(\boldsymbol{\theta}, \mathbf{X}, \Psi)$, in order to quantify the uncertainty in the system-level response Z . This can be done by Monte Carlo sampling or other preferred stochastic analysis methods. Thus, similar to Sect. 4, it can be seen that the tools of conditional probability and total probability are directly useful for integrating verification, validation, and calibration and, thereby, aid in the quantification of the system-level prediction uncertainty.

7 Numerical Example: Models with Type-II Interaction

7.1 Problem Description

A multilevel structural dynamics challenge problem provided by Sandia National Laboratories [101] is used to illustrate the methodology developed in Sect. 6. As shown in Fig. 11.12, Level 1 contains three mass-spring-damper dynamic components in series, and a sinusoidal force input $p = 300 \sin(500t)$ is applied to m_1 . At Level 2, the dynamic system is mounted on a beam supported by a hinge at one end and a spring at the other end; a sinusoidal force input $p = 300 \sin(350t)$ is applied on the beam. The configuration of the system level is the same as Level 2, but the

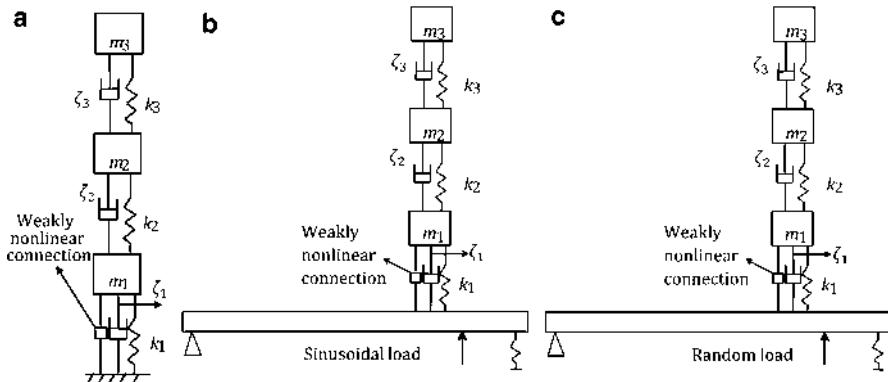


Fig. 11.12 Structural dynamics challenge problem. (a) Level 1. (b) Level 2. (c) System level

input is a random process loading (indicating difference in usage condition). Here Level 1 and Level 2 are defined as lower levels, and experimental data are assumed to be available only at the lower levels. All levels share six model parameters: three spring stiffnesses $k_i (i = 1, 2, 3)$ and three damping ratios $\zeta_i (i = 1, 2, 3)$, and they are assumed to be deterministic but unknown parameters, which are to be calibrated. The units of all quantities are nondimensional.

Suppose ten experiments are conducted at each of Level 1 and Level 2, and the displacement, velocity, and acceleration history at each degree of freedom are recorded. Six quantities at each lower level are extracted from these records as the synthetic experimental data in model calibration and validation:

- (1) $A_i (i = 1, 2, 3)$: the maximum acceleration in the i th mass;
- (2) $D_i (i = 1, 2, 3)$: the energy dissipated by the i th damper in 1000 time units

The data points for each quantity from the first five tests are selected as calibration data and the rest five as validation data.

Computational models for the three levels have been established. The method to solve the dynamic problem at Level 1 can be found in structural dynamics textbooks [102], and the computational models using the finite element method for Level 2 and the system level are provided by Sandia National Laboratories [101].

Since the model input at each level is fixed, the input-dependent model error is an unknown deterministic value. Thus the parameters to be calibrated in this example are the spring stiffnesses $k_i (i = 1, 2, 3)$, the damping ratios $\zeta_i (i = 1, 2, 3)$, model error δ , and the output measurement error standard deviation σ_m if the data of the corresponding quantity are used in model calibration. Based on expert opinion, suppose the prior distribution of each k_i and ζ_i is assumed to be lognormal with a coefficient of variation of 10% and mean values of $\mu_{k_1} = 5000$, $\mu_{k_2} = 9000$, $\mu_{k_3} = 8000$, $\mu_{\zeta_i} = 0.025 (i = 1, 2, 3)$. The prior distribution of model error is assumed to be uniform, i.e., $\delta \sim U(a, b)$ and the prior of σ_m is Jeffrey's prior $f'(\sigma) \propto 1/\sigma$.

The objective in this numerical example is to quantify the uncertainty in the prediction of maximum acceleration at m_3 in the system level, by using available models and experimental data.

Since as many as six quantities are measured, we can choose any combination of these six quantities in the analysis. Measurement data on more output quantities reduce the uncertainty in the system output prediction, but the computational effort will also increase and each quantity will bring two more related terms (δ and σ) for calibration. For the sake of brevity, only the calibration and validation results using the test data for all six quantities are provided below. But a plot showing the reduction in the uncertainty of system output prediction with the increase of output quantity measurements is also provided at the end.

7.2 Results and Analysis

In order to reduce the computational effort, Gaussian process (GP) surrogate models are established to replace the computational models for all the output quantities. The surrogate model uncertainty introduced by the GP models is incorporated in model calibration and validation, as explained in Sects. 2 and 3.1. The calibration results of k_i and ζ_i using the calibration data of the six output quantities at different levels are shown in Fig. 11.13, including all the PDFs needed in Eq. (11.32). As more data are used in the calibration, the uncertainty of the model parameters will decline. Thus Fig. 11.13 shows that the posterior distributions using the data at both levels always have less uncertainty than those using data at a single level. The difference between the posterior distributions within each sub-figure also indicates that the posterior distribution is a best-fitting result in the sense of representing that particular data set, but we do not yet know how to combine these alternatives in the subsequent prediction. This is answered by model validation and relevance analysis.

Next model validation is performed using the stochastic model reliability metric with multivariate output. The tolerance for each quantity is chosen to be 15% of the validation data. Level 2 is expected to have lower model reliability value for two main factors:

1. The discretization error at Level 2 due to a limited number of finite elements for the beam (41 in this example). But this factor is not effective here since the data at Level 2 are synthetic data generated using the computational model, meaning that the difference between the computational model and the physics model is ignored. This factor will come into play if experimental data instead of synthetic data are used.
2. The coupling between the beam and the damped mass-spring system brings stronger nonlinearity at Level 2. Under the same number of training points, the GP surrogate model at Level 2 has more surrogate uncertainty (larger GP model prediction variance) than the GP surrogate model at Level 1. This factor is included in the numerical example.

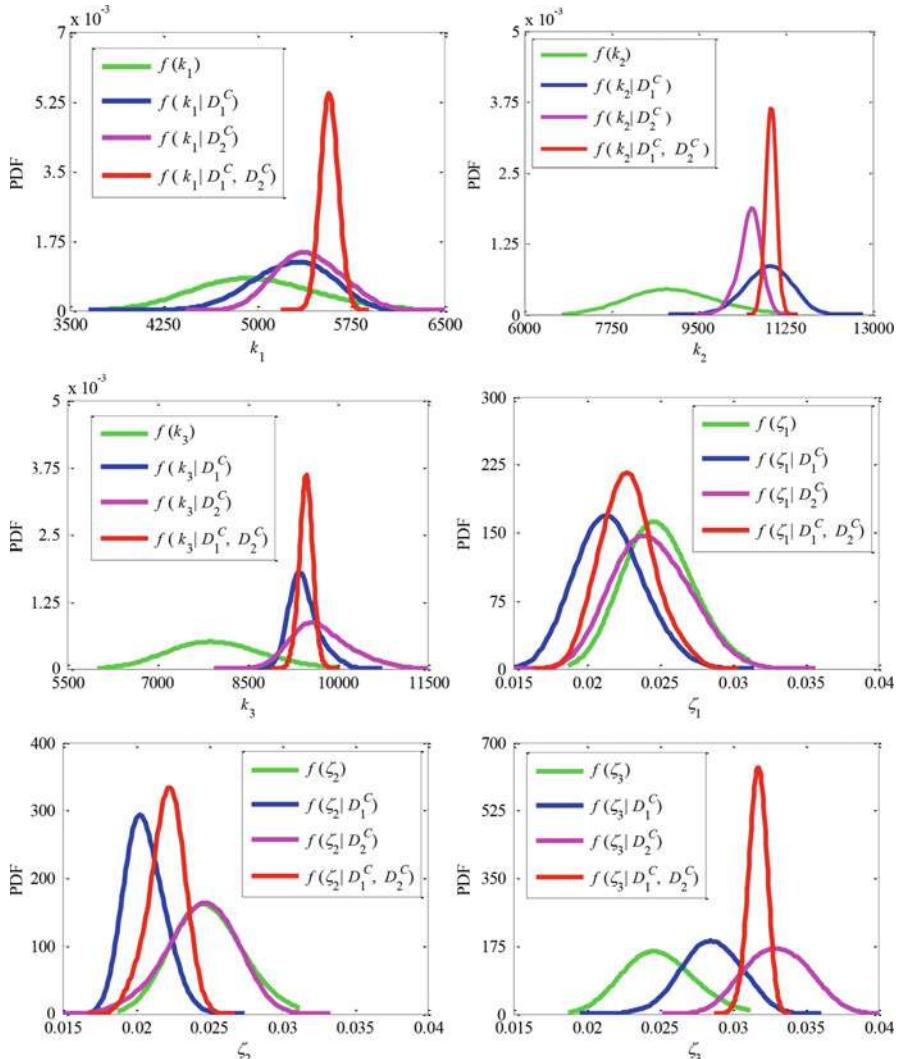
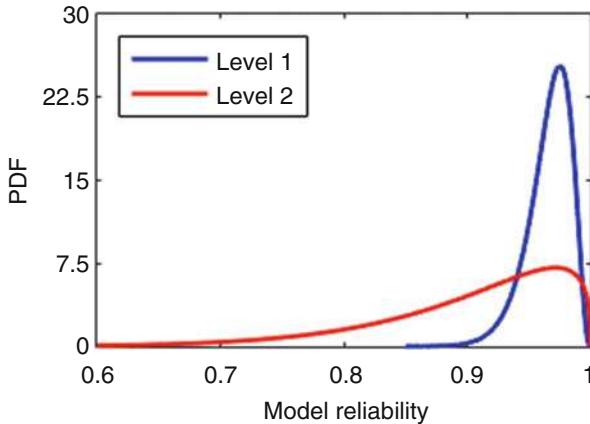


Fig. 11.13 Posterior distributions of model parameters

The model reliability values given by the validation data from each validation test are listed in Table 11.1, which indicate lower model reliability at Level 2. In Fig. 11.14, these values are used to construct the distributions of model reliability at Level 1 and Level 2 using the method of moments. However, even though the model at Level 1 has higher model reliability than the model at Level 2, the configuration at Level 2 is closer to the system level of interest. Therefore relevance analysis also needs to be considered.

Table 11.1 Model reliability values

Validation test	1	2	3	4	5
Model reliability at Level 1	0.9702	0.958	0.9398	0.9828	0.9800
Model reliability at Level 2	0.9616	0.8564	0.9208	0.9796	0.7904

**Fig. 11.14** Distribution of model reliability

The relevance index of each lower level to the system level is computed using the iterative algorithm in Sect. 6.2. The initial values of relevance indices for both lower levels are set as 1. The algorithm converges after three iterations for Level 1 and after five iterations for Level 2. The results are $P(S_1) = 0.5785$ and $P(S_2) = 0.8971$. This result means that Level 2 is more relevant to the system level, which is consistent with our intuition since Level 2 has the same structural configuration as the system and differs only in the load input (sinusoidal vs. random process). Compared with the result of model validation, Level 2 has a lower value of model reliability but higher relevance index.

Based on all the information from calibration, validation, and relevance analyses, the integrated distributions of all six model parameters are constructed in Fig. 11.15 using Eqs. (11.32) and (11.33). Figure 11.15 also shows the result by considering validation only (no relevance) using the previous roll-up method in Ref. [98], but stochastic model reliability metric is still used. It is shown that the new roll-up method is more conservative than the previous one, since we add one more criterion of relevance during the generation of samples from the posterior distribution.

The system output is predicted by propagating the integrated distribution of model parameters through the computational model at the system level. Figure 11.16 gives not only the prediction using the data of all six quantities but also the prediction by other combinations of quantities whose names are shown in the legend. The mean values and variances of the predictions are shown in Table 11.2.

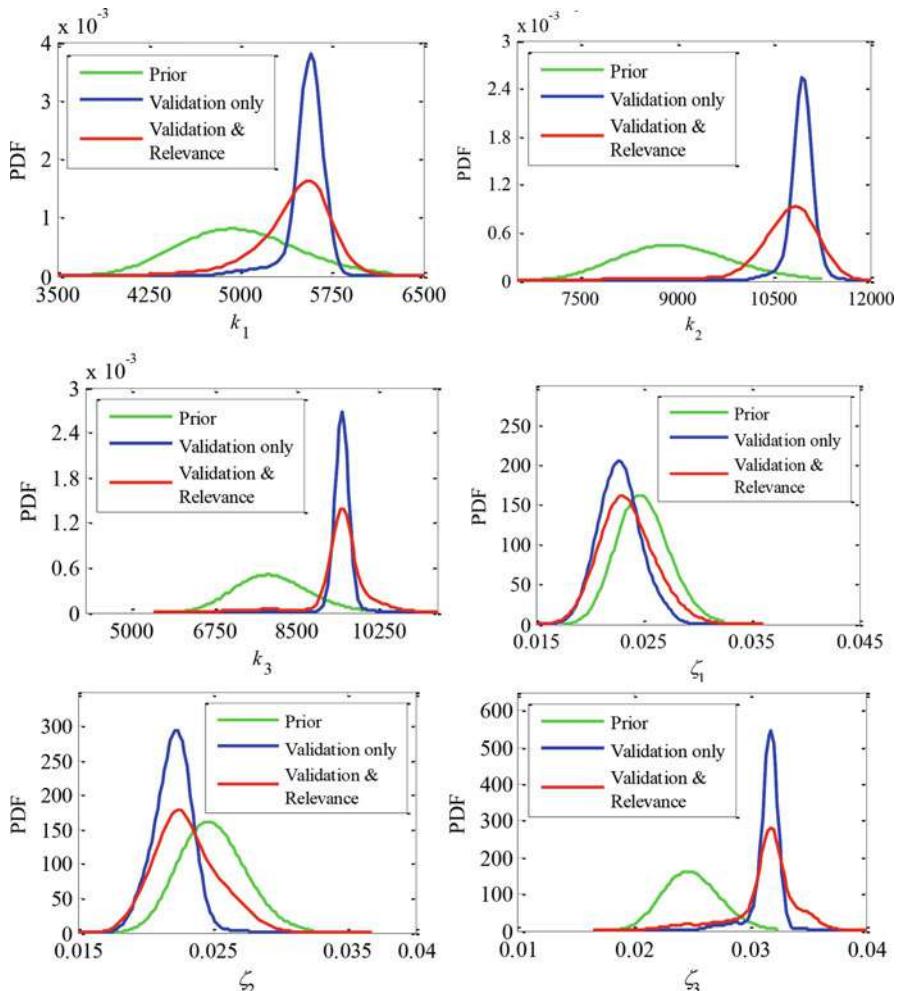


Fig. 11.15 Integrated distributions of model parameters

Table 11.2 Mean values and variances of predictions

Number of quantities	1	2	3	4	5	6
Mean values	710	713	690	632	655	656
Variance	12,202	10,499	10,959	4868	5432	2301

As more quantities are employed, the mean value of prediction decreases from 712 to 656, and the variance shows an overall decreasing tendency, but not monotonic (the variance increases slightly when the number of outputs considered rises from 2 to 3 and from 4 to 5).

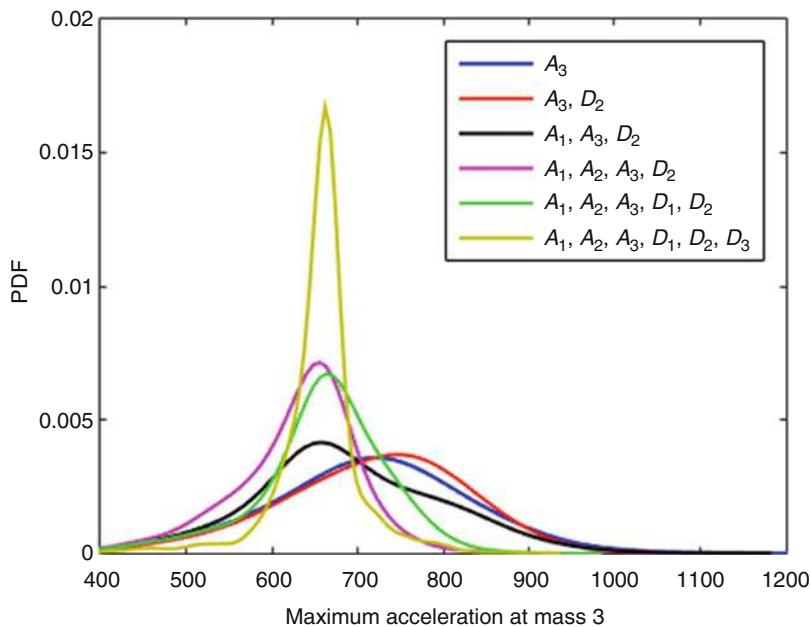


Fig. 11.16 System output prediction

8 Conclusion

Verification, validation, and calibration are significant activities in the process of model development. While methods for individual activities have been addressed in the past, the quantification of the combined effect of these activities on the overall system-level prediction uncertainty is addressed in this chapter, for single-level and multilevel models.

This topic is of specific importance in systems which are studied using multiple models where data may be available only at lower levels, and it may be desired to quantify the uncertainty in the system-level prediction using lower-level data. This chapter discussed a Bayesian network-based methodology to integrate the various uncertainty quantification activities, including verification, validation, and calibration, performed at lower levels, and rigorously account for their effects on the system-level prediction uncertainty. The Bayesian network is first used to connect the multiple models, the corresponding inputs, parameters, outputs, error estimates, and all available data. During the verification procedure, the solution approximation errors are quantified and accounted for. Both deterministic and stochastic errors are properly included, and the model is corrected before calibration and validation. Two independent sets of test data are considered: the first set is used to calibrate the model parameters and the second set is used to validate the calibrated model. The principles of conditional probability and total probability are then used to integrate

the results of calibration and validation in order to compute the overall uncertainty in the model-based prediction.

The integration methodology is first developed for single-level models and then extended to multilevel systems that consist of interacting models. Two types of interactions are discussed in detail: (1) Type-I interaction, where the output of a lower-level model becomes an input to the higher-level model, and (2) Type-II interaction, where models and experiments at various levels of reduced complexity are used to infer system model parameters. While verification is performed before calibration and validation in both the cases, in order to account for the results of verification during calibration and validation, the procedure for the integration of calibration and validation results at lower levels is different for Type-I and Type-II interactions; in the former case, the key idea is to compute the unconditional PDF of the *output* of the lower-level model, whereas in the latter case, the key idea is to compute the unconditional PDF of the system *model parameters*. If a system-level prediction is based on models with both types of interactions (both Type-I and Type-II), then the unconditional PDFs of the intermediate output and the parameters can both be used to compute the uncertainty in the overall system-level prediction uncertainty.

The multilevel roll-up methodology described in this chapter offers considerable promise toward the quantification of margins and uncertainties in multilevel system prediction. While calibration and validation have previously been performed independently at individual levels, this methodology systematically integrates all such activities in order to compute the system-level prediction uncertainty, thereby aiding in risk-informed decision-making with all available information. Only two types of interactions between multiple models were considered in this chapter.

An important related issue is the extrapolation of the model to application conditions under which experiments may not have been performed. Often, there are two types of extrapolation. The first type is where the model is validated at certain input values, but prediction needs to be performed at other input values that are not contained in the validation domain. The second type of extrapolation is where validation is performed using a simplified system (with restricted features, physics, etc.) and the desired prediction is of the original system. While regression-based techniques have been developed for the first type of extrapolation [27], the second type of extrapolation could be represented through a Type-II configuration discussed in Sects. 6 and 7. The methods discussed in this chapter can only be used for the propagation of uncertainty to the extrapolation domain *as long as there is no change in the physics of the system between validation and extrapolation domains*.

References

1. Urbina, A., Mahadevan, S., Paez, T.L.: Quantification of margins and uncertainties of complex systems in the presence of aleatoric and epistemic uncertainty. Reliab. Eng. Syst. Saf. **96**(9), 1114–1125 (2011)

2. Sankararaman, S., Ling, Y., Mahadevan, S.: Uncertainty quantification and model validation of fatigue crack growth prediction. *Eng. Fract. Mech.* **78**(7), 1487–1504 (2011)
3. Sankararaman, S., Mahadevan, S.: Model parameter estimation with imprecise and unpaired data. *Inverse Probl. Sci. Eng.* **20**(7), 1017–1041 (2012)
4. Sankararaman, S., Mahadevan, S.: Model validation under epistemic uncertainty. *Reliab. Eng. Syst. Saf.* **96**(9), 1232–1241 (2011)
5. Ling, Y., Mahadevan, S.: Quantitative model validation techniques: new insights. *Reliab. Eng. Syst. Saf.* **111**, 217–231 (2013)
6. Sankararaman, S., Mahadevan, S.: Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. *Reliab. Eng. Syst. Saf.* **96**(7), 814–824 (2011)
7. Jeffrey, H.: Theory of Probability. Oxford University Press, Oxford (1998)
8. Sankararaman, S., Ling, Y., Shantz, C., Mahadevan, S.: Uncertainty quantification in fatigue crack growth prognosis. *Int. J. Progn. Heal. Manag.* **2**(1), 15 (2011)
9. Sankararaman, S., Ling, Y., Shantz, C., Mahadevan, S.: Inference of equivalent initial flaw size under multiple sources of uncertainty. *Int. J. Fatigue* **33**(2), 75–89 (2011)
10. Sankararaman, S., Ling, Y., Mahadevan, S.: Statistical inference of equivalent initial flaw size with complicated structural geometry and multi-axial variable amplitude loading. *Int. J. Fatigue* **32**(10), 1689–1700 (2010)
11. Sankararaman, S., McLemore, K., Mahadevan, S., Bradford, S.C., Peterson, L.D.: Test resource allocation in hierarchical systems using bayesian networks. *AIAA J.* **51**(3), 537–550 (2013)
12. Mullins, J., Li, C., Mahadevan, S., Urbina, A.: Optimal Selection of Calibration and Validation Test Samples under Uncertainty. In: IMAC XXXII, Orlando, pp. 391–401 (2014)
13. Li, C., Mahadevan, S.: Sensitivity Analysis for Test Resource Allocation. In: IMAC XXXIII, Orlando (2015)
14. Mullins, J., Li, C., Sankararaman, S., Mahadevan, S.: Probabilistic integration of validation and calibration results for prediction level uncertainty quantification: application to structural dynamics. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston (2013)
15. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. *J. R. Stat. Soc.* **63**(3), 425–464 (2001)
16. Sankararaman, S., Mahadevan, S.: Comprehensive framework for integration of calibration, verification and validation. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, pp. 1–12 (2012)
17. Li, C., Mahadevan, S.: Uncertainty quantification and output prediction in multi-level problems. In: 16th AIAA Non-Deterministic Approaches Conference, National Harbor (2014)
18. Li, C., Mahadevan, S.: Role of calibration, validation, and relevance in multi-level uncertainty integration. *Reliab. Eng. Syst. Saf.* **148**, 32–43 (2016)
19. Sankararaman, S., Mahadevan, S.: Likelihood-based approach to multidisciplinary analysis under uncertainty. *J. Mech. Des.* **134**(3), 031008 (2012)
20. Babuska, I., Oden, J.T.T.: Verification and validation in computational engineering and science: basic concepts. *Comput. Methods Appl. Mech. Eng.* **193**(36–38), 4057–4066 (2004)
21. Roy, C.J.: Review of code and solution verification procedures for computational simulation. *J. Comput. Phys.* **205**(1), 131–156 (2005)
22. AIAA: Guide for the verification and validation of computational fluid dynamics simulations. American Institute of Aeronautics and Astronautics (AIAA), no. AIAA G-077-1998 (1998)
23. Defense Modelling and Simulation Office, Verification, Validation, and accreditation (VV & A) recommend practices guide, Alexandria (1998)
24. Oberkampf, W.L., Blottner, F.G.: Issues in computational fluid dynamics code verification and validation. *AIAA J* **36**(5), 687–695 (1998)
25. Oberkampf, W.L., Trucano, T.G.G.: Verification and validation in computational fluid dynamics. *Prog. Aerosp. Sci.* **38**(3), 209–272 (2002)
26. Benay, R., Chanetz, B., Delery, J.: Code verification/validation with respect to experimental data banks. *Aerosp. Sci. Technol.* **7**(4), 239–262 (2003)

27. Roy, C.J., Oberkampf, W.L.: A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Comput. Methods Appl. Mech. Eng.* **200**(25), 2131–2144 (2011)
28. Roache, P.J.: Verification of codes and calculations. *Aiaa J.* **36**(5), 696–702 (1998)
29. Roache, P.J.: Verification and Validation in Computational Science and Engineering. Hermosa Publishers, Albuquerque (1998)
30. Oberkampf, W.L., Trucano, T.G., Hirsch, C.: Verification, validation, and predictive capability in computational engineering and physics. *Appl. Mech. Rev.* **57**(5), 345–384 (2004)
31. Roy, C.J., McWherter-Payne, M.A., Oberkampf, W.L.: Verification and validation for laminar hypersonic flowfields, part 1: verification. *Aiaa J.* **41**(10), 1934–1943 (2003)
32. Rebba, R., Mahadevan, S., Huang, S.: Validation and error estimation of computational models. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1390–1397 (2006)
33. Liang, B., Mahadevan, S.: Error and uncertainty quantification and sensitivity analysis in mechanics computational models. *Int. J. Uncertain. Quantif.* **1**(2), 147–161 (2011)
34. Rangavajhala, S., Sura, V.S., Hombal, V.K., Mahadevan, S.: Discretization error estimation in multidisciplinary simulations. *AIAA J.* **49**(12), 2673–2683 (2011)
35. Ferziger, J., Peric, M.: Computational Methods for Fluid Dynamics. Springer, Berlin (1996)
36. Ainsworth, M., Oden, J.T.T.: A posteriori error estimation in finite element analysis. *Comput. Methods Appl. Mech. Eng.* **142**(1–2), 1–88 (1997)
37. Oberkampf, W.L., DeLand, S.M., Rutherford, B.M., Diegert, K.V., Alvin, K.F.: Error and uncertainty in modeling and simulation. *Reliab. Eng. Syst. Saf.* **75**(3), 333–357 (2002)
38. Haldar, A., Mahadevan, S.: Probability, Reliability, and Statistical Methods in Engineering Design. John Wiley, New York (2000)
39. Ghanem, R., Spanos, P.D.: Polynomial chaos in stochastic finite elements. *J. Appl. Mech.* **57**(1)(89), 197–202 (1990)
40. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations, vol. 12. Cambridge university press, Cambridge/New York (2003)
41. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT, Cambridge (2006)
42. Richards, S.A.: Completed Richardson extrapolation in space and time. *Commun. Numer. Methods Eng.* **13**(7), 573–582 (1997)
43. Xu, P., Su, X., Mahadevan, S., Li, C., Deng, Y.: A non-parametric method to determine basic probability assignment for classification problems. *Appl. Intell.* **41**(3), 681–693 (2014)
44. Babuska, I., Rheinboldt, W.C.: A posteriori error estimates for the finite element method. *Int. J. Numer. Methods Eng.* **12**(10), 1597–1615 (1978)
45. Demkowicz, L., Oden, J.T., Strouboulis, T.: Adaptive finite elements for flow problems with moving boundaries. part I: variational principles and a posteriori estimates. *Comput. Methods Appl. Mech. Eng.* **46**(2), 217–251 (1984)
46. Rasmussen, C.E.: Evaluation of Gaussian processes and other methods for non-linear regression. PhD dissertation, University of Toronto, 1996
47. Rasmussen, C.E.: The infinite Gaussian mixture model. In: NIPS, Denver, vol. 12, pp. 554–560 (1999)
48. Rasmussen, C.E.: Gaussian processes in machine learning. In: Bousquet, O., VonLuxburg, U., Ratsch, G. (eds.) Advanced Lectures on Machine Learning, vol. 3176, pp. 63–71 (2004)
49. Santner, T.J., Williams, B.J., Notz, W.I.: The design and analysis of computer experiments. Springer, Dordrecht/New York (2013)
50. Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M.: Efficient global reliability analysis for nonlinear implicit performance functions. *Aiaa J.* **46**(10), 2459–2468 (2008)
51. McFarland, J.M.: Uncertainty Analysis for Computer Simulations through Validation and Calibration. Vanderbilt University, Nashville (2008)
52. Cressie, N.: Spatial Statistics. John Wiley, New York (1991)
53. Chiles, J.-P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty, vol. 344. Wiley-Interscience, New York (1999)

54. Wackernagel, H.: *Multivariate Geostatistics: An Introduction with Applications*. Springer, Berlin/New York (2003)
55. Trucano, T.G., Swiler, L.P., Igusa, T., Oberkampf, W.L., Pilch, M.: Calibration, validation, and sensitivity analysis: what's what. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1331–1357 (2006)
56. Seber, G.A.F., Wild, C.J.: *Nonlinear Regression*. Wiley, New York (1989)
57. Edwards, A.W.F.: *Likelihood*. Cambridge University Press, Cambridge, UK (1972)
58. Pawitan, Y.: *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, Oxford/New York (2001)
59. Leonard, T., Hsu, J.: *Bayesian Methods*. Cambridge University Press, Cambridge (2001)
60. Lee, P.: *Bayesian Statistics*, 3rd edn. Arnold, London (2004)
61. Malinverno, A., Briggs, V.A.: Expanded uncertainty quantification in inverse problems: hierarchical Bayes and empirical Bayes. *Geophysics* **69**(4), 1005–1016 (2004)
62. Park, I., Amarchinta, H.K., Grandhi, R.V.: A Bayesian approach for quantification of model uncertainty. *Reliab. Eng. Syst. Saf.* **95**(7), 777–785 (2010)
63. Oliver, T.A., Moser, R.D.: Accounting for uncertainty in the analysis of overlap layer mean velocity models. *Phys. Fluids* **24**(7), 075108 (2012)
64. Arendt, P.D., Apley, D.W., Chen, W.: Quantification of model uncertainty: calibration, model discrepancy, and identifiability. *J. Mech. Des.* **134**(10), 100908 (2012)
65. Ling, Y., Mullins, J.G., Mahadevan, S.: Options for the inclusion of model discrepancy in Bayesian calibration. In: 16th AIAA Non-Deterministic Approaches Conference, National Harbor. American Institute of Aeronautics and Astronautics (2014)
66. Liu, F., Bayarri, M.J., Berger, J.O.: Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Anal.* **4**(1), 119–150 (2009)
67. Sankaranarayanan, S.: *Uncertainty Quantification and Integration in Engineering Systems*. Vanderbilt University, Nashville (2012)
68. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: *Markov chain Monte Carlo in practice*. Chapman and Hall, London (1996)
69. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953)
70. Gilks, W.R., Wild, P.: Adaptive rejection sampling for Gibbs sampling. *Appl. Stat.* **41**(2), 337–348 (1992)
71. Neal, R.M.: Slice sampling. *Ann. Stat.* **31**(3), 705–767 (2003)
72. American Society of Mechanical Engineers: Guide for Verification and Validation in Computational Solid Mechanics, p. 53. American Society of Mechanical Engineers, New York (2006)
73. Coleman, H.W., Stern, F.: Uncertainties and CFD code validation. *J. Fluids Eng. Asme* **119**(4), 795–803 (1997)
74. Oberkampf, W.L., Barone, M.F.: Measures of agreement between computation and experiment: validation metrics. *J. Comput. Phys.* **217**(1), 5–36 (2006)
75. Ferson, S., Oberkampf, W.L., Ginzburg, L.: Model validation and predictive capability for the thermal challenge problem. *Comput. Methods Appl. Mech. Eng.* **197**(29–32), 2408–2430 (2008)
76. Hills, R.G., Leslie, I.H.: Statistical validation of engineering and scientific models: validation experiments to application. Sandia National Labs., Albuquerque/Livermore (2003).
77. Urbina, A., Paez, T.L., Hasselman, T.K., Wathugala, G.W., Yap, K.: Assessment of model accuracy relative to stochastic system behavior. In: Proceedings of 44th AIAA Structures, Structural Dynamics, Materials Conference, Norfolk, pp. 7–10 (2003)
78. Gelfand, A.E., Dey, D.K.: Bayesian model choice: asymptotics and exact calculations. *J. R. Stat. Soc. Ser. B-Methodol.* **56**(3), 501–514 (1994)
79. Geweke, J.: Bayesian model comparison and validation. *Am. Econ. Rev.* **97**(2), 60–64 (2007)
80. Zhang, R.X., Mahadevan, S.: Bayesian methodology for reliability model acceptance. *Reliab. Eng. Syst. Saf.* **80**(1), 95–103 (2003)
81. Mahadevan, S., Rebba, R.: Validation of reliability computational models using Bayes networks. *Reliab. Eng. Syst. Saf.* **87**(2), 223–232 (2005)

82. Rebba, R., Mahadevan, S.: Computational methods for model reliability assessment. *Reliab. Eng. Syst. Saf.* **93**(8), 1197–1207 (2008)
83. Sankararaman, S., Mahadevan, S.: Assessing the reliability of computational models under uncertainty. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston, pp. 1–8 (2013)
84. Thacker, B.H., Paez, T.L.: A simple probabilistic validation metric for the comparison of uncertain model and test results. In: ASME Verification and Validation Symposium, Las Vegas (2013)
85. Liu, Y., Chen, W., Arendt, P., Huang, H.-Z.: Toward a better understanding of model validation metrics. *J. Mech. Des.* **133**(7), 071005 (2011)
86. Roache, P.J.: *Fundamentals of Verification and Validation*. Hermosa Press, Socorro (2009)
87. Oberkampf, W.L., Roy, C.C.J.: *Verification and Validation in Scientific Computing*. Cambridge University Press, New York (2010)
88. O'Hagan, A.: Fractional Bayes Factors for Model Comparison. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **57**(1), 99–138 (1995)
89. Jiang, X., Mahadevan, S.: Bayesian risk-based decision method for model validation under uncertainty. *Reliab. Eng. Syst. Saf.* **92**(6), 707–718 (2007)
90. Cha, S.: Comprehensive survey on distance/similarity measures between probability density functions. *Int. J. Math. Model. METHODS Appl. Sci.* **1**(4) (2007)
91. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L.: The Mahalanobis distance. *Chemom. Intell. Lab. Syst.* **50**(1), 1–18 (2000)
92. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global Sensitivity Analysis: The Primer*. John Wiley, Chichester (2008)
93. Sobol', I.M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **55**(1–3), 271–280 (2001)
94. Li, C., Mahadevan, S.: Global sensitivity analysis for system response prediction using auxiliary variable method. In: 17th AIAA Non-Deterministic Approaches Conference, Kissimmee (2015)
95. Li, C., Mahadevan, S.: Relative contributions of aleatory and epistemic uncertainty sources in time series prediction. *Int. J. Fatigue* **82**, 474–486 (2016)
96. Singhal, A.: Modern information retrieval: a brief overview. *IEEE Data Eng. Bull.* **24**(4), 35–43 (2001)
97. Van Horn, K.S.: Constructing a logic of plausible inference: a guide to Cox's theorem. *Int. J. Approx. Reason.* **34**(1), 3–24 (2003)
98. Sankararaman, S., Mahadevan, S.: Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliab. Eng. Syst. Saf.* **138**, 194–209 (2015)
99. Li, C., Mahadevan, S.: Uncertainty quantification and integration in multi-level problems. In: IMAC XXXII, Orlando, vol. 3 (2014)
100. Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. *Ann. Math. Stat.* **27**(3), 832–837 (1956)
101. Red-Horse, J.R.R., Paez, T.L.L.: Sandia National Laboratories validation workshop: Structural dynamics application. *Comput. Methods Appl. Mech. Eng.* **197**(29–32), 2578–2584 (2008)
102. Chopra, A.K.: *Dynamics of Structures: Theory and Applications to Earthquake Engineering*, 4th edn. Prentice Hall, Englewood Cliffs (2011)

Yijie Dylan Wang and C. F. Jeff Wu

Abstract

Cubic splines are commonly used in numerical analysis. It has also become popular in the analysis of computer experiments, thanks to its adoption by the software JMP 8.0.2 2010. In this chapter, a Bayesian version of the cubic spline method is proposed, in which the random function that represents prior uncertainty about y is taken to be a specific stationary Gaussian process and y is the output of the computer experiment. A Markov chain Monte Carlo (MCMC) procedure is developed for updating the prior given the observed y values. Simulation examples and a real data application are given to show that the proposed Bayesian method performs better than the frequentist cubic spline method and the standard method based on the Gaussian correlation function.

Keywords

Gaussian process • Markov chain Monte Carlo (MCMC) • Kriging • Nugget • Uncertainty quantification

Contents

1	Introduction	478
2	A Brief Review on Kriging	480
3	Bayesian Cubic Spline	481
3.1	The Prior and Posterior Processes	481
3.2	Nugget Parameter	483

Y.D. Wang (✉)

Blizzard Entertainment, Irvine, CA, USA

e-mail: dylan.jie@gmail.com

C.F.J. Wu

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

e-mail: jeffwu@isye.gatech.edu

3.3 Extension to High Dimensions	484
4 Simulation Study and Results	484
5 Application	488
6 Conclusions	492
References	494

1 Introduction

Because of the advances in complex mathematical models and fast computation, computer experiments have become popular in engineering and scientific investigations. Computer simulations can be much faster or less costly than running physical experiments. Furthermore, physical experiments can be hard to conduct or even infeasible when only rare events, like landslides or hurricanes, are observed. There are many successful applications of computer experiments as reported in the literature. The Gaussian process (GP) has been used as the main tool for modeling computer experiments. See the books by Santner, Williams, and Notz [31]; Fang, Li, and Sudjianto [8]; as well as the November 2009 issue of *Technometrics*, which was devoted to computer experiments.

First, the GP model is introduced. Suppose an experiment involves k factors $\mathbf{x} = (x_1, \dots, x_k)^t$ and n computer runs are performed at $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The input can be written as the $n \times k$ matrix $D = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t$. The corresponding response values are the vector $\mathbf{Y}_D = (y_1, \dots, y_n)^t$. The GP model assumes that

$$y(\mathbf{x}) = \mathbf{b}' \mathbf{f}(\mathbf{x}) + Z(\mathbf{x}), \quad (12.1)$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_s(\mathbf{x}))^t$ is a vector of s known regression functions, $\mathbf{b} = (b_1, \dots, b_s)^t$ is a vector of unknown coefficients, and $Z(\mathbf{x})$ is a stationary GP with mean zero, variance σ^2 , and isotropic correlation function $\text{corr}(y(\mathbf{x}_1), y(\mathbf{x}_2)) = R(\mathbf{x}_1, \mathbf{x}_2) = R(\|\mathbf{x}_1 - \mathbf{x}_2\|)$. For the GP model in (12.1), the best linear unbiased predictor (BLUP) of $y(\mathbf{x})$ is an interpolator, which will be shown in (12.5).

One popular choice of the correlation function is the Gaussian correlation function, which is the product exponential correlation function with power two. For the one-dimension case, it can be written as

$$R(d) = \exp(-\theta d^2), \quad (12.2)$$

where $d = \|\mathbf{x}_1 - \mathbf{x}_2\|$ is the distance between two input values \mathbf{x}_1 and \mathbf{x}_2 and θ is the scale parameter. It has been used in many applications [1, 24, 28, 29] and software including JMP 8.0.2 2010. However, a process $y(\mathbf{x})$ with (12.2) as the correlation function has the property that its realization on an arbitrarily small, continuous interval determines the realization on the whole real line. This global influence of local data is considered unrealistic and possibly misleading in some applications [6, p. 54]. This property will be referred to hereafter as *global prediction*. Another

well-known correlation function is the Matérn family (Matérn, 1960). For the one-dimension case, it is a two-parameter family:

$$R(d) = \{2^{v-1} \Gamma(v)\}^{-1} (d/\phi)^v K_v(d/\phi),$$

where $K_v(\cdot)$ denotes a modified Bessel function of order $v > 0$ and $\phi > 0$ is a scale parameter for the distance d . As $v \rightarrow \infty$, the Matérn correlation function converges to (12.2).

Another commonly used interpolation method is the spline. An order- s spline with knots ξ_i , $i = 1, \dots, l$ is a piecewise polynomial of order s and has continuous derivatives up to order $s - 2$. A cubic spline has $s = 4$. The GP may also be viewed as a spline in a reproducing kernel Hilbert space, with the reproducing kernel given by its covariance function [32]. The main difference between them is in the interpretation. While the spline is driven by a minimum norm interpolation based on a Hilbert space structure, the GP is driven by minimizing the expected squared prediction error based on a stochastic model.

In this chapter, emphasis is placed on the cubic spline by considering it in the GP framework via the cubic spline correlation function [4, 31]:

$$R(d) = \begin{cases} 1 - 6\left(\frac{d}{\theta}\right)^2 + 6\left(\frac{|d|}{\theta}\right)^3, & \text{if } |d| < \frac{\theta}{2}, \\ 2\left(1 - \frac{|d|}{\theta}\right)^3, & \text{if } \frac{\theta}{2} \leq |d| < \theta, \\ 0, & \text{if } |d| \geq \theta, \end{cases} \quad (12.3)$$

where $\theta > 0$ is the scale parameter. Currin et al. [5] showed that the BLUP with the function in (12.3) as the correlation function gives the usual cubic spline interpolator. An advantage of the cubic spline correlation is that θ can be made small, which permits prediction to be based on data in a local region around the predicting location [31, p. 38]. This property shall be referred to hereafter as *local prediction*.

In this chapter, a Bayesian version of the Gaussian process approach is introduced for the cubic spline correlation function given in (12.3). One advantage of Bayesian prediction is that the variability of $y(\mathbf{x})$ given observations can be used to provide measures of posterior uncertainty, and designs can be sought to minimize the expected uncertainty [29, 36]. Some empirical studies have shown the superiority of Gaussian processes over the other interpolating techniques, including splines (see Laslett [18]). Here, some potential advantages of using Bayesian cubic spline in the GP model compared to the power exponential correlation function (12.2) is illustrated through simulation studies.

The chapter is organized as follows. In the following section, a brief review of the kriging technique based on GP models is provided. A methodological section is then presented introducing a Bayesian version of the cubic spline method, abbreviated as BCS. This section also outlines methodology for the nugget parameter in BCS for situations where numerical and estimation stability is required. A summary procedure for BCS is also given, and its extension to high dimensions is discussed.

Next, a simulation section compares BCS with two competing procedures: a GP based on the cubic spline correlation function in (12.3), abbreviated as CS, and a GP based on the power exponential Gaussian correlation function in (12.2), abbreviated as GC (CS and GC are explained in more detail in the review section). The performance of BCS and GC is then compared on some real data in an application section. Finally, remarks on the importance of BCS and future research directions are given in a concluding section.

2 A Brief Review on Kriging

The GP model has been used in geostatistics and is known as kriging [3, 6, 20]. Kriging is used to analyze spatially referenced data which have the following characteristics [3]: The observed values y_i are at a discrete set of sampling locations $\mathbf{x}_i, i = 1, \dots, n$, within a spatial region. The observations y_i are statistically related to the values of an underlying continuous spatial phenomenon. Sacks et al. [29] proposed kriging as a technique for developing meta models from a computer experiment. Computer experiments produce a response for a given set of input variables. Our methodology development only considers deterministic computer experiments, i.e., the code produces identical answers if run twice using the same set of inputs.

Suppose the goal is to predict the output of a function $y(\mathbf{x})$ at an untried location \mathbf{x} , given the observed y values at $D = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t$. For the Gaussian process in (12.1), the best linear unbiased estimator (BLUE) of \mathbf{b} is

$$\hat{\mathbf{b}} = (\mathbf{f}'_D \mathbf{R}_D(\hat{\theta})^{-1} \mathbf{f}_D)^{-1} \mathbf{f}'_D \mathbf{R}_D(\hat{\theta})^{-1} \mathbf{Y}_D, \quad (12.4)$$

where $\mathbf{f}_D = \mathbf{f}(D) = (\mathbf{f}(\mathbf{x}_i))_{\mathbf{x}_i \in D}$, the dependence on θ is now explicitly indicated in the notation, and $\hat{\theta}$ is an estimate of θ . The BLUP of $\mathbf{Y}_0 = y(\mathbf{x}_0)$ at $\mathbf{x}_0 \in \mathbb{R}$ is

$$\hat{\mathbf{Y}}_0 = \hat{\mathbf{b}}' \mathbf{f}_0 + \mathbf{R}_0(\hat{\theta})^t \mathbf{R}_D(\hat{\theta})^{-1} (\mathbf{Y}_D - \hat{\mathbf{b}}' \mathbf{f}_D), \quad (12.5)$$

where $\hat{\mathbf{b}}$ is given in (12.4), $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0)$, and $\mathbf{R}_0 = (R(\mathbf{x}_0 - \mathbf{x}_1), \dots, R(\mathbf{x}_0 - \mathbf{x}_n))^t$ is the $n \times 1$ vector of correlations between \mathbf{Y}_D and \mathbf{Y}_0 . Letting $\boldsymbol{\mu}_D = \mathbf{b}' \mathbf{f}_D$ and $\boldsymbol{\mu}_0 = \mathbf{b}' \mathbf{f}_0$, then (12.5) becomes

$$\hat{\mathbf{Y}}_0 = \hat{\boldsymbol{\mu}}_0 + \mathbf{R}_0(\hat{\theta})^t \mathbf{R}_D(\hat{\theta})^{-1} (\mathbf{Y}_D - \hat{\boldsymbol{\mu}}_D).$$

One way to estimate θ and σ^2 is by maximum likelihood. Maximum likelihood is a commonly used method for estimating parameters in both computer experiments

and spatial process models [4, 19, 29, 30, 34]. For the power exponential correlation function in (12.2), the estimate of σ^2 yields

$$\hat{\sigma}^2(\theta) = \frac{1}{n}(\mathbf{Y}_D - \boldsymbol{\mu}_D)' \mathbf{R}_D(\theta)^{-1} (\mathbf{Y}_D - \boldsymbol{\mu}_D).$$

Estimation of θ is usually done by a constrained iterative search. This method will be referred to as kriging based on the power exponential Gaussian correlation (GC). If the cubic spline correlation function is adopted in (12.3), the correlation parameter θ is both a scale and truncation parameter. In this case, the estimation method of θ is based on the restricted maximum likelihood method (REML). REML [25] was proposed as a method of obtaining less biased estimates of the variance and covariance parameters than the (unrestricted) maximum likelihood method. This method will be referred to as kriging based on the cubic spline correlation function (CS).

3 Bayesian Cubic Spline

3.1 The Prior and Posterior Processes

With $\mathbf{Y}_D \sim \mathcal{N}(\boldsymbol{\mu}_D, \sigma^2 \mathbf{R}_D(\theta))$, the Bayesian framework for the cubic spline method can now be developed, where \mathbf{R} is the correlation matrix based on the function in (12.3). First assign the non-informative priors to $\boldsymbol{\mu}_D$ and θ and the conjugate prior to σ^2 , and further assume that these priors are independent with each other:

$$\begin{aligned}\boldsymbol{\mu}_D &\propto 1, \\ \sigma^2 | \alpha, \beta &\sim \text{InverseGamma}(\alpha, \beta), \\ \theta | a &\sim \mathcal{U}(0, a), \\ \beta &\propto 1/\beta,\end{aligned}$$

where θ follows the uniform distribution in $(0, a)$ and β^{-1} has a non-informative prior over $(0, \infty)$. Here θ can be viewed as the *knot location* parameter in the spline literature. In the Bayesian spline literature [7, 33], it is a common practice to assign a uniform prior to the knot location parameter. The hyperparameter a is fixed as

$$a = \max_{\mathbf{x}_i, \mathbf{x}_j \in D} \|\mathbf{x}_i - \mathbf{x}_j\|.$$

The reason for choosing this particular a is because the function in (12.3) is truncated and equals zero for $|d| \geq \theta$. Since the local prediction property is desired for the GP, a is chosen to be the largest distance among the \mathbf{x} values in D . A simulation study (not reported here) shows that a larger value of a does not change

overall performance in estimation. Because an unknown shape parameter α will bring unnecessary complication in the computation, α is assumed to be fixed and known.

MCMC was used to perform the Bayesian computation [11, 27]. It samples from probability distributions by constructing a Markov chain that has the desired distribution as its equilibrium distribution. Gibbs sampling can be implemented to obtain the posterior distribution of μ_D and β :

$$\begin{aligned}\mu_D \mid \mathbf{Y}_D, \theta, \alpha, \beta &\propto \int \mathbb{P}(\mathbf{Y}_D \mid \mu_D, \theta, \sigma^2) \mathbb{P}(\mu_D) \mathbb{P}(\sigma^2 \mid \alpha, \beta) d\sigma \\ &\propto \int \mathcal{N}(\mu_D, \mathbf{R}_D(\theta, \sigma^2)) \times \text{InverseGamma}(\alpha, \beta) d\alpha d\beta,\end{aligned}\tag{12.6}$$

$$\begin{aligned}\beta \mid \mathbf{Y}_D, \mu_D, \theta, \alpha &\propto \int \mathbb{P}(\mathbf{Y}_D \mid \mu_D, \theta, \sigma^2) \mathbb{P}(\beta) \mathbb{P}(\sigma^2 \mid \alpha, \beta) d\sigma \\ &\propto \int \mathcal{N}(\mu_D, \mathbf{R}_D(\theta, \sigma^2)) \times \beta^{-1} \times \text{InverseGamma}(\alpha, \beta) d\alpha d\beta.\end{aligned}$$

However, the parameters θ are embedded into the covariance function \mathbf{R} in (12.3) and have no posterior distribution in closed form. Hence, posterior samples of θ are obtained using the Metropolis-Hastings (MH) algorithm [14, 22]. The MH algorithm works by generating a sequence of sample values in such a way that, as more and more sample values are produced, the distribution of values more closely approximates the desired distribution. Specifically, at each iteration, the algorithm picks a candidate for the next sample value based on the current sample value. Then, with some probability, the candidate is either accepted (in which case the candidate value is used in the next iteration) or rejected (in which case the candidate value is discarded and the current value is reused in the next iteration).

Specifically, a new value of θ (denoted as θ_{new}) is sampled using a normal density with respect to the existing θ (denoted as θ_{old}). The normal densities work as a *jumping* distribution, because they choose a new sample value based on the current sample. In theory, any arbitrary jumping probability density $Q(\delta_{\text{old}} \mid \delta_{\text{new}})$ can work, where δ is the parameter of interest. Here, a symmetric jumping density $Q(\delta_{\text{old}} \mid \delta_{\text{new}}) = Q(\delta_{\text{new}} \mid \delta_{\text{old}})$ is chosen for simplicity. The variance term in the normal distribution is the jumping size from the old sample to the new sample of the MH algorithm. Here, the variance is chosen to be one. As this jumping size gets smaller, the deviation of new parameters from previous ones should get smaller. The jumping distribution is therefore

$$\log \theta_{\text{new}} \sim \mathcal{N}(\log \theta_{\text{old}}, 1).\tag{12.7}$$

After getting θ_{new} , the acceptance ratio is defined as

$$\begin{aligned} r_1 &= \frac{\mathbb{P}(\mathbf{Y}_D \mid \boldsymbol{\mu}_D, \theta_{\text{new}}, \sigma^2) \mathbb{P}(\theta_{\text{new}} \mid a)}{\mathbb{P}(\mathbf{Y}_D \mid \boldsymbol{\mu}_D, \theta_{\text{old}}, \sigma^2) \mathbb{P}(\theta_{\text{old}} \mid a)} \\ &= \frac{|\mathbf{R}_D(\theta_{\text{new}}, \sigma^2)|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{Y}_D - \boldsymbol{\mu}_D)' \mathbf{R}_D(\theta_{\text{new}}, \sigma^2)^{-1} (\mathbf{Y}_D - \boldsymbol{\mu}_D)\right) \mathbb{1}_{\theta_{\text{new}} \in [0, a]}}{|\mathbf{R}_D(\theta_{\text{old}}, \sigma^2)|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{Y}_D - \boldsymbol{\mu}_D)' \mathbf{R}_D(\theta_{\text{old}}, \sigma^2)^{-1} (\mathbf{Y}_D - \boldsymbol{\mu}_D)\right) \mathbb{1}_{\theta_{\text{old}} \in [0, a]}}. \end{aligned}$$

If $r_1 < 1$, then θ_{new} is accepted with probability r_1 . Otherwise, if $r_1 \geq 1$, then θ_{new} is accepted.

3.2 Nugget Parameter

One possible problem with the kriging approach is the potential numerical instability in the computation of the inverse of the correlation matrix in (12.5). This happens when the correlation matrix is nearly singular. Numerical instability is serious because it can lead to large variability and poor performance of the predictor. The simplest and perhaps most appealing way is to add a nugget effect in the GP modeling. In the spatial statistic literature [3], a nugget effect is introduced to compensate for local discontinuities in an underlying stochastic process. A well-known precursor is the ridge regression in linear regression analysis. Gramacy and Lee [13] gave justifications for the use of nugget in GP modeling for deterministic computer experiments. Here, the option of adding a nugget parameter in GP model is considered by using ridge regression.

Consider the Gaussian model:

$$Y \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R}(\sigma^2, \theta) + \tau^2 I),$$

where τ^2 is the nugget parameter. Adding the matrix $\tau^2 I$ to \mathbf{R} makes the covariance matrix nonsingular and helps stabilize the parameter estimate. MH sampling can be used to estimate τ^2 by letting $\gamma^2 = \tau^2/\sigma^2$ and assigning a uniform $[0, \kappa]$ prior on γ^2 , where κ is fixed and known. The correlation matrices \mathbf{R}_D and \mathbf{R}_0 introduced at the start of this section can then be replaced by $\mathbf{V}_D = \mathbf{R}_D + \gamma^2 I$ and $\mathbf{V}_0 = \mathbf{R}_0 + \gamma^2 I$. To use MH sampling for γ^2 , the jumping distribution

$$\log \gamma_{\text{new}}^2 \sim \mathcal{N}(\log \gamma_{\text{old}}^2, 1), \quad (12.8)$$

is used, with the acceptance ratio

$$\begin{aligned} r_2 &= \frac{\mathbb{P}(\mathbf{Y}_D \mid \boldsymbol{\mu}_D, \gamma_{\text{new}}^2, \theta, \sigma^2) \mathbb{P}(\gamma_{\text{new}}^2 \mid \kappa)}{\mathbb{P}(\mathbf{Y}_D \mid \boldsymbol{\mu}_D, \gamma_{\text{old}}^2, \theta, \sigma^2) \mathbb{P}(\gamma_{\text{old}}^2 \mid \kappa)} \\ &= \frac{|\mathbf{V}_D(\gamma_{\text{new}}^2, \theta, \sigma^2)|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{Y}_D - \boldsymbol{\mu}_D)' \mathbf{V}_D(\gamma_{\text{new}}^2, \theta, \sigma^2)^{-1} (\mathbf{Y}_D - \boldsymbol{\mu}_D)\right) \mathbb{1}_{\gamma_{\text{new}}^2 \in [0, \kappa]}}{|\mathbf{V}_D(\gamma_{\text{old}}^2, \theta, \sigma^2)|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{Y}_D - \boldsymbol{\mu}_D)' \mathbf{V}_D(\gamma_{\text{old}}^2, \theta, \sigma^2)^{-1} (\mathbf{Y}_D - \boldsymbol{\mu}_D)\right) \mathbb{1}_{\gamma_{\text{old}}^2 \in [0, \kappa]}}. \end{aligned}$$

In our computation, the criterion introduced by Peng and Wu [26] is used to determine whether or not to include a nugget effect. The condition number of a square matrix is used as the primary measure of singularity, that is, the ratio of its maximum eigenvalue over its minimum eigenvalue. In particular, the Linear Algebra PACKage (LAPACK), reciprocal condition estimator in MATrix LABoratory (MATLAB) is employed to determine whether the covariance matrix \mathbf{R} is ill-conditioned. If $(\kappa_1(\mathbf{R}))^{-1} < 10\epsilon$, where $\epsilon = 2^{-8}$ is the floating-point relative accuracy, then \mathbf{R} is ill-conditioned and the nugget effect is introduced into the model. Otherwise, the nugget effect is set to be zero.

With the option of adding a nugget parameter, the steps to perform the Bayesian cubic spline are summarized as follows:

1. Set initial values for μ_D , θ , σ^2 and let $\gamma^2 = 0$.
2. Calculate $\kappa_1(\mathbf{R})$.
3. If $(\kappa_1(\mathbf{R}))^{-1} \geq 10\epsilon$, where ϵ is a specified constant, set $\gamma^2 = 0$; sample μ_D and θ from (12.6) and (12.7), respectively. If the parameters do not converge, go back to step 2.
4. If $(\kappa_1(\mathbf{R}))^{-1} < 10\epsilon$, use $\mathbf{V} = \mathbf{R} + \gamma^2 I$ instead of \mathbf{R} and sample μ_D , θ , and γ^2 from (12.6), (12.7), and (12.8), respectively. Repeat this step until convergence.
5. Calculate the estimate of Y_0 using (12.5) with μ_D , θ , and γ^2 .

In step 1, the posterior mode is chosen as the starting value. This choice is sensible and can avoid the need of doing “burn-in” in MCMC. See its justification in Geyer [10].

3.3 Extension to High Dimensions

For multiple dimensions, let $\mathbf{x} \in \mathbb{R}^k$ and assume the correlation function $\mathbf{R}(\mathbf{x}_i, \mathbf{x}_j) = \prod_{t=1}^k \mathbf{R}_t(\mathbf{x}_{i,t} - \mathbf{x}_{j,t}) = \prod_{t=1}^k \mathbf{R}_t(d_t)$, where \mathbf{x}_i and \mathbf{x}_j are in \mathbb{R}^k , d_t is the distance between \mathbf{x}_i and \mathbf{x}_j on the t th dimension, and \mathbf{R}_t is the correlation function for the t th dimension.

The multidimensional spline correlation function $\mathbf{R}(d)$ is the product of the one-dimension spline correlation function with individual parameter θ_t estimated for each dimension [2, 35]. The corresponding Bayesian computation is done by doing MH sampling for each dimension until convergence. Our criterion for convergence is when the change of $\|\theta\|$ between consecutive iterations of the MCMC computation is smaller than 10^{-4} . In our simulation studies, θ converges within 3 min.

4 Simulation Study and Results

First, the performance of the proposed Bayesian cubic spline (BCS) method is compared with two other methods: GC and CS (described in the review section). The criterion for evaluating the performance of the estimators is the integrated mean squared error (IMSE), defined as

$$\text{IMSE}(\hat{f}) = \int_{\Omega} (\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 d\mathbf{x},$$

where f and \hat{f} are respectively the true function values and estimated values and Ω is the region of the \mathbf{x} values. The following mean squared error (MSE) is a finite-sample approximation to the IMSE:

$$\text{MSE}(\hat{f}) = \frac{1}{m} \sum_{i=1}^m (\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i))^2, \quad (12.9)$$

where m is the number of randomly selected points $\{\mathbf{x}_i\}$ from Ω . Three choices of the true function $f(x)$ are considered in Examples 1–3, which range from low to high dimensions and from smooth to non-smooth functions.

Example 1.

$$f_1(\mathbf{x}) = \{1 - \exp(-.5/x_2)\} \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}.$$

This two-dimensional function is from Currin et al. [5], where $\mathbf{x} \in [0, 1]^2$ and $f_1 \in [4.1, 13.8]$. The function f_1 is further scaled into $[0, 1]$. Currin et al. [5] studied a 16-run design in their paper. Four designs are considered here: 4^2 design (16 runs) with levels (.125, .375, .628, .875) [15] and (0, .3333, .6667, 1) [5], 5^2 design (25 runs) with levels (0, .25, .5, .75, 1), and 6^2 design (36 runs) with levels (0, 0.2, 0.4, 0.6, 0.8, 1). Four types of noise ϵ are added to $f_1(\mathbf{x})$: $\mathcal{U}(0, 0)$ (no noise), $\mathcal{U}(0, .2)$, $\mathcal{U}(0, .5)$, and $\mathcal{U}(0, 1)$. As the range of the noise increases from 0 to 1, the function $f_1(\mathbf{x}) + \epsilon$ becomes more rugged. It allows us to compare the performance of the three methods as the true function becomes less smooth.

For noise based on $\mathcal{U}(0, 0)$ (and $\mathcal{U}(0, .2)$, $\mathcal{U}(0, .5)$ and $\mathcal{U}(0, 1)$ respectively), the simulation is conducted as follows. First, a noise is randomly sampled from $\mathcal{U}(0, 0)$ (and $\mathcal{U}(0, .2)$, $\mathcal{U}(0, .5)$, and $\mathcal{U}(0, 1)$). For each simulation, the noise is fixed and denoted as $\{\epsilon_1, \dots, \epsilon_n\}$. Here n , the number of design points, is 16, 16, 25, and 36, respectively, for the four designs. Second, the values of $\{f_1(\mathbf{x}_1), \dots, f_1(\mathbf{x}_n)\}$ are calculated. Then the values of $\{f_1(\mathbf{x}_1) + \epsilon_1, \dots, f_1(\mathbf{x}_n) + \epsilon_n\}$ are treated as the response values by GC, BCS, and CS in parameter estimation. The purpose of this step is to facilitate the study of the robustness of estimation against noise of various sizes. Then, the MSE (see (12.9)) is calculated on $m = 100$ of \mathbf{x} randomly sampled from $[0, 1]^2$. The errors $\{\epsilon_1, \dots, \epsilon_n\}$ and inputs $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ are sampled repeatedly and independently for each simulation, and the average of the MSE values from 500 simulations for each noise and design setting is recorded in Table 12.1. For each simulation setting, the method with the smallest average MSE is highlighted in boldface.

The MCMC iterations of the BCS terminate if the change of parameter estimate is smaller than 10^{-4} . The running time takes about 20 s for an Intel Xeon CPU with 2.66 GHz and 3.00 GB of RAM to reach convergence. The power exponential method performs best when the noise is small ($\mathcal{U}(0, 0)$ and $\mathcal{U}(0, .2)$) or the design

Table 12.1 Average MSE values for GC, BCS, and CS predictors in Example 1

		$\mathcal{U}(0, 0)$	$\mathcal{U}(0, .2)$	$\mathcal{U}(0, .5)$	$\mathcal{U}(0, 1)$
$4^2(J)$	GC	1.0034	1.0437	1.7192	1.8951
	BCS	1.0901	1.1024	1.6722	1.8642
	CS	1.1036	1.2518	1.6714	1.9854
$4^2(C)$	GC	1.1223	1.3495	1.6545	2.4491
	BCS	1.1651	1.4475	1.5819	2.1270
	CS	1.1710	1.4322	1.6318	2.1476
5^2	GC	0.9087	1.0186	1.3516	1.4845
	BCS	1.0112	1.0574	1.2416	1.3391
	CS	1.0481	1.1204	1.2665	1.5816
6^2	GC	0.2171	0.2588	0.5604	0.6352
	BCS	0.2716	0.3079	0.7008	0.6893
	CS	0.2942	0.2769	0.7479	0.8042

size is large (36 run). This is because the example is relatively smooth when noise is small, and the function f_1 contains the exponential term $\exp(-.5/x_2)$, which is best captured by the nonzero exponential correlation function. GC also benefits from the larger sample size of 6^2 , which helps to stabilize the estimate. For relatively small designs (16 and 25 run) with large noise ($\mathcal{U}(0, .5)$ and $\mathcal{U}(0, 1)$), CS and BCS perform better than GC. When the design is small and noise is large, the response surface tends to be very rugged, and there is not enough data for GC to estimate the surface with good precision. A localized estimate like CS and BCS with truncated correlation function gives smaller MSE. BCS in most cases outperforms CS. The over-smoothing property of GC can result in large estimation errors as will be shown in Example 2 and in the application section.

Example 2.

$$f_2(x) = 0.3 \exp^{-1.4x} |\cos(10\pi x)| + 3x.$$

This is a one-dimension function and contains a non-smooth term $|\cos(10\pi x)|$. Here f_2 is scaled into $[0, 1]$. As in Example 1, four types of random noise are added to f_2 , and 5, 10, 20, and 30 design points (i.e., $\{x_1, \dots, x_n\}$ values) are uniformly sampled from $[0, 1]$. In each simulation, noise is sampled and fixed, denoted as $\{\epsilon_1, \dots, \epsilon_n\}$, where $n = 5$ (and 10, 20, and 30, respectively). Then 5 (and 10, 20, and 30) design locations $\{x_1, \dots, x_n\}$ are uniformly sampled from $[0, 1]$. The values of $\{f_2(x_1) + \epsilon_1, \dots, f_2(x_n) + \epsilon_n\}$ are used as the response values. The MSE for each simulation is calculated on $m = 100$ randomly sampled x values in $[0, 1]$. For each design, this procedure is repeated 500 times by taking random samples of $\{\epsilon_i\}_{i=1}^n$ and $\{\mathbf{x}_i\}_{i=1}^n$. The average MSE based on the 500 simulations is given in Table 12.2 for each noise and design setting. Again, the method with the smallest average MSE is highlighted in boldface.

Table 12.2 Average MSE values for GC, BCS, and CS predictors in Example 2

		$\mathcal{U}(0, 0)$	$\mathcal{U}(0, .2)$	$\mathcal{U}(0, .5)$	$\mathcal{U}(0, 1)$
	GC	0.0026	0.2857	0.7214	0.8147
5	BCS	0.0018	0.0772	0.2135	0.2740
	CS	0.0021	0.0518	0.2768	0.2853
	GC	0.0017	0.0389	0.1626	0.3260
10	BCS	0.0013	0.0261	0.0508	0.1892
	CS	0.0013	0.0259	0.0591	0.1964
	GC	0.0015	0.0092	0.1443	0.1547
20	BCS	0.0004	0.0057	0.0721	0.1208
	CS	0.0011	0.0063	0.1125	0.1248
	GC	0.0011	0.0049	0.0754	0.1853
30	BCS	6.70E-04	0.0032	0.0341	0.1807
	CS	7.20E-04	0.0028	0.0596	0.2005

In all cases, CS and BCS beat GC even when no noise is added to the true function. BCS performs better than CS in most cases. CS performs better than BCS in four cases, three of which the difference is not significant. GC gives much worse results when the design size is small (5, 10) and the noise is large ($\mathcal{U}(0, .5)$ and $\mathcal{U}(0, 1)$). This is due to the global prediction property of GC. For non-smooth functions, this can bring in unnecessarily large errors. On the other hand, the better performance of CS and BCS benefits from their local prediction property.

Example 3.

$$f_3(\mathbf{x}) = \frac{2\pi x_1(x_2 - x_3)}{\ln(x_4/x_5)[1 + \frac{2x_1x_6}{\ln(x_4/x_5)x_5^2x_7} + \frac{x_1}{x_8}]}$$

This is an eight-dimensional smooth function from Morris et al. [23], where $x_1 \in [63070, 115600]$, $x_2 \in [990, 1110]$, $x_3 \in [700, 820]$, $x_4 \in [100, 5000]$, $x_5 \in [.05, .15]$, $x_6 \in [1120, 1680]$, $x_7 \in [9855, 12046]$, and $x_8 \in [63.1, 116]$. It is also referred to as the “borehole” data in the literature. Here, each input x_1, \dots, x_8 , as well as the function f_3 , is scaled into $[0, 1]$. Morris et al. [23] proposed a 10-run design with two levels 0 and 1 based on the maximin distance criterion (see Table 12.3). In the study, this 10-run design together with 10-, 20-, and 50-run Latin hypercube designs [21] is used. A n -run Latin hypercube design in $[0, 1]^k$ is based on the Latin hypercube sampling. For each dimension, each of the n values is sampled randomly and independently from each interval $(0, 1/n), \dots, (1 - 1/n, 1)$ and randomly permutes the n values. Here the maximin criterion is employed in choosing the Latin hypercubes, i.e., maximizing the minimum distance between points. As before, four types of noise are added to the true function. In each simulation, after the noise $\{\epsilon_1, \dots, \epsilon_n\}$ is sampled, one Latin hypercube design is generated, denoted by $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where $n = 10, 20, 50$. Then apply GC, CS, and BCS to $\{f_1(\mathbf{x}_1) + \epsilon_1, \dots, f_1(\mathbf{x}_n) + \epsilon_n\}$ for parameter estimation. The MSE is

Table 12.3 A maximin distance design in $[0, 1]^8$ for $n = 10$ [23]

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	1	0	0	1	0	1	1
1	1	1	1	0	0	1	0
1	0	0	1	1	0	0	0
0	1	0	0	1	1	0	0
1	1	0	1	0	1	0	1
0	1	1	0	0	0	0	1
0	0	1	1	1	0	1	1
0	0	0	0	0	1	1	1
0	0	1	1	0	1	0	0
1	0	1	0	1	1	1	0

Table 12.4 MSE values for GC, BCS, and CS predictors in Example 3

		$\mathcal{U}(0, 0)$	$\mathcal{U}(0, .2)$	$\mathcal{U}(0, .5)$	$\mathcal{U}(0, 1)$
	GC	0.0386	0.0457	0.0695	0.1850
10	BCS	0.0490	0.0516	0.0855	0.1608
	CS	0.0501	0.0672	0.0884	0.1691
	GC	0.0277	0.0473	0.0721	0.1821
10LH	BCS	0.0473	0.0655	0.0890	0.1542
	CS	0.0458	0.0632	0.0876	0.1409
	GC	0.0013	0.0125	0.0405	0.1714
20LH	BCS	0.0067	0.0366	0.0784	0.1509
	CS	0.0077	0.0298	0.0868	0.1621
	GC	5.10E-04	0.0096	0.0311	0.1355
50LH	BCS	0.0039	0.0117	0.0520	0.1428
	CS	0.0043	0.0159	0.0581	0.1523

calculated based on $m = 5000$ random samples $\{\mathbf{x}_i\}_{i=1}^{5000}$ in $[0, 1]^8$. The simulations are repeated 1000 times, and the average MSE values are given in Table 12.4. The running time for each simulation of BCS is less than 2 min on the same machine.

The results are similar to those of Example 1. This is expected as they are both smooth functions. GC gives best results among the three methods when the noise is small ($\mathcal{U}(0, 0)$ and $\mathcal{U}(0, .2)$) or the sample size is large (50LH). BCS and CS perform well when sample size is small (10 and 20) and the noise is large ($\mathcal{U}(0, 1)$). BCS generally outperforms CS. Noting that x_4 and x_5 appear in the f_3 function through the \ln function, an alternative simulation is run by taking the log transformation of x_4 and x_5 and rescaling them into $[0, 1]$. Since it does not change the overall picture of comparisons, these results are not reported in the paper.

5 Application

Instead of simulating from known functions, another comparison study is presented using the methane combustion data from Mitchell and Morris [23]. Table 12.5 shows

Table 12.5 Methane combustion data

Run	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
1	0	0	0	0.5	1	1	0.25	7.9315
2	0.25	0.5	0.5	0.75	0	1	0	6.2171
3	0	1	0	0.25	0	0	1	7.8535
4	0.5	0.5	0.75	0	1	0.25	0	7.5708
5	0	0.75	0.75	1	1	1	0.5	6.3491
6	1	0	1	0.25	0	1	0	5.3045
7	0	1	0	0.75	1	0.5	1	8.5372
8	0.75	0.25	0	1	1	0	1	7.871
9	0.5	0.75	0.25	0	0.25	0.5	0.5	7.8725
10	0.25	1	0.75	0.75	0.5	0	0.25	6.593
11	0.5	0	1	0.25	1	0.75	1	6.2131
12	1	0	0	0.5	0.5	0.5	1	7.6311
13	1	0.5	1	0.75	0	0.25	0.5	5.109
14	0	1	0.25	0.25	0.75	1	0	8.4206
15	1	1	0	1	0.25	0	0.5	7.2242
16	0.5	0	0.25	1	0	0.25	0.75	6.0216
17	1	1	1	1	0.5	1	0	5.3495
18	1	1	0.5	0.25	0	1	1	6.0325
19	1	0	1	0	0.75	0	0.25	6.4065
20	0.5	1	0.75	1	0.25	0.75	1	5.5674
21	0.25	0	0.5	0.25	0.25	0	1	6.5214
22	0	0.5	0.5	0	0.75	0.5	0.75	7.7907
23	0.25	0	0	0.25	0	0.75	0.5	7.3542
24	0.75	0.75	1	0	0	0	0	5.8651
25	0.25	0	1	0.5	0.75	0.5	0	6.4489
26	0.75	1	0.25	0.75	1	0.75	0.25	7.6225
27	0	1	1	0.5	0.25	1	0.5	5.8572
28	1	0.5	1	0	1	1	0.5	6.5656
29	1	0.25	1	1	1	0.25	0	5.7137
30	0	0	0	1	0.25	1	1	6.5603
31	0.5	0	0.5	0	0.75	1	0.25	7.5044
32	1	0	0.75	0.75	0.5	0.75	0.25	5.8721
33	1	0	0	0	1	0.25	0	8.206
34	1	0.5	0	1	0	0.75	1	6.3746
35	0.25	0.5	1	0.75	0.5	1	1	5.4478
36	1	0.75	0	0	0.5	1	0.75	7.6953
37	0.5	0	1	0	0	0.5	0.75	5.3423
38	0.5	1	0	1	0	1	0.25	6.4493
39	1	0.25	0	0.5	0	0.5	0	6.8957
40	0.75	0.5	0.25	0.5	1	1	1	7.5563
41	0.75	0.75	0.25	0.5	0	0.25	1	6.7549
42	0.75	0	0.75	0.75	0	1	1	5.0056

(continued)

Table 12.5 (continued)

43	0	0.25	0.25	1	0.75	0.25	0.75	7.4006
44	1	0.25	0.75	0	0.25	0	1	5.6656
45	0.5	0.5	0.5	0.5	0.5	0	0	7.4111
46	0.75	1	0.75	0.25	0.25	0.75	0.25	6.7111
47	1	0.5	0.25	0.25	0.75	0.75	0	7.9182
48	1	0.25	0.5	1	1	1	0.5	6.2543
49	0	0.75	0.5	0.75	0.25	0.25	0.5	6.7319
50	0.25	1	0.25	1	0.75	1	0.75	6.9749

Table 12.6 Average $\hat{\theta}$ values using GC and BCS

	Method	% of input	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	$\hat{\theta}_5$	$\hat{\theta}_6$	$\hat{\theta}_7$
D20	GC	90	0.1137	0.0368	0.2697	0.0606	0.0712	0.0101	0.0137
D20	GC	80	0.1892	0.0878	0.4891	0.0928	0.1082	0.0348	0.0647
D20	GC	50	0.4092	0.2977	0.7952	0.3177	0.3580	0.2476	0.2681
D20	BCS	90	4.1345	4.0635	3.6870	3.8788	3.8440	4.2943	3.3749
D20	BCS	80	4.5264	4.5190	3.6424	3.8407	4.5733	4.6059	4.0459
D20	BCS	50	4.1244	4.8617	3.8581	3.4646	4.2030	5.5327	3.1355
D30	GC	90	0.0958	0.0318	0.1582	0.0335	0.0700	0.0079	0.0144
D30	GC	80	0.1216	0.0647	0.3231	0.0667	0.1017	0.0292	0.0256
D30	GC	50	0.2293	0.1874	0.5877	0.1790	0.2252	0.1315	0.1348
D30	BCS	90	3.7533	4.9328	4.6481	3.6454	5.2628	4.6647	2.0442
D30	BCS	80	3.4020	5.2822	5.4022	3.4501	5.8690	4.6797	2.5237
D30	BCS	50	4.5891	4.9748	5.8604	3.4380	5.5703	4.9078	2.4123
D50	GC	90	0.1297	0.0441	0.1057	0.0394	0.0529	0.0112	0.0129
D50	GC	80	0.1189	0.0404	0.1676	0.0506	0.0741	0.0093	0.0124
D50	GC	50	0.1200	0.0449	0.2086	0.0641	0.0956	0.0123	0.0155
D50	BCS	90	3.3356	4.9131	4.9231	3.2469	3.8125	3.5908	5.0959
D50	BCS	80	3.3427	5.4804	4.3651	3.4601	3.6676	3.3073	4.9948
D50	BCS	50	3.8516	4.7773	4.5888	3.6960	3.8341	3.3936	5.3051

its 50-run design. In addition, Mitchell and Morris gave 20-, 30-, 40-, 50-, and 7-run variable maximin designs in their paper. The first 20, 30, 40, and 50 runs in Table 12.5 consist of these designs, denoted as D20, D30, D40, and D50. The response y is the logarithm of the ignition delay time.

Before conducting the comparison, a careful data analysis is performed to show some feature of the data. For D20, D30, and D50, 90%, 80%, and 50% of the original data are taken as the response values for GC and BCS to estimate θ . The average values of $\hat{\theta}_j$, $j = 1, \dots, 7$, based on 100 simulations, are calculated and given in Table 12.6 for each setting. For each design, the value of $\hat{\theta}$ from GC increases as the number of input data decreases, while the $\hat{\theta}$ values for BCS are more stable. The divergent behavior between GC and BCS for this data can be partially explained by their respective global and local prediction properties. First, note that a larger $\hat{\theta}$ value indicates a smoother surface. As the size of input data gets smaller, the data

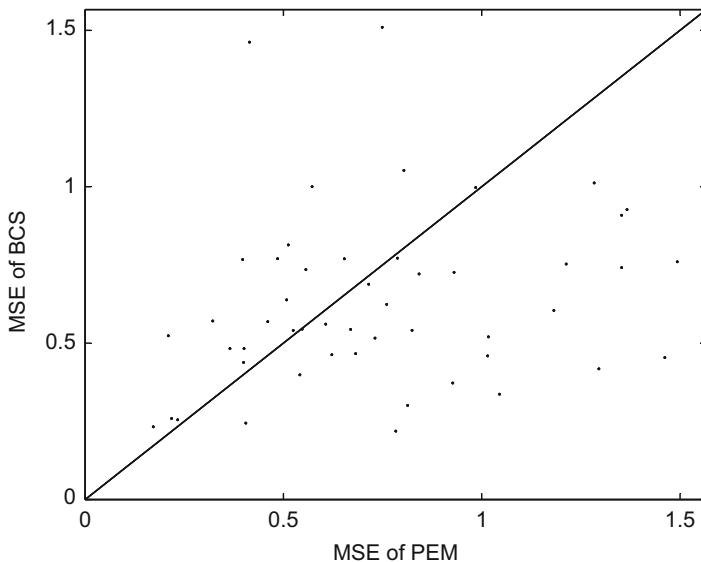


Fig. 12.1 MSE of 40 Training Data in D50

points are spread more thinly in the design region $[0, 1]^7$. The fitted response surface by GC will become more smooth due to its global prediction property. The change will not be as dramatic for BCS, thanks to its local prediction property. Even though the ruggedness of the true response surface is not known, this divergent behavior seems to suggest that BCS is a better method for the data. This is confirmed in the next study based on cross-validation.

The same data and design settings are then used to run cross-validations on D50, D40, and D30 for each of the three methods. One round of cross-validation involves partitioning the data into complementary subsets, performing the analysis on one subset (called the *training set*) and validating the analysis on the other subset (called the *validation set*). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds [9, 17]. Each time, a fixed number of data out of D50 (and D40, D30 respectively) are used for model fitting. The remaining data are used to calculate the MSE in (12.9). Because CS gives much larger MSE in each case, only the MSE results for GC and BCS are reported. For D50, the results of training data size of 40 and 30 are plotted in Figs. 12.1 and 12.2. In each figure, one dot indicates the MSE from GC versus the MSE from BCS for a given design. The reference line of 45° indicates that the two designs are equally good since they render the same MSE. When the majority of the dots are below the line, it means BCS has smaller MSE. This is evident in Fig. 12.2. GC gives some very bad predictions with MSE as high as 5.5 (dots in the right bottom of Fig. 12.2), while the majority of MSE of BCS center around 1.5. The average of MSE from 100 simulations for each cross-validation

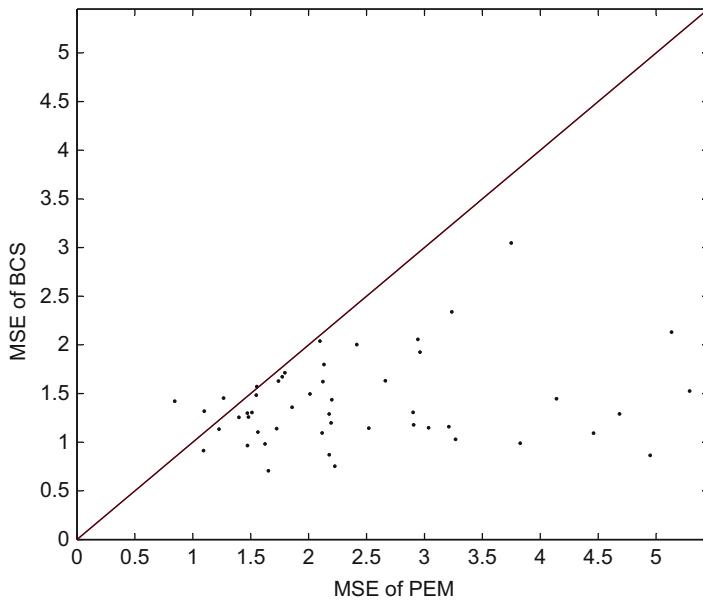


Fig. 12.2 MSE of 30 training data in D50

Table 12.7 Comparison of GC and BCS

Design	Training Data Size	MSE(GC)	MSE(BCS)	% Improvement
D50	40	0.6237	0.6524	-4.6
D50	30	2.4980	1.3908	44.3
D40	30	0.8812	0.7132	19.1
D40	25	2.8516	1.5590	45.3
D30	25	1.6108	0.7888	51.0
D30	20	1.9498	1.2729	34.7

setting for BCS and GC and the percentage of BCS outperforming GC are given in Table 12.7. There is a relatively larger difference between GC and BCS when the training data size is relatively small (20 for D30, 25 for D30, and 30 for D50). This is probably caused by the global prediction and over-smoothing properties of GC. There is no systematic pattern in the percent improvement figures, possibly due to the small number of simulations.

6 Conclusions

The cubic spline is widely used in numerical approximation. In GP modeling, the use of the cubic spline correlation function in (12.3) can lead to a sparse correlation matrix with many zero off-diagonal elements. By comparison, the two commonly

used correlation functions, the Matérn family and the power exponential correlation, do not enjoy this property. A sparse correlation matrix can reduce the cost of computation and enhance its stability. The viability of cubic spline for computer experiment applications received a further boost when JMP 8.0.2 2010 (and its update 11.1.1 2014) provided the power exponential correlation and the cubic spline correlation as its *only* two choices in GP modeling. The prominence that JMP gives to the cubic spline was one motivation for us to develop a Bayesian version of the cubic spline method. By putting a prior on the parameters in the cubic spline correlation function in (12.3), Bayesian computation can be performed by using MCMC. The Bayesian cubic spline should outperform its frequentist counterpart because of its smoothness and shrinkage properties. It also provides posterior estimates, which enable statistical inference on the parameters of interest.

In this chapter, a comparison of BCS with CS and GC is performed in three simulation examples and application to real data. Other correlation functions with compact support have also been considered, such as those from the spherical family:

$$R(d) = \begin{cases} 1 - \frac{3|d|}{2\theta} + \frac{1}{2}\left(\frac{|d|}{\theta}\right)^3, & \text{if } |d| \leq \theta, \\ 0, & \text{if } |d| > \theta. \end{cases}$$

Because the performance of the frequentist version of the spherical family is similar to that of the cubic spline, these results are omitted in the chapter. In the three simulation examples, BCS outperforms CS in most cases. GC performs the best when the true function is smooth or the data size is large. BCS and CS perform better than GC when the true function is rugged and the data size is relatively small. This difference in performance can be explained by the local prediction property of BCS and CS and the global prediction property of GC. Recall that, in global prediction, the prediction at any location is influenced by far-away locations (though with less weights). This leads to over-smoothing, which is not good for a rugged surface. Local prediction does not suffer from this as the prediction depends only on nearby locations. In the real data application, BCS outperforms GC in most choices of design. In summary, when the response surface is non-smooth and/or the input dimension is high, the BCS method can have potential advantages and should be considered for adoption in data analysis.

There are other methods that also attempt to balance between local and global prediction. See, for examples, Kaufman et al. [16] and Gramacy and Apley [12], although the main purpose of these two papers was to leverage sparsity to build emulators for large computer experiments. Comparisons of our proposed method with these and other methods in the literature will require a separate study, and it lies outside our original scope of comparing the Bayesian and the frequentist methods for cubic spline.

Some issues need to be considered in future work. When the dimension is high, the parameter estimation is based on the MH sampling which can be costly. Grouping parameters to reduce computation is an alternative. Also, only non-

informative priors were considered in this chapter. If more information is available, informative prior assignments should be considered.

This work was part of Y. Wang's doctoral thesis at Georgia Tech under the supervision of Jeff Wu. The authors would like to thank Roshan Joseph and Simon Mak for their valuable comments. The research was supported by ARO grant W911NF-14-1-002 and DOE grant DE-SC0010548.

References

1. Abrahamsen, P.: A Review of Gaussian Random Fields and Correlation Functions. *Norsk Regnesentral/Norwegian Computing Center* (1997)
2. Chen, Z., Gu, C., Wahba, G.: Comment on “linear smoothers and additive models”. *Ann. Stat.* **17**(3), 515–521 (1989)
3. Cressie, N.: *Statistics for Spatial Data*. Wiley, Chichester/New York (1992)
4. Currin, C., Mitchell, T., Morris, M., Ylvisaker, D.: A Bayesian approach to the design and analysis of computer experiments. *ORNL-6498* (1988)
5. Currin, C., Mitchell, T., Morris, M., Ylvisaker, D.: Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Stat. Assoc.* **86**(416), 953–963 (1991)
6. Diggle, P.J., Ribeiro, P.J.: *Model-Based Geostatistics*. Springer, New York (2007)
7. DiMatteo, I., Genovese, C.R., Kass, R.E.: Bayesian curve-fitting with free-knot splines. *Biometrika* **88**(4), 1055–1071 (2001)
8. Fang, K.-T., Li, R., Sudjianto, A.: *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC, Boca Raton (2010)
9. Geisser, S.: *Predictive Inference: an Introduction*, vol. 55. CRC Press, New York (1993)
10. Geyer, C.J.: *Introduction to MCMC*. Chapman & Hall, Boca Raton (2011)
11. Gill, J.: *Bayesian Methods: A Social and Behavioral Sciences Approach*. CRC Press, Boca Raton (2002)
12. Gramacy, R.B., Apley, D.W.: Local Gaussian process approximation for large computer experiments. *J. Comput. Graph. Stat.* **24**(2), 561–578 (2014)
13. Gramacy, R.B., Lee, H.K.H.: Cases for the nugget in modeling computer experiments. *Stat. Comput.* **22**(3), 713–722 (2012)
14. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970)
15. Joseph, V.R.: Limit kriging. *Technometrics* **48**(4), 458–466 (2006)
16. Kaufman, C.G., Bingham, D., Habib, S., Heitmann, K., Frieman, J.A.: Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *Ann. Appl. Stat.* **5**(4), 2470–2492 (2011)
17. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence, vol. 14, pp. 1137–1145. Lawrence Erlbaum Associates Ltd (1995)
18. Laslett, G.M.: Kriging and splines: an empirical comparison of their predictive performance in some applications. *J. Am. Stat. Assoc.* **89**(426), 391–400 (1994)
19. Mardia, K.V., Marshall, R.J.: Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika* **71**(1), 135–146 (1984)
20. Matheron, G.: Principles of geostatistics. *Econ. Geol.* **58**(8), 1246–1266 (1963)
21. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245 (1979)
22. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953)

23. Morris, M.D., Mitchell, T.J., Ylvisaker, D.: Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics* **35**(3), 243–255 (1993)
24. O'Hagan, A., Kingman, J.F.C.: Curve fitting and optimal design for prediction. *J. R. Stat. Soc. Ser. B (Methodolog.)* **2**, 1–42 (1978)
25. Patterson, H.D., Thompson, R.: Recovery of inter-block information when block sizes are unequal. *Biometrika* **58**(3), 545–554 (1971)
26. Peng, C.-Y., Wu, C.F.J.: On the choice of nugget in kriging modeling for deterministic computer experiments. *J. Comput. Graph. Stat.* **23**(1), 151–168 (2014)
27. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, vol. 319. Citeseerx, New York (2004)
28. Sacks, J., Schiller, S.: Spatial designs. *Stat. Decis. Theory Relat. Topics IV* **2**(32), 385–399 (1988)
29. Sacks, J., Schiller, S., Welch, W.J.: Designs for computer experiments. *Technometrics* **31**(1), 41–47 (1989)
30. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
31. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2003)
32. Wahba, G.: Spline Models for Observational Data, vol. 59. Society for Industrial and Applied Mathematics, Philadelphia (1990)
33. Wang, X.: Bayesian free-knot monotone cubic spline regression. *J. Comput. Graph. Stat.* **17**(2), 518–527 (2008)
34. Wecker, W.E., Ansley, C.F.: The signal extraction approach to nonlinear regression and spline smoothing. *J. Am. Stat. Assoc.* **78**(381), 81–89 (1983)
35. Ylvisaker, D.: Designs on random fields. *Surv. Stat. Des. Linear Models* **37**(6), 593–607 (1975)
36. Ylvisaker, D.: Prediction and design. *Ann. Stat.* **52**(17), 1–19 (1987)

Propagation of Stochasticity in Heterogeneous Media and Applications to Uncertainty Quantification

13

Guillaume Bal

Abstract

This chapter reviews several results on the derivation of asymptotic models for solutions to partial differential equations (PDE) with highly oscillatory random coefficients. We primarily consider elliptic models with random diffusion or random potential terms. In the regime of small correlation length of the random coefficients, the solution may be described either as the sum of a leading, deterministic, term, and random fluctuations, whose macroscopic law is described, or as a random solution of a stochastic partial differential equation (SPDE). Several models for such random fluctuations or SPDEs are reviewed here.

The second part of the chapter focuses on potential applications of such macroscopic models to uncertainty quantification and effective medium models. The main advantage of macroscopic models, when they are available, is that the small correlation length parameter no longer appears. Highly oscillatory coefficients are typically replaced by (additive or multiplicative) white noise or fractional white noise forcing. Quantification of uncertainties, such as, for instance, the probability that a certain function of the PDE solution exceeds a certain threshold, then sometimes takes an explicit, solvable, form. In most cases, however, the propagation of stochasticity from the random forcing to the PDE solution needs to be estimated numerically. Since (fractional) white noise oscillates at all scales (as a fractal object), its numerical discretization involves a large number of random degrees of freedom, which renders accurate computations extremely expensive. Some remarks on a coupled polynomial chaos – Monte Carlo framework and related concentration (Efron-Stein) inequalities to numerically solve problems with small-scale randomness conclude this chapter.

G. Bal (✉)

Department of Applied Physics and Applied Mathematics, Columbia University, New York,
NY, USA

e-mail: gb2030@columbia.edu

Keywords

Concentration inequalities • Equations with random coefficients • Propagation of stochasticity • Uncertainty quantification

Contents

1	Introduction	498
2	Propagation of Stochasticity for Elliptic Equations	499
2.1	Asymptotic Random Fluctuations in One Dimension	500
2.2	Large Deviations in One Dimension	502
2.3	Some Remarks on the Higher-Dimensional Case	505
3	Equations with a Random Potential	506
3.1	Perturbation Theory for Bounded Random Potentials	506
3.2	Homogenization Theory for Large Random Potentials	510
3.3	Convergence to Stochastic Limits for Long-Range Random Potentials	512
4	Applications to Uncertainty Quantification	514
4.1	Application to Effective Medium Models	515
4.2	Concentration Inequalities and Coupled PCE-MC Framework	515
5	Conclusions	517
	References	518

1 Introduction

Consider a problem modeled by a partial differential equation of the form

$$L([q], x, u) = 0, \quad x \in X \subset \mathbb{R}^d, \quad (13.1)$$

where L is a polynomial in u and its partial derivatives involving a set of parameters $[q]$. When $[q]$ is not known precisely, it may prove useful to model it as a random object, whose statistics are hopefully reasonably well understood. The stochasticity in $[q]$ then propagates to the solution u by means of the constraint (13.1). Decomposing the random variable $[q]$ in a basis of orthogonal polynomials for a measure describing randomness, we can invoke, for instance, the theory of polynomial chaos expansions (PCE), to obtain and numerically compute the decomposition of the random solution u in that basis. The reader is referred to articles in this handbook for an extensive list of presentations and examples.

A major computational difficulty with PCE is that the dimension of the problem is mainly that of the number of degrees of freedom required to represent the random object $[q]$. On a large (spatial or spatio-temporal) domain, that number often proves so overwhelming that direct computations are not feasible. On the other hand, in many situations, an accurate calculation of this propagation of stochasticity may not be necessary. Indeed, the solution u may often be described at a macroscopic scale at which the fine-scale structure of the parameter $[q]$ has been *averaged*. A typical setting for such an averaging process is that of homogenization theory. What we learn from such a theory is that the rapid fluctuations of $[q]$ asymptotically have an *effective* influence on u , essentially by an application of the law of large numbers.

The residual stochasticity in u coming from such rapid fluctuations is then expected to appear as a correction to the law of large numbers, for instance, as an application of a central limit result.

These remarks lead us to the modified modeling of (13.1) given by

$$L([q_0], [q_\varepsilon], x, u_\varepsilon) = 0, \quad x \in X \subset \mathbb{R}^d, \quad (13.2)$$

where $\varepsilon \ll 1$ is a small parameter, $[q_0]$ now represents the low-frequency part of $[q]$ that may be treated numerically by a PCE method, and $[q_\varepsilon]$ is the high-frequency (frequencies larger than $O(\varepsilon^{-1})$) component of $[q]$ whose influence cannot be estimated precisely numerically. The choice of ε is typically driven by computational issues: the number of degrees of freedom characterizing $[q_0]$ is as large as can be computationally managed. It remains to address the influence of $[q_\varepsilon]$, which here we propose to treat asymptotically as $\varepsilon \rightarrow 0$.

As compelling as this program may be, it remains to analyze the asymptotic propagation of stochasticity from the rapidly oscillating coefficients $[q_\varepsilon]$ to the solution u_ε . Unfortunately, the residual stochasticity in u_ε is well understood only for a small class of models. The objective of this contribution is to review the results that have been obtained for some of these models and to stress their applications in uncertainty quantification.

The next section considers elliptic equations of the form $-\nabla \cdot a_\varepsilon \nabla u_\varepsilon = f$ on a domain in \mathbb{R}^d with appropriate boundary conditions. The theory of propagation of stochasticity is well understood in dimension $d = 1$, with current research rapidly shedding light on the higher-dimensional cases. In the subsequent section, we consider equations of the form $P u_\varepsilon + q_\varepsilon u_\varepsilon = 0$ with P typically a (linear) elliptic or parabolic operator. For such models, more is known. As will become clear in the sequel, the propagation of stochasticity critically depends on the random structure of the potential q_ε . The presentation follows in the review paper [9], to which we refer the reader for additional details. The last sections concern the applications of such asymptotic results and their derivation to the field of effective medium models and uncertainty quantification.

2 Propagation of Stochasticity for Elliptic Equations

Consider the elliptic equation

$$\begin{aligned} -\nabla \cdot a\left(\frac{x}{\varepsilon}, \omega\right) \nabla u_\varepsilon &= f, \quad x \in X \subset \mathbb{R}^d \\ u &= g, \quad x \in \partial X, \end{aligned} \quad (13.3)$$

with $a(y, \omega)$ a stationary ergodic random field bounded above and below by positive constants. It is then known that u_ε converges strongly, e.g., in the L^2 sense, to a deterministic, homogenized limit u^* satisfying the same equation as above with a replaced by an effective, deterministic, constant, diffusion tensor a^* ; see, e.g., [29, 31, 33].

As we mentioned earlier, the appearance of this effective tensor may be seen as an application of the law of large numbers. The residual fluctuations in u_ε thus need to be found in the corrections to the homogenized limit u^* . In dimension $d \geq 2$, significantly less is known on the asymptotic structure of $u_\varepsilon - u^*$ than in the much simpler setting with $d = 1$, which we consider first.

2.1 Asymptotic Random Fluctuations in One Dimension

Let us define $a_\varepsilon(x, \omega) = a(\frac{x}{\varepsilon}, \omega)$. When $d = 1$, the above elliptic equation takes the form

$$-\frac{d}{dx} \left(a_\varepsilon(x, \omega) \frac{d}{dx} u_\varepsilon \right) = f(x) \quad \text{in } (0, 1), \quad u_\varepsilon(0, \omega) = 0, \quad u_\varepsilon(1, \omega) = g. \quad (13.4)$$

Here, $a(x, \omega)$ is a stationary ergodic random process satisfying the ellipticity condition $0 < \alpha_0 \leq a(x, \omega) \leq \alpha_0^{-1}$ a.e. for $(x, \omega) \in \mathbb{R} \times \Omega$ where $(\Omega, \mathcal{F}, \mathbb{P})$ is an abstract probability space.

With $F(x) = \int_0^x f(y) dy$, the solution is written using *explicit integrals* of the random coefficient a :

$$u_\varepsilon(x, \omega) = \int_0^x \frac{c_\varepsilon(\omega) - F(y)}{a_\varepsilon(y, \omega)} dy, \quad c_\varepsilon(\omega) = \frac{g + \int_0^1 \frac{F(y)}{a_\varepsilon(y, \omega)} dy}{\int_0^1 \frac{1}{a_\varepsilon(y, \omega)} dy}. \quad (13.5)$$

The stochasticity of u_ε is explicitly characterized by integrals of the random process $a_\varepsilon^{-1}(y, \omega)$. As an application of the law of large numbers, we obtain that u_ε converges strongly in $L^2((0, 1) \times \Omega)$ to its deterministic limit u^* solution of

$$-\frac{d}{dx} \left(a^* \frac{d}{dx} u^* \right) = f(x) \quad \text{in } (0, 1), \quad u^*(0, \omega) = 0, \quad u^*(1, \omega) = g, \quad (13.6)$$

with $a^* = (\mathbb{E}\{a^{-1}(0, \cdot)\})^{-1}$ the harmonic mean of $a(0, \cdot)$. We also have the explicit expression

$$u^*(x) = \int_0^x \frac{c^* - F(y)}{a^*} dy, \quad c^* = ga^* + \int_0^1 F(y) dy. \quad (13.7)$$

The residual stochasticity captured by $u_\varepsilon - u^*$ now depends on the statistical properties of a or equivalently those of $\varphi(x, \omega) := \frac{1}{a(x, \omega)} - \frac{1}{a^*}$. Let R be the correlation function of the stationary random process φ :

$$R(x) = \mathbb{E}\{\varphi(0)\varphi(x)\}. \quad (13.8)$$

The stochastic structure of u_ε strongly depends on the behavior of $R(x)$ as $x \rightarrow \infty$. When $R(x)$ is integrable, then a central limit theory applies. Define $\sigma^2 := \int_{-\infty}^{\infty} R(y)dy > 0$. Then under the additional constraint that φ is strongly mixing, it was shown in [14] that $u_\varepsilon - u^*$ had a variance of order ε and more precisely converged to a Gaussian process after appropriate rescaling:

$$\frac{u_\varepsilon - u^*}{\sqrt{\varepsilon}}(x) \xrightarrow{\varepsilon \rightarrow 0} \sigma \int_0^1 K(x, y)dW_y, \quad (13.9)$$

where $W(y)$ is Brownian motion on $(0, 1)$ and

$$K(x, y) = (\mathbf{1}_{[0,x]}(y) - x)(c^* - F(y)).$$

The above convergence holds in distribution in the space of continuous functions $C[0, 1]$ and may be seen as a functional central limit theorem. When R is integrable, we thus obtain the limit u^* as an application of the law of large numbers and the asymptotic behavior of the random fluctuations $u_\varepsilon - u^*$ beyond homogenization as an application of a central limit.

When $R(x)$ is not integrable, the random variables that are summed in (13.5) are too strongly correlated for the central limit to hold. In some situations, a limiting behavior for $u_\varepsilon - u^*$ can still be obtained. Let us assume that

$$\varphi(x) = \Phi(g_x) \quad (13.10)$$

where g_x is a stationary Gaussian process with mean zero and variance one and Φ is a bounded function such that

$$\begin{aligned} V_0 &= \mathbb{E}\{\Phi(g_0)\} = \int \Phi(g) \frac{e^{-\frac{g^2}{2}}}{\sqrt{2\pi}} dg = 0, \\ V_1 &= \mathbb{E}\{g_0 \Phi(g_0)\} = \int g \Phi(g) \frac{e^{-\frac{g^2}{2}}}{\sqrt{2\pi}} dg > 0. \end{aligned} \quad (13.11)$$

We assume that the correlation function of g :

$$R_g(y) = \mathbb{E}\{g_x g_{x+y}\},$$

decays slowly and is of the form

$$R_g(y) \sim \kappa_g y^{-\alpha} \text{ as } y \rightarrow \infty, \quad (13.12)$$

where $\kappa_g > 0$ and $\alpha \in (0, 1)$. Then we can show [7] that

$$R(y) := \mathbb{E}\{\varphi(x)\varphi(x + y)\} \sim \kappa y^{-\alpha} \text{ as } y \rightarrow \infty \quad \text{with } \kappa = \kappa_g V_1^2. \quad (13.13)$$

We observe that $R(y)$ is no longer integrable. In this setting, we obtain [7] that

$$\frac{u^\varepsilon(x) - u^*(x)}{\varepsilon^{\frac{\alpha}{2}}} \xrightarrow{\varepsilon \rightarrow 0} \sqrt{\frac{\kappa}{H(2H-1)}} \int_{\mathbb{R}} K(x, y) dW_y^H, \quad (13.14)$$

in the space of continuous functions $\mathcal{C}[0, 1]$, where $K(x, y)$ is as above and W_y^H is a fractional Brownian motion with Hurst index $H = 1 - \frac{\alpha}{2}$.

We thus observe that the random fluctuations are of variance $\varepsilon^\alpha \gg \varepsilon$, which is larger than in the case of an integrable correlation function and in fact could be arbitrarily close to $\varepsilon^0 = 1$. Moreover, they are conveniently represented as a stochastic integral with respect to a fractional Brownian motion such that the correlation function of dW_y^H also decays like $y^{-\alpha}$ as $y \rightarrow \infty$.

Note that $\kappa = 0$ when $V_1 = 0$. In such a case, we can also sometimes exhibit a limit for $u_\varepsilon - u^*$, which is no longer Gaussian. Let us assume that $V_0 = V_1 = 0$ and that

$$V_2 = \mathbb{E}\{g_0^2 \Phi(g_0)\} = \int g^2 \Phi(g) \frac{e^{-\frac{g^2}{2}}}{\sqrt{2\pi}} dg > 0, \quad (13.15)$$

in other words, Φ is of Hermite rank 2. Defining $\beta = 2\alpha$, we then observe for $\alpha \in (0, \frac{1}{2})$ [26] that

$$R(y) := \mathbb{E}\{\varphi(x)\varphi(x+y)\} \sim \kappa y^{-\beta} \text{ as } y \rightarrow \infty \quad \text{with } \kappa = \frac{1}{2} \kappa_g^2 V_2^2, \quad (13.16)$$

and obtain the convergence result

$$\frac{u^\varepsilon(x) - u^*(x)}{\varepsilon^{\frac{\beta}{2}}} \xrightarrow{\varepsilon \rightarrow 0} \frac{V_2 \kappa_g}{2} \int_{\mathbb{R}} K(x, y) dR_D(y), \quad (13.17)$$

in the space of continuous functions $\mathcal{C}[0, 1]$, where $K(x, y)$ is as above and $R_D(y)$ is a Rosenblatt process with $D = \frac{\beta}{2} = \alpha$ [36]. The result holds for $\beta \in (0, 1)$ and thus mimics that obtained in (13.14) with a fractional Brownian motion replaced by a non-Gaussian Rosenblatt process.

2.2 Large Deviations in One Dimension

For small enough ε , the homogenized solution u^* captures the bulk of the solution u_ε . The corrector attempts to capture some statistics of the term $u_\varepsilon - u^*$. The above corrector result for integrable correlation R shows that for any $\ell > 0$

$$\mathbb{P}\left[\frac{u_\varepsilon(x) - u^*(x)}{\sqrt{\varepsilon}} \geq \ell\right] \xrightarrow{\varepsilon \rightarrow 0} \mathbb{P}\left[v := \sigma \int_0^1 K(x, y) dW_y \geq \ell\right]. \quad (13.18)$$

More generally, we may ask whether

$$\mathbb{P}[u_\varepsilon(x) \geq \ell] \approx \mathbb{P}[u^*(x) + \sqrt{\varepsilon} v \geq \ell]. \quad (13.19)$$

The answer to the above question is in general negative for $u^*(x) < \ell = O(1)$. This review follows the presentation in [8], to which the reader is referred for additional details, and presents some relevant results in the analysis of (13.19). Let us first recall a few definitions.

Definition 1 (Rate functions). A *rate function* I is a lower semicontinuous mapping (such that for all $\alpha \in [0, \infty)$, the sublevel set $\Psi_I(\alpha) := \{x, I(x) \leq \alpha\}$ is closed) $I : \mathbb{R}^n \rightarrow [0, \infty]$. A *good rate function* is a rate function for which all the sublevel sets $\Psi_I(\alpha)$ are compact.

Definition 2. We say that a family of random vectors $Y_\varepsilon \in \mathbb{R}^n$ satisfy a large deviation principle (LDP) with rate function I if for all $\Gamma \subset \mathbb{R}^n$

$$-\inf_{y \in \Gamma^\circ} I(y) \leq \liminf_{\varepsilon \rightarrow 0} \varepsilon \log P[Y_\varepsilon \in \Gamma] \leq \limsup_{\varepsilon \rightarrow 0} \varepsilon \log P[Y_\varepsilon \in \Gamma] \leq -\inf_{y \in \bar{\Gamma}} I(y).$$

Above, Γ° , $\bar{\Gamma}$ denote the interior and closure of Γ .

Rather than directly analyzing (13.19), we consider the simpler limit for $\ell > \mathbb{E}\{u_\varepsilon\}$:

$$\lim_{\varepsilon \rightarrow 0} \varepsilon \log \mathbb{P}[u_\varepsilon > \ell] = -I_{u_\varepsilon}(\ell) \quad (13.20)$$

assuming such a limit exists. Note that such a limit implies that for all $\delta > 0$ and $\varepsilon < \varepsilon_0(\delta)$, we have

$$e^{-\frac{1}{\varepsilon}(I_{u_\varepsilon}(\ell)+\delta)} \leq \mathbb{P}[u_\varepsilon > \ell] \leq e^{-\frac{1}{\varepsilon}(I_{u_\varepsilon}(\ell)-\delta)}.$$

In $\dim d = 1$, we verify from the explicit above formulas that the solution $u_\varepsilon(x) = g(Z_\varepsilon)$ for $g(z_1, \dots, z_4) = -z_1 + z_2 z_3 z_4^{-1}$ and

$$Z_{\varepsilon,j} = \int_0^1 \frac{H_j(s)}{a_\varepsilon(s)} ds, \quad \mathbf{H} := (H_1, \dots, H_4) = (f 1_{(0,x)}, f, 1_{(0,x)}, 1). \quad (13.21)$$

Let us now describe the main steps of the process leading to a characterization of the rate function $I_{u_\varepsilon}(\ell)$ and first recall two results.

Theorem 1 (Gärtner-Ellis [19]). Suppose

$$\Lambda(\lambda) := \lim_{\varepsilon \rightarrow 0} \varepsilon \log \mathbb{E} e^{\varepsilon^{-1} \lambda \cdot Z_\varepsilon}$$

exists as an extended real number. Furthermore suppose that Λ is essentially smooth, lower semicontinuous and that the origin belongs to the interior of $\mathcal{D}_\Lambda := \{x : \Lambda(x) < \infty\}$. Then Z_ε satisfies an LDP with good convex rate function Λ^* defined by

$$\Lambda^*(\ell) := \sup_{\lambda \in \mathbb{R}^n} [\lambda \cdot \ell - \Lambda(\lambda)].$$

The above theorem allows us to characterize the rate function I_{Z_ε} of the oscillatory integrals in (13.21). From this, we deduce the rate function of I_{u_ε} by means of the following contraction principle:

Theorem 2 (Contraction principle). Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuous and $I : \mathbb{R}^n \rightarrow [0, \infty]$ is a good rate function for the family of random variables Z_ε and associated measures μ_ε ($\mu_\varepsilon(A) = P[Z_\varepsilon \in A]$). For $y \in \mathbb{R}^m$, define

$$I'(y) := \inf\{I(x) \text{ s.t. } x \in \mathbb{R}^n, y = f(x)\}.$$

Then I' is a good rate function controlling the LDP associated with the measures $\mu_\varepsilon \circ f^{-1}$ ($\mu_\varepsilon \circ f^{-1}(B) = P[f(Z_\varepsilon) \in B]$).

In other words, the rate function for u_ε is given by

$$I_{u_\varepsilon}(\ell) = \inf_{z \in g^{-1}\{\ell\}} I_{Z_\varepsilon}(z).$$

It thus remains to obtain a characterization of the rate function of Z_ε . Such a characterization is not straightforward and has been carried out in [8] for a few examples of random media.

Consider (13.4) with $a_\varepsilon(x, \omega) = a_0(x) + b(\frac{x}{\varepsilon}, \omega)$, with $a_0(x)$, as a slight generalization of the coefficients consider so far, allowed to depend on x . The formulas in (13.5) still hold. We assume that

$$b(y, \omega) = v_b \sum_{j=1}^{\infty} \theta_j \mathbf{1}_{[n-1, n)}(y), \quad \theta_i \sim_{i.i.d.} \pi_\theta, \quad |\theta| < 1.$$

Here, a_0 and v_b are chosen so that $0 < v_1 < a_\varepsilon(x, \omega) < v_2$. For a random variable Y , we define the logarithmic moment generating function

$$\mathcal{L}(Y, \lambda) := \log \mathbb{E}[e^{\lambda Y}].$$

Then we have the following result: [8]

Theorem 3. *With g , Z_ε , and $\mathbf{H}(s)$, given as above, define*

$$\Lambda(\lambda) := \int_0^1 \mathcal{L}\left(\frac{1}{a_0(s) + v_b \theta}, \lambda \cdot \mathbf{H}(s)\right) ds.$$

Then $\Lambda \in C^\infty(\mathbb{R}^4)$ is a convex function such that when a_ε is defined as above,

$$\lim_{\varepsilon \rightarrow 0} \varepsilon \log \mathbb{E} e^{\varepsilon^{-1} \lambda \cdot Z_\varepsilon} = \Lambda(\lambda).$$

Moreover, for fixed ξ , Z_ε satisfies a large deviation principle with good convex rate function

$$\Lambda^*(\ell) := \sup_{\lambda \in \mathbb{R}^4} [\lambda \cdot \ell - \Lambda(\lambda)],$$

and u_ε satisfies a large deviation principle with good rate function

$$I_{u_\varepsilon}(\ell) := \inf_{z \in g^{-1}\{\ell\}} \Lambda^*(z).$$

The above example and further examples presented in [8] show that a large deviation principle may be obtained for u_ε and asymptotically characterize the probability $\mathbb{P}[u_\varepsilon > \ell]$. However, the derivation of such a rate function is not straightforward and depends on the whole law of the random coefficient a_ε and not only its correlation function as in the application of central limit results.

2.3 Some Remarks on the Higher-Dimensional Case

The above results strongly exploit the integral representation (13.5), which is only available in the one-dimensional setting. Fewer results exist for the higher-dimensional problem, concerning the propagation of stochasticity from $a_\varepsilon(x, \omega) = a(\frac{x}{\varepsilon}, \omega)$ to u_ε in the elliptic equation $-\nabla \cdot a_\varepsilon \nabla u_\varepsilon = f$ on $X \subset \mathbb{R}^d$, augmented with, say, Dirichlet conditions on ∂X .

As we already indicated, homogenization theory states that u_ε converges to a deterministic solution u^* when the diffusion coefficient $a(x, \omega)$ is a stationary, ergodic (bounded above and below by positive constants) function [29, 31, 33]. Homogenization is obtained by introducing a (vector-valued) corrector χ_ε such that $v_\varepsilon := u_\varepsilon - u^* - \varepsilon \chi_\varepsilon \cdot \nabla u^*$ converges to 0 in the strong H^1 sense. In the periodic setting and away from boundaries, $\varepsilon \chi_\varepsilon \cdot \nabla u^*$ also captures the main contribution of the fluctuations $u_\varepsilon - u^*$ with $v_\varepsilon = o(\varepsilon)$ in the L^2 sense [11]. In the random setting, such results no longer hold. It remains true that v_ε converges to 0 in the H^1 sense, but it is no longer necessary of order $O(\varepsilon)$ in the L^2 sense, as may be observed

in the one-dimensional setting. Moreover, $\varepsilon \chi_\varepsilon \cdot \nabla u^*$ may no longer be the main contribution to the error $u_\varepsilon - u^*$, as shown in, e.g., [27].

Concerning the random fluctuations $u_\varepsilon - u^*$, Yurinskii [37] gave the first statistical error estimate, a nonoptimal rate of convergence to homogenization. Recent results, borrowing from (Naddaf, A., Spencer, T.: Estimates on the variance of some homogenization problems. Unpublished Manuscript, 1998), provide optimal rates of convergence of u_ε to its deterministic limit [23–25] in the discrete and continuous settings for random coefficients with short range (heuristically corresponding to a correlation function R with compact support). In [1, 16], rates of convergence for fully nonlinear equations are also provided.

The limiting law of the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$ is the object of current active research. Using central limit results of [17], it was shown in [32] that the (normalized) fluctuations of certain functionals of u_ε were indeed Gaussian. Central limit results for the effective conductance have been obtained in [12] in the setting of small conductance contrast and using a martingale CLT method. Convergence of the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$ was obtained in the discrete setting in [27].

Rather than describing in detail the results obtained in this rapidly evolving field, we consider in the following section a simpler multidimensional problem with a random (zeroth-order) potential for which a more complete theory is available. We come back to the above elliptic problem briefly in the section devoted to the applications to uncertainty quantifications.

3 Equations with a Random Potential

3.1 Perturbation Theory for Bounded Random Potentials

In this section, we consider linear equations with a random potential of the form

$$P(x, D)u_\varepsilon + q_\varepsilon u_\varepsilon = f, \quad x \in X \tag{13.22}$$

with $u_\varepsilon = 0$ on ∂X , where $P(x, D)$ is a deterministic self-adjoint, elliptic, differential operator and X an open-bounded domain in \mathbb{R}^d . Here, $q_\varepsilon(x, \omega) = q(\frac{x}{\varepsilon}, \omega)$ with q a bounded function. When q defined on $(\Omega, \mathcal{F}, \mathbb{P})$ is ergodic and stationary, its high oscillations ensure that it has a limited influence on u_ε . Define u the solution to

$$P(x, D)u = f, \quad x \in X, \quad u = 0 \text{ on } \partial X, \tag{13.23}$$

which we assume is unique and is defined as

$$u(x) = \mathcal{G}f(x) := \int_X G(x, y)f(y)dy, \tag{13.24}$$

for a Schwartz kernel $G(x, y)$, which we assume is nonnegative, real valued, and symmetric so that $G(x, y) = G(y, x)$.

Then u_ε converges, for instance, in $L^2(X \times \Omega)$, to the unperturbed solution u . We are next interested in a macroscopic characterization of the fluctuations $u_\varepsilon - u$. These fluctuations may be decomposed into the superposition of a deterministic corrector $\mathbb{E}\{u_\varepsilon\} - u$ and the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$. The latter contribution dominates when the Green's function $G(x, y)$ is a little more than square integrable in the sense that

$$C_\eta := \sup_{x \in X} \left(\int_X |G(x, y)|^{2+\eta} dy \right)^{\frac{1}{2+\eta}} < \infty \quad \text{for some } \eta > 0. \quad (13.25)$$

The above constraint is satisfied for $P(x, D) = -\nabla \cdot a(x)\nabla + \sigma(x)$ for $a(x)$ bounded and coercive and $\sigma(x) \geq 0$ bounded in dimension $d \leq 3$.

Under sufficient conditions on the decorrelation properties of $q(x, \omega)$, we obtain that $u_\varepsilon - u$ is well approximated by a central limit theory as in the preceding section. We describe the results obtained in [2]; see also [21]. The main idea is to decompose $u_\varepsilon - u$ as a sum of stochastic integrals that can be analyzed explicitly as in the one-dimensional case and negligible higher-order terms. We now present some details of the derivation of such a decomposition.

We define $\tilde{q}_\varepsilon(\mathbf{x}, \omega) = q(\frac{\mathbf{x}}{\varepsilon}, \omega)$, where $q(\mathbf{x}, \omega)$ is a mean zero, strictly stationary, process defined on an abstract probability space $(\Omega, \mathcal{F}, \mathbb{P})$ [15]. We assume that $q(\mathbf{x}, \omega)$ has an integrable correlation function $R(x) = \mathbb{E}\{q(0)q(x)\}$. We also assume that $q(x, \omega)$ is strongly mixing in the following sense. For two Borel sets $A, B \subset \mathbb{R}^d$, we denote by \mathcal{F}_A and \mathcal{F}_B the sub- σ algebras of \mathcal{F} generated by the field $q(x, \omega)$ for $x \in A$ and $x \in B$, respectively. Then we assume the existence of a $(\rho-)$ mixing coefficient $\varphi(r)$ such that

$$\left| \frac{\mathbb{E}\{(\eta - \mathbb{E}\{\eta\})(\xi - \mathbb{E}\{\xi\})\}}{(\mathbb{E}\{\eta^2\}\mathbb{E}\{\xi^2\})^{\frac{1}{2}}} \right| \leq \varphi(d(A, B)) \quad (13.26)$$

for all (real-valued) square integrable random variables η on $(\Omega, \mathcal{F}_A, \mathbb{P})$ and ξ on $(\Omega, \mathcal{F}_B, \mathbb{P})$. Here, $d(A, B)$ is the Euclidean distance between the Borel sets A and B . We then assume that $\varphi^{\frac{1}{2}}(r)$ is bounded and $r^{d-1}\varphi^{\frac{1}{2}}(r)$ is integrable on \mathbb{R}^+ . We also assume that $q(x, \omega)$ is finite ($dx \times \mathbb{P}$)-a.s. and that $\mathbb{E}\{q^6(0, \cdot)\}$ is bounded. This results allows us to show [2, Lemma 3.2] that

$$\mathbb{E}\{\|\mathcal{G}\tilde{q}_\varepsilon \mathcal{G}\tilde{q}_\varepsilon\|_{\mathcal{L}(L^2(X))}^2\} \leq C\varepsilon^d. \quad (13.27)$$

The equation for u_ε may be formally recast as

$$u_\varepsilon = \mathcal{G}f - \mathcal{G}q_\varepsilon \mathcal{G}f + \mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon u_\varepsilon. \quad (13.28)$$

The above equation may not be invertible for all realizations, even if \mathcal{G} is bounded. We are not interested in the analysis of such possible resonances here and thus modify the definition of our random field q_ε . Let $0 < \rho < 1$. We denote by $\Omega_\varepsilon \subset \Omega$ the set where $\|\mathcal{G}\tilde{q}_\varepsilon \mathcal{G}\tilde{q}_\varepsilon\|_{\mathcal{L}(L^2(X))}^2 > \rho$. We deduce from (13.27) that $\mathbb{P}(\Omega_\varepsilon) \leq C\varepsilon^d$. We thus modify \tilde{q}_ε as

$$q_\varepsilon(\cdot, \omega) = \begin{cases} \tilde{q}_\varepsilon(\cdot, \omega), & \omega \in \Omega \setminus \Omega_\varepsilon, \\ 0, & \omega \in \Omega_\varepsilon, \end{cases} \quad (13.29)$$

and now assume that the above q_ε is a reasonable representation of the physical heterogeneous potential. Note that the process q_ε is no longer necessarily stationary or ergodic. However, since the set of bad realizations Ω_ε is small, all subsequent calculations involving q_ε can be performed using \tilde{q}_ε up to a negligible correction. Now, almost surely, $\|\mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon\|_{\mathcal{L}(L^2(X))}^2 < \rho < 1$ and u_ε is well defined in $L^2(X)$ \mathbb{P} -a.s. Moreover, we observe that

$$(I - \mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon)(u_\varepsilon - u) = -\mathcal{G}q_\varepsilon \mathcal{G}f + \mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon \mathcal{G}f. \quad (13.30)$$

Since $\mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon$ is small thanks to (13.27), we verify that $\mathbb{E}\{\|\mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon(u_\varepsilon - u)\|\} \leq C\varepsilon^d$ is also small.

The analysis of $u_\varepsilon - u$ therefore boils down to that of $\mathcal{G}q_\varepsilon \mathcal{G}f$ and $\mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon \mathcal{G}f$, which are integrals of stochastic field q_ε of a similar nature to those obtained in the preceding section. When (13.25) holds, we obtain that the former term dominates the latter. It thus remains to analyze $\mathcal{G}q_\varepsilon u$, which up to a negligible contribution, is the same as $\mathcal{G}q(\cdot, \omega)u$. This integral may be analyzed as in the one-dimensional setting considered in the preceding section to obtain [2]:

Theorem 4. *Let q satisfy the hypotheses mentioned above. Then we have that*

$$\frac{u_\varepsilon - u}{\varepsilon^{\frac{d}{2}}}(x) \xrightarrow{\varepsilon \rightarrow 0} -\sigma \int_X G(x, y)u(y)dW_y, \quad (13.31)$$

in distribution weakly in space where $\sigma^2 = \int_{\mathbb{R}^d} \mathbb{E}\{q(0)q(x)\}dx < \infty$ and dW_y is a standard multiparameter Wiener measure on \mathbb{R}^d .

Convergence in distribution weakly in space means the following (see below Theorem 5 for a stronger convergence result). Let $\{M_j\}_{1 \leq j \leq J}$ be a finite family of sufficiently smooth functions and define $u_{1\varepsilon} = \varepsilon^{-\frac{d}{2}}(u_\varepsilon - u)$ and $\mathcal{N}(x)$ the right-hand side in (13.31). Then the random vector $(u_{1\varepsilon}, M_j)_{1 \leq j \leq J}$, where (\cdot, \cdot) is the usual inner product on $L^2(X)$, converges in distribution to its limit $(\mathcal{N}, M_j)_{1 \leq j \leq J}$.

When the Green's function $G(x, y)$ is not square integrable, then the deterministic corrector $\mathbb{E}\{u_\varepsilon\} - u$ may be of the same order as or larger than the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$. Assuming that $\mathcal{G}q_\varepsilon \mathcal{G}q_\varepsilon$ can still be controlled, then Theorem 4 can be generalized to this setting under additional assumptions on the

random coefficient $q(x, \omega)$. We refer to [10] for such a theory when the operator P is the square root of the Laplacian, which finds applications in cell biology and the diffusion of molecules through heterogeneous membranes.

Assuming now that the random potential has a slowly decaying correlation function, we expect the random fluctuations $u_\varepsilon - u$ to be significantly larger. Let g_x be a stationary-centered Gaussian random field with unit variance and a correlation function that has a heavy tail

$$R_g(x) = \mathbb{E}\{g_0 g_x\} \sim \kappa_g |x|^{-\alpha} \quad \text{as } |x| \rightarrow \infty$$

for $\kappa_g > 0$ and some $0 < \alpha < d$. Let then $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ bounded (and sufficiently small) so that

$$\mathbb{E}\{\Phi(g_0)\} = \int_{\mathbb{R}} \Phi(g) \frac{e^{-\frac{1}{2}g^2}}{\sqrt{2\pi}} dg = 0, \quad \kappa = \kappa_g (\mathbb{E}\{g_0 \Phi(g_0)\})^2 > 0.$$

We also assume that $\hat{\Phi}(\xi)$, the Fourier transform of Φ , decays sufficiently rapidly so that $\hat{\Phi}(\xi)(1 + |\xi|^3)$ is integrable. We also assume that the Green's function of the operator P satisfies $|G(x, y)| \leq C|x - y|^{-(d-\beta)}$ for some $\alpha < 4\beta$. This condition essentially ensures that the deterministic corrector $\mathbb{E}\{u_\varepsilon\} - u$ is smaller than the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$. Let us assume that $V(x) = \Phi(g_x)$. Then Theorem 4 generalizes to the following result [6]:

Theorem 5. *With the aforementioned hypotheses on the operator P and random potential q , we obtain that*

$$\frac{u_\varepsilon - \mathbb{E}\{u_\varepsilon\}}{\varepsilon^{\frac{\alpha}{2}}} \xrightarrow{\varepsilon \rightarrow 0} - \int_X G(x, y) u(y) W^\alpha(dy), \quad (13.32)$$

in distribution weakly in space, where $W^\alpha(dy)$ is formally defined as $\dot{W}^\alpha(y)dy$ with $\dot{W}^\alpha(y)$ a centered Gaussian random field such that $\mathbb{E}\{\dot{W}^\alpha(x) \dot{W}^\alpha(y)\} = \kappa |x - y|^{-\alpha}$.

The above “weak in space” convergence may often be improved. Consider, for instance, the case of $P(x, D) = -\Delta + 1$ in dimension $d \leq 3$. Then we can show [6, Theorem 2.7] that $Y_\varepsilon := \varepsilon^{-\frac{\alpha}{2}}(u_\varepsilon - \mathbb{E}\{u_\varepsilon\})$ converges in distribution in the space of functions $L^2(X)$ to its limit Y given on the right-hand side of (13.32). This more precise statement means that for any continuous map f from $L^2(X)$ to \mathbb{R} , we have that

$$\mathbb{E}\{f(Y_\varepsilon)\} \xrightarrow{\varepsilon \rightarrow 0} \mathbb{E}\{f(Y)\}, \quad (13.33)$$

so that, for instance, the L^2 norm of Y_ε converges to that of Y . See [6] for some generalizations of the above convergence result.

3.2 Homogenization Theory for Large Random Potentials

In the preceding section, the elliptic problems involved a highly oscillatory potential q_ε satisfying bounds independent of ε . We saw that the limit of the random solution u_ε was given by the solution u obtained by replacing q_ε by its ensemble average. Such a centered potential is therefore not sufficiently strong to have an influence on the leading term u as $\varepsilon \rightarrow 0$.

In this and the following two sections, we consider the more strongly stochastic case where the potential is rescaled such that it has an influence of order $O(1)$ on the limit as $\varepsilon \rightarrow 0$, assuming the latter exists. In this section, we consider results obtained by a diagrammatic expansion method that converges only for Gaussian potentials q_ε . With this restriction in mind, consider the problem

$$\begin{aligned} \frac{\partial u_\varepsilon}{\partial t} + P(D)u_\varepsilon - \frac{1}{\varepsilon^\beta} q\left(\frac{x}{\varepsilon}\right)u_\varepsilon &= 0, \quad t \geq 0, \quad x \in \mathbb{R}^d \\ u_\varepsilon(0, x) &= u_0(x), \quad x \in \mathbb{R}^d, \end{aligned} \tag{13.34}$$

where $d \geq 1$ is spatial dimension, $P(D) = (-\Delta)^{\frac{m}{2}}$ for some $m > 0$, and $q(x)$ is a stationary-centered Gaussian field with correlation function $R(x) = \mathbb{E}\{q(0)q(x)\}$. We assume the initial condition u_0 sufficiently smooth, deterministic, and compactly supported.

The limit of u_ε and the natural choice of β depend on the decorrelation properties of q . When the correlation function of q decays sufficiently rapidly, then averaging effects are sufficiently efficient to imply that u_ε converges to a deterministic solution u . However, when the correlation function of q decays slowly, stochasticity persists in the limit, and u may be shown to be the solution of a stochastic partial differential equation with multiplicative noise. The threshold rate of decay of the correlation is as follows. Define the power spectrum of q as the Fourier transform (up to a factor $(2\pi)^d$) of the correlation function

$$(2\pi)^d \hat{R}(\xi) = \int_{\mathbb{R}^d} e^{-ix \cdot \xi} R(x) dx. \tag{13.35}$$

When it is finite, let us define

$$\rho := \int_{\mathbb{R}^d} \frac{\hat{R}(\xi)}{|\xi|^m} d\xi. \tag{13.36}$$

When the above quantity is finite, then u_ε converges to the deterministic solution of

$$\begin{aligned} \left(\frac{\partial}{\partial t} + P(D) - \rho \right) u(t, x) &= 0, \quad x \in \mathbb{R}^d, \quad t > 0, \\ u(0, x) &= u_0(x), \quad x \in \mathbb{R}^d. \end{aligned} \tag{13.37}$$

When the above integral diverges (because of the behavior of the integrand at $\xi = 0$), then u_ε converges to a stochastic limit described in (13.43) below.

In the case of convergence to a deterministic limit, we have the following result:

Theorem 6. *Let $m < d$ and $R(x)$ be an integrable function or a bounded function such that $R(x) \sim \kappa|x|^{-p}$ as $|x| \rightarrow \infty$ with $m < p < d$. Let us choose $\beta = \frac{m}{2}$.*

Let $T > 0$ sufficiently small. Then there exists a solution to (13.34) $u_\varepsilon(t) \in L^2(\Omega \times \mathbb{R}^d)$ uniformly in $0 < \varepsilon < \varepsilon_0$ for all $t \in [0, T]$. Moreover, let us assume that $\hat{R}(\xi)$ is of class $C^\gamma(\mathbb{R}^d)$ for some $0 < \gamma$ and let $u(t, x)$ be the unique solution in $L^2(\mathbb{R}^d)$ to (13.37). Then, we have the convergence result

$$\|u_\varepsilon(t) - u(t)\|_{L^2(\Omega \times \mathbb{R}^d)} \xrightarrow{\varepsilon \rightarrow 0} 0, \quad (13.38)$$

uniformly in $0 < t < T$.

More precise rates of convergence are given in [4, Theorem 1]. A similar result of convergence holds in the critical dimension $d = m$ with $R(x)$ integrable. In such a case, ε^β has to be chosen as $\varepsilon^{\frac{m}{2}} |\ln \varepsilon|^{\frac{1}{2}}$ [4]. The same method shows that for any choice of potential rescaling $\beta < \frac{m}{2}$, then ρ is replaced by $\varepsilon^{m-2\beta}\rho$ in (13.37) so that u_ε converges uniformly in time on compact intervals $(0, T)$ (with no restriction on T then) to the unperturbed solution u of (13.37) with ρ replaced by 0.

The residual stochasticity of u_ε can be computed explicitly in the diagrammatic expansion. Let us separate $u_\varepsilon - u$ as $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$ and $\mathbb{E}\{u_\varepsilon\} - u$. The latter contribution is a deterministic corrector, which could be larger than the random fluctuations. We refer to [4] for its size and how it may be computed. For the random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$, we have the following convergence result:

Theorem 7. *Under the hypotheses of Theorem 6 and defining $p := d$ when R is integrable, we have*

$$\frac{u_\varepsilon - \mathbb{E}\{u_\varepsilon\}}{\varepsilon^{\frac{p-m}{2}}} \xrightarrow{\varepsilon \rightarrow 0} u_1, \quad (13.39)$$

in distribution and weakly in space, where u_1 is the unique solution of the following stochastic partial differential equation (SPDE) with additive noise

$$\begin{aligned} \left(\frac{\partial}{\partial t} + P(D) - \rho \right) u_1(t, x) &= \sigma u \dot{W}, \quad x \in \mathbb{R}^d, \quad t > 0, \\ u_1(0, x) &= 0, \quad x \in \mathbb{R}^d, \end{aligned} \quad (13.40)$$

where σ is a constant and \dot{W} is a centered Gaussian random field such that

$$\begin{aligned} \sigma^2 &= \int_{\mathbb{R}^d} R(x) dx, & \mathbb{E}\{\dot{W}(x) \dot{W}(x+y)\} &= \delta(y), & p &= d \\ \sigma^2 &= (2\pi)^d \lim_{\xi \rightarrow 0} |\xi|^{d-p} \hat{R}(\xi), & \mathbb{E}\{\dot{W}(x) \dot{W}(x+y)\} &= c_p |y|^{-p}, & m < p < d. \end{aligned} \quad (13.41)$$

Here, we have defined the normalizing constant $c_{\mathfrak{p}} = \frac{\Gamma(\frac{\mathfrak{p}}{2})}{2^{d-\mathfrak{p}} \pi^{\frac{d}{2}} \Gamma(\frac{d-\mathfrak{p}}{2})}$.

The proof of these results may be found in [4] with some extensions in [5]. The convergence result in Theorem 6 was extended to the case of Schrödinger equations (with $\frac{\partial}{\partial t}$ replaced by $i \frac{\partial}{\partial t}$) to arbitrary times $0 < t < T < \infty$ in [39] using the unitarity of the unperturbed solution operator and the decomposition introduced in [20].

3.3 Convergence to Stochastic Limits for Long-Range Random Potentials

The behavior of u_ε is different when the correlation function decays slowly or when $d < \mathfrak{m}$. When \mathfrak{p} tends to \mathfrak{m} , we observe that the random fluctuations (13.39) become of order $O(1)$ and we thus expect the limit of u_ε , when it exists, to be stochastic.

Theorem 8. *Let either $\mathfrak{m} > d$ and $R(x)$ is an integrable function, in which case, we set $\mathfrak{p} = d$, or R is a bounded function such that $R(x) \sim \kappa|x|^{-\mathfrak{p}}$ as $|x| \rightarrow \infty$ with $0 < \mathfrak{p} < \mathfrak{m}$. Let us choose $\beta = \frac{\mathfrak{p}}{2}$.*

Then there exists a solution to (13.34) $u_\varepsilon(t) \in L^2(\Omega \times \mathbb{R}^d)$ uniformly in $0 < \varepsilon < \varepsilon_0$ and $t \in [0, T]$ for all $T > 0$. Moreover, we have the convergence result

$$u_\varepsilon \xrightarrow{\varepsilon \rightarrow 0} u, \quad (13.42)$$

in distribution and in the space of square integrable functions $L^2(\mathbb{R}^d)$, where u is the unique solution (in an appropriate dense subset of $L^2(\mathbb{R}^d \times \Omega)$ uniformly in time) of the following SPDE with multiplicative noise

$$\begin{aligned} \left(\frac{\partial}{\partial t} + P(D) \right) u(t, x) &= \sigma u \dot{W}, \quad x \in \mathbb{R}^d, \quad t > 0, \\ u(0, x) &= u_0(x), \quad x \in \mathbb{R}^d, \end{aligned} \quad (13.43)$$

where σ and \dot{W} are given in (13.41).

The derivation of the above result is presented in [3] with some extensions in [5]. In low dimensions $d < \mathfrak{m}$ and in arbitrary dimension $d \geq \mathfrak{m}$ when the correlation function decays sufficiently slowly that $0 < \mathfrak{p} < \mathfrak{m}$, we observe that the solution u_ε remains stochastic in the limit $\varepsilon \rightarrow 0$. Note that we are in a situation where the integral in (13.36) is infinite. A choice of $\beta = \frac{\mathfrak{m}}{2}$ would generate too large a random potential. Smaller, but with a heavier tail, potentials corresponding to $\beta = \frac{\mathfrak{p}}{2} < \frac{\mathfrak{m}}{2}$ generate an influence of order $O(1)$ on the (limiting) solution u . Any choice $\beta < \frac{\mathfrak{p}}{2}$ would again lead u_ε to converge (in the strong $L^2(\Omega \times \mathbb{R}^d)$ sense then) to the unperturbed solution u of (13.43) with $\sigma = 0$.

Note that it is not obvious that the (Stratonovich) product $u\dot{W}$ is defined *a priori*. \dot{W} is a distribution and as a consequence, u is also singular. It turns out that in order to make sense of a solution to (13.43), we need either a sufficiently low dimension d ensuring that $e^{-tP(D)}$ is an efficient smoothing operator or a sufficiently slow decay $p < m$ ensuring that \dot{W} with statistics recalled in (13.41) is sufficiently regular. When $d < m$ or $m < p$, then the product of the two distributions $u\dot{W}$ in (13.43) cannot be defined as a distribution. From a physical point of view, we may not need such SPDE models since u_ε then converges to the deterministic solution in (13.37) with its random fluctuations described by the well-defined SPDE with additive noise (13.40); see [18] for a general treatment and references to SPDEs.

As for the case of convergence to a deterministic solution, similar results may be obtained for the Schrödinger equation (with $\frac{\partial}{\partial t}$ above replaced by $i\frac{\partial}{\partial t}$); see [30, 38].

The results presented above extend to the setting of time-dependent Gaussian potentials

$$\begin{aligned} \frac{\partial u_\varepsilon}{\partial t} + P(D)u_\varepsilon - \frac{1}{\varepsilon^\beta} q\left(\frac{t}{\varepsilon^\gamma}, \frac{x}{\varepsilon}\right)u_\varepsilon = 0, & \quad t \geq 0, \quad x \in \mathbb{R}^d \\ u_\varepsilon(0, x) = u_0(x), & \quad x \in \mathbb{R}^d, \end{aligned} \quad (13.44)$$

with $0 \leq \gamma \leq m$ and β now chosen as a function of the correlation properties of q , γ , and m . When $\gamma \geq m$, then the temporal fluctuations dominate the spatial fluctuations, and β should be chosen as $\beta = \frac{\gamma}{2}$ when q is sufficiently mixing; see, for instance, [35] when $m = 2$ in one dimension of space for a general mixing coefficient q .

When $0 \leq \gamma \leq m$, then both the spatial and temporal fluctuations of V contribute to the stochasticity of the solution u_ε . Let us define $R(t, x) = \mathbb{E}\{q(s, y)s(s+t, y+x)\}$ the correlation function of q and assume the decay properties

$$R(t, x) \sim \frac{\kappa}{|x|^p t^\beta} \quad \text{as } |x|, t \rightarrow \infty.$$

We restrict ourselves to the setting $0 < \beta < 1$ and $0 < p < d$ with formally $\beta = 1$ when R is integrable in time (uniformly in space) and $p = d$ when R is integrable in space (uniformly in time). Then when p and β are sufficiently small, we again obtain that u_ε converges to the solution of a SPDE, while it converges to a homogenized, deterministic, solution otherwise.

More precisely, when $\beta m + p < m$, then we should choose $\beta = \frac{1}{2}(p + \gamma\beta)$ and u_ε then converges to a SPDE of the form (13.43) with \dot{W} replaced by a spatio-temporal fractional Brownian motion with asymptotically the same correlation function as $R(t, x)$, i.e., such that

$$\mathbb{E}\{\dot{W}(s, x)\dot{W}(s+t, x+y)\} = \frac{c_{p,\beta}}{|y|^p |t|^\beta}, \quad (13.45)$$

for an appropriate constant $c_{p,\beta}$.

When $b m + p > m$, then u_ε converges instead to a homogenized solution given by (13.37). We should choose $\beta = \frac{1}{2}((1-b)m + \gamma b)$ and ρ as

$$\rho = \lim_{\varepsilon \rightarrow 0} \varepsilon^{d-2\beta} \int_0^\infty \int_{\mathbb{R}^d} e^{-t|\xi|^m} \hat{R}\left(\frac{t}{\varepsilon^\gamma}, \varepsilon \xi\right) d\xi dt,$$

with $(2\pi)^d \hat{R}(t, \xi)$ the Fourier transform of $R(t, x)$ with respect to the second variable. We recognize in $e^{-t|\xi|^m}$ the Fourier transform of the fundamental solution of the unperturbed operator $\frac{\partial}{\partial t} + P(D)$. The random fluctuations $u_\varepsilon - \mathbb{E}\{u_\varepsilon\}$ are still given by u_1 solution of the SPDE (13.40) with \dot{W} the spatio-temporal fractional Brownian motion given by (13.45).

We refer to [5] for additional details on these results, which use a diagrammatic expansion that can be applied only to Gaussian potentials. Many results remain valid for non-Gaussian potentials as well; see, e.g., [28, 34, 35].

4 Applications to Uncertainty Quantification

The preceding sections presented problems with different asymptotic models of propagation of stochasticity from the random coefficients to the PDE solutions. Heuristically, we obtain effective medium properties as an application of the law of large numbers and a central limit correction when the random coefficients have sufficiently short-range correlation. For long-range correlations, the random fluctuations of the PDE solution depend more directly on the decay at infinity of the correlation function. In specific cases, we were able to display Gaussian or non-Gaussian asymptotic behaviors for the random fluctuations. In the case of large random potentials, long-range correlations had a more pronounced effect: randomness could not be averaged efficiently and the limiting solution remained stochastic, typically the solution of a stochastic PDE with multiplicative noise.

Such results have direct applications in the quantification of uncertainty. Consider for concreteness the equation

$$-\partial_x \left(a\left(\frac{x}{\varepsilon}, \omega\right) \partial_x u_\varepsilon \right) = f \quad \text{in } (0, 1),$$

with a fluctuation theory given by (13.14) (or (13.9)) $u_\varepsilon(x) = u(x) + \varepsilon^{\frac{\alpha}{2}} \sigma \int_0^1 K(x, t) dW_t^H + r_\varepsilon(x)$, where $\varepsilon^{-\frac{\alpha}{2}} r_\varepsilon$ converges to 0 (in probability in the uniform norm).

From this, we deduce asymptotic results of the form

$$\mathbb{P}\left[u_\varepsilon(x) \geq u^*(x) + \varepsilon^{\frac{\alpha}{2}} \ell\right] = \mathbb{P}\left[\sigma \int_0^1 K(x, t) dW_t^H \geq \ell\right] + o(1). \quad (13.46)$$

We also observed in the one-dimensional case that such results no longer held for $\varepsilon^{\frac{\alpha}{2}} \ell$ of order $O(1)$, where more complex large deviation results needed to be developed.

4.1 Application to Effective Medium Models

How such asymptotic results may be used for computational purposes is less clear. Consider again the above one-dimensional model. We can formally write the limiting model (with dW_t^H formally written as $\dot{W}^H dt$)

$$-\partial_x \left(\frac{a^*}{1 + \varepsilon^{\frac{\alpha}{2}} \tau \dot{W}^H} \partial_x \tilde{u}_\varepsilon \right) = f, \quad (a^*)^{-1} = \mathbb{E}a^{-1},$$

and verify that asymptotically, u_ε and \tilde{u}_ε have the same limiting random fluctuations. The above model is heuristic, and $\varepsilon^{\frac{\alpha}{2}} \tau \dot{W}^H$ should be appropriately smoothed out and truncated to preserve the ellipticity of the diffusion coefficient in the above equation, without significantly changing the law of the fluctuations for $\varepsilon \ll 1$.

From this formal expression preserving the leading contribution to the stochasticity of u_ε , we can draw two conclusions: (i) upscaling of diffusion coefficient involves a small-scale limit taking the form of white noise (when $\alpha = H = \frac{1}{2}$) in short-range case or colored noise (when $\alpha > \frac{1}{2}$) in the long-range case and (ii) the small-scale structure of “homogenized” coefficient makes it very difficult to solve such equations by polynomial chaos expansions since the rapid spatial fluctuations of $\dot{W}^H(x)$ requires a very large number of polynomials in the expansion.

4.2 Concentration Inequalities and Coupled PCE-MC Framework

Let us come back to the general problem (13.2) presented in the introduction and the propagation of stochasticity from $q_0 = q_0(x, \xi)$ and $q_\varepsilon = q_\varepsilon(x, \zeta)$ to $u_\varepsilon = u_\varepsilon(x, \xi, \zeta)$. We assume here that q_0 corresponds to the low-frequency component of the random coefficients and q_ε to their high-frequency component, which means that ξ is relatively *low dimensional*, whereas ζ lives in a high-dimensional space. The asymptotic models presented earlier in this paper allow us to estimate the influence of the parameters ζ on the distribution of u_ε . Unfortunately, the latter propagation can be obtained theoretically only in a very limited class of problems.

When such theoretical results are not available, we would like to devise computational tools that respect the above multi-scale decomposition. One such model consists of combining a PCE method to model the effect of the random variables ξ with a Monte Carlo (MC) method to estimate that of ζ .

Although the theoretical asymptotic results mentioned in the preceding sections involve technical derivations that are problem specific, a central idea underlying the validity of several of them is the following type of inequalities, which we will refer to as Efron-Stein inequalities. We follow here the presentation in [13]. Let

us assume that $\zeta = (\zeta_1, \dots, \zeta_n)$ with n large, which is the case in the asymptotic regimes considered earlier, where typically $n \sim \varepsilon^{-d}$ in dimension d . Now let us assume that the variables ζ_j are *independent* and that they each have an influence on the solution u proportional to their physical volume of order $\varepsilon^d \sim n^{-1}$ and hence small. This means that

$$\sup_{\zeta, \zeta'_i} |u(\zeta_1, \dots, \zeta_n) - u(\zeta_1, \dots, \zeta_{i-1}, \zeta'_i, \zeta_{i+1}, \dots, \zeta_n)| \leq \frac{c_i}{n}, \quad 1 \leq i \leq n, \quad (13.47)$$

for some positive constants c_i , $1 \leq i \leq n$. In other words, variations of ζ_i has an influence on the PDE solution $u(\zeta; \xi, x)$ (say uniformly in (ξ, x)) to simplify the presentation) bounded by c_i/n . If $Z = u(\zeta)$ was the mean of the random variables ζ_j , i.e., $\frac{1}{n} \sum_{j=1}^n \zeta_j$, then the central limit would indicate that Z was approximately Gaussian with variance given by $\frac{1}{n^2} \sum_{j=1}^n \sigma_j^2$ with σ_j the variance of ζ_j . The Efron-Stein inequality [13] states that a similar estimate on the variance holds for a large class of *nonlinear* functionals $Z = u(\zeta)$, including those satisfying the bounded differences constraint (13.47) and for which we have:

$$\text{Var}(Z) \leq \frac{c^2}{2n}, \quad c^2 = \frac{1}{n} \sum_{j=1}^n c_j^2. \quad (13.48)$$

The above result is consistent with those presented in (13.9) in the one-dimensional setting and in Theorem 4, although the latter results do not require a stochastic model involving independent random variables. The results obtained for long-range random fluctuations in Theorem 5, for instance, correspond to random variables that have an effect that is larger than their physical volume. However, another similar averaging mechanism ensures that the global effect of n variables decays as n increases.

Results such as (13.48) do not provide an explicit characterization of the variance of $u(\zeta)$ but rather an upper bound. In general, the effect of these n random variables is not straightforward to describe and hence needs to be obtained by computational means. In settings, where an estimate such as (13.48) may be obtained, we claim that the Monte Carlo method is particularly well suited. Indeed, for $\zeta^{(k)}$, $1 \leq k \leq K$ realizations of ζ and $Z_k = u(\zeta^{(k)})$, we obtain for the empirical mean $S = \frac{1}{K} \sum_{k=1}^K Z_k$ using (13.48) that

$$\text{Var}(S) \leq \frac{c^2}{2Kn},$$

in other words is small even for moderately large values of K provided that n is large.

Coming back to $u = u(x; \xi, \zeta)$ as the solution of an equation of the form

$$L([q_0(\xi)], [q_\varepsilon(\zeta)], x, u) = 0, \quad (13.49)$$

we may envision a PCE-MC coupled method, where ξ involves those random variables for which a bound of the form (13.47) does not hold, which is typically the case if ξ models randomness in the low-frequency component q_0 of the constitutive coefficients, and ζ modeling the high-frequency part q_ε are those (ideally independent) random variables that (ideally) we know as not having an effect larger than the (small) physical volume they represent. Let then u_k for $1 \leq k \leq K$ be the solution of

$$L([q_0(\xi)], [q_\varepsilon(\zeta_k)], x, u_k) = 0.$$

It may formally be written as

$$u_k(x; \xi) = \mathcal{F}_k(x, [q_0(\xi)]),$$

which may then be approximated by representing the solution operator \mathcal{F}_k as a suitable polynomial in the low-frequency random variables ξ (e.g., polynomial chaos expansion); see also [22].

5 Conclusions

The first part of this chapter reviewed several macroscopic models that describe the propagation of stochasticity from (highly oscillatory) random coefficients to random solutions of partial differential equations. Although not exhaustive, the set of examples covered above displays the main features of the problem: (i) the explicit, analytic derivation of macroscopic model is a difficult task that has been completed for a relatively small set of examples and (ii) the propagation of stochasticity depends on the correlation function of the random fluctuations. In the presence of long-range fluctuations, stochasticity may still dominate in the limit of vanishing correlation length and the macroscopic model takes the form of a stochastic partial differential equation. For short-range fluctuations in some models and independent of the correlation function in other models, averaging (law of large number) effects dominate the propagation process, and the limiting model takes the form of a homogenized, effective medium equation. The residual stochasticity may then be characterized as a central limit correction to homogenization when randomness is sufficiently short range. In the presence of long-range fluctuations, the random fluctuations of the solutions are often described by means of integrals of fractional white noise.

The advantages of analytically tractable macroscopic models in uncertainty quantification are clear. When they can be derived, such models provide an explicit expression for the probability density of many functionals of the PDE solution of interest. Moreover, these explicit models typically involve a very small number of parameters, such as the integral or the asymptotic decay of the correlation function.

A notable exception is the case of large deviation results (see, e.g., Theorem 3), which involve the fine structure of the random process. Finally, these models show that the random fluctuations of PDE solutions typically involve functionals of (fractional) white noise.

The presence of such fractal processes is, however, problematic when the propagation of stochasticity needs to be performed computationally: white noise requires a large number of degrees of freedom to be modeled accurately. In such a situation, the last part of this chapter presented a combined PCE-MC computational framework in which large-scale random coefficients are treated by polynomial chaos expansions, while the large number of small-scale coefficients is handled by Monte Carlo. In the event that each small-scale random coefficient has a small influence on the final solution, concentration (Efron-Stein) inequalities, which are consistent with a central limit scaling, show that the collective influence of these random coefficients has a relatively small variance that can be accurately predicted by a reasonable number of MC samples.

References

1. Armstrong, S.N., Smart, C.K.: Quantitative stochastic homogenization of elliptic equations in nondivergence form. *Arch. Ration. Mech. Anal.* **214**, 867–911 (2014)
2. Bal, G.: Central limits and homogenization in random media. *Multiscale Model. Simul.* **7**(2), 677–702 (2008)
3. Bal, G.: Convergence to SPDEs in Stratonovich form. *Commun. Math. Phys.* **212**(2), 457–477 (2009)
4. Bal, G.: Homogenization with large spatial random potential. *Multiscale Model. Simul.* **8**(4), 1484–1510 (2010)
5. Bal, G.: Convergence to homogenized or stochastic partial differential equations. *Appl. Math. Res. Express* **2011**(2), 215–241 (2011)
6. Bal, G., Garnier, J., Gu, Y., Jing, W.: Corrector theory for elliptic equations with long-range correlated random potential. *Asymptot. Anal.* **77**, 123–145 (2012)
7. Bal, G., Garnier, J., Motsch, S., Perrier, V.: Random integrals and correctors in homogenization. *Asymptot. Anal.* **59**(1–2), 1–26 (2008)
8. Bal, G., Ghanem, R., Langmore, I.: Large deviation theory for a homogenized and “corrected” elliptic ode. *J. Differ. Equ.* **251**(7), 1864–1902 (2011)
9. Bal, G., Gu, Y.: Limiting models for equations with large random potential: a review. *Commun. Math. Sci.* **13**(3), 729–748 (2015)
10. Bal, G., Jing, W.: Corrector theory for elliptic equations in random media with singular Green’s function. Application to random boundaries. *Commun. Math. Sci.* **9**(2), 383–411 (2011)
11. Bensoussan, A., Lions, J.-L., Papanicolaou, G.C.: Homogenization in deterministic and stochastic problems. In: *Symposium on Stochastic Problems in Dynamics*, University of Southampton, Southampton, 1976, pp. 106–115. Pitman, London (1977)
12. Biskup, M., Salvi, M., Wolff, T.: A central limit theorem for the effective conductance: linear boundary data and small ellipticity contrasts. *Commun. Math. Phys.* **328**, 701–731 (2014)
13. Boucheron, S., Lugosi, G., Bousquet, O.: Concentration inequalities. In: *Advanced Lectures on Machine Learning*. Volume 3176 of *Lecture Notes in Computer Science*, pp. 208–240. Springer, Berlin (2004)
14. Bourgeat, A., Piatnitski, A.: Estimates in probability of the residual between the random and the homogenized solutions of one-dimensional second-order operator. *Asymptot. Anal.* **21**, 303–315 (1999)

15. Breiman, L.: Probability. Volume 7 of Classics in Applied Mathematics. SIAM, Philadelphia (1992)
16. Caffarelli, L.A., Souganidis, P.E.: Rates of convergence for the homogenization of fully nonlinear uniformly elliptic PDE in random media. *Invent. Math.* **180**, 301–360 (2010)
17. Chatterjee, S.: Fluctuations of eigenvalues and second order Poincaré inequalities. *Prob. Theory Relat. Fields* **143**, 1–40 (2009)
18. Da Prato, G., Zabczyk, J.: Stochastic Equations in Infinite Dimensions. Cambridge University Press, Cambridge (2008)
19. Dembo, A., Zeitouni, O.: Large Deviations Techniques and Applications. Applications of Mathematics. Springer, New York (1998)
20. Erdős, L., Yau, H.T.: Linear Boltzmann equation as the weak coupling limit of a random Schrödinger Equation. *Commun. Pure Appl. Math.* **53**(6), 667–735 (2000)
21. Figari, R., Orlandi, E., Papanicolaou, G.: Mean field and Gaussian approximation for partial differential equations with random coefficients. *SIAM J. Appl. Math.* **42**(5), 1069–1077 (1982)
22. Ghanem, R.G.: Hybrid stochastic finite elements and generalized Monte Carlo simulation. *Trans. ASME* **65**, 1004–1009 (1998)
23. Gloria, A., Otto, F.: An optimal variance estimate in stochastic homogenization of discrete elliptic equations. *Ann. Probab.* **39**, 779–856 (2011)
24. Gloria, A., Otto, F.: An optimal error estimate in stochastic homogenization of discrete elliptic equations. *Ann. Appl. Probab.* **22**, 1–28 (2012)
25. Gloria, A., Otto, F.: An optimal variance estimate in stochastic homogenization of discrete elliptic equations. *ESAIM Math. Model. Numer. Anal.* **48**, 325–346 (2014)
26. Gu, Y., Bal, G.: Random homogenization and convergence to integrals with respect to the Rosenblatt process. *J. Differ. Equ.* **253**(4), 1069–1087 (2012)
27. Gu, Y., Mourrat, J.-C.: Scaling limit of fluctuations in stochastic homogenization. *Probab. Theory Relat. Fields* (2015, to appear)
28. Hairer, M., Pardoux, E., Piatnitski, A.: Random homogenization of a highly oscillatory singular potential. *Stoch. Partial Differ. Equ.* **1**, 572–605 (2013)
29. Jikov, V.V., Kozlov, S.M., Oleinik, O.A.: Homogenization of Differential Operators and Integral Functionals. Springer, New York (1994)
30. Komorowski, T., Nieznaj, E.: On the asymptotic behavior of solutions of the heat equation with a random, long-range correlated potential. *Potential Anal.* **33**(2), 175–197 (2010)
31. Kozlov, S.M.: The averaging of random operators. *Math. Sb. (N.S.)* **109**, 188–202 (1979)
32. Nolen, J.: Normal approximation for a random elliptic equation. *Probab. Theory Relat. Fields* **159**, 661–700 (2014)
33. Papanicolaou, G.C., Varadhan, S.R.S.: Boundary value problems with rapidly oscillating random coefficients. In: Random Fields, Esztergom, 1979, Volumes I and II. Colloquia Mathematica Societatis János Bolyai, vol. 27, pp. 835–873. North Holland, Amsterdam/New York (1981)
34. Pardoux, E., Piatnitski, A.: Homogenization of a singular random One dimensional PDE. *GAKUTO Int. Ser. Math. Sci. Appl.* **24**, 291–303 (2006)
35. Pardoux, E., Piatnitski, A.: Homogenization of a singular random one-dimensional PDE with time-varying coefficients. *Ann. Probab.* **40**, 1316–1356 (2012)
36. Taqqu, M.S.: Weak convergence to fractional Brownian motion and to the Rosenblatt process. *Probab. Theory Relat. Fields* **31**, 287–302 (1975)
37. Yurinskii, V.V.: Averaging of symmetric diffusion in a random medium. *Siberian Math. J.* **4**, 603–613 (1986). English translation of: *Sibirsk. Mat. Zh.* **27**(4), 167–180 (1986, Russian)
38. Zhang, N., Bal, G.: Convergence to SPDE of the Schrödinger equation with large, random potential. *Commun. Math. Sci.* **5**, 825–841 (2014)
39. Zhang, N., Bal, G.: Homogenization of a Schrödinger equation with large, random, potential. *Stoch. Dyn.* **14**, 1350013 (2014)

Polynomial Chaos: Modeling, Estimation, and Approximation

14

Roger Ghanem and John Red-Horse

Abstract

Polynomial chaos decompositions (PCE) have emerged over the past three decades as a standard among the many tools for uncertainty quantification. They provide a rich mathematical structure that is particularly well suited to enabling probabilistic assessments in situations where interdependencies between physical processes or between spatiotemporal scales of observables constitute credible constraints on system-level predictability. Algorithmic developments exploiting their structural simplicity have permitted the adaptation of PCE to many of the challenges currently facing prediction science. These include requirements for large-scale high-resolution computational simulations implicit in modern applications, non-Gaussian probabilistic models, and non-smooth dependencies and for handling general vector-valued stochastic processes. This chapter presents an overview of polynomial chaos that underscores their relevance to problems of constructing and estimating probabilistic models, propagating them through arbitrarily complex computational representations of underlying physical mechanisms, and updating the models and their predictions as additional constraints become known.

Keywords

Polynomial chaos expansions • Stochastic analysis • Stochastic modeling • Uncertainty quantification

R. Ghanem (✉)

Department of Civil and Environmental Engineering, University of Southern California,
Los Angeles, CA, USA
e-mail: rghanem@usc.edu

J. Red-Horse

Engineering Sciences Center, Sandia National Laboratories, Albuquerque, NM, USA
e-mail: jreddho@sandia.gov

Contents

1	Introduction	522
2	Mathematical Setup	526
3	Polynomial Chaos	527
4	Representation of Stochastic Processes	532
5	Polynomial Chaos with Random Coefficients: Model Error	534
6	Adapted Representations of PCE	537
7	Stochastic Galerkin Implementation of PCE	539
7.1	Nonintrusive Evaluation of the Stochastic Galerkin Solution	539
7.2	Adapted Preconditioners for the Stochastic Galerkin Equations	541
8	Embedded Quadratures for Stochastic Coupled Physics	542
9	Constructing Stochastic Processes	545
10	Conclusion	547
	References	547

1 Introduction

Uncertainty quantification is an age-old scientific endeavor that has been reshaped in recent years in response to emerging technologies relevant to sensing and computing. Indeed, scientists' ability to make experimental observations of physical nature over a wide range of length scales is now matched by their ability to numerically resolve comprehensive mathematical formulations of the relevant physical phenomena and their interactions. This convergence of technologies has raised expectations that prediction science is now equipped to support critical decisions that have long eluded rigorous analysis. Examples of these decisions permeate such fields as climate science, material science, manufacturing, urban science, coupled interacting infrastructures, and reaction kinetics, to name only a few. An examination of the unresolved logical, conceptual, and technical questions relevant to prediction science identifies uncertainties as hurdles at several stages of the analysis process. These uncertainties enter at the very beginning with the selection of the mathematical model that is chosen to represent the physics-based constraints in the scenario of interest. Uncertainty characterizations for the parameters contained in that model are themselves models of information and are subject to additional, information-induced limitations, such as those associated with the conversion of available raw information into inference about the parameters in question. These conversions include least squares, maximum likelihood estimation methods, Bayesian strategies, or maximum entropy methods. Additional uncertainties are induced by limits on the amount of experimental evidence or in the particular form this evidence takes. All of these are factors that influence the inverse problems associated with parameter estimation. Yet another source of uncertainty stems from approximating the “push forward” of assumed input uncertainties into outputs of interest using the mathematical model describing the physics. In science and engineering, these mathematical models consist of mixtures of ordinary and partial differential equations, of integral equations, and of algebraic equations that are

eventually further constrained through an optimization problem involving chance constraints.

A critical challenge for organizing this mathematical medley is to formulate a logical, consistent, and operational perspective together with associated mathematical constructs for characterizing all the uncertainties present in a typical prediction/assessment/decision workflow.

On the logical side, a mathematical theory should reflect a clear interpretation of uncertainty against which the behavior of corresponding models can be validated. Such an interpretation must be grounded in physics and technology, since uncertainty can be reduced by augmenting information through modeling and sensing. Mathematical models of uncertainty should therefore be parameterized so as to behave consistently as new knowledge is acquired. In many instances this may require the derivation of new physics models and not merely probabilistic re-parameterization of existing models. Random operator models provide a hint as to what these new models may look like, as they provide stochastic operator-valued perturbations to deterministic models [13, 62, 81].

One of the critical objectives of uncertainty quantification is to strike a rigorously quantifiable balance between the weight of experimental evidence, the magnitude of numerical errors, and the credibility of related decisions. The first of these is associated with the paucity and quality of experimental data as well as the choice of physical and mathematical models. Numerical errors can be attributed to discretization errors, convergence tolerance of algorithms (e.g., linear and nonlinear solvers), and finite (numerical) sample statistics. Credibility of decisions is clearly increased as various errors are assessed and their influence on said decisions estimated. Clearly, the significance of this credibility is a function of the criticality of the object of decision and subsequent consequences of unplanned failure or suboptimal performance.

The polynomial chaos expansion (PCE) methodology [40] is an approach that emerged over the past 20 years with the promise of the simultaneous and consistent mathematical characterization of data errors and numerical errors by relating them both to an analogous treatment grounded in white noise calculus. This approach essentially embeds the problem in a high-dimensional setting that is sufficiently structured to describe all uncertain parameters and all square-integrable mappings on them. A challenge with this approach has been the “curse of dimensionality” that stems from the need to characterize arbitrary nonlinear mappings in high-dimensional spaces and which is manifested in the form of high-order multivariate polynomial expansions in high dimensions. This challenge is exacerbated manyfold in the context of decision-making and design, both manifestations of an optimization problem that iterates on an already very expensive stochastic function evaluator. Mathematical problems with uncertainty can generally be understood, and treated, as involving alternate independent possible realities. This perspective is very appealing in that it permits the formulation of this class of problems as a collection of standard deterministic problems. Imposing a probabilistic structure on the uncertainty is then manifested by a statistical treatment of this collection of problems. Thus if the general solution to these problems is symbolically denoted by

$u(x, \omega)$, where $x \in \mathcal{D}$ and $\omega \in \Omega$ reference, respectively, the “usual” deterministic domain of u and the sample space of the underlying experiment, one is led to approximating the solution, independently, for each value of ω . The ω -related errors can subsequently be analyzed through a statistical analysis of these approximations. Within the same probabilistic framework, mathematical problems with uncertainty have an alternative formulation. Specifically, $u(x, \omega)$ can be viewed as a function of two variables, and approximation schemes over a corresponding product space can then be pursued. Pursuing this line of inquiry requires a detailed understanding of the structure of the function spaces associated with the domains \mathcal{D} and Ω and the behavior of operators acting on them.

Operators of mathematical physics and other dynamical systems are generally construed as mappings between function spaces, with the domain of the mapping typically defined by the boundary and initial data. The smoothness and stability of the solutions to these equations are then explored in terms of the properties of these function spaces and operators. As indicated above, the introduction of uncertainty in the mathematical formulation replaces the standard operators with new ones defined on new function spaces with a domain that now represents the extended data which includes probabilistic reference to the parameters.

Even the simplest linear operators of mathematical physics exhibit nonlinear dependence on their parameters. These new operators on the extended domain are thus generally nonlinear, and methods of nonlinear analysis have been brought to bear on their mathematical exploration. The initial development of nonlinear analysis tools was fashioned for deterministic nonlinear differential equations and consisted of Taylor expansions that were generalized to infinite dimensions by Volterra [100]. The extension of these ideas to nonlinear problems with stochastic forcing was initiated by Wiener [102] and Itô [49], culminating in the construction of polynomial chaos decompositions (also known as multiple Wiener integrals) and the Wiener-Itô-Segal isomorphism ultimately laying the foundation for the development of white noise calculus [46, 47]. In all of these approaches, the essential mathematical challenge was the need for new measure and convergence constructs suitable for infinite dimensional analysis. Wiener began to work on this by first developing an abstraction of Brownian motion [102], extending the work of Paul Lévy [55], and eventually setting up an analysis framework in the particular infinite dimensional space endowed with the Gaussian measure [103]. Following the seminal work of Cameron and Martin [18] in which convergence of infinite dimensional approximations in Wiener’s Gaussian space was established, a flurry of activity ensued applying Wiener’s ideas to various nonlinear systems. In particular, a series of PhD dissertations at the Research Laboratory of Electronics at MIT explored a wide range of mathematical issues that were focused mainly on physical realizability of these systems [15, 17, 31]. Issues of causal discretization of the Brownian motion using Laguerre polynomials were investigated as were issues of adaptive refinement in the probability space, a precursor of recent multielement and h -type refinements of polynomial chaos expansions [59, 101]. Inspired by the pioneering work of Volterra and Wiener, nonlinear problems from across

science and engineering were successfully tackled, and a very strong foundation for nonlinear system theory was forged [19, 48, 49, 51, 52, 65, 75].

Extensions of Wiener's formalism to non-Gaussian measures have been pursued with some success. In particular, those ideas were readily extended to functionals of Lévy processes [50, 78, 79, 104]. Methods based on Jacobi fields [58] and bi-orthogonal expansions [3, 53] have provided a systematic approach for these extensions to other processes [12, 54].

Interest in treating problems with random coefficients, as opposed to problems with random external forcing, was initially motivated by waves propagating in heterogeneous media [13, 16, 82]. This work relied upon perturbation methods and low-order Taylor expansions as essential means for treating the nonlinearities introduced by parametric dependence. The adaptation of these expansions to general random operators soon followed in the form of stochastic Green's functions [1, 2] and Neumann expansions of the inverse operator [107]. Integration of these approaches into finite element formalisms enabled their application to a much wider range of problems [11, 44, 45, 57, 63, 80, 87]. In spite of their algorithmic and computational simplicity, issues of convergence and statistical significance limited the applicability of these methods to problems with relatively small dependence on the random coefficients. It is critical to note here that during this time period, Monte Carlo sampling methods were challenged with limited computational resources, thus motivating the development of alternative formalisms.

The adaptation of the Wiener-based approaches to problems of parametric dependence required a change of perspective from Wiener and Volterra's initial efforts. Parametric noise as motivated by the initial physical problems described above fluctuates in space and does not exhibit the non-anticipative behavior, i.e., the clear distinction between past and future, implicit in time-dependent processes. Pioneering efforts in this direction were carried out by Ghanem and Spanos [40] where the Karhunen-Loëve expansion of a correlated Gaussian process was used to embed the problem in an infinite dimensional Gaussian space. This embedding enabled the use of the polynomial chaos development of Wiener to represent the various stochastic processes involved. That work also introduced a Galerkin projection scheme to integrate polynomial chaos expansions into an approximation formalism for algebraic and differential equations with random coefficients. Extensions of that work with basis adaptation [56], hybrid expansions [32], geometric uncertainties [35], dynamic model reduction [39], and nonlinear partial differential equations and coupled physics [24, 33, 36, 60] were also developed around that time. Given its reliance on Wiener's discretization of Brownian motion, these initial applications of polynomial chaos expansions to parametric uncertainty were limited to parameters with Gaussian measure or models that are simple transformation of Gaussian processes, such as lognormal processes. The Karhunen-Loëve expansion provides a natural discretization of a stochastic process into a denumerable set of jointly distributed random variables. Other settings with denumerable random variables are common in science and engineering and are typically associated with a collection of generally dependent scalar or vector random variables used to parameterize

the problem. In these cases, a specialization of the Stone-Weierstrass theorem [92] can be readily used, resulting in a tensor product construction using one-dimensional orthogonal polynomials. For stochastically independent dimensions, these polynomials would be orthogonal with respect to the probability density function corresponding to their respective dimensions, giving rise to the so-called generalized polynomial chaos (gPC) [105]. While gPC were initially constructed to pair weights and polynomials from the Askey scheme, this limitation is unnecessary and can be easily relaxed using standard arguments from the theory of polynomial approximations [94]. Corrections to account for statistical dependence when only a finite number of random variables are retained for the analysis (finite stochastic dimension) have also been introduced [84]. It should be emphasized that these recent additions to the polynomial chaos literature do not extend the results of Wiener or the Cameron-Martin theorem as the main challenge in that earlier work was the discretization and limiting behavior of an infinite dimensional object, namely, the Brownian motion. It should also be noted that since this more recent work emphasizes finite stochastic dimensions, it has served to facilitate critical linkages between stochastic analysis, numerical analysis, statistics, and applications which almost certainly explains the magnitude of its continuing impact on the field of uncertainty quantification.

Several contributions to this Uncertainty Quantification Handbook report on current research related to various aspects of polynomial chaos in uncertainty quantification. The present article reviews a selection of recent developments related to estimating, propagating, and updating polynomial chaos representations in the context of model-based predictions.

2 Mathematical Setup

We will be concerned with the characterization of random variables and processes directly and their behavior under either deterministic or stochastic transformations. Information for these random entities often originates from data acquired through experimentally obtained measurements. Regardless of whether or not such data are available, a large part of the characterization process is the development of appropriate uncertainty models. In science and engineering applications, the transformations under consideration typically represent equilibrium and conservation laws as embodied by algebraic or differential equations with random coefficients and forcing or a combination of such equations. Some minimal mathematical structure is required for describing this setting in a manner that is conducive to the useful analysis of related problems.

Let (Ω, \mathcal{F}, P) denote the probability triple that defines the mathematical context for an experiment. Random variables are defined as measurable functions from Ω into a measurable space, which is itself a probability triple. The measure induced by the mapping in this new probability triple is known as the distribution of the random variable. Thus, if X denotes a random variable, $X : (\Omega, \mathcal{F}) \mapsto (\mathbb{G}, \mathcal{G}(X))$ where \mathbb{G} is a topological vector space and $\mathcal{G}(X)$ is that subset of the Borel σ -algebra \mathcal{G}

on \mathbb{G} induced by X , then the measure or distribution of X is denoted by $\mu_X(A)$. This distribution is such that, for every event $A \subset \mathcal{G}(X)$, its measure is given by $\mu_X(A) = P(X^{-1}(A))$. The requirement that X be measurable ensures that $X^{-1}(A) \in \mathcal{F}$. The induction process for μ_X using the above definition is often referred to as a “push forward” operation on P , here through the mapping X . The collection of all P -measurable random variables from (Ω, \mathcal{F}) to $(\mathbb{G}, \mathcal{G}(X))$ with finite expectation defines the space of \mathbb{G} -valued integrable functions. Although the space \mathbb{G} is most often identified with the real number line, \mathbb{R}^1 , other realizations of \mathbb{G} are useful, as, for example, in characterizing model errors where nonparametric methods based on random matrices have been used [21, 43, 83, 86]. Characterizing these random variables as L_1 maps is typically constrained by data, and measures of proximity between different representations are relegated to a comparison of probability measures.

In addition to random variables defined on the original probability triple, the action on these random variables of various operators describing the physics of the problem is also often of interest. Accordingly, denote by $L_2(\mathbb{G}, \mathbb{V})$ by the space of μ -square-integrable functions from topological vector space, \mathbb{G} , to another topological vector space, \mathbb{V} , where μ is a probability measure on \mathbb{G} . Here, both \mathbb{G} and \mathbb{V} are assumed to be equipped with their respective Borel σ -fields, \mathcal{G} and \mathcal{V} . We are specially interested in the case where the range, \mathbb{V} , of these random variables is a function space corresponding to the solution of a partial differential equation, or a finite dimensional Euclidean space, or a reproducing kernel Hilbert space. The probability measure on \mathbb{V} induced by an element of $L_2(\mathbb{G}, \mathbb{V})$ is therefore also of interest. We should note, however, that under sufficient smoothness conditions on the mapping, the measure on \mathbb{V} will generally be absolutely continuous [14] with respect to the measure μ on \mathbb{G} , permitting one to express the probability of all events on $(\mathbb{V}, \mathcal{V})$ in terms of the probability of corresponding events on $(\mathbb{G}, \mathcal{G})$ and hence on (Ω, \mathcal{F}) . The L_2 structure of the mappings $\mathbb{G} \mapsto \mathbb{V}$ is usually associated with their deterministic nature as they are induced primarily by conservation laws and so-called first principles.

Clearly, attributing model uncertainty to these laws and principles necessitates the co-mingling of L_1 and L_2 structure of function spaces. Some aspects of this coupling are described subsequently in this article.

3 Polynomial Chaos

We adopt a polynomial chaos expansion (PCE) [40] form for the uncertainty models as they provide the means to develop flexible representations of random variables that can be easily constrained either by observations or by governing equations. These representations serve also as highly efficient generators of random variables, permitting on-the-fly sampling from complex distributions and the real-time implementation of Approximate Bayesian Computation (ABC) methods [26, 95]. A polynomial chaos decomposition of a random variable, X , involves two stages.

In the first of these stages, the input set of system random variables, $X \in \mathbb{G}$, is described as function of an underlying set of basic random variables, $\xi \in \mathbb{R}^d$, which we refer to as the “germ” of the expansion [4, 40, 84]. The probability description of ξ is assumed to be given by a probability density function, ρ_ξ , and the functional dependence is denoted by $X = X(\xi)$. The second stage consists of developing this general functional dependence in a polynomial expansion with respect to the germ that is convergent in $L_2(\mathbb{R}^d, \mathbb{G}, \rho_\xi)$ and estimating the corresponding expansion coefficients. The structure of this L_2 space was detailed elsewhere [84].

The polynomial chaos expansion (PCE) of random variable X thus takes the form

$$X = \sum_{|\alpha| > 0} X_\alpha \psi_\alpha(\xi), \quad (14.1)$$

where $\alpha = (\alpha_1 \dots, \alpha_d)$, $|\alpha| = \sum_{i=1}^d \alpha_i$, and $\{\psi_\alpha\}$ are polynomials orthonormal with respect to the measure of ξ . The coefficients can thus be readily expressed as

$$X_\alpha = \int_{\mathbb{R}^d} X(\xi) \psi_\alpha(\xi) \rho_\xi d\xi, \quad (14.2)$$

where the mathematical expectation indicated by this last integral can be shown to be the scalar product in $L_2(\mathbb{R}^d, \mathbb{G}, \rho_\xi)$ between X and ψ_α and will be denoted as $\langle X \psi_\alpha \rangle_\xi$. While the case where ξ is a discretization of the Brownian motion would align this development with Wiener’s theory, the infinite dimensional problem is usually discretized and truncated in a preprocessing step as part of modeling and data assimilation. This preprocessing obviates the need for an infinite dimensional analysis allowing far simpler concepts from multivariate polynomial approximation to be utilized. The summation in the above expansion is typically truncated after some polynomial order p . To be consistent with current research on efficient PCE representations, however, we will replace such a truncation with a projection on a subspace spanned by an indexing set \mathcal{I} and thus replace Eq. (14.1) with the following equation:

$$X = \sum_{\alpha \in \mathcal{I}} X_\alpha \psi_\alpha(\xi). \quad (14.3)$$

It is understood, of course, that convergence is only achieved in the limit of infinite summation.

If X is only available through experimental observations, then a functional dependence of X on random variables ξ can be constructed to match the statistics of the observed data, through, for example, a Rosenblatt transform [73, 76] or a Cornish-Fisher expansion [20, 29, 93]. In this case, the coefficients X_α can be viewed as statistics of the data [23, 38] and their estimation as a task of statistical inference which can be accomplished using standard methods of statistical estimation such as Maximum Likelihood [25, 37], Maximum Entropy [22], or Bayesian procedures [8, 77]. We will elaborate further on this point later in this chapter.

If X is the solution of some governing equation with random parameters $\xi \in \mathbb{R}^d$, then its functional dependence on ξ is through the governing equation. Much of recent research on polynomial chaos methods has been to effectively express this dependence, expanding it in compressed and projected polynomial forms or compositions of such forms. In many instances, the combined mathematical structure associated with the governing equations and the polynomial chaos expansions permits a rigorous analysis of these approximations. Other sections in this chapter and several other contributions to this Handbook expand further on this point.

Regardless of whether the PCE is constrained by experimental data, by governing equations, or a composition of both operations, it embeds the variable X into a d -dimensional space defined by the random variables ξ , with d often termed the stochastic dimension of the approximation. In particular, the joint probability density function of ξ is critical for defining orthogonality and projections in that space. The PCE thus provides a parameterization of random variables that construes them as the output of a nonlinear filter representing either hidden unobservable dynamics or explicitly modeled physical processes.

Furthermore, the form of the PCE is crucial as it facilitates the use of the representation as generator for the random variable, as well as for ease of coupling across mathematical and computational models. In some situations, the underlying variables are stochastic processes, resulting in an infinite dimensional polynomial representation. The coefficients in this expansion are deterministic with a similar mathematical character to that of X , that is, they are scalar, vector, function, etc. The range of PCE representations includes random variables that possess probability measures with bounded support, are multimodal, or exhibit various degrees of skewness and peak sharpness; there is no need for the regularity for a probability density function to exist for them. In fact, the probability triples for the range spaces of these random variables include push forward induced distributions that can be shown to converge to a fixed distribution, μ_X , as the PCE expansion itself converges, using machinery of the function space L_2 . This is a key aspect of the PCE methodologies since it means that rigorous error analysis is possible, even for functionals on product spaces that include uncertainty subdomains. An additional consequence of this is that these approximations are representations that are known to also converge in probability.

We will assume that a system of possibly coupled operators, parameterized with random variables $X_i \in \mathbb{G}_i$, ($i = 1, \dots, r$), yields a random variable $U \in \mathbb{H}$ and that further processing of U yields $Q \in \mathbb{V}$, the final quantity of interest (QoI) to be used in decision or design settings. In several cases of interest, \mathbb{H} will refer to a function space associated with the solution of the coupled operator equations, while \mathbb{V} will be identified with \mathbb{R}^n or even more simply with \mathbb{R} .

In computational settings, each $X_i \in \mathbb{G}_i$ will be expressed in terms of a finite dimensional $\xi_i \in \mathbb{R}^{d_i}$. Thus if X_i is a stochastic process, ξ_i would denote its discretization, possibly through a Karhunen-Loëve expansion; or if X_i is a correlated random vector, then ξ_i could denote either its associated Karhunen-Loëve random variables or its map through the Rosenblatt transform or a similar measure transport map [22, 72]. While it is not imperative for the components

of ξ_i to be statistically independent [5–7, 84], they will be assumed to be so in the sequel in order to simplify notation. Carrying out this substitution of X_i in terms of ξ_i results in a system of governing equations that is parameterized with $\xi = (\xi_1, \dots, \xi_r) \in \mathbb{R}^d$ where $d = d_1 + \dots + d_r$. In general, then, we have the following situation:

$$\begin{aligned} h(U, X(\xi)) &= 0, & h : \mathbb{H} \times \mathbb{R}^d &\mapsto \mathbb{R}^m \\ Q = q(U(\xi)), & & q : \mathbb{R}^d &\mapsto \mathbb{R}^n. \end{aligned} \quad (14.4)$$

A common step in a UQ analysis procedure entails characterizing U and Q as PCEs. This essentially consists in evaluating the coefficients in the following two expansions:

$$U = \sum_{\alpha \in \mathcal{I}} U_\alpha \psi_\alpha \quad Q = \sum_{\alpha \in \mathcal{I}} Q_\alpha \psi_\alpha. \quad (14.5)$$

At first sight it would seem that the QoI, Q , should be expressed in terms of ξ . This would require the characterization of a nonlinear multivariate map, a task closely associated with the curse of dimensionality.

Capitalizing on the simplicity of Q relative to X_i and U has been shown to lead to adapted algorithms that significantly enhance the efficiency of associated computational models. This topic will be further elaborated in a separate section below.

Two algorithmic methodologies have been pursued for characterizing the PCE of U and Q . In a first of these, an approximation for U is substituted into the governing equations, and the resulting error is constrained to be orthogonal to the operator input space which yields the following equation:

$$\left\langle \psi_k h \left(\sum_{j \in \mathcal{I}} U_j \psi_j, X \right) \right\rangle_\xi = 0, \quad \forall k \in \mathcal{I}. \quad (14.6)$$

This last equation is a system of coupled nonlinear equations for the coefficients U_j in the PCE representation of U . Linearizing h with respect to U , which is typically done as part of nonlinear solution algorithms, results in the following where l denotes the linearized operator:

$$\sum_{j \in \mathcal{I}} \langle \psi_k \psi_j l(U_j, X) \rangle_\xi = 0, \quad \forall k \in \mathcal{I}. \quad (14.7)$$

In some instances, l may be linear in X , while in other instances it may be expressed in the form

$$l(U_j, X) = \sum_{i \in \mathcal{I}} l_i(U_j) \psi_i(\xi), \quad l_i(U_j) = \langle l(U_j, X), \psi_i \rangle_\xi. \quad (14.8)$$

In either case, Eq. (14.7) can be rewritten in the form

$$\sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} \langle \psi_i \psi_j \psi_k \rangle_{\xi} l_i(U_j) = 0, \quad \forall k \in \mathcal{I}, \quad (14.9)$$

which is a system of linear equations to be solved for the coefficients U_j . These equations can consist of a combination of algebraic, integral, and differential equations. For the case where the operator h belongs to a limited class of partial differential equations, the projection in Eq. (14.6) can be combined with the deterministic projection, such as a finite element projection, associated with discretizing the differential operator. Once a PCE decomposition for U has been computed, an expansion for Q can be achieved by applying the mapping q on the PCE of U .

The class of algorithm just described is typically referred to as the “intrusive approach,” involving the solution of a new system of equations obtained from a stochastic Galerkin projection on the operator input space. Accordingly, there is generally a significant code development that accompanies the development of these new system equations. Other characteristics of the approach are that the stochastic integration is determined exactly since the expectation of the triple products in Eq. (14.9) is available either analytically or numerically. This fact means that a significant portion of the approximation error in the stochastic result for the response originates in the formation of the linearized building blocks, l_i , in Eq. (14.9). Also, in Eq. (14.9), the global system matrices for the PCE coefficients in the case of discretized deterministic operators are compound; each coefficient vector, U_j , is the size of the discretized deterministic solution. The result is that the system matrices for the stochastic solution can be extraordinarily large. Finally, this approach is a generalized Fourier method and as for generalized Galerkin methods permits a priori and some a posteriori error analyses [9, 10]. A perspective on recent methods for solving the (very) large system of linear algebraic equations associated with Eq. (14.9) is presented subsequently in this article.

The second class of procedures for characterizing the PCE of Q is based on projecting in response space, that is, on Q and its PCE approximation. These procedures exploit orthogonality of the polynomials $\{\psi_{\alpha}\}$, yielding an expression for Q_{α} in the form

$$Q_{\alpha} = \langle Q \psi_{\alpha} \rangle_{\xi}, \quad \alpha \in \mathcal{I} \quad (14.10)$$

which can be approximated as the following quadrature:

$$Q_{\alpha} \approx \sum_{r=1}^{nq} Q(\xi^{(r)}) \psi_{\alpha}(\xi^{(r)}) w_q, \quad (14.11)$$

where $\{\xi^{(r)}\}$ are the nq quadrature points in \mathbb{R}^d and $\{w_q\}$ their associated weights. The number of quadrature points required in the above approximation depends

on the dimension of ξ and on the required level of fidelity. Using tensorized quadrature rules in d , dimensions quickly becomes prohibitive, and adapted rules have been developed [30, 38, 41, 106] and are elaborated in other contributions to this Handbook. The procedures described by Eq. (14.11) require only function evaluations from deterministic codes and are thus often referred to as “nonintrusive.” Further, since PCE expansions for parameters are not incorporated into the operator itself, they can generally be implemented more readily into existing deterministic analysis codes, where the error is carried in the expansion coefficients for the stochastic system response, which are projections based on expectation operators shown in Eq. (14.10). Regardless in the limit where all response calculations are equally accurate and by using the general L_2 theoretical structure, the solutions using either of the procedure classes will yield the same final stochastic answer. Other nonintrusive approaches have been developed that rely on interpolation and least squares arguments.

4 Representation of Stochastic Processes

Stochastic processes are significant in the context of uncertainty quantification as they are often crucial for describing stochastic inputs that vary in space or time, or both, as well as for characterizing spatially and temporally varying solutions of differential equations.

A number of mathematical conceptualizations can be pursued in describing stochastic processes that, while all consistent, are mathematically nuanced usually differing by their identification of sets of measure zero and can be adapted to nuanced requirements of various applications. Thus, the standard approach for describing a stochastic process, as a set of indexed random variables, emphasizes the topological and metric properties of the indexing set at the expense of the sample path properties of the process [91]. An alternative approach consists of describing stochastic processes by constructing probability measures on function spaces, thus implicitly describing several sample path properties [69]. The reproducing kernel Hilbert space (RKHS) associated with the process is a natural common by-product of either of these two constructions.

Clearly, a particular representation for stochastic processes is already embedded in the PCE by taking the random variable X to be an element in a function space. This is typical of situations where the stochastic process represents the solution of a stochastic differential equation where dependence of the process on the underlying variables ξ is inherited from the parameterization of the governing equations by these variables. In this case, the statistical properties of the stochastic process are completely determined by the initial parameterization of the governing equation.

In other situations, a stochastic process is inferred either from a statistical inverse problem or from experimental observations of functionals of the process, typically in the form local averages of the process. In this case, a covariance kernel of the process can be estimated, and the corresponding RKHS can be associated with the process. A by-product of this mathematical construction is

the Karhunen-Loëve expansion of the stochastic process in terms of the eigen-decomposition of its covariance kernel. The Karhunen-Loëve expansion provides a mean-square convergent representation of the stochastic process, permitting its characterization in terms of a denumerable set of jointly dependent uncorrelated random variables [40, 70, 84].

Thus, the covariance operator C_X of the random variable X can be defined as the bilinear map $C_X : \mathbb{G}' \mapsto \mathbb{G}$ defined implicitly by

$$(f, C_X g)_{\mathbb{G}', \mathbb{G}} = E \{(f, X)_{\mathbb{G}', \mathbb{G}} (g, X)_{\mathbb{G}', \mathbb{G}}\} = E \{f(X)g(X)\}, \quad f, g \in \mathbb{G}' \quad (14.12)$$

where $(., .)_{\mathbb{G}', \mathbb{G}}$ denotes duality pairing between \mathbb{G}' and \mathbb{G} and the second equality is shown for notational clarity. The bilinear form given by Eq. (14.12) defines a scalar product which transforms \mathbb{G}' into a Hilbert space \mathbb{H}_X , namely, the RKHS associated with X . This, in turn, determines the isometric map, $\eta = (f, X)_{\mathbb{G}', \mathbb{G}}$, from \mathbb{H}_X to Hilbert space \mathbb{H}_V such that $\|\eta\|_{\mathbb{H}_V} = \|f\|_{\mathbb{H}_X}$. Let $\{f_i\}$ denote an orthonormal basis in \mathbb{H}_X , then it can be shown [70] that the representation

$$X = \sum_{i=1}^{\infty} \eta_i X_i, \quad (14.13)$$

where $\eta_i = (f_i, X)_{\mathbb{G}', \mathbb{G}}$ and $X_i = C_X f_i$, is mean-square convergent in the weak topology of \mathbb{G} . In this expansion, the set $\{\eta_i\}$ is clearly orthonormal in \mathbb{H}_V . For the case in which the space \mathbb{G} is itself a Hilbert space, the duality pairing defines a scalar product in \mathbb{G} denoted by $(., .)$. The covariance operator is then a positive Hilbert-Schmidt operator admitting a convergent expansion in terms of its orthonormal eigenvectors which form a complete orthonormal system in \mathbb{G} . In that case, the expansion given by Eq. (14.13) can be rewritten in the form

$$X = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \eta_i e_i \quad (14.14)$$

where η_i has the same meaning as above and where

$$(v, C_X e_i) = \lambda_i (v, e_i), \quad \forall v \in \mathbb{G}. \quad (14.15)$$

Convergence is both pointwise and in mean square. Equation (14.14) is known as the Karhunen-Loëve expansion of random variable X . Clearly, the random variables η_i can be expressed in terms of the random variable X in the form

$$\eta_i = \frac{1}{\sqrt{\lambda_i}} (e_i, X), \quad (14.16)$$

thus providing an explicit transformation from realizations of X to realizations of η_i . These realizations can be used to estimate the joint density function of the

random vector $\boldsymbol{\eta}$ [22,23]. For the special case where the Hilbert space \mathbb{G} is identified with $L_2(T)$ for some subset T of a metric space, the eigenproblem specified in Eq.(14.15) is replaced by

$$\int_T k(t,s)e_i(s)ds = \lambda_i e_i(t), \quad t \in T \quad (14.17)$$

where $k(t,s) = E\{X(t)X(s)\}$ is the covariance kernel of X . In addition, for the case where $k(t,s) = R(t-s)$ for some symmetric function R , the rate of decay of the eigenvalues of Eq.(14.17) is related to the smoothness of the function R and the decay of its Fourier transform at infinity [71]. If domain T is also unbounded, then the point spectrum from Eq.(14.17) becomes continuous, and the Karhunen-Loève expansion is replaced by an integral representation in terms of the independent increments of a Brownian motion [42]. Recent developments have permitted the expression of the Karhunen-Loève random variables, $\{\eta_i\}$, as polynomials in independent random variables ξ [23,27]

$$\boldsymbol{\eta} = \sum_{\alpha} \boldsymbol{\gamma}_{\alpha} \psi_{\alpha}(\xi), \quad (14.18)$$

simultaneously permitting both their sampling and their integration with other simulation codes that rely on polynomial chaos representations for their input parameters.

A challenge with inferring a stochastic process from data is the very large number of constraints required on any probabilistic representation of the process. In the case of a Karhunen-Loève expansion, these constraints take the form of specifying the covariance operator and also specifying the joint probability density function of the random variables $\boldsymbol{\eta}$. Under the added assumption of a Gaussian process, $\boldsymbol{\eta}$ is a vector of independent standard Gaussian variables. Even in this overly simplified setting, any assumption concerning the form of the covariance function presumes the ability to observe the process over a continuum of scales within the domain T of the process, a feat that is likely to remain elusive. Alternative procedures for characterizing stochastic processes that are not predicated on knowledge of covariance functions are required for stochastic models to bridge the gap between predictions and reality. This point is addressed in a subsequent section in this chapter.

5 Polynomial Chaos with Random Coefficients: Model Error

As with any probabilistic representation, a PCE describes a set of knowledge that is necessarily incomplete. As the state of knowledge changes, for example, by acquiring additional experimental evidence and by revising the underlying physical assumptions or the associated mathematical and measurement instruments, the PCE representations of any quantity of interest (QoI) should be updated accordingly.

Given its structure, there are several manners in which the PCE can be viewed as a prior model, and the associated update method can take on at least one of the following three distinct forms. First, the germ can be updated, thus assimilating the new knowledge into the motive source of uncertainty. This can be achieved by either updating the measure of the same germ or by introducing new components into the germ, thus increasing the underlying stochastic dimension of the PCE. The first of these germ update methods maintains the same physical conceptualization of uncertainty, while modifying its measure. The second germ update approach introduces new observables and in the process necessitates more elaborate physical interpretations, typically through multiscale and multiphysics formulations.

A second prior update method is to update the numerical values of the coefficients in the PCE with the goal of minimizing some measure of discrepancy between predictions and observations. Even with such a deterministic update of the coefficients, predictions from the PCE remain probabilistic in view of their dependence on the stochastic germ. This second update approach can be achieved through a Bayesian MAP, a Maximum Likelihood estimation, or even a least squares setting.

A third alternative update method consists of maintaining the probability measure of the germ at its prior value while updating a probabilistic model of the coefficients. This last option presupposes that a prior model of the PCE coefficients can be constructed.

Each of these three update alternatives is associated with a distinctly different perspective on interpreting and managing uncertainty.

A unique feature of a PCE that differentiates it among other probabilistic representations is that both parameters, X , the solution to governing equations, U , and the QoI, Q , are all represented with respect to the same germ ξ . This gives rise to interesting additional options for updating prior PCE models. Specifically, the representation of X can be updated and subsequently propagated through the model to obtain a correspondingly updated representation of Q . Alternatively, the representation of Q can itself be directly updated with the implicit understanding that the coefficients in the PCE of Q already encapsulate a numerical resolution of the prior governing equations.

Equation (14.3) is thus replaced by the following equation:

$$X = \sum_{\alpha \in \mathcal{I}} X_\alpha(\mu) \psi_\alpha(\xi), \quad (14.19)$$

where the new random variables $\mu \in \mathbb{R}^{n_v}$ describe the uncertainty about the probabilistic model. Typically, each X_α depends on a proper subset of μ . The QoI can thus be expressed as $Q(\xi, \mu)$ to underscore its dependence on the (d, n_v) -dimensional random vectors (ξ, μ) . Clearly, the uncertainty captured by the random variables, μ , can be reduced by assimilating more data into the estimation problem, and increasing the size of ξ suggests more subscale details. In the limit, the asymptotic distributions of X_α and their posterior probability densities become concentrated. The uncertainty reflected by the ξ variables, on the other hand, is a property of the selected uncertainty model form and characterizes the limits on

predictability of the model. Thus, while the underlying mathematical uncertainty model of the physical processes is only parameterized by ξ , the quantity of interest Q can be represented in the form

$$Q = q(\xi, \mu) = \sum_{\alpha \in \mathcal{I}} Q_\alpha(\mu) \psi_\alpha(\xi), \quad (14.20)$$

where $\{\psi_\alpha\}$ are polynomials orthogonal with respect to the measure of ξ . Dependence of the coefficients Q_α on μ reflects their sensitivity to modeling errors [8, 37, 38, 85, 96] and can be inferred as part of their statistical estimation. Once the probability measure of Q_α has been constructed, dependence of Q on μ can in turn be expressed through a Rosenblatt transform and then in its own polynomial chaos decomposition in terms of polynomials orthogonal with respect to the joint measure of μ . This results in the following expression:

$$Q(\xi, \mu) = \sum_{\alpha \in \mathcal{I}} \sum_{\beta \in \mathcal{I}} Q_{\alpha, \beta} \phi_\beta(\mu) \psi_\alpha(\xi). \quad (14.21)$$

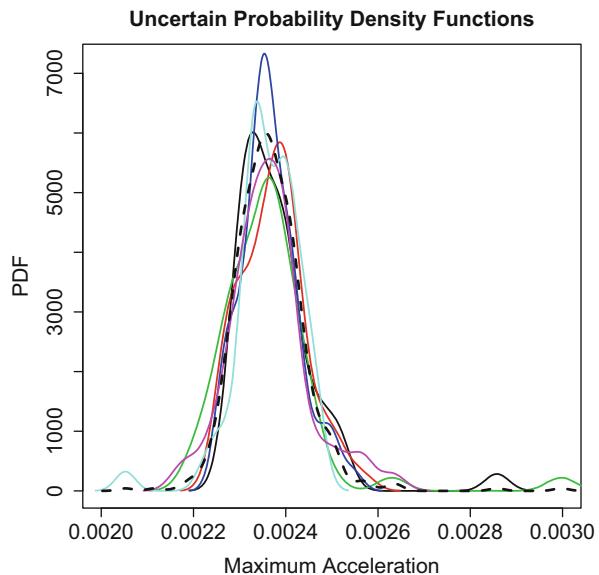
Equations (14.20) and (14.21) provide two different, yet consistent, representations. Given the orthogonality of the polynomials appearing in these representations, the coefficients Q_α can be obtained via projection and quadrature in the form

$$Q_\alpha(\mu) = \sum_{r=1}^{nq} w_r q(\xi^{(r)}) \psi_\alpha(\xi^{(r)}), \quad (14.22)$$

where w_r are quadrature weights, while $\xi^{(r)}$ are multidimensional abscissas: realizations of the random vectors ξ associated with the integration rules used. The quantity $q(\xi^{(r)})$ appearing in Eq. (14.22) is obtained as the solution of a deterministic operator evaluated at realizations $\xi^{(r)}$.

Figure 14.1 shows results associated with an implementation of the foregoing analysis [38]. In that implementation, properties of a mechanical device are estimated from a limited number of samples, each observed at a finite number of spatial locations, and represented mathematically using a Karhunen-Loëve representation. The material itself consists of foam, which exhibits an intricate microstructure resulting in an inherent uncertainty (labeled by ξ). Additional uncertainty due to limited sample size and coverage induces uncertainty in the covariance kernel of the material properties which result in a stochastic covariance function and associated eigenvalues and eigenvectors. This uncertainty, labeled by μ , is then propagated into the behavior of the mechanical device which is subjected to a mechanical shock. The Maximum Acceleration within the device, during the shock event, is then a stochastic process, indexed by both ξ and μ . For each realization of μ , the Maximum Acceleration is a random variable with a probability measure that can be readily inferred from its PCE representation. Figure 14.1 shows a family of such density functions as μ is itself sampled from its own distribution function. It is

Fig. 14.1 Scatter in probability density function associated with randomness in coefficients



clear from this figure that inferences about Maximum Acceleration, in particular near the tail of its distribution, are quite sensitive to additional samples of material properties.

6 Adapted Representations of PCE

A PCE of total degree p in d random variables is an expansion with P terms where

$$P = \frac{(d + p)!}{d! p!}. \quad (14.23)$$

This factorial growth in the number of terms has motivated the pursuit of compact representations which typically select a subset of terms in the expansions. Another recent approach, reviewed in this section, capitalizes on properties of Gaussian germs to developed representations that are uniquely adapted to specific quantities of interest. This approach is based on the observation that irrespective of how large the stochastic dimension is, the quantity of interest q is typically very low dimensional, spanning a manifold of significantly lower intrinsic dimension than the ambient Euclidean space. Recently, algorithms were developed for identifying two such manifolds, namely, a one-dimensional subspace and a sequence of one-dimensional subspaces [98]. Identifying these manifolds involves computing an isometry, A , in Gaussian space that rotates the Gaussian germ so as to minimize the concentration of the measure of the QoI away from the manifold. For problems involving non-Gaussian germ, a mapping back to Gaussian variables is typically

implemented as a preprocessing step. This step is both highly efficient and accurate. According to this basis adaptation approach, Eq. (14.5) is first rewritten as follows:

$$Q = Q_0 + \sum_{i=1}^d Q_i \xi_i + \sum_{|\alpha|>1} Q_\alpha \psi_\alpha(\xi). \quad (14.24)$$

Under conditions of a Gaussian germ ξ , the first summation in this last equation is a Gaussian random variable. The idea then is to rotate the germ ξ into a new germ η where the first coordinate, η_1 , is aligned with $\sum_{i=1}^d Q_i \xi_i$. To that end, let η be such a transformed germ expressed as

$$\eta = A\xi, \quad (14.25)$$

where A is an isometry in \mathbb{R}^d such that $\eta_1 = \sum_{i=1}^d Q_i \xi_i$. The other rows of A can be completed through a Gram-Schmidt orthogonalization procedure which may be supplemented by additional constraints. The same PCE for Q can then be expressed either in terms of ξ or in terms of η leading to the following two expressions:

$$\begin{aligned} Q &= Q_0 + \sum_{i=1}^d Q_i \xi_i + \sum_{|\alpha|>1} Q_\alpha \psi_\alpha(\xi) \\ &= Q_0 + Q_1^A \eta_1 + \sum_{|\alpha|>1} Q_\alpha^A \psi_\alpha^A(\xi) \end{aligned} \quad (14.26)$$

where $\psi_\alpha^A(\xi) = \psi_\alpha(A\xi)$. Equation (14.26) can be rewritten as follows:

$$Q = Q_0 + Q_1^A \eta_1 + \sum_{i=2}^p Q_i^A \psi_i(\eta_1) + \sum_{|\alpha|>1} Q_\alpha^A \psi_\alpha(\eta), \quad (14.27)$$

where the second summation involves polynomials in the random variables (η_2, \dots, η_d) , while the terms prior to that reflect a p order one-dimensional expansion in η_1 . A one-dimensional approximation of Q is then obtained by neglecting the last summation. Thus knowledge of the Gaussian (i.e., linear) components of a PCE representation permits the construction of a rotation of the germ such that a significant portion of the QoI probabilistic content is concentrated along a single coordinate. One justification for the possibility of a one-dimensional construction, specially when Q is a scalar quantity, is as follows. Let $F_Q(q)$ denote the distribution function of Q and $\Phi(\eta)$ denote the one-dimensional standard Gaussian distribution function. One then has

$$Q = F_Q^{-1}(\Phi(\eta)), \quad (14.28)$$

where the equality is in distribution and the equation provides a map from a Gaussian variate to a Q -variate. According to the Skorokhod representation theorem [14], a version of η can be defined in the same probability triple as Q . Thus η appearing in the inverse CDF expression above can be defined in the linear span of the germ ξ . Other constructions of the isometry, A , with alternative probabilistic arguments and justifications have also been presented [98].

The construction of the adapted bases requires an initial investment of resources for the evaluation of the “Gaussian” components of the QoI Q . These components are used to construct the isometry, A , and thus the dominant direction η_1 that is adapted to Q . Following that, a high-order expansion in η_1 is sought and can be evaluated using either intrusive or nonintrusive procedures.

An extension to the case where Q is a stochastic process (i.e., infinite dimensional) has also been implemented [99].

7 Stochastic Galerkin Implementation of PCE

A salient feature of the mathematical setup underlying PCE is the L_2 setting induced by the germ. This permits L_2 -convergent representations of random variables in contradistinction to representations of their statistical moments or distributions. A by-product of this L_2 structure is the ability to characterize the convergence of solutions to stochastic operator equations in an appropriate stochastic operator norm. For bounded linear operators, this results in an immediate extension of Céa’s lemma and a corresponding interpretation of the Lax-Milgram theorem [9]. This is typically implemented according to the so-called “intrusive” approach described in Eqs. (14.6), (14.7), (14.8), and (14.9).

These intrusive implementations face two challenges, both of which were touched upon briefly earlier in this chapter. First, the intrusive nature of the algorithms presents an implementation challenge as it necessitates the development of new software or significant revisions to existing code. Second, the size of these linear problems is much larger than the size of the corresponding deterministic problems, thus requiring careful attention to algorithmic implementations and the development of adapted preconditioners. These two challenges are briefly discussed in this section.

7.1 Nonintrusive Evaluation of the Stochastic Galerkin Solution

Equation (14.9) can be rewritten as

$$\sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} \langle \psi_i \psi_j \psi_k \rangle_\xi \mathbf{L}_i \mathbf{U}_j = \mathbf{f}_k, \quad \forall k \in \mathcal{I}, \quad (14.29)$$

where the deterministic linear operators $\{l_i\}$ have been projected on a suitable finite dimensional space and are represented by the associated matrices $\{\mathbf{L}_i\}$ with \mathbf{U}_j and \mathbf{f}_k denoting, respectively, the corresponding projections of U_j and of boundary conditions. We further distinguish a special case where Eq. (14.8) can be written as

$$l(\mathbf{U}_j, \mathbf{X}) = \sum_{i=0}^d l_i(\mathbf{U}_j) \xi_i, \quad l_i(\mathbf{U}_j) = \langle l(\mathbf{U}_j, \mathbf{X}), \xi_i \rangle_{\xi}, \quad (14.30)$$

for which case Eq. (14.29) becomes

$$\sum_{j \in \mathcal{I}} \sum_{i=0}^d \langle \xi_i \psi_j \psi_k \rangle_{\xi} \mathbf{L}_i \mathbf{U}_j = \mathbf{f}_k, \quad k \in \mathcal{I}. \quad (14.31)$$

Equations (14.29) and (14.31) can be rewritten as

$$\sum_{j \in \mathcal{I}} \mathbf{L}_{jk} \mathbf{U}_j = \mathbf{f}_k, \quad k \in \mathcal{I}, \quad \mathbf{L}_{jk} = \sum_i c_{ijk} \mathbf{L}_i \quad (14.32)$$

where c_{ijk} is equal to either $\langle \xi_i \psi_j \psi_k \rangle_{\xi}$ or $\langle \psi_i \psi_j \psi_k \rangle_{\xi}$ and the summation over i extends either over \mathcal{I} or over $(0, \dots, d)$, depending on whether Eq. (14.29) or (14.31) is being represented. It is clear from this last equation that matrix-vector (MATVEC) multiplications involving \mathbf{L}_{jk} can be affected using MATVEC operations on \mathbf{L}_i , thus alleviating the need to store the block matrices \mathbf{L}_{jk} , $j, k \in \mathcal{I}$ [67]. Equation (14.29) can be rearranged to take the following form [34]:

$$\mathbf{L}_0 \mathbf{U}_k = \mathbf{f}_k - \sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}^+} c_{ijk} \mathbf{L}_i \mathbf{U}_j, \quad k \in \mathcal{I} \quad (14.33)$$

where \mathcal{I}^+ denotes the set \mathcal{I} without 0. This last equation serves two purposes. First, it describes one of the most robust preconditioners for the system given by Eq. (14.29), namely, a preconditioning via the inverse of the mean operator \mathbf{L}_0 [67]. Second, it provides a path for evaluating the solution of the intrusive approach via a nonintrusive algorithm. Specifically, as long as an analysis code exists for solving the mean operator, Eq. (14.33) describes how to evaluate the polynomial chaos coefficients \mathbf{U}_k by updating the right-hand side of the equations and iterating until convergence is achieved. Block-diagonal preconditioning has also been proposed [67, 68] as well as preconditioning with truncated PCE solutions [88, 89].

7.2 Adapted Preconditioners for the Stochastic Galerkin Equations

Given the desirable mathematical properties of a stochastic Galerkin solution, the associated matrices have recently received significant attention [28, 61, 74]. A key role in their block structure is played by the constants c_{ijk} . In general, there are two types of block patterns associated, respectively, with Eqs. (14.31) and (14.29). The first type is typically regarded as block sparse with a typical block structure depicted in Fig. 14.2a. It is noted that, for general forms of the set \mathcal{I} , the block sparse matrix has additional structure, as seen in the figure, whereby larger blocks around its diagonal are themselves block diagonal. This is a consequence of the recursion formula for orthogonal polynomials. The second type, block dense, corresponds to $c_{ijk} = \langle \psi_i \psi_j \psi_k \rangle$, and its block structure is depicted in Fig. 14.2b.

The structure of the global stochastic Galerkin matrix is induced by the matrix c_P with entries $c_{jk}^P = \sum_{i \in \mathcal{I}} c_{ijk}$ where $j, k \in \mathcal{I}$. To fix the presentation, let us consider some ℓ -th order polynomial expansion, such that $1 \leq \ell \leq p$. Thus the set \mathcal{I} consists of all d -dimensional multi-indices such that $|\ell| \leq p$. It is easy to see that the corresponding coefficient matrix c_ℓ will have a hierarchical structure

$$c_\ell = \begin{bmatrix} c_{\ell-1} & b_\ell^T \\ b_\ell & d_\ell \end{bmatrix}, \quad \ell = 1, \dots, p, \quad (14.34)$$

where $c_{\ell-1}$ are the first principal submatrices corresponding to the $(\ell - 1)$ -th order polynomial expansion. We note that even though the matrices c_ℓ are symmetric, the global stochastic Galerkin matrix will be symmetric only if each one of the matrices

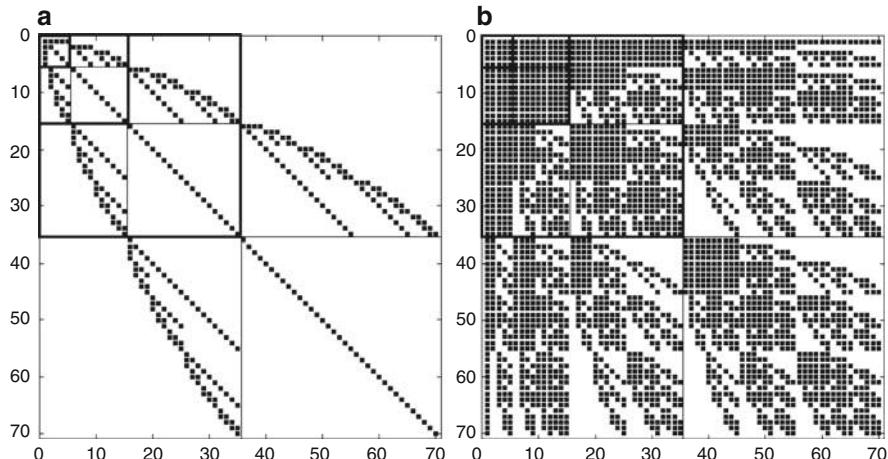


Fig. 14.2 Typical sparsity patterns of the coefficient matrices for different c_{ijk} . (a) $c_{ijk} = \langle \xi_i \psi_j \psi_k \rangle$. (b) $c_{ijk} = \langle \psi_i \psi_j \psi_k \rangle$

\mathbf{L}_i is symmetric. Clearly, all matrices \mathbf{L}_i will have the same sparsity pattern. In either case, the block sparse or the block dense, the linear system (14.29) can be written as

$$\mathbf{A}_P \mathbf{U}_P = \mathbf{f}_P, \quad (14.35)$$

where the global Galerkin matrix \mathbf{A}_P has the hierarchy specified as

$$\mathbf{A}_\ell = \begin{bmatrix} \mathbf{A}_{\ell-1} & \mathbf{B}_\ell \\ \mathbf{C}_\ell & \mathbf{D}_\ell \end{bmatrix}, \quad \ell = P, \dots, 1, \quad (14.36)$$

and $\mathbf{A}_0 = \mathbf{L}_0$. Note that in general, $\mathbf{C}_\ell = \mathbf{B}_\ell^T$, for $\ell = 1, \dots, P$. It is clear from Fig. 14.2a that the decomposition (14.36) provides a path for a hierarchical Schur complement solution for the block sparse case [90]. In the block dense case, this approach still provides an effective preconditioner [89, 90]. The number of linear solves required by these preconditioners is equal to the number of terms included in the PCE (P in Eq. (14.23)). Additional computational effort is required for MATVEC operations required for evaluating the Schur complements and for iterating in the block dense case.

It is important to note that algorithms adapted to the peculiar structure of the global stochastic Galerkin matrices continue to improve the efficiency of these methods.

8 Embedded Quadratures for Stochastic Coupled Physics

As indicated previously, several nonintrusive approaches for evaluating the PCE representations to the solution of stochastic equations are challenged with the need to evaluate high-dimensional integrals. Many problems of recent interest involve the numerical solution of large-scale coupled problems, such as multistage, multiphysics, and multiscale, which at first sight seem to significantly exacerbate this difficulty. On closer inspection, however, several of these problems are equipped with structure, often inherited from the physics and system-level requirements, whereby the bulk of the uncertainty remains localized within each subproblem, thus reducing the dimension of the stochastic information linking the subproblems and consequently alleviating the overall computational burden. This idea was recently explored and observed to yield orders of magnitude reduction in computational prerequisites [5–7].

A first step is to imagine the collection of subsystems as forming a graph, with each node i having parent nodes P_i . Denoting the output from the i th model by $u^{(i)}$, we typically have the following structure (with overloading of notation):

$$u^{(i)}(\mathbf{X}) = u^{(i)}(\mathbf{X}_i, u^{P_i}), \quad \mathbf{X} \in \mathbb{R}^d, \quad \mathbf{X}_i \in \mathbb{R}^{d_i} \quad (14.37)$$

where u^{P_i} refers to the set of stochastic outputs over P_i . This notation highlights the fact that significant smoothing takes place as uncertainty propagates through each model component in the overall system model. It is thus clear that rather than carrying out the high-dimensional integrals over d -space, a subspace which contains the fluctuations of X_i together with those of u^{P_i} should be sufficient. It is a challenge to efficiently characterize the probability measure on this space and to compute the associated quadrature rules. Instead, we will rely on an embedded quadrature approach [6]. A first step in this approach is to reduce the functions u^{P_i} , whenever they are described as random fields, via their Karhunen-Loève decomposition. Let the number of terms retained for these functions be equal to t_i . Clearly, these terms will generally be dependent, and hence the current approach, which is described next, is conservative as they are presumed independent. Let \mathcal{Q}_d denote the coordinates of a quadrature rule in d -space. We then identify a quadrature rule $\mathcal{Q}_{d_i+t_i}$ in the space of dimension $d_i + t_i$ that is a subset of \mathcal{Q}_d with an L_1 constraint on the weights. We demonstrate this idea for a two-model problem below.

Consider two physical processes, u and v , governed by the following functional relationships:

$$f(\kappa, \mu(v), u) = 0, \quad g(\chi, v(u), v) = 0 \quad (14.38)$$

where κ and χ denote random parameters implicit to the first and second physics relationships, respectively. We will assume that these have been discretized into random vectors using, for instance, a Karhunen-Loève expansion. Also, $\mu(v)$ and $v(u)$ are additional parameters that describe the coupling between the two physical processes. A complete solution of the problem would require an analysis over a stochastic space that is large enough to characterize κ and χ simultaneously. We will approach this challenge in two steps. First, we reduce the stochastic character of each of u and v through decorrelation, by describing their respective dominant Karhunen-Loève expansions. Our goal is to express u , v , μ , and ν . To begin, we take

$$u = \sum_{\alpha=0}^{K_u} u_{\alpha} \hat{\xi}_{\alpha}, \quad v = \sum_{\alpha=0}^{K_v} v_{\alpha} \hat{\xi}_{\alpha}, \quad (14.39)$$

where $\hat{\xi}$ and $\hat{\zeta}$ are the Karhunen-Loève variables which, for now, are assumed to be independent. Note that each of these variables can be described independently using an inverse CDF mapping, as a function of a Gaussian variable. This results in the following representations:

$$\begin{aligned} u &= \sum_{\alpha=0}^{L_u} u_{\alpha} E_{\alpha}(\xi), & v &= \sum_{\alpha=0}^{L_v} v_{\alpha} E_{\alpha}(\zeta) \\ v(u) &= \sum_{\alpha=0}^{L_{\mu}} m_{\alpha} E_{\alpha}(\xi), & \mu(v) &= \sum_{\alpha=0}^{L_{\nu}} n_{\alpha} E_{\alpha}(\zeta) \end{aligned} \quad (14.40)$$

where $\xi \in \mathbb{R}^{K_u}$ and $\zeta \in \mathbb{R}^{K_v}$ are independent Gaussian random vectors, E_α are normalized multidimensional Hermite polynomials, and (L_v, L_μ) are such that all summations are converged within specified tolerance. The solutions to Eq. (14.38) can thus be written as

$$\begin{aligned} u(\kappa, \chi) &\equiv u(\kappa, \xi) = \sum_{\beta=0}^{L_v} \sum_{\alpha=0}^K u_{\alpha\beta} \psi_\alpha(\kappa) E_\beta(\xi) \\ v(\kappa, \chi) &\equiv v(\xi, \chi) = \sum_{\beta=0}^{L_\mu} \sum_{\alpha=0}^K v_{\alpha\beta} \Phi_\alpha(\chi) E_\beta(\xi) \end{aligned} \quad (14.41)$$

where ψ_α and Φ_β are multidimensional polynomials orthonormal with respect to the probability density of κ and χ , respectively, and the upper limit on the summation, K , is sufficient for the representation of dependence of u and v and κ and χ . In these representations, the deterministic coefficients for each solution, $u_{\alpha\beta}$ and $v_{\alpha\beta}$, depend on the solution of the other problem. Given the orthogonality of all the above polynomials, the coefficients $u_{\alpha\beta}$ and $v_{\alpha\beta}$ can be obtained as follows:

$$u_{\alpha\beta}(v) = \mathbb{E}\{u \psi_\alpha E_\beta\}, \quad v_{\alpha\beta}(u) = \mathbb{E}\{v \Phi_\alpha E_\beta\}, \quad (14.42)$$

which are evaluated as multidimensional integrals. We address the important challenge of evaluating these integrals efficiently by taking advantage of their composite structure. We elaborate on that point next.

Consider a function $f(\kappa_1, \dots, \kappa_p)$ of p parameters where κ_i is a μ_i -square-integrable \mathbb{R}^{m_i} -valued random variable with density relative to Lebesgue measure given by μ_i , and let the joint density of all the κ_i parameters be denoted by μ_κ . Thus, $\kappa_i : (\Omega, \mathcal{F}, P) \mapsto \mathbb{R}^{m_i}$ and $f : \mathbb{R}^{m_1} \times \dots \times \mathbb{R}^{m_p} \mapsto \mathbb{R}^m$. We are interested in evaluating the mathematical expectation, $E\{f\}$, of the function f , where in general f can be decomposed as

$$f(\kappa_1, \dots, \kappa_p) = g(\kappa_1, \dots, \kappa_p) h(\kappa_1, \dots, \kappa_p). \quad (14.43)$$

If the function h is an indicator function for a set A , then $E\{f\}$ evaluates the probability of g being in A . We are also interested in situations where the function, h , is an orthogonal polynomial in the random variables, κ_i , in which case $E\{f\}$ evaluates to the polynomial chaos coefficient in the expansion of g in a basis consisting of these polynomials. In some instances, we will also be interested in situations where the random variables κ are themselves functions of another set of random variables $\xi \in \mathbb{R}^N$. Thus, in general, we will consider

$$\begin{aligned} I = \mathbb{E}\{f\} &= \int_{\mathbb{R}^m} f(\kappa) \mu_\kappa(\kappa) d\kappa = \int_{\mathbb{R}^N} f(\kappa(\xi)) \mu_\xi(\xi) d\xi \approx \sum_{q=1}^{nq} w_q f(\xi^q), \\ \xi^q &\in \mathbb{R}^N, f \in \mathbb{V}, \end{aligned} \quad (14.44)$$

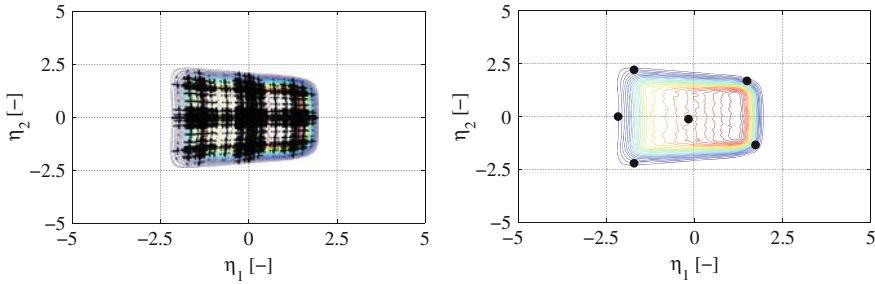


Fig. 14.3 Quadrature points in original space and embedded quadratures in reduced space

where \mathbb{V} denotes a functional space specified by the physics of the problem, μ_ξ is the probability density function of the N -dimensional random variable ξ , and (w_q, ξ_q) are weight/coordinate pairs of some particular quadrature rule. The integral I can thus be evaluated either over \mathbb{R}^m with respect to the measure of κ or over \mathbb{R}^N with respect to the measure of ξ . In most cases of interest, quadrature with respect to ξ , while easier to compute (since the ξ are usually statistically independent), is in a much higher dimension. On the other hand, quadrature with respect to κ , while over a much lower dimension, is with respect to a dependent measure and does not conform to standard quadrature rules. Recent algorithms [6] select quadrature points for κ -integration as a very small subset from the points required for the ξ -integration. The selection is based on an optimality requirement for the L_1 norm of the weights of the selected subset. Figure 14.3a, b show a comparison between the required quadrature points using full tensor products and the method that was just described. The results in these figures pertain to an application involving stationary transport of neutrons with thermal coupling [5]. The figures show the projection onto a two-dimensional plane of all quadrature points used for evaluating the PCE coefficients of the temperature field, to within similar accuracy. The number of required quadrature points is reduced from over a 1000 to 6.

9 Constructing Stochastic Processes

As already mentioned above when discussing Karhunen-Loëve expansions, the infinite dimensional nature of stochastic processes gives rise to modeling and computational challenges. For instance, when selecting the form for covariance functions of stochastic processes, the behavior of the processes at very small and very large length scales is implicitly decided by the selection. The statistical measure and dependence between these scales must be determined from additional observations of the process. In many instances, the complexity inherited from the infinite dimensional construction of the stochastic process belies a regularity that permits its assimilation into design and decision processes. One rational

approach to simplify the construction of stochastic processes without giving up mathematical rigor can be gleaned from multiscale modeling. For example, emerging properties on any scale can often be deduced from fluctuations on a finer scale. Thus, if we are in a position to characterize fluctuations on the finest scale in terms of a collection of experimentally observed random variables, emerging properties on coarser scales will be possible to deduce through an upscaling procedure. Since any result is dependent on the specific upscaling procedure adopted, it is not unique. However, that result does have the important benefit of having produced a stochastic process for derived properties that are commensurate with that particular procedure. An additional benefit for such a construction is the ability to deduce several coarse-scale processes as dependent stochastic processes [97] related through their dependence on the same germ.

As an example, we consider the problem of two-dimensional flow past circular thermal inclusions [97]. Uncertainty in the system is considered by modeling the thermal conductivity of the discs in terms of ten random variables. Fluid flow and heat transfer at the fine scale are described with the Navier-Stokes equations augmented by conservation of energy and utilized temperature-dependent fluid viscosity and fluid density. At the coarse scale, a Darcy-Forchheimer model of a porous medium is constructed, and the spatial variability of its hydraulic and thermal properties are deduced from the fine scale through homogenization.

The stochastic fine-scale problem, with ten-dimensional stochastic input, was solved using quadrature rules as implemented in Albany [66]. This results in a polynomial chaos characterization of the temperature and flow fields in the fine scale.

The coarse-scale permeability tensor, k_{ij} , was computed using volume averaging [64]. Thus, at low velocities, Darcy's law can be used to compute the permeability as

$$-\frac{\partial \langle P \rangle_V}{\partial x_i} = \mu k_{ij}^{-1} \langle u_j \rangle_V \quad (14.45)$$

where $\langle P \rangle_V$ and $\langle u \rangle$ are volume averaged pressure and velocity obtained from fine-scale solution. The volume average is computed as

$$\langle a \rangle = \frac{1}{V} \int_V a dV. \quad (14.46)$$

In this manner, a PCE representation is constructed of the coarse-scale permeability and conductivity tensors in terms of the germ describing the fine-scale properties. This construction can serve to generate realizations of the process as well as to investigate the correlation structure induced by fine-scale variability and prevailing physical processes. Furthermore, since both coarse-scale properties are described in terms of the same germ, their statistical dependence is built into their models.

10 Conclusion

A number of analogies have been drawn between polynomial chaos representations and a number of other procedures including response surface models, surrogate models, reduced models, mean-square approximations, hierarchical models, white noise models, stochastic differential equations, and infinite dimensional analysis. The standing of PCE methods at such a diverse intersection underscores its versatility and its coherent mathematical structure.

The basis of polynomial chaos formalism is the direct approximation, and coincidentally the parametrization, of stochastic variables and processes. This formalism provides a particular probabilistic packaging of evidence and constraints that is well adapted to computational and algorithmic requirements of evolving large-scale computational problems. The underlying mathematical framework enables a seamless extension of numerical analysis from deterministic to stochastic problems, while also providing a general enough characterization to encompass general multivariate stochastic processes. The historical roots of PCE methods in infinite dimensional probabilistic analysis provide them with sufficient mathematical structure and flexibility to characterize a very wide range of uncertainties and to serve as proper representations for prior and posterior knowledge as postulated by probabilistic updating rules. Finally, their foundation in functional analysis equips them naturally for innovation at the interface between physics modeling, computational science, and statistical science.

This flexibility in modeling and representation comes at a cost, namely, the scientific or engineering problem is embedded into a Euclidean space that is prone to the so-called curse of dimensionality. In spite of a number of recent methods for tackling this challenge, for instance, in the guise of basis adaptation along both spatial and stochastic coordinates, significant challenges remain. An important issue to be addressed by PCE approaches, as well as other UQ representations and methodologies, is that of sufficient complexity. Specifically, ascertaining the sensitivity of ultimate decisions and designs to increases in mathematical complexity is a nascent idea, which must be addressed with regard to, simultaneously, the complexity of physics models, probabilistic and statistical models, and numerical discretization and solvers.

References

1. Adomian, G.: Stochastic Green's functions. In: Bellman, R. (ed.) *Proceedings of Symposia in Applied Mathematics. Volume 16: Stochastic Processes in Mathematical Physics and Engineering*. American Mathematical Society, Providence (1964)
2. Adomian, G.: *Stochastic Systems*. Academic, New York (1983)
3. Albeverio, S., Daletsky, Y., Kondratiev, Y., Streit, L.: Non-Gaussian infinite dimensional analysis. *J. Funct. Anal.* **138**, 311–350 (1996)
4. Arnst, M., Ghanem, R.: Probabilistic equivalence and stochastic model reduction in multiscale analysis. *Comput. Methods Appl. Mech. Eng.* **197**(43–44), 3584–3592 (2008)

5. Arnst, M., Ghanem, R., Phipps, E., Red-Horse, J.: Dimension reduction in stochastic modeling of coupled problems. *Int. J. Numer. Methods Eng.* **92**, 940–968 (2012)
6. Arnst, M., Ghanem, R., Phipps, E., Red-Horse, J.: Measure transformation and efficient quadrature in reduced-dimensional stochastic modeling of coupled problems. *Int. J. Numer. Methods Eng.* **92**, 1044–1080 (2012)
7. Arnst, M., Ghanem, R., Phipps, E., Red-Horse, J.: Reduced chaos expansions with random coefficients in reduced-dimensional stochastic modeling of coupled problems. *Int. J. Numer. Methods Eng.* **97**(5), 352–376 (2014)
8. Arnst, M., Ghanem, R., Soize, C.: Identification of Bayesian posteriors for coefficients of chaos expansions. *J. Comput. Phys.* **229**(9), 3134–3154 (2010)
9. Babuška, I., Tempone, R., Zouraris, G.E.: Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.* **42**(2), 800–825 (2005)
10. Babuška, I., Tempone, R., Zouraris, G.E.: Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Eng.* **194**(12–16), 1251–1294 (2005)
11. Benaroya, H., Rehak, M.: Finite element methods in probabilistic structural analysis: a selective review. *Appl. Mech. Rev.* **41**(5), 201–213 (1988)
12. Berezansky, Y.M.: Infinite-dimensional non-Gaussian analysis and generalized translation operators. *Funct. Anal. Appl.* **30**(4), 269–272 (1996)
13. Bharucha-Reid, A.T.: On random operator equations in Banach space. *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astr. Phys.* **7**, 561–564 (1959)
14. Billingsley, P.: Probability and Measure. Wiley Interscience, New York (1995)
15. Bose, A.G.: A theory of nonlinear systems. Technical report 309, Research Laboratory of Electronics, MIT (1956)
16. Boyce, E.W., Goodwin, B.E.: Random transverse vibration of elastic beams. *SIAM J.* **12**(3), 613–629 (1964)
17. Brilliant, M.B.: Theory of the analysis of nonlinear systems. Technical report 345, Research Laboratory of Electronics, MIT (1958)
18. Cameron, R.H., Martin, W.T.: The orthogonal development of nonlinear functions in a series of Fourier-Hermite functionals. *Ann. Math.* **48**, 385–392 (1947)
19. Chorin, A.: Hermite expansions in Monte-Carlo computation. *J. Comput. Phys.* **8**, 472–482 (1971)
20. Cornish, E., Fisher, R.: Moments and cumulants in the specification of distributions. *Rev. Int. Stat. Inst.* **5**(4), 307–320 (1938)
21. Das, S., Ghanem, R.: A bounded random matrix approach for stochastic upscaling. *SIAM J. Multiscale Model. Simul.* **8**(1), 296–325 (2009)
22. Das, S., Ghanem, R., Finette, S.: Polynomial chaos representation of spatio-temporal random fields from experimental measurements. *J. Comput. Phys.* **228**(23), 8726–8751 (2009)
23. Das, S., Ghanem, R., Spall, J.: Sampling distribution for polynomial chaos representation of data: a maximum-entropy and fisher information approach. *SIAM J. Sci. Comput.* **30**(5), 2207–2234 (2008)
24. Debusschere, B., Najm, H., Matta, A., Knio, O., Ghanem, R., Le Maître, O.: Protein labeling reactions in electrochemical microchannel flow: numerical simulation and uncertainty propagation. *Phys. Fluids* **15**(8), 2238–2250 (2003)
25. Descelliers, C., Ghanem, R., Soize, C.: Maximum likelihood estimation of stochastic chaos representation from experimental data. *Int. J. Numer. Methods Eng.* **66**(6), 978–1001 (2006)
26. Diggle, P., Gratton, R.: Monte Carlo methods of inference for implicit statistical models. *J. R. Stat. Soc. Ser. B* **46**, 193–227 (1984)
27. Doostan, A., Ghanem, R., Red-Horse, J.: Stochastic model reduction for chaos representations. *Comput. Methods Appl. Mech. Eng.* **196**, 3951–3966 (2007)
28. Ernst, O.G., Ullmann, E.: Stochastic Galerkin matrices. *SIAM J. Matrix Anal. Appl.* **31**(4), 1848–1872 (2010)
29. Fisher, R., Cornish, E.: The percentile points of distributions having known cumulants. *Technometrics* **2**(2), 209–225 (1960)

30. Ganapathysubramanian, B., Zabaras, N.: Sparse grid collocation methods for stochastic natural convection problems. *J. Comput. Phys.* **225**, 652–685 (2007)
31. George, D.A.: Continuous nonlinear systems. Technical report 355, Research Laboratory of Electronics, MIT (1959)
32. Ghanem, R.: Hybrid stochastic finite elements: coupling of spectral expansions with Monte Carlo simulations. *ASME J. Appl. Mech.* **65**, 1004–1009 (1998)
33. Ghanem, R.: Scales of fluctuation and the propagation of uncertainty in random porous media. *Water Resour. Res.* **34**(9), 2123–2136 (1998)
34. Ghanem, R., Abas, J.: A general purpose library for stochastic finite element computations. In: Bathe, J. (ed.) Second MIT Conference on Computational Mechanics, Cambridge (2003)
35. Ghanem, R., Brzakala, V.: Stochastic finite element analysis for randomly layered media. *ASCE J. Eng. Mech.* **122**(4), 361–369 (1996)
36. Ghanem, R., Dham, S.: Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transp. Porous Media* **32**, 239–262 (1998)
37. Ghanem, R., Doostan, A., Red-Horse, J.: A probabilistic construction of model validation. *Comput. Methods Appl. Mech. Eng.* **197**, 2585–2595 (2008)
38. Ghanem, R., Red-Horse, J., Benjamin, A., Doostan, A., Yu, A.: Stochastic process model for material properties under incomplete information (AIAA 2007–1968). In: 48th AIAA/ASME/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Honolulu, 23–26 Apr 2007. AIAA (2007)
39. Ghanem, R., Sarkar, A.: Reduced models for the medium-frequency dynamics of stochastic systems. *JASA* **113**(2), 834–846 (2003)
40. Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach. Springer, New York (1991). Revised edition by Dover Publications, (2003)
41. Ghiocel, D., Ghanem, R.: Stochastic finite element analysis of seismic soil-structure interaction. *J. Eng. Mech.* **128**(1), 66–77 (2002)
42. Gikhman, I., Skorohod, A.: The Theory of Stochastic Processes I. Springer, Berlin (1974)
43. Guilleminot, J., Soize, C., Ghanem, R.: Stochastic representation for anisotropic permeability tensor random fields. *Int. J. Numer. Anal. Methods Geomech.* **36**, 1592–1608 (2012)
44. Hart, G.C., Collins, J.D.: The treatment of randomness in finite element modelling. In: SAE Shock and Vibrations Symposium, Los Angeles, pp. 2509–2519 (1970)
45. Hasselman, T.K., Hart, G.C.: Modal analysis of random structural systems. *ASCE J. Eng. Mech.* **98**(EM3), 561–579 (1972)
46. Hida, T.: White noise analysis and nonlinear filtering problems. *Appl. Math. Optim.* **2**, 82–89 (1975)
47. Hida, T., Kuo, H.-H., Potthoff, J., Streit, L.: White Noise: An Infinite Dimensional Calculus. Kluwer Academic Publishers, Dordrecht/Boston (1993)
48. Imamura, T., Meecham, W.: Wiener-Hermite expansion in model turbulence in the late decay stage. *J. Math. Phys.* **6**(5), 707–721 (1965)
49. Itô, K.: Multiple Wiener integrals. *J. Math. Soc. Jpn.* **3**(1), 157–169 (1951)
50. Itô, K.: Spectral type of shift transformations of differential process with stationary increments. *Trans. Am. Math. Soc.* **81**, 253–263 (1956)
51. Jahedi, A., Ahmadi, G.: Application of Wiener-Hermite expansion to nonstationary random vibration of a Duffing oscillator. *ASME J. Appl. Mech.* **50**, 436–442 (1983)
52. Kallianpur, G.: Stochastic Filtering Theory. Springer, New York (1980)
53. Klein, S., Yasui, S.: Nonlinear systems analysis with non-Gaussian white stimuli: General basis functionals and kernels. *IEEE Tran. Inf. Theory* **IT-25**(4), 495–500 (1979)
54. Kondratiev, Y., Da Silva, J., Streit, L., Us, G.: Analysis on Poisson and Gamma spaces. *Infinite Dimens. Anal. Quantum Probab. Relat. Top.* **1**(1), 91–117 (1998)
55. Lévy, P.: Leçons d'Analyses Fonctionnelles. Gauthier-Villars, Paris (1922)
56. Li, R., Ghanem, R.: Adaptive polynomial chaos simulation applied to statistics of extremes in nonlinear random vibration. *Probab. Eng. Mech.* **13**(2), 125–136 (1998)
57. Liu, W.K., Besterfield, G., Mani, A.: Probabilistic finite element methods in nonlinear structural dynamics. *Comput. Methods Appl. Mech. Eng.* **57**, 61–81 (1986)

58. Lytvynov, E.: Multiple Wiener integrals and non-Gaussian white noise: a Jacobi field approach. *Methods Funct. Anal. Topol.* **1**(1), 61–85 (1995)
59. Le Maître, O., Najm, H., Ghanem, R., Knio, O.: Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Phys.* **197**(2), 502–531 (2004)
60. Le Maître, O., Reagan, M., Najm, H., Ghanem, R., Knio, O.: A stochastic projection method for fluid flow. II: random process. *J. Comput. Phys.* **181**, 9–44 (2002)
61. Matthies, H.G., Keese, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**(12–16), 1295–1331 (2005). Special Issue on Computational Methods in Stochastic Mechanics and Reliability Analysis
62. Meidani, H., Ghanem, R.: Uncertainty quantification for Markov chain models. *Chaos* **22**(4) (2012)
63. Nakagiri, S., Hisada, T.: Stochastic finite element method applied to structural analysis with uncertain parameters. In: Proceeding of the International Conference on FEM, pp. 206–211 (1982)
64. Nakayama, A., Kuwahara, F., Umemoto, T., Hayashi, T.: Heat and fluid flow within an anisotropic porous medium. *Trans. ASME* **124**, 746–753 (2012)
65. Ogura, H.: Orthogonal functionals of the Poisson process. *IEEE Trans. Inf. Theory* **IT-18**(4), 473–481 (1972)
66. Pawłowski, R., Phipps, R., Salinger, A., Owen, S., Ciebert, C., Stalen, A.: Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part II: application to partial differential equations. *Sci. Program.* **20**(3), 327–345 (2012)
67. Pellissetti, M.F., Ghanem, R.G.: Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Adv. Eng. Softw.* **31**(8–9), 607–616 (2000)
68. Powell, C.E., Elman, H.C.: Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA J. Numer. Anal.* **29**(2), 350–375 (2009)
69. Pugachev, V., Sinitsyn, I.: *Stochastic Systems: Theory and Applications*. World Scientific, River Edge (2001)
70. Red-Horse, J., Ghanem, R.: Elements of a functional analytic approach to probability. *Int. J. Numer. Methods Eng.* **80**(6–7), 689–716 (2009)
71. Reichel, L., Trefethen, L.: Eigenvalues and pseudo-eigenvalues of toeplitz matrices. *Linear Algebra Appl.* **162**, 153–185 (1992)
72. Rosenblatt, M.: Remarks on a multivariate transformation. *Ann. Math. Stat.* **23**, 470–472 (1952)
73. Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. *Ann. Math. Stat.* **27**, 832–837 (1956)
74. Rosseel, E., Vandewalle, S.: Iterative solvers for the stochastic finite element method. *SIAM J. Sci. Comput.* **32**(1), 372–397 (2010)
75. Rugh, W.J.: *Nonlinear System Theory: The Volterra-Wiener Approach*. Johns Hopkins University Press, Baltimore (1981)
76. Sakamoto, S., Ghanem, R.: Simulation of multi-dimensional non-Gaussian non-stationary random fields. *Probab. Eng. Mech.* **17**(2), 167–176 (2002)
77. Sargsyan, K., Najm, H., Ghanem, R.: On the statistical calibration of physical models. *Int. J. Chem. Kinet.* **47**(4), 246–276 (2015)
78. Schoutens, W.: *Stochastic Processes and Orthogonal Polynomials*. Springer, New York (2000)
79. Segall, A., Kailath, T.: Orthogonal functionals of independent-increment processes. *IEEE Trans. Inf. Theory* **IT-22**(3), 287–298 (1976)
80. Shinozuka, M., Astill, J.: Random eigenvalue problem in structural mechanics. *AIAA J.* **10**(4), 456–462 (1972)
81. Skorohod, A.V.: *Random linear operators*. Reidel publishing company, Dordrecht (1984)
82. Sobczyk, K.: *Wave Propagation in Random Media*. Elsevier, Amsterdam (1985)
83. Soize, C.: A nonparametric model of random uncertainties for reduced matrix models in structural dynamics. *Probab. Eng. Mech.* **15**(3), 277–294 (2000)
84. Soize, C., Ghanem, R.: Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM J. Sci. Comput.* **26**(2), 395–410 (2004)

85. Soize, C., Ghanem, R.: Reduced chaos decomposition with random coefficients of vector-valued random variables and random fields. *Comput. Methods Appl. Mech. Eng.* **198**(21–26), 1926–1934 (2009)
86. Soize, C., Ghanem, R.: Data-driven probability concentration and sampling on manifold. *J. Comput. Phys.* **321**, 242–258 (2016)
87. Soong, T.T., Bogdanoff, J.L.: On the natural frequencies of a disordered linear chain of n degrees of freedom. *Int. J. Mech. Sci.* **5**, 237–265 (1963)
88. Sousedik, B., Elman, H.: Stochastic Galerkin methods for the steady-state Navier-Stokes equations. *J. Comput. Phys.* **316**, 435–452 (2016)
89. Sousedik, B., Ghanem, R.: Truncated hierarchical preconditioning for the stochastic Galerkin FEM. *Int. J. Uncertain. Quantif.* **4**(4), 333–348 (2014)
90. Sousedik, B., Ghanem, R., Phipps, E.: Hierarchical schur complement preconditioner for the stochastic Galerkin finite element methods. *Numer. Linear Algebra Appl.* **21**(1), 136–151 (2014)
91. Steinwart, I., Scovel, C.: Mercer’s theorem on general domains: on the interaction between measures, kernels, and RKHSs. *Constr. Approx.* **35**, 363–417 (2012)
92. Stone, M.: The genralized Weierstrass approximation theorem. *Math. Mag.* **21**(4), 167–184 (1948)
93. Takemura, A., Takeuchi, K.: Some results on univariate and multivariate Cornish-Fisher expansion: algebraic properties and validity. *Sankhyā* **50**, 111–136 (1988)
94. Tan, W., Guttman, I.: On the construction of multi-dimensional orthogonal polynomials. *Metron* **34**, 37–54 (1976)
95. Tavaré, S., Balding, D., Griffiths, R., Donnelly, P.: Inferring coalescence times from dna sequence data. *Genetics* **145**, 505–518 (1997)
96. Thammisetty, C., Khodabakhshnejad, A., Jabbari, N., Aminzadeh, F., Ghanem, R., Rose, K., Disenhofer, C., Bauer, J.: Multiscale stochastic representation in high-dimensional data using Gaussian processes with implicit diffusion metrics. In: Ravela, S., Sandu, A. (eds.) *Dynamic Data-Driven Environmental Systems Science*. Lecture Notes in Computer Science, vol. 8964. Springer (2015). doi:10.1007/978-3-319-25138-7_15
97. Tipireddy, R.: Stochastic Galerkin projections: solvers, basis adaptation and multiscale modeling and reduction. PhD thesis, University of Southern California (2013)
98. Tipireddy, R., Ghanem, R.: Basis adaptation in homogeneous chaos spaces. *J. Comput. Phys.* **259**, 304–317 (2014)
99. Tsilifis, P., Ghanem, R.: Reduced Wiener chaos representation of random fields via basis adaptation and projection. *J. Comput. Phys.* (2016, submitted)
100. Volterra, V.: *Theory of Functionals and of Integral and Integro-Differential Equations*. Blackie & Son, Ltd., Glasgow (1930)
101. Wan, X., Karniadakis, G.: Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.* **28**(3), 901–928 (2006)
102. Wiener, N.: Differential space. *J. Math. Phys.* **2**, 131–174 (1923)
103. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**(4), 897–936 (1938)
104. Wintner, A., Wiener, N.: The discrete chaos. *Am. J. Math.* **65**, 279–298 (1943)
105. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**, 619–644 (2002)
106. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
107. Yamazaki, F., Shinozuka, M., Dasgupta, G.: Neumann expansion for stochastic finite-element analysis. *ASCE J. Eng. Mech.* **114**(8), 1335–1354 (1988)

Part III
Forward Problems

Ilias Bilionis and Nicholas Zabaras

Abstract

Classic non-intrusive uncertainty propagation techniques, typically, require a significant number of model evaluations in order to yield convergent statistics. In practice, however, the computational complexity of the underlying computer codes limits significantly the number of observations that one can actually make. In such situations the estimates produced by classic approaches cannot be trusted since the limited number of observations induces additional epistemic uncertainty. The goal of this chapter is to highlight how the Bayesian formalism can quantify this epistemic uncertainty and provide robust predictive intervals for the statistics of interest with as few simulations as one has available. It is shown how the Bayesian formalism can be materialized by employing the concept of a Gaussian process (GP). In addition, several practical aspects that depend on the nature of the underlying response surface, such as the treatment of spatiotemporal variation, and multi-output responses are discussed. The practicality of the approach is demonstrated by propagating uncertainty through a dynamical system and an elliptic partial differential equation.

Keywords

Epistemic uncertainty • Expensive computer code • Expensive computer simulations Gaussian process • Uncertainty propagation

Contents

1	Introduction	556
2	Methodology	557

I. Bilionis (✉)

School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA
e-mail: ibilion@purdue.edu; <http://www.predictivesciencelab.org>

N. Zabaras

Warwick Centre for Predictive Modelling, University of Warwick, Coventry, UK
e-mail: nzabaras@gmail.com; <http://www.zabaras.com>

2.1	Physical Models	557
2.2	Computer Emulators	558
2.3	The Uncertainty Propagation Problem	559
2.4	The Bayesian Approach to Uncertainty Propagation	560
2.5	Gaussian Process Regression	561
2.6	Training the Parameters of the Gaussian Process	568
2.7	Multi-output Gaussian Processes	570
2.8	Sampling Possible Surrogates	572
2.9	Semi-analytic Formulas for the Mean and the Variance	578
3	Numerical Examples	580
3.1	Synthetic One-Dimensional Example	580
3.2	Dynamical System Example	582
3.3	Partial Differential Equation Example	583
4	Conclusions	593
	References	596

1 Introduction

National laboratories, research groups, and corporate R&D departments have spent decades in time and billions in dollars to develop realistic multi-scale/multi-physics computer codes for a wide range of engineered systems such as aircraft engines, nuclear reactors, and automobile vehicles. The driving force behind the development of these models has been their potential use for designing systems with desirable properties. However, this potential has been hindered by the inherent presence of uncertainty attributed to the lack of knowledge about initial/boundary conditions, material properties, geometric features, the form of the models, as well as the noise present in the experimental data used in model calibration.

This chapter focuses on the task of propagating parametric uncertainty through a given computer code ignoring the uncertainty introduced by discretizing an ideal mathematical model. This task is known as the *uncertainty propagation* (UP) problem. Even though the discussion is limited to the UP problem, all the ideas presented can be extended to the problems of model calibration and design optimization under uncertainty, albeit this is beyond the scope of this chapter.

The simplest approach to the solution of the UP problem is the Monte Carlo (MC) approach. When using MC, one simply samples the parameters, evaluates the model, and records the response. By post-processing the recorded responses, it is possible to quantify any statistic of interest. However, obtaining convergent statistics via MC for computationally expensive realistic models is not feasible, since one may be able to run only a handful of simulations.

The most common way of dealing with expensive models is to replace them with inexpensive surrogates. That is, one evaluates the model on a set of design points and then tries to develop an accurate representation of the response surface based on what he observes. The most popular such approach is to expand the response in a generalized polynomial chaos basis (gPC) [4, 51–53] and approximate its coefficients with a Galerkin projection computed with a quadrature rule, e.g., sparse grids [44]. In relatively low-dimensional parametric settings, these techniques

outperform MC by orders of magnitude. In addition, it is possible to prove rigorous convergence results for gPC. However, the quality of the estimates when using a very limited number of simulations is questionable. The main reason is that they do not attempt to quantify the epistemic uncertainty induced by this very fact.

The quantification of the epistemic uncertainty induced by the limited number of simulations requires statistical methodologies and, specifically, a Bayesian approach. The first statistical approach to computer code surrogate building was put forward by Currin et al. [16] and Sacks et al. [42], both using Gaussian processes (GP). This work was put in a Bayesian framework by Currin et al. [17] and Welch et al. [50]. The first fully Bayesian treatment of the UP problem has its roots in the Bayes-Hermite quadrature of O'Hagan [36], and it was put on the modern context in O'Hagan et al. [38] and Oakley and O'Hagan [34]. Of great relevance is the Gaussian emulation machine for sensitivity analysis (GEM-SAM) software of O'Hagan and Kennedy [37]. The work of the authors in [7, 10] constitutes a continuation along this path. The present chapter is a comprehensive review of the Bayesian approach to UP as of now.

The outline of the chapter is as follows. It starts with a generic description of physical models and the computer emulators used for their evaluation, followed by the definition of the UP problem. Then, it discusses the Bayesian approach to UP introducing the concept of a Bayesian surrogate and showing how the epistemic uncertainty induced by limited observations can be represented. Then, it shows how the Bayesian framework can be materialized using GPs, by providing practical guidelines for the treatment of spatiotemporal multi-output responses, training the hyper-parameters of the model, and quantifying epistemic uncertainty due to limited data by sampling candidate surrogates. The chapter ends with three demonstrative examples, a synthetic one-dimensional example that clarifies some of the introduced concepts, a dynamical system with uncertain initial conditions, and a stochastic partial differential equation.

2 Methodology

2.1 Physical Models

A *physical model* is mathematically equivalent to a multi-output function,

$$\mathbf{f}: \mathcal{X} = \mathcal{X}_s \times \mathcal{X}_t \times \mathcal{X}_{\xi} \rightarrow \mathcal{Y}, \quad (15.1)$$

where $\mathcal{X}_s \subset \mathbb{R}^{d_s}$, with $d_s = 0, 1, 2$, or 3 , is the *spatial* domain; $\mathcal{X}_t \subset \mathbb{R}^{d_t}$, with $d_t = 0$, or 1 , is the *time* domain; $\mathcal{X}_{\xi} \subset \mathbb{R}^{d_{\xi}}$ is the *parameter* domain; and $\mathcal{Y} \subset \mathbb{R}^{d_y}$ is the *output* space. Note that under this notation, $d_s = 0$, or $d_t = 1$, is interpreted as if $\mathbf{f}(\cdot)$ has no spatial or time components, respectively. One thinks of $\mathbf{f}(\mathbf{x}_s, t, \boldsymbol{\xi})$ as the model response at the spatial location $\mathbf{x}_s \in \mathcal{X}_s$ at time $t \in \mathcal{X}_t$ when the parameters $\boldsymbol{\xi} \in \mathcal{X}_{\xi}$ are used. The parameters, $\boldsymbol{\xi}$, should specify everything that is

required in order to provide a complete description of the system. This covers any physical parameters, boundary conditions, external forcings, etc.

2.1.1 Example: Dynamical Systems

Let $\mathbf{f}(t, \xi) = \mathbf{z}(t; \mathbf{z}_0(\xi_1), \xi_2)$ be the solution of the initial value problem

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{h}(\mathbf{z}, \xi_2), \\ \mathbf{z}(0) &= \mathbf{z}_0(\xi_1),\end{aligned}\tag{15.2}$$

with $\xi = (\xi_1, \xi_2)$, and $\mathbf{h} : \mathbb{R}^q \times \mathcal{X}_\xi \rightarrow \mathbb{R}^q$. Here ξ_1 and ξ_2 are any parameters that affect the dynamics and the initial conditions, respectively. One has $n_s = 0$, $n_t = 1$, and $d_y = q$.

2.1.2 Example: Partial Differential Equation

In flow through porous media applications, ξ parametrizes the permeability and the porosity fields, e.g., via a Karhunen-Loeve Expansion (KLE). Here, $d_s = 3$, $d_t = 1$, and $d_y = 4$, with

$$\mathbf{f}(\mathbf{x}_s, t, \xi) = (p(\mathbf{x}_s, t, \xi), v_1(\mathbf{x}_s, t, \xi), v_2(\mathbf{x}_s, t, \xi), v_3(\mathbf{x}_s, t, \xi)),\tag{15.3}$$

where $p(\mathbf{x}_s, t, \xi)$ and $v_i(\mathbf{x}_s, t, \xi)$, $i = 1, 2, 3$ is the pressure and the i -th component of the velocity field of the fluid, respectively.

2.2 Computer Emulators

A *computer emulator*, $\mathbf{f}_c(\cdot)$, of a physical model, $\mathbf{f}(\cdot)$, is a function that reports the physical model at a given set of spatial locations,

$$\mathbf{X}_s = \{\mathbf{x}_{s,1}, \dots, \mathbf{x}_{s,n_s}\},\tag{15.4}$$

and at specific times,

$$\mathbf{X}_t = \{t_1, \dots, t_{n_t}\},\tag{15.5}$$

for any realization of the parameters, ξ . That is,

$$\mathbf{f}_c : \mathcal{X}_\xi \rightarrow \mathcal{Y}^{n_s n_t} \subset \mathbb{R}^{n_s n_t d_y},\tag{15.6}$$

with

$$\mathbf{f}_c(\xi) = (\mathbf{f}(\mathbf{x}_{s,1}, t_1, \xi)^T \dots \mathbf{f}(\mathbf{x}_{s,1}, t_{n_t}, \xi)^T \dots \mathbf{f}(\mathbf{x}_{s,n_s}, t_1, \xi)^T \dots \mathbf{f}(\mathbf{x}_{s,n_s}, t_{n_t}, \xi)^T)^T.\tag{15.7}$$

For example, for the case of dynamical systems, \mathbf{X}_t may be the time steps on which the numerical integration routine reports the solution. In the porous flow example \mathbf{X}_s may be the centers of the cells of a finite volume scheme and \mathbf{X}_t the time steps on which the numerical integration reports the solution.

Alternatively, the computer code represents the physical model in an internal basis, e.g., spectral elements. In this case,

$$\mathbf{f}(\mathbf{x}_s, t, \xi) \approx \sum_{i=1}^{n_c} \mathbf{c}_i(t, \xi) \psi_i(\mathbf{x}_s), \quad (15.8)$$

where $\psi_i(\mathbf{x}_s)$ are known spatial basis functions. Then, one may think of the computer code as the function that returns the coefficients $\mathbf{c}_i(t, \xi) \in \mathbb{R}^{d_y}$, i.e.,

$$\mathbf{f}_c : \mathcal{X}_\xi \rightarrow \mathbb{R}^{n_c n_t d_y} \quad (15.9)$$

$$\mathbf{f}_c(\xi) = (\mathbf{c}_1(t_1, \xi)^T \dots \mathbf{c}_1(t_{n_t}, \xi)^T \dots \mathbf{c}_{n_c}(t_1, \xi)^T \dots \mathbf{c}_{n_c}(t_n, \xi)^T)^T. \quad (15.10)$$

2.3 The Uncertainty Propagation Problem

In all problems of physical relevance, the parameters, ξ , are not known explicitly. Without loss of generality, uncertainty about ξ may be represented by assigning to it a probability density, $p(\xi)$. Accordingly, ξ is referred to as the *stochastic* input. The goal of uncertainty propagation is to study the effects of $p(\xi)$ on the model output $\mathbf{f}(\mathbf{x}_s, t, \xi)$. Usually, it is sufficient to be able to compute low-order statistics such as the *mean*,

$$\mathbb{E}_\xi[\mathbf{f}(\cdot)](\mathbf{x}_s, t) = \int \mathbf{f}(\mathbf{x}_s, t, \xi) p(\xi) d\xi, \quad (15.11)$$

the *covariance* matrix function:

$$\begin{aligned} \mathbb{C}_\xi[\mathbf{f}(\cdot)]((\mathbf{x}_s, t); (\mathbf{x}'_s, t')) &= \mathbb{E}_\xi \left[\{\mathbf{f}(\mathbf{x}_s, t, \xi) - \mathbb{E}_\xi[\mathbf{f}(\cdot)](\mathbf{x}_s, t)\} \right. \\ &\quad \left. \{\mathbf{f}(\mathbf{x}'_s, t', \xi) - \mathbb{E}_\xi[\mathbf{f}(\cdot)](\mathbf{x}'_s, t')\}^T \right], \end{aligned} \quad (15.12)$$

the *variance* of component i as a function of space and time:

$$\mathbb{V}_{\xi,i}[\mathbf{f}(\cdot)](\mathbf{x}_s, t) = \mathbb{C}_{\xi,ii}[\mathbf{f}(\cdot)]((\mathbf{x}_s, t); (\mathbf{x}'_s, t')), \quad (15.13)$$

for $i = 1, \dots, d_y$, low-dimensional full statistics, e.g., point-wise probability densities of one component of the response.

The focus of this chapter is restricted to *non-intrusive* uncertainty propagation methods. These techniques estimate the statistics of the physical model using the computer code $\mathbf{f}_c(\cdot)$ as a black box. In particular, a fully Bayesian approach using Gaussian processes is developed.

2.4 The Bayesian Approach to Uncertainty Propagation

Assume that one has made n simulations and collected the following data set:

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n, \quad (15.14)$$

where $\mathbf{x}_i \in \mathcal{X}$, i.e., $\mathbf{x}_i = (\mathbf{x}_{s,i}, \xi_i, t_i)$, and $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$. The problem is to estimate the statistics of the response, \mathbf{y} , using only the simulations in \mathcal{D} .

Classic approaches to uncertainty propagation use \mathcal{D} to build a surrogate surface $\hat{\mathbf{f}}(\mathbf{x})$ of the original model $\mathbf{f}(\mathbf{x})$. Then, they characterize the uncertainty on the response \mathbf{y} by propagating the uncertainty of the stochastic inputs, ξ , through the surrogate. In some cases, e.g., gPC [26], this can be done analytically. In general, since the surrogate surface is cheap to evaluate, the uncertainty of the stochastic inputs, ξ , is propagated through via a simple Monte Carlo procedure [31, 41]. Such a procedure can provide point estimates of any statistic. However, what can one say about the accuracy of these estimates? This question becomes important especially when the number of simulations, n , is very small. The Bayesian approach to uncertainty propagation can address this issue, by providing confidence intervals for the estimated statistics.

2.4.1 Bayesian Surrogates

The Bayesian approach is based on the idea of a *Bayesian surrogate*. A Bayesian surrogate is a probability measure on the space of surrogates which is compatible with one's prior beliefs about the nature of $\mathbf{f}(\mathbf{x})$ as well as the data \mathcal{D} . A precise mathematical meaning of these concepts is given in the Gaussian process section. For the moment – and without loss of generality – assume that one has a parameterized family of surrogates, $\hat{\mathbf{f}}(\cdot; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a finite dimensional random variable, with PDF $p(\boldsymbol{\theta})$. Intuitively, think of $\hat{\mathbf{f}}(\cdot; \boldsymbol{\theta})$ as a candidate surrogate with parameters $\boldsymbol{\theta}$ and that, before observing any data, $\boldsymbol{\theta}$ may take any value compatible with the *prior* probability $p(\boldsymbol{\theta})$. In addition, let $p(\mathcal{D}|\boldsymbol{\theta})$ be the *likelihood* of the simulations under the model. Using Bayes rule, one may characterize his state of knowledge about $\boldsymbol{\theta}$ via the *posterior* PDF:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}, \quad (15.15)$$

where the normalization constant,

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}, \quad (15.16)$$

is known as the *evidence*. The posterior of $\boldsymbol{\theta}$, Equation (15.15), neatly encodes everything one has learned about the true response, $\mathbf{f}(\cdot)$, after seeing the simulations in \mathcal{D} . How can one use this information to characterize his state of knowledge about the statistics of $\mathbf{f}(\cdot)$?

2.4.2 Predictive Distribution of Statistics

A *statistic* is an operator $\mathcal{Q}[\cdot]$ that acts on the response surface. Examples of statistics are the mean, $\mathbb{E}_{\xi}[\cdot]$, of Equation (15.11); the covariance, $\mathbb{C}_{\xi}[\cdot]$, of Equation (15.12); and the variance, $\mathbb{V}_{\xi}[\cdot]$, of Equation (15.13). Using the posterior of $\boldsymbol{\theta}$ (see Equation (15.15)), the state of knowledge about an arbitrary statistic $\mathcal{Q}[\cdot]$ is characterized via

$$p(\mathcal{Q}|\mathcal{D}) = \int \delta \left(\mathcal{Q} - \mathcal{Q}[\hat{\mathbf{f}}(\cdot; \boldsymbol{\theta})] \right) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}. \quad (15.17)$$

Equation (15.17) contains everything that is known about the value of $\mathcal{Q}[\cdot]$, given the observations in \mathcal{D} . The quantity $p(\mathcal{Q}|\mathcal{D})$ is known as the *predictive distribution* of the statistic $\mathcal{Q}[\cdot]$ given \mathcal{D} . The uncertainty in $p(\mathcal{Q}|\mathcal{D})$ corresponds to the *epistemic uncertainty* induced by one's limited-data budget. The Bayesian approach is the only one that can naturally characterize this epistemic uncertainty.

Equation (15.17) applies generically to any Bayesian surrogate and to any statistic $\mathcal{Q}[\cdot]$. For the case of a Gaussian process surrogate, it is possible to develop semi-analytic approximations to Equation (15.17) when $\mathcal{Q}[\cdot]$ is the mean or the variance of the response. However, in general one has to think of Equation (15.17) in a generative manner in the sense that it enables sampling of possible statistics in a two-step procedure:

1. Sample a $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\mathcal{D})$ of Equation (15.15).
2. Evaluate $\mathcal{Q}[\hat{\mathbf{f}}(\cdot; \boldsymbol{\theta})]$.

2.5 Gaussian Process Regression

For simplicity, the section starts by developing the theory for one-dimensional outputs, i.e., $n_y = 1$.

2.5.1 Modeling Prior Knowledge About the Response

An experienced scientist or engineer has some knowledge about the response function, $f(\cdot)$, even before running any simulations. For example, he might know that $f(\cdot)$ cannot exceed, or be smaller than, certain values, that it satisfies translation invariance, that it is periodic along certain inputs, etc. This knowledge is known as *prior knowledge*.

Prior knowledge can be *precise*, e.g., the response is exactly twice differentiable, the period of the first input is 2π , etc., or it can be *vague*, e.g., the probability that the period T takes any particular value is $p(T)$, the probability that the length-scale ℓ_1 of the first input takes any particular value is $p(\ell_1)$, etc. When one is dealing with vague prior knowledge, he may refer to it as *prior belief*. Almost always, one's prior knowledge about a computer code is a prior belief.

Prior knowledge about $f(\cdot)$ can be modeled by a probability measure on the space of functions from \mathcal{X} to \mathbb{R} . This probability measure encodes one's prior

beliefs, in the sense that it assigns probability one to the set of functions that are consistent with it. A Gaussian process is a great way to represent this probability measure.

A Gaussian process is a generalization of a multivariate Gaussian random variable to infinite dimensions [39]. In particular, $f(\cdot)$ is a Gaussian process with *mean function* $m : \mathcal{X} \rightarrow \mathbb{R}$ and *covariance function* $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, i.e.,

$$f(\cdot) \sim \text{GP}(f(\cdot)|m(\cdot), k(\cdot, \cdot)), \quad (15.18)$$

if and only if for any collection of input points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{X}$, the corresponding outputs $\mathbf{Y} = \{y_1 = f(\mathbf{x}_1), \dots, y_n = f(\mathbf{x}_n)\}$ are distributed according to:

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}_n(\mathbf{Y}|\mathbf{m}(\mathbf{X}), \mathbf{k}(\mathbf{X}, \mathbf{X})), \quad (15.19)$$

where $\mathbf{m}(\mathbf{X}) = (m(\mathbf{x}_i))_i$, $\mathbf{k}(\mathbf{X}, \mathbf{X}') = (k(\mathbf{x}_i, \mathbf{x}'_j))_{ij}$, and $\mathcal{N}_n(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are the probability density function of a n -dimensional multivariate normal random variable with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

But how does Equation (15.18) encode one's prior knowledge about the code? It does so through the choice of the mean and the covariance functions. The mean function can be an arbitrary function. Its role is to encode any generic trends about the response. The covariance function can be any semi-positive-definite function. It is used to model the signal strength of the response and how it varies across \mathcal{X} , the similarity (correlation) of the response at two distinct input points, noise, regularity, periodicity, invariance, and more. The choice of mean and covariance functions is discussed more elaborately in what follows.

Vague prior knowledge, i.e., prior beliefs, may be modeled by parameterizing the mean and covariance functions and assigning prior probabilities to their parameters. In particular, the following generic forms for the mean and covariance functions are considered:

$$m : \mathcal{X} \times \Psi_m \rightarrow \mathbb{R}, \quad (15.20)$$

and

$$k : \mathcal{X} \times \mathcal{X} \times \Psi_k \rightarrow \mathbb{R}, \quad (15.21)$$

respectively. Here

$$\Psi_m \subset \mathbb{R}^{d_m} \text{ and } \Psi_k \subset \mathbb{R}^{d_k}, \quad (15.22)$$

and for all $\boldsymbol{\psi}_k \in \Psi_k$ the function $k(\cdot, \cdot; \boldsymbol{\psi}_k)$ is positive definite. The parameters $\boldsymbol{\psi}_m \in \Psi_m$ and $\boldsymbol{\psi}_k \in \Psi_k$, of the mean and covariance functions, respectively, are

known as *hyper-parameters* of the model. Using this notation, the most general model considered in this work is:

$$f(\cdot) | \psi_m, \psi_k \sim \text{GP}(f(\cdot) | m(\cdot; \psi_m), k(\cdot, \cdot; \psi_k)), \quad (15.23)$$

$$\psi_m \sim p(\psi_m), \quad (15.24)$$

$$\psi_k \sim p(\psi_k). \quad (15.25)$$

For notational economy one writes:

$$\psi := \{\psi_m, \psi_k\}, \quad (15.26)$$

and

$$p(\psi) := p(\psi_m)p(\psi_k). \quad (15.27)$$

Choosing the Mean Function

The role of the mean function is to model one's prior beliefs about the existence of systematic trends. One of the most common choices for the mean function is the *generalized linear model*:

$$m(\mathbf{x}; \mathbf{b}) = \mathbf{b}^T \mathbf{h}(\mathbf{x}), \quad (15.28)$$

where $\mathbf{h} : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$ is an arbitrary function and $\mathbf{b} =: \psi_m \in \Psi_m = \mathbb{R}^{d_m}$, $d_m = d_h$, are the hyper-parameters known as *weights*. A popular prior for the weights, \mathbf{b} , is the improper, “non-informative,” uniform:

$$p(\psi_m) = p(\mathbf{b}) \propto 1. \quad (15.29)$$

Another commonly used prior is the multivariate normal:

$$p(\mathbf{b} | \mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) = \mathcal{N}_{d_m}(\mathbf{b} | \mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}), \quad (15.30)$$

where $\mu_{\mathbf{b}} \in \mathbb{R}^{d_m}$, and $\Sigma_{\mathbf{b}} \in \mathbb{R}^{d_m \times d_m}$ is positive definite. It can be shown that in both choices, Equation (15.29) or Equation (15.30), it is actually possible to integrate \mathbf{b} out of the model [15]. Some examples of generalized linear models are:

1. The *constant* mean function, $d_h = 1$,

$$\mathbf{h}(\mathbf{x}) = 1; \quad (15.31)$$

2. The *linear* mean function, $d_h = d + 1$,

$$\mathbf{h}(\mathbf{x}) = (1, x_1, \dots, x_d); \quad (15.32)$$

3. The *generalized polynomial chaos* (gPC) mean function in which $\mathbf{h}(\cdot) = (h_1(\cdot), \dots, h_{d_h}(\cdot))$, with the $h_i : \mathbb{R}^d \rightarrow \mathbb{R}, i = 1, \dots, d_h$ being polynomials with degree up to ρ and orthogonal with respect to a measure $\mu(\cdot)$, i.e.

$$\int h_i(x)h_j(x)d\mu(x) = \delta_{ij}. \quad (15.33)$$

For uncertainty propagation applications, $\mu(\cdot)$ is usually a probability measure. For many well-known probability measures, the corresponding gPC is known [53]. For arbitrary probability measures, the polynomials can only be constructed numerically, e.g., [24]. Excellent Fortran code for the construction of orthogonal polynomials can be found in [25]. An easy-to-use Python interface is available by Bilionis [6].

4. The *Fourier* mean function defined for $d = 1$ in which $\mathbf{h}(\cdot) = (h_1(\cdot), \dots, h_{d_h}(\cdot))$, with the $h_i : \mathbb{R} \rightarrow \mathbb{R}, i = 1, \dots, d_h$ being trigonometric functions supporting certain frequencies ω_i , i.e.,

$$h_{2i}(x) = \sin(\omega_i x), \text{ and } h_{2i+1}(x) = \cos(\omega_i x). \quad (15.34)$$

Choosing the Covariance Function

Discussing the choice of the covariance function in great depth is beyond the scope of the chapter. Instead, the interested reader is pointed to the excellent account by Rasmussen and Williams [39]. Here, some remarkable aspects that relate directly to representing prior beliefs for computer codes are discussed:

1. Modeling measurement noise. Assume that measurements of $f(\mathbf{x})$ are noisy and that this noise is Gaussian with variance σ^2 . GPs can account for this fact if one adds a Kronecker delta-like function to the covariance, i.e., if one works with a covariance of the form:

$$k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}_k) = k_0(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}_{k_0}) + \sigma^2 \delta(\mathbf{x} - \mathbf{x}'). \quad (15.35)$$

Note that $\delta(\mathbf{x} - \mathbf{x}')$ here is one if and only if \mathbf{x} and \mathbf{x}' correspond to exactly the same measurement. Even though most computer simulators do not return noisy outputs, some do, e.g., molecular dynamics simulations. So, being able to model noise is useful. Apart from that, it turns out that including a small σ^2 , even when there is no noise, is beneficial because it can improve the stability of the computations. When σ^2 is included for this reason, it is known as a “nugget” or as a “jitter.” In addition to the improved stability, the nugget can also lead to better predictive accuracy [27].

2. Modeling regularity. It can be shown that the regularity properties of the covariance function are directly associated to the regularity of the functions sampled from the probability measure induced by the GP (Equations (15.18) and (15.23)). For example, if $k(\mathbf{x}, \mathbf{x}; \boldsymbol{\psi}_k)$ is continuous at \mathbf{x} , then samples $f(\cdot)$ from Equation (15.23) are continuous almost surely (a.s.) at \mathbf{x} . If $k(\mathbf{x}, \mathbf{x}; \boldsymbol{\psi})$ is ρ

times differentiable at \mathbf{x} , then samples $f(\cdot)$ from Equation (15.23) are ρ times differentiable a.s. at \mathbf{x} .

3. Modeling invariance. If $k(\cdot, \cdot; \psi_k)$ is invariant with respect to a transformation T , i.e., $k(\mathbf{x}, T\mathbf{x}'; \psi_k) = k(T\mathbf{x}, \mathbf{x}'; \psi_k) = k(\mathbf{x}, \mathbf{x}'; \psi_k)$, then samples $f(\mathbf{x})$ from Equation (15.23) are invariant with respect to the same transformation a.s. In particular, if $k(\cdot, \cdot; \psi)$ is periodic, then samples $f(\mathbf{x})$ from Equation (15.23) are periodic a.s.
4. Modeling additivity. Assume that the covariance function is additive, i.e.,

$$k(\mathbf{x}, \mathbf{x}'; \psi_k) = \sum_{i=1}^d k_i(x_i, x'_i; \psi_{k_i}) + \sum_{1 \leq i < j \leq d} k_{ij}((x_i, x_j), (x'_i, x'_j); \psi_{k_{ij}}) + \dots, \quad (15.36)$$

with $\psi_k = \{\psi_{k_i}\} \cup \{\psi_{k_{ij}}\}$. If $f_i(\mathbf{x}), f_{ij}(\mathbf{x}), \dots$ are samples from Equation (15.23) with covariances $k_i(\cdot, \cdot; \psi_{k_i}), k_{ij}(\cdot, \cdot; \psi_{k_{ij}}), \dots$, respectively, then

$$f(\mathbf{x}) = \sum_{i=1}^d f_i(x_i) + \sum_{1 \leq i < j \leq d} f_{ij}(x_i, x_j) + \dots,$$

is a sample from Equation (15.23) with the additive covariance defined in Equation (15.36). These ideas can be used to deal effectively with high-dimensional inputs [22, 23].

The most commonly used covariance function for representing one's prior knowledge about a computer code is of the form of Equation (15.35) with $k_0(\cdot, \cdot; \psi_{k_0})$ being the squared exponential (SE) covariance function:

$$k_0(\mathbf{x}, \mathbf{x}'; \psi_{k_0}) = s^2 \exp \left\{ - \sum_{i=1}^d \frac{(x_i - x'_i)^2}{2\ell_i^2} \right\}, \quad (15.37)$$

where $\psi_{k_0} = \{s, \ell_1, \dots, \ell_d\}$. Note that $s > 0$ may be interpreted as the *signal strength* and $\ell_i > 0$ as the *length scale* of the $i = 1, \dots, d$ input dimension. Prior beliefs about $\psi_k = \{\sigma^2\} \cup \{\psi_{k_0}\}$ are modeled by:

$$p(\psi_k) = p(\sigma)p(\psi_{k_0}), \quad (15.38)$$

with

$$p(\psi_{k_0}) = p(s) \prod_{i=1}^d p(\ell_i). \quad (15.39)$$

Typically, $p(\sigma), p(s), p(\ell_1), \dots, p(\ell_d)$ are chosen to be Jeffreys priors or exponential probability densities with fixed parameters.

2.5.2 Conditioning on Observations of the Response

As seen in the previous section, one's prior knowledge about the response can be modeled in terms of a generic GP defined by Equations (15.23), (15.24), and (15.25). Now, assume that one makes n simulations at inputs $\mathbf{x}_1, \dots, \mathbf{x}_n$ and that he observes the responses $y_1 = f(\mathbf{x}_1), \dots, y_n = f(\mathbf{x}_n)$. Write $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and $\mathbf{Y} = \{y_1, \dots, y_n\}$ for the observed inputs and outputs, respectively. Abusing the mathematical notation slightly, the symbol \mathcal{D} is used to denote \mathbf{X} and \mathbf{Y} collectively (see Equation (15.14)). We refer to \mathcal{D} as the (*observed*) data. How does the observation of \mathcal{D} alter one's state of knowledge about the response surface?

The answer to the aforementioned question comes after a straightforward application of Bayes rule and use of Kolmogorov's theorem on the existence of random fields. One's state of knowledge is characterized by a new GP,

$$f(\cdot) | \mathcal{D}, \psi \sim \text{GP}\left(f(\cdot) | m^*(\cdot; \psi, \mathcal{D}), k^*(\cdot, \cdot; \psi, \mathcal{D})\right), \quad (15.40)$$

with mean function:

$$m^*(\mathbf{x}; \psi, \mathcal{D}) := m(\mathbf{x}; \psi_m) + \mathbf{k}(\mathbf{x}, \mathbf{X}; \psi_k) \mathbf{k}(\mathbf{X}, \mathbf{X}; \psi_k)^{-1} (\mathbf{Y} - \mathbf{m}(\mathbf{X}; \psi_m)), \quad (15.41)$$

covariance function:

$$k^*(\mathbf{x}, \mathbf{x}'; \psi, \mathcal{D}) := k(\mathbf{x}, \mathbf{x}'; \psi_k) - \mathbf{k}(\mathbf{x}, \mathbf{X}; \psi_k) \mathbf{k}(\mathbf{X}, \mathbf{X}; \psi_k)^{-1} \mathbf{k}(\mathbf{X}, \mathbf{x}'; \psi_k), \quad (15.42)$$

and the *posterior* of the hyper-parameters:

$$p(\psi | \mathcal{D}) = \frac{p(\mathcal{D} | \psi) p(\psi)}{p(\mathcal{D})}, \quad (15.43)$$

where

$$p(\mathcal{D} | \psi) := p(\mathbf{Y} | \mathbf{X}, \psi) = \mathcal{N}_n(\mathbf{Y} | \mathbf{m}(\mathbf{X}; \psi_m), \mathbf{k}(\mathbf{X}, \mathbf{X}; \psi_m)), \quad (15.44)$$

is the *likelihood* of \mathcal{D} induced by the defining property of the GP (Equation (15.19)) and $p(\mathcal{D}) = \int p(\mathcal{D} | \psi) p(\psi) d\psi$ the evidence.

2.5.3 Treating Space and Time by Using Separable Mean and Covariance Functions

The input \mathbf{x} may include spatial \mathbf{x}_s ; time, t ; and stochastic, ξ , components (see Equation (15.1)). A computer code reports the response for a given ξ (see Equation (15.7)) at a fixed set of n_s spatial locations, \mathbf{X}_s , and n_t time instants \mathbf{X}_t (see Equations (15.4) and (15.5), respectively). Now suppose that n_ξ observations of ξ are to be made. Then, the size of the covariance matrix $\mathbf{k}(\mathbf{X}, \mathbf{X}; \psi_k)$ used in Equation (15.19) becomes $(n_\xi n_s n_t) \times (n_\xi n_s n_t)$. Since the cost of inference and prediction is $O((n_s n_t n_\xi)^3)$, one encounters insurmountable computational issues even for moderate values of n_ξ, n_s , and n_t . In an attempt to remedy this problem,

simplifying assumptions must be made. Namely, one has to assume that the mean is a generalized linear model (see Equation (15.28)) with a *separable* set of basis functions, $\mathbf{h}(\cdot)$, and that the covariance function is also separable.

The mean function $\mathbf{h} : \mathcal{X}_s \times \mathcal{X}_t \times \mathcal{X}_\xi \rightarrow \mathbb{R}^{d_h}$ is separable if it can be written as:

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}_s(\mathbf{x}_s) \otimes \mathbf{h}_t(t) \otimes \mathbf{h}_\xi(\xi), \quad (15.45)$$

where “ \otimes ” denotes the Kronecker product, $\mathbf{h}_s : \mathcal{X}_s \rightarrow \mathbb{R}^{d_{h_s}}$, $\mathbf{h}_t : \mathcal{X}_t \rightarrow \mathbb{R}^{d_{h_t}}$, and $\mathbf{h}_\xi : \rightarrow \mathbb{R}^{d_{h_\xi}}$, with $d_h = d_{h_s} d_{h_t} d_{h_\xi}$.

The covariance function, $k : \mathcal{X}_s \times \mathcal{X}_t \times \mathcal{X}_\xi \rightarrow \mathbb{R}$, is separable if it can be written as:

$$k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}_k) = k_s(\mathbf{x}_s, \mathbf{x}'_s; \boldsymbol{\psi}_{k,s}) k_t(t, t'; \boldsymbol{\psi}_{k,t}) k_\xi(\xi, \xi'; \boldsymbol{\psi}_{k,\xi}), \quad (15.46)$$

where $k_s(\cdot, \cdot; \boldsymbol{\psi}_{k,s})$, $k_t(\cdot, \cdot; \boldsymbol{\psi}_{k,t})$, and $k_\xi(\cdot, \cdot; \boldsymbol{\psi}_{k,\xi})$ are spatial, time, and stochastic domain covariance functions, respectively; $\boldsymbol{\psi}_{k,s}$, $\boldsymbol{\psi}_{k,t}$, and $\boldsymbol{\psi}_{k,\xi}$ are the corresponding hyper-parameters; and $\boldsymbol{\psi}_k = \{\boldsymbol{\psi}_{k,s}, \boldsymbol{\psi}_{k,t}, \boldsymbol{\psi}_{k,\xi}\}$. Then, as shown in Bilionis et al. [10], the covariance matrix can be written as the Kronecker product of a spatial, a time, and a stochastic covariance, i.e.,

$$\mathbf{k}(\mathbf{X}, \mathbf{X}; \boldsymbol{\psi}_k) = \mathbf{k}_s(\mathbf{X}_s, \mathbf{X}_s; \boldsymbol{\psi}_{k,s}) \otimes \mathbf{k}_t(\mathbf{X}_t, \mathbf{X}_t; \boldsymbol{\psi}_{k,t}) \otimes \mathbf{k}_\xi(\mathbf{X}_\xi, \mathbf{X}_\xi; \boldsymbol{\psi}_{k,\xi}). \quad (15.47)$$

Exploiting the fact that factorizations (e.g., Cholesky or QR) of a matrix formed by Kronecker products are given by the Kronecker products of the factorizations of the individual matrices [47], inference and predictions can be made in $O(n_s^3) + O(n_t^3) + O(n_\xi^3)$ time. For the complete details of this approach, the reader is directed to the appendix of Bilionis et al. [10]. It is worth mentioning that exactly the same idea was used by Stegle et al. [46].

2.5.4 Treating Space and Time by Using Output Dimensionality Reduction

In this section an alternative way to deal with space and time inputs, first introduced by Higdon et al. [30], is presented. Assume that the computer code reports the response at specific spatial locations and time instants exactly as in the previous subsection (see also Equation (15.7)). The idea is to perform a dimensionality reduction on the output matrix $\mathbf{Y} \in \mathbb{R}^{n_\xi \times (n_s n_t)}$,

$$\mathbf{Y} = \begin{pmatrix} \mathbf{f}_c(\xi_1)^T \\ \mathbf{f}_c(\xi_{n_\xi})^T \end{pmatrix}, \quad (15.48)$$

and then learn the map between ξ the reduced variables. For notational convenience, let $n_y = n_s n_t$ denote the number of outputs of the code, $\mathbf{y} \in \mathbb{R}^{n_y}$ the full output, and $\mathbf{z} \in \mathbb{R}^{n_z}$ the reduced output. The choice of the right dimensionality reduction map is an open research problem. Principal component analysis (PCA) [12, Chapter 12] is

going to be used for the identification of the dimensionality reduction map. Consider the empirical covariance matrix $\mathbf{C} \in \mathbb{R}^{n_y \times n_y}$:

$$\mathbf{C} = \frac{1}{n_\xi - 1} \sum_{i=1}^{n_\xi} (\mathbf{Y}_i - \mathbf{m})(\mathbf{Y}_i - \mathbf{m})^T, \quad (15.49)$$

where \mathbf{Y}_i is the i -th row of the output matrix \mathbf{Y} and $\mathbf{m} \in \mathbb{R}^{n_y}$ is the empirical mean of the observed outputs:

$$\mathbf{m} = \frac{1}{n_\xi} \sum_{i=1}^{n_\xi} \mathbf{Y}_i. \quad (15.50)$$

One proceed by diagonalizing \mathbf{C} :

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \quad (15.51)$$

where $\mathbf{V} \in \mathbb{R}^{n_y \times n_y}$ contains the eigenvectors of \mathbf{C} as columns, and $\mathbf{D} \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix with the eigenvalues of \mathbf{C} on its diagonal. The PCA connection between \mathbf{y} and \mathbf{z} (reconstruction map) is given by:

$$\mathbf{y} = \mathbf{V}\mathbf{D}^{1/2}\mathbf{z} + \mathbf{m}. \quad (15.52)$$

In this equation, the reduced outputs corresponding to very small eigenvalues can be eliminated to a very good approximation. Typically, one keeps n_z eigenvalues of \mathbf{C} so that 95%, or more, of the observed variance of \mathbf{y} is explained. This is achieved by removing columns from \mathbf{V} and columns and rows from \mathbf{D} . The inverse of Equation (15.52) (reduction map) is given by:

$$\mathbf{z} = \mathbf{V}^T \mathbf{D}^{-1/2}(\mathbf{y} - \mathbf{m}). \quad (15.53)$$

The map between ξ and the reduced variables \mathbf{z} can be learned by any of the multi-output GP regression techniques to be introduced in subsequent sections.

2.6 Training the Parameters of the Gaussian Process

Training the parameters of a GP requires the characterization of the posterior distribution $p(\psi | \mathcal{D})$. A very general way to describe this posterior is via a *particle approximation*. A particle approximation is a collection of weights, $w^{(s)}$, and samples, $\psi^{(s)}$, with which one may represent the posterior as:

$$p(\psi | \mathcal{D}) \approx \sum_{s=1}^S w^{(s)} \delta(\psi - \psi^{(s)}). \quad (15.54)$$

Here, $w^{(s)} \geq 0$ and $\sum_{s=1}^S w^{(s)} = 1$. The usefulness of Equation (15.54) relies on the fact that it allows one to approximate expectations with respect to $p(\psi|\mathcal{D})$ in a straightforward manner.

$$\mathbb{E}_{\psi}[g(\psi)] := \int g(\psi) p(\psi|\mathcal{D}) d\psi \approx \sum_{s=1}^S w^{(s)} g(\psi^{(s)}). \quad (15.55)$$

There are various ways in which a particle approximation can be constructed. The discussion below includes the most common approaches.

2.6.1 Maximization of the Likelihood

The most widespread approach to training the GP parameters is to obtain a point estimate of the hyper-parameters by maximizing the likelihood of the data as given in Equation (15.44) [39, Ch. 5]. This is known as the maximum likelihood estimator (MLE) of the hyper-parameters:

$$\psi_{\text{MLE}}^* = \arg \max_{\psi \in \Psi} p(\mathcal{D}|\psi). \quad (15.56)$$

The MLE can be thought as single-particle approximation, i.e., $S = 1, w^{(1)} = 1, \psi^{(1)} = \psi_{\text{MLE}}^*$,

$$p(\psi|\mathcal{D}) \approx \delta(\psi - \psi_{\text{MLE}}^*), \quad (15.57)$$

if the prior is relatively flat and the posterior is sharply peaked around ψ_{MLE}^* .

2.6.2 Maximization of the Posterior

If the prior information is not flat but the posterior is expected to be sharply peaked, then it is preferable to use the maximum a posteriori (MAP) estimate of the hyper-parameters:

$$\psi_{\text{MAP}}^* = \arg \max_{\psi \in \Psi} p(\psi|\mathcal{D}), \quad (15.58)$$

where the posterior $p(\psi|\mathcal{D})$ was defined in Equation (15.43). This is also a single-particle approximation:

$$p(\psi|\mathcal{D}) \approx \delta(\psi - \psi_{\text{MAP}}^*). \quad (15.59)$$

2.6.3 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) techniques [29, 31, 33] can be employed to sample from the posterior of Equation (15.43). These techniques result in a series of uncorrelated samples $\psi^{(s)}, s = 1, \dots, S$ from Equation (15.43). The particle approximation is built by picking $w^{(s)} = 1/S, s = 1, \dots, S$. MCMC is useful when the posterior is unimodal.

2.6.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a nonlinear extension of Kalman filters [19, 21]. These algorithms can be also used in order to sample from the posterior of a Bayesian model. For example, Bilionis and Zabaras [9], Bilionis et al. [11], Wan and Zabaras [48] use them to sample from the posterior of a stochastic inverse problem. One introduces a family of probability densities on ψ parameterized by γ :

$$p(\psi | \gamma, \mathcal{D}) \propto p(\psi | \mathcal{D})^\gamma p(\psi). \quad (15.60)$$

Notice that for $\gamma = 0$ one obtains the prior and for $\gamma = 1$ one obtains the posterior. The idea is to start a particle approximation $\left\{ w_0^{(s)}, \psi_0^{(s)} \right\}_{s=1}^S$ from the prior ($\gamma = 0$) which is very easy to sample from and gradually move it to the posterior ($\gamma = 1$). This can be easily achieved if there is an underlying MCMC routine for sampling from Equation (15.60). In addition, the schedule of γ 's can be picked adaptively. The reader is directed to the appendix of Bilionis and Zabaras [9] for the complete details. SMC-based methodologies are suitable for multimodal posteriors. Another very attractive attribute is that they are embarrassingly parallelizable.

2.7 Multi-output Gaussian Processes

In what follows, the treatment of generic d_y -dimensional response functions $\mathbf{f}(\cdot)$ is considered. The collection of all observed outputs is denoted by $\mathbf{Y}_d = (\mathbf{y}_1^T, \dots, \mathbf{y}_n^T) \in \mathbb{R}^{n \times d_y}$. The proper treatment of multiple-outputs in a computationally efficient way is an open research problem. Even though there exist techniques that attempt to capture nontrivial dependencies between the various outputs, e.g., [2, 3, 13, 46], here the focus is on computationally simple techniques that treat outputs as either independent or linearly dependent. These techniques tend to be the easiest to use in practice.

2.7.1 Completely Independent Outputs

In this model the outputs are completely independent, i.e.,

$$\mathbf{f}(\cdot) | \psi \sim \prod_{i=1}^{d_y} \text{GP} \left(f_i(\cdot) | m_i(\cdot; \psi_{m,i}), k_i(\cdot, \cdot; \psi_{k,i}) \right), \quad (15.61)$$

with

$$\psi = \bigcup_{i=1}^{d_y} \{\psi_{m,i}, \psi_{k,i}\}, \quad (15.62)$$

where $m_i(\cdot, \cdot)$, $k_i(\cdot, \cdot)$ are the mean and the covariance function and $\psi_{m,i}$, $\psi_{k,i}$ the parameters of the mean and the covariance function, respectively, of the i -th output of $\mathbf{f}(\cdot)$. The likelihood of the model is:

$$\begin{aligned} p(\mathcal{D} | \psi) &= p(\mathbf{Y} | \mathbf{X}, \psi) \\ &= \prod_{i=1}^{d_y} p(\mathbf{y}_i | \mathbf{X}, \psi_i), \end{aligned} \quad (15.63)$$

where $\psi_i = \{\psi_{m,i}, \psi_{k,i}\}$ and $p(\mathbf{y}_i | \mathbf{X}, \psi_i)$ is a multivariate Gaussian exactly the same as the one in Equation (15.19) with $m_i(\cdot; \psi_{m,i})$ and $k_i(\cdot, \cdot; \psi_{k,i})$ instead of $m(\cdot; \psi_m)$ and $k(\cdot, \cdot; \psi_k)$, respectively. A typical choice of a prior for ψ assumes a priori independence of the various parameters, i.e.,

$$p(\psi) = \prod_{i=1}^{d_y} (p(\psi_{m,i}) p(\psi_{k,i})) . \quad (15.64)$$

There is nothing special about this model. Essentially, it is equivalent to carrying out a Gaussian process regression independently on each one of the outputs.

2.7.2 Independent, but Similar, Outputs

In this model, the outputs are correlated only via the parameters of the covariance function, i.e.,

$$\mathbf{f}(\cdot) | \psi \sim \prod_{i=1}^{d_y} \text{GP}(f_i(\cdot) | m(\cdot; \psi_{m,i}), \psi_{s,i}^2 k(\cdot, \cdot; \psi_k)) , \quad (15.65)$$

with

$$\psi = \{\psi_k\} \cup \bigcup_{i=1}^{d_y} \{\psi_{m,i}, \psi_{s,i}\} , \quad (15.66)$$

where $\psi_{m,i}$ and $\psi_{s,i}$ are the parameters of the mean and the signal strength, respectively, of the i -th outputs, and ψ_k are the parameters of the covariance function shared by all outputs. The likelihood of this model is given by an equation similar to Equation (15.63) with an appropriate mean and covariance function. The prior of ψ is assumed to have an a priori independence structure similar to Equation (15.64). The advantage of this approach compared with the fully independent approach is that all outputs share the same covariance function, and, hence, its computational complexity is the same as that of a single-output Gaussian process regression.

2.7.3 Linearly Correlated Outputs

Conti and O'Hagan [15] introduce a simple multi-output model in which the outputs are correlated using a positive-definite covariance matrix $\Sigma \in \mathbb{R}^{d_y \times d_y}$ via a d_y -dimensional Gaussian random field, i.e.,

$$\mathbf{f}(\cdot) | \psi \sim \text{GP}(\mathbf{f}(\cdot) | \mathbf{m}(\cdot; \psi_m), k(\cdot, \cdot; \psi_k), \Sigma) , \quad (15.67)$$

with

$$\psi = \{\psi_m, \psi_k, \Sigma\} , \quad (15.68)$$

where $\mathbf{m}(\cdot; \psi_m)$ is the mean-vector function with d_y -outputs, and $k(\cdot, \cdot; \psi_k)$ is a common covariance function. Equation (15.67) essentially means that, a priori,

$$\mathbb{E}[\mathbf{f}(\mathbf{x})|\psi] = \mathbf{m}(\mathbf{x}; \psi_m), \quad (15.69)$$

and

$$\mathbb{C}[\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')|\psi] = k(\mathbf{x}, \mathbf{x}'; \psi_k) \Sigma. \quad (15.70)$$

Notice that there is no need to include a signal parameter in the covariance function, since it will be absorbed by Σ . Therefore, in this model, one can identically take the signal strength of the response to be identically equal to one. Equation (15.70) provides the intuitive meaning of Σ as the matrix capturing the linear part of the correlations of the various outputs. In this case, the likelihood is given via a matrix normal [18]:

$$\begin{aligned} p(\mathcal{D}|\psi) &= p(\mathbf{Y}|\mathbf{X}, \psi) \\ &= \mathcal{N}_{n \times d_y} (\mathbf{Y} | \mathbf{m}(\mathbf{X}; \psi_m), \Sigma, \mathbf{k}(\mathbf{X}, \mathbf{X}; \psi_k)). \end{aligned} \quad (15.71)$$

The prior of ψ is assumed to have the form:

$$p(\psi) = p(\psi_m)p(\psi_k)p(\Sigma), \quad (15.72)$$

with the prior of Σ being “non-informative” [15]:

$$p(\Sigma) \propto |\Sigma|^{-\frac{d_y+1}{2}}. \quad (15.73)$$

Alternatively, Σ could follow an inverse Wishart distribution [28]. As shown in [10] and [14], if in addition to the prior assumptions for Σ , the mean $m(\cdot; \psi_m)$ is chosen to be a generalized linear model with a priori flat weights ψ_m , then both Σ and ψ_m can be integrated out of the model analytically. This feature makes inference in this model as computationally efficient as single-output Gaussian process regression (GPR). For the complete details of this approach, the reader is directed to [10].

2.8 Sampling Possible Surrogates

The purpose of this section is to demonstrate how the posterior Gaussian process defined by Equations (15.40) and (15.43) can be represented as $\hat{f}(\cdot; \theta)$, where θ is a finite dimensional random variable with probability density $p(\theta|\mathcal{D})$. There are two ways in which this can be achieved. The first way is based on a truncated Karhunen-Loëve expansion [26, 32, 45] (KLE) of the Equation (15.40). The second way is

based on the idea introduced in O'Hagan et al. [38], further discussed in Oakley and O'Hagan [34, 35], and revisited in Bilionis et al. [10] as well as in Chen et al. [14]. In both approximations, $\hat{f}(\cdot; \boldsymbol{\theta})$ is given *analytically*. In particular, it can be expressed as

$$\hat{f}(\mathbf{x}; \boldsymbol{\theta}) = m^*(\mathbf{x}; \boldsymbol{\psi}) + \mathbf{k}^*(\mathbf{x}, \mathbf{X}_d; \boldsymbol{\psi}) \mathbf{C}(\boldsymbol{\psi}) \boldsymbol{\omega}, \quad (15.74)$$

where $m^*(\cdot; \boldsymbol{\psi})$ is the posterior mean given in Equation (15.41) and $\mathbf{k}^*(\mathbf{x}, \mathbf{X}_d; \boldsymbol{\psi})$ is the posterior cross covariance, defined generically via

$$\mathbf{k}^*(\mathbf{X}, \mathbf{X}'; \boldsymbol{\psi}) := \begin{pmatrix} k^*(\mathbf{x}_1, \mathbf{x}'_1; \boldsymbol{\psi}) & \dots & k^*(\mathbf{x}_1, \mathbf{x}'_{n'}; \boldsymbol{\psi}) \\ \vdots & \ddots & \vdots \\ k^*(\mathbf{x}_n, \mathbf{x}'_1; \boldsymbol{\psi}) & \dots & k^*(\mathbf{x}_n, \mathbf{x}'_{n'}; \boldsymbol{\psi}) \end{pmatrix}, \quad (15.75)$$

with $k^*(\cdot, \cdot; \boldsymbol{\psi})$ being the posterior covariance defined in Equation (15.42). The parameters $\boldsymbol{\theta}$ that characterize the surrogate surface $\hat{f}(\mathbf{x}; \boldsymbol{\theta})$ are:

$$\boldsymbol{\theta} = \{\boldsymbol{\psi}, \boldsymbol{\omega}\}, \quad (15.76)$$

with

$$\boldsymbol{\omega} = (\omega_1, \dots, \omega_{d_\omega}), \quad (15.77)$$

being distributed as a standard normal random vector, i.e.,

$$p(\boldsymbol{\omega}) = \mathcal{N}_{d_\omega}(\boldsymbol{\omega} | \mathbf{0}_{d_\omega}, \mathbf{I}_{d_\omega}). \quad (15.78)$$

The posterior probability density of $\boldsymbol{\theta}$, $p(\boldsymbol{\theta} | \mathcal{D})$, is given by:

$$p(\boldsymbol{\theta} | \mathcal{D}) = p(\boldsymbol{\psi} | \mathcal{D}) p(\boldsymbol{\omega}), \quad (15.79)$$

with $p(\boldsymbol{\psi} | \mathcal{D})$ being the posterior of the hyper-parameters (see Equation (15.43)). The matrix

$$\mathbf{X}_d = (\mathbf{x}_{d,1}^T \dots \mathbf{x}_{d,n_d}^T)^T, \quad (15.80)$$

contains n_d *design* points in \mathcal{X} , and $\mathbf{C}(\boldsymbol{\psi}) \in \mathbb{R}^{n_d \times d_\omega}$ is a matrix that corresponds to a factorization of the posterior covariance function of Equation (15.43) over the design points \mathbf{X}_d . The optimal choice of the design points, \mathbf{X}_d , is an open research problem. Below, some heuristics are provided. These depend on how one actually constructs the various quantities of Equation (15.74).

2.8.1 The Karhunen-Loève Approach for Constructing $\hat{f}(\cdot; \theta)$

Let $\omega_\ell, \ell = 1, \dots$ be standard normal random variables. Consider the eigenvalues $\lambda_i(\psi)$ and eigenfunctions $\phi_\ell(\cdot; \psi)$ of the posterior covariance function $k^*(\cdot, \cdot; \psi)$ of Equation (15.42). That is:

$$\int k^*(\mathbf{x}, \mathbf{x}'; \psi) \phi_\ell(\mathbf{x}; \psi) d\mathbf{x} = \lambda_\ell(\psi) \phi_\ell(\mathbf{x}; \psi), \quad (15.81)$$

for $\ell = 1, \dots$. Then, the posterior Gaussian process defined by Equation (15.40) can be written as:

$$\hat{f}(\mathbf{x}; \psi, \omega_1, \dots) = m^*(\mathbf{x}; \psi) + \sum_{\ell=1}^{\infty} \omega_\ell \sqrt{\lambda_\ell(\psi)} \phi_\ell(\mathbf{x}; \psi). \quad (15.82)$$

Typically, one truncates the series to a finite-order d_ω and writes

$$\hat{f}(\mathbf{x}; \theta) = m^*(\mathbf{x}; \psi) + \sum_{\ell=1}^{d_\omega} \omega_\ell \sqrt{\lambda_\ell(\psi)} \phi_\ell(\mathbf{x}; \psi), \quad (15.83)$$

where θ and ω are as in Equations (15.76) and (15.77), respectively. The probability density of θ , $p(\theta | \mathcal{D})$, is defined via Equations (15.79) and (15.78).

Equation (15.81) is a Fredholm integral eigenvalue problem. In general, this equation cannot be solved analytically. A very recent study of the numerical techniques that can be used for the solution of this problem can be found in Betz et al. [5]. This work relies on the Nyström approximation [20, 40] and follows Betz et al. [5] closely in its development.

Start by approximating the integral on the left-hand side of Equation (15.81) by

$$\sum_{j=1}^{n_d} w_j k^*(\mathbf{x}, \mathbf{x}_{d,j}; \psi) \phi_\ell(\mathbf{x}_{d,j}; \psi) \approx \lambda_\ell(\psi) \phi_\ell(\mathbf{x}; \psi), \quad (15.84)$$

where $\{\mathbf{x}_{d,j}, w_j\}_{j=1}^{n_d}$ is a suitable quadrature rule. Notice that in this approximation the design points of Equation (15.80) correspond to the quadrature points. The simplest quadrature rule would be a Monte Carlo type of rule, in which the points $\mathbf{x}_{d,j}$ are randomly picked in \mathcal{X} and $w_j = \frac{1}{n_d}$. Other choices could be based on tensor products of one-dimensional rules or a sparse grid quadrature rule [44]. As shown later on, for the special – but very common – case of a separable covariance function, the difficulty of the problem can be reduced dramatically.

The next step in the Nyström approximation is to solve Equation (15.84) at the quadrature points:

$$\sum_{j=1}^{n_d} w_j k^*(\mathbf{x}_{d,i}, \mathbf{x}_{d,j}; \psi) \phi_\ell(\mathbf{x}_{d,j}; \psi) \approx \lambda_\ell(\psi) \phi_\ell(\mathbf{x}_{d,i}; \psi), \quad (15.85)$$

for $i = 1, \dots, n_d$. Therefore, one needs to solve the generalized eigenvalue problem:

$$\mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \boldsymbol{\psi}) \mathbf{W} \mathbf{v}_\ell = \lambda_\ell \mathbf{v}_\ell, \quad (15.86)$$

where $k^*(\mathbf{X}_d, \mathbf{X}_d; \boldsymbol{\psi})$ is the posterior covariance matrix on the integration points \mathbf{X}_d (see Equation (15.75)), and $\mathbf{W} = \text{diag}(w_1, \dots, w_{n_d})$. It is easy to see that the solution of Equation (15.86) can be obtained by first solving the regular eigenvalue problem

$$\mathbf{B} \tilde{\mathbf{v}}_\ell = \lambda_\ell \tilde{\mathbf{v}}_\ell, \quad (15.87)$$

where

$$\mathbf{B} = \mathbf{W}^{\frac{1}{2}} \mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \boldsymbol{\psi}) \mathbf{W}^{\frac{1}{2}}, \quad (15.88)$$

and noticing that

$$\mathbf{v}_\ell = \frac{1}{\sqrt{w_\ell}} \tilde{\mathbf{v}}_\ell, \quad (15.89)$$

or

$$\phi_\ell(\mathbf{x}_{d,j}; \boldsymbol{\psi}) \approx \frac{1}{\sqrt{w_\ell}} \tilde{\mathbf{v}}_\ell. \quad (15.90)$$

Plugging this into Equation (15.84) and solving for $\phi_\ell(\mathbf{x})$, it is seen that

$$\phi_\ell(\mathbf{x}; \boldsymbol{\psi}) \approx \frac{1}{\lambda_\ell} \sum_{j=1}^{n_d} \sqrt{w_j} \tilde{\mathbf{v}}_{\ell,j} k^*(\mathbf{x}_{d,j}, \mathbf{x}). \quad (15.91)$$

Typically, the KLE is truncated at some $d_\omega \leq n_d$ such that the $\alpha\%$ of the energy of the field is captured, e.g., $\alpha = 90\%$. That is, one may pick d_ω so that:

$$\sum_{\ell=1}^{d_\omega} \lambda_\ell = \alpha \sum_{\ell=1}^{n_d} \lambda_\ell. \quad (15.92)$$

With this choice, the matrix $\mathbf{C}(\boldsymbol{\psi})$, Equation (15.74), is given by:

$$\mathbf{C}(\boldsymbol{\psi}) = \mathbf{W}^{\frac{1}{2}} \tilde{\mathbf{V}} \mathbf{A}^{-\frac{1}{2}}, \quad (15.93)$$

where $\mathbf{A} = \text{diag}(\lambda_1, \dots, \lambda_{d_\omega})$ and $\tilde{\mathbf{V}}$ is the first d_ω eigenvectors of \mathbf{B} of Equation (15.88), i.e.,

$$\tilde{\mathbf{V}} = (\tilde{\mathbf{v}}_1 \dots \tilde{\mathbf{v}}_{d_\omega}), \quad (15.94)$$

is the matrix of the first d_ω eigenvectors as columns.

2.8.2 The O'Hagan Approach for Constructing $\hat{f}(\cdot; \theta)$

Consider the \mathbf{X}_d design points defined in Equation (15.80). Let $\mathbf{Y}_d \in \mathbb{R}^{nd \times 1}$ be the random variable corresponding to the unobserved output of the simulation on the design points \mathbf{X}_d . That is,

$$\mathbf{Y}_d | \mathcal{D}, \psi, \mathbf{X}_d \sim \mathcal{N}_{nd} (\mathbf{Y}_d | \mathbf{m}^*(\mathbf{X}_d; \psi), \mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi)), \quad (15.95)$$

where

$$\mathbf{m}^*(\mathbf{X}_d; \psi) = (m^*(\mathbf{x}_{d,1}; \psi) \dots m^*(\mathbf{x}_{d,n_d}; \psi)), \quad (15.96)$$

with $m^*(\cdot; \psi)$ being the posterior mean given defined in Equation (15.41), and

$$\mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi) = \begin{pmatrix} k^*(\mathbf{x}_{d,1}, \mathbf{x}_{d,1}; \psi) & \dots & k^*(\mathbf{x}_{d,1}, \mathbf{x}_{d,n_d}; \psi) \\ \vdots & \ddots & \vdots \\ k^*(\mathbf{x}_{d,n}, \mathbf{x}_{d,1}; \psi) & \dots & k^*(\mathbf{x}_{d,n}, \mathbf{x}_{d,n}; \psi) \end{pmatrix}, \quad (15.97)$$

with $k^*(\cdot, \cdot; \psi)$ being the posterior covariance defined in Equation (15.42). Finally, let us condition the posterior Gaussian process of Equation (15.40) on the hypothetical observations $\{\mathbf{X}_d, \mathbf{Y}_d\}$. One has:

$$f(\cdot) | \mathcal{D}, \psi, \mathbf{X}_d, \mathbf{Y}_d \sim \text{GP}(f(\cdot) | \mathbf{m}^{**}(\cdot; \psi, \mathbf{Y}_d), k^{**}(\cdot, \cdot; \psi, \mathbf{Y}_d)), \quad (15.98)$$

where the mean is given by:

$$\mathbf{m}^{**}(\mathbf{x}; \psi, \mathbf{Y}_d) = m^*(\mathbf{x}; \psi) + \mathbf{k}^*(\mathbf{x}, \mathbf{X}_d; \psi) \mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi)^{-1} (\mathbf{Y}_d - \mathbf{m}^*(\mathbf{X}_d; \psi)), \quad (15.99)$$

and the covariance is given by:

$$k^{**}(\mathbf{x}, \mathbf{x}'; \psi) = k^*(\mathbf{x}, \mathbf{x}'; \psi) - \mathbf{k}^*(\mathbf{x}, \mathbf{X}_d) \mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi)^{-1} \mathbf{k}^*(\mathbf{X}_d, \mathbf{x}). \quad (15.100)$$

The idea of [38] was that if n_d is sufficiently large, then $k^{**}(\mathbf{x}, \mathbf{x}')$ is very small and, thus, negligible. In other words, if \mathbf{X}_d is dense enough, then all the probability mass of Equation (15.98) is accumulated around the mean $\mathbf{m}^{**}(\cdot; \psi, \mathbf{Y}_d)$. Therefore, one may think of the mean $\mathbf{m}^{**}(\cdot; \psi, \mathbf{Y}_d)$ as a sample surface from Equation (15.40).

To make the connection with Equation (15.74), let $\mathbf{L}^*(\psi)$ be the Cholesky decomposition of the covariance $\mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi)$, Equation (15.100),

$$\mathbf{k}^*(\mathbf{X}_d, \mathbf{X}_d; \psi) = \mathbf{L}^*(\psi) (\mathbf{L}^*(\psi))^T.$$

From Equation (15.95), one may express \mathbf{Y}_d as

$$\mathbf{Y}_d = \mathbf{m}^*(\mathbf{X}_d; \psi) + \mathbf{L}^*(\psi) \boldsymbol{\omega},$$

for a $\boldsymbol{\omega} \sim \mathcal{N}_{d_\omega}(\boldsymbol{\omega} | \mathbf{0}_{n_d}, \mathbf{I}_{n_d})$. Therefore, one may rewrite Equation (15.99) as:

$$m^{**}(\mathbf{x}; \boldsymbol{\psi}; \boldsymbol{\omega}) = m^*(\mathbf{x}; \boldsymbol{\psi}) + \mathbf{k}^*(\mathbf{x}, \mathbf{X}_d; \boldsymbol{\psi}) (\mathbf{L}^*(\boldsymbol{\psi}))^{T,-1} \boldsymbol{\omega}.$$

From this, the matrix $\mathbf{C}(\boldsymbol{\psi})$ of Equation (15.74) should be:

$$\mathbf{C}(\boldsymbol{\psi}) = (\mathbf{L}^*(\boldsymbol{\psi}))^{T,-1}. \quad (15.101)$$

However, since n_d is expected to be quite large, it is not a good idea to use all n_d design points in \mathbf{X}_d to build a functional sample. Apart from the increased computational complexity, the Cholesky of the large covariance matrix of Equation (15.100) introduces numerical instabilities. A heuristic that can be used to construct a subset of design points, $\mathbf{X}_{d,s}$, from the original, dense, set of design points, \mathbf{X}_d , without sacrificing accuracy, is discussed. The idea is to iteratively select the points of \mathbf{X}_d with maximum variance, until the maximum variance falls below a threshold $\epsilon > 0$. The algorithm is as follows:

1. Start with

$$\mathbf{X}_{d,s} = \{\}.$$

2. If

$$|\mathbf{X}_{d,s}| = n_d,$$

then STOP. Otherwise, CONTINUE.

3. Find

$$i^* = \arg \max_{1 \leq j \leq n_d} k^{**}(\mathbf{x}_{d,i}, \mathbf{x}_{d,i}; \boldsymbol{\psi}, \mathbf{X}_{d,s}),$$

where $k^{**}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}, \mathbf{X}_{d,s})$ is the covariance function defined in Equation (15.100) if $\mathbf{X}_{d,s}$ is used instead of \mathbf{X}_d .

4. If

$$k^{**}(\mathbf{x}_{d,i^*}, \mathbf{x}_{d,i^*}; \boldsymbol{\psi}, \mathbf{X}_{d,s}) > \epsilon,$$

then

$$\mathbf{X}_{d,s} \leftarrow \mathbf{X}_{d,s} \cup \{\mathbf{x}_{d,i^*}\},$$

and GO TO 2. Otherwise, STOP.

Notice that when one includes a new point \mathbf{x}_{d,i^*} , he has to compute the Cholesky decomposition of the covariance matrix $\mathbf{k}^*(\mathbf{X}_{d,s} \cup \{\mathbf{x}_{d,i^*}\}, \mathbf{X}_{d,s} \cup \{\mathbf{x}_{d,i^*}\}; \boldsymbol{\psi})$. This can be done efficiently using rank-one updates of the covariance matrix (see Seeger [43]).

2.9 Semi-analytic Formulas for the Mean and the Variance

It is quite obvious how Equation (15.74) can be used to obtain samples from the predictive distribution, $p(\mathcal{Q}|\mathcal{D})$, of a statistic of interest $\mathcal{Q}[\cdot]$. Thus, using a Monte Carlo procedure, one can characterize one's uncertainty about any statistic of the response surface. This could become computationally expensive in the case of high dimensions and many observations, albeit less expensive than evaluating these statistics using the simulator itself. Fortunately, as shown in this section, it is actually possible to evaluate exactly the predictive distribution for the mean statistic, Equation (15.11), since it turns out to be Gaussian. Furthermore, it is possible to derive the predictive mean and variance of the covariance statistic Equation (15.12). In this subsection, it is shown that the predictive distribution for the mean statistic Equation (15.11) is actually Gaussian.

2.9.1 One-Dimensional Output with No Spatial or Time Inputs

Assume a one-dimensional output $d_y = 1$ and that there are no spatial or time inputs, i.e., $d_s = d_t = 0$. In this case, one has $\mathbf{x} = \xi$, $\mathbf{X}_d = \Xi_d$, and he may simply rewrite Equation (15.74) as:

$$\hat{f}(\xi; \theta) = m^*(\xi; \psi) + \mathbf{k}^*(\xi, \Xi_d; \psi) \mathbf{C}(\psi) \omega, \quad (15.102)$$

where θ , ψ , and ω are as before. Taking the expectation of this over $p(\xi)$, one obtains:

$$\mathbb{E}_{\xi}[\hat{f}(\cdot; \theta)] = \mathbb{E}_{\xi}[m^*(\cdot; \psi)] + \mathbb{E}_{\xi}[\mathbf{k}^*(\xi, \Xi_d; \psi)] \mathbf{C}(\psi) \omega. \quad (15.103)$$

Since ω is a standard normal random variable (see Equation (15.78)), it can be integrated out from Equation (15.103) to give:

$$\mathbb{E}_{\xi}[f(\cdot)|\mathcal{D}, \psi] \sim \mathcal{N}\left(\mathbb{E}_{\xi}[f(\cdot)|\mu_{\mu_f}(\psi), \sigma_{\mu_f}^2(\psi)], \right) \quad (15.104)$$

where

$$\mu_{\mu_f}(\psi) := \mathbb{E}_{\xi}[m^*(\cdot; \psi)], \quad (15.105)$$

and

$$\sigma_{\mu_f}^2(\psi) := \| \mathbf{C}^T(\psi) \epsilon^*(\Xi_d; \psi) \|^2, \quad (15.106)$$

with

$$\epsilon^*(\Xi_d; \psi) := \mathbb{E}_{\xi}[\mathbf{k}^*(\Xi_d, \cdot; \psi)]. \quad (15.107)$$

All these quantities are expressible in terms of expectations of the covariance function with respect to $p(\xi)$:

$$\epsilon(\Xi'; \psi_k) := \mathbb{E}_\xi [\mathbf{k}(\Xi', \cdot; \psi_k)]. \quad (15.108)$$

Indeed, from Equation (15.41) one gets:

$$\mu_{\mu_f}(\psi) = \mathbb{E}_\xi [m(\cdot; \psi_m)] + \epsilon(\Xi; \psi_k)^T \mathbf{k}(\Xi, \Xi; \psi_k)^{-1} (\mathbf{Y} - \mathbf{m}(\Xi; \psi_m)), \quad (15.109)$$

and from Equation (15.42)

$$\epsilon^*(\Xi_d; \psi) = \epsilon(\Xi_d; \psi_k) - \mathbf{k}(\Xi_d, \Xi; \psi_k) \mathbf{k}(\Xi, \Xi; \psi_k)^{-1} \epsilon(\Xi; \psi_k). \quad (15.110)$$

For the case of a SE covariance (see Equation (15.37)) combined with a Gaussian or uniform distribution $p(\xi)$, [34] and [7], respectively, show that Equation (15.108) can be computed analytically. As shown in [8], for the case of an arbitrary separable covariance as well as arbitrary independent random variables ξ , Equation (15.108) can be computed efficiently by doing d_y one-dimensional integrals.

2.9.2 One-Dimensional Output with Spatial and/or Time Inputs

Consider the case of one-dimensional output, i.e., $d_y = 1$, with possible spatial and/or time inputs, i.e., $d_s, d_t \geq 0$. In this generic case, $\mathbf{x} = (\mathbf{x}_s, t, \xi)$.

It is possible to use the particular form of Equation (15.74) to derive semi-analytic formulas for some of the statistics. Let us start by considering the mean statistic. One has

$$\mathbb{E}_\xi [\hat{f}(\cdot; \theta)](\mathbf{x}_s, t) = \mathbb{E}_\xi [m^*(\mathbf{x}_s, t, \xi; \psi)] + \mathbb{E}_\xi [\mathbf{k}^*((\mathbf{x}_s, t, \xi), \mathbf{X}_d; \psi)] \mathbf{C}(\psi) \boldsymbol{\omega}. \quad (15.111)$$

In other words, it has been shown that if $f(\cdot)$ is a Gaussian process, then its mean $\mathbb{E}_\xi [f(\cdot)](\cdot)$ is a Gaussian process:

$$\mathbb{E}_\xi [f(\cdot)](\cdot) | \mathcal{D}, \psi \sim \text{GP}(\mathbb{E}_\xi [f(\cdot)](\cdot) | m_{\text{mean}}(\cdot; \psi), k_{\text{mean}}(\cdot, \cdot; \psi)), \quad (15.112)$$

with mean function:

$$m_{\text{mean}}(\mathbf{x}_s, t) = \mathbb{E}_\xi [m^*(\mathbf{x}_s, t, \xi; \psi)], \quad (15.113)$$

and covariance function:

$$\begin{aligned} k_{\text{mean}}((\mathbf{x}_s, t), (\mathbf{x}'_s, t')) \\ = \mathbb{E}_\xi [\mathbf{k}^*((\mathbf{x}_s, t, \xi), \mathbf{X}_d; \psi)] \mathbf{C}(\psi) \mathbf{C}(\psi)^T \mathbb{E}_\xi [\mathbf{k}^*((\mathbf{x}'_s, t', \xi), \mathbf{X}_d; \psi)]^T. \end{aligned} \quad (15.114)$$

Note, that if the stochastic variables in ξ are independent and the covariance function $k(\mathbf{x}, \mathbf{x}'; \psi)$ is separable with respect to the ξ_i 's, then all these quantities can be computed efficiently with numerical integration.

Equations similar to Equation (15.111) can be derived without difficulty for the covariance statistic of Equation (15.12) as well as for the variance statistic of Equation (15.13) [10, 14]. In contrast to the mean statistic, however, the resulting random field is not Gaussian. That is, an equation similar to Equation (15.112) does not hold.

3 Numerical Examples

3.1 Synthetic One-Dimensional Example

In this synthetic example, the ability of the Bayesian approach to characterize one's state of knowledge about the statistics with a very limited number of observations is demonstrated. To keep things simple, start with no space/time inputs ($d_s = d_t = 0$), one stochastic variable $d_\xi = 0$, and one output $d_y = 0$. That is, the input is just $x = \xi$. Consider $n = 7$ arbitrary observations, $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$, which are shown as crosses in Fig. 15.1a. The goal is to use these seven observations to learn the underlying response function $y = f(x)$ and characterize one's state of knowledge about the mean $\mathbb{E}[f(\cdot)]$ (Equation (15.11)), the variance $\mathbb{V}[f(\cdot)]$ (Equation (15.13)), and the induced probability density function in the response y :

$$p(y) = \int \delta(y - f(x)) p(x) dx,$$

where $p(x)$ is the input probability density, taken to be a Beta(10, 2) and shown in Fig. 15.1b.

The first step is to assign a prior Gaussian process to the response (Equation (15.18)). This is done by picking a zero mean and an SE covariance function (Equation (15.37)) with no nugget, $\sigma^2 = 0$ in Equation (15.35), and fixed signal and length-scale parameters to $s = 1$ and $\ell = 0.1$, respectively. These choices represent one's prior beliefs about the underlying response function $y = f(x)$.

Given the observations in \mathcal{D} , the updated state of knowledge is characterized by the posterior GP of Equation (15.40). The posterior mean function, $m^*(\cdot)$ of Equation (15.41), is the dashed blue line of Fig. 15.1a. The shaded gray area of the same figure corresponds to a 95% predictive interval about the mean. This interval is computed using the posterior covariance function, $k^*(\cdot, \cdot)$ of Equation (15.42). Specifically, the point predictive distribution at x is

$$p(y|x, \mathcal{D}) \sim \mathcal{N}\left(m^*(x), (\sigma^*(x))^2\right),$$

where $\sigma^*(x) = \sqrt{k^*(x, x)}$ and, thus, the 95% predictive interval at x is given, approximately, by $(m^*(x) - 1.96\sigma^*(x), m^*(x) + 1.96\sigma^*(x))$. The posterior mean can be thought of as a point estimation of the underlying response surface.

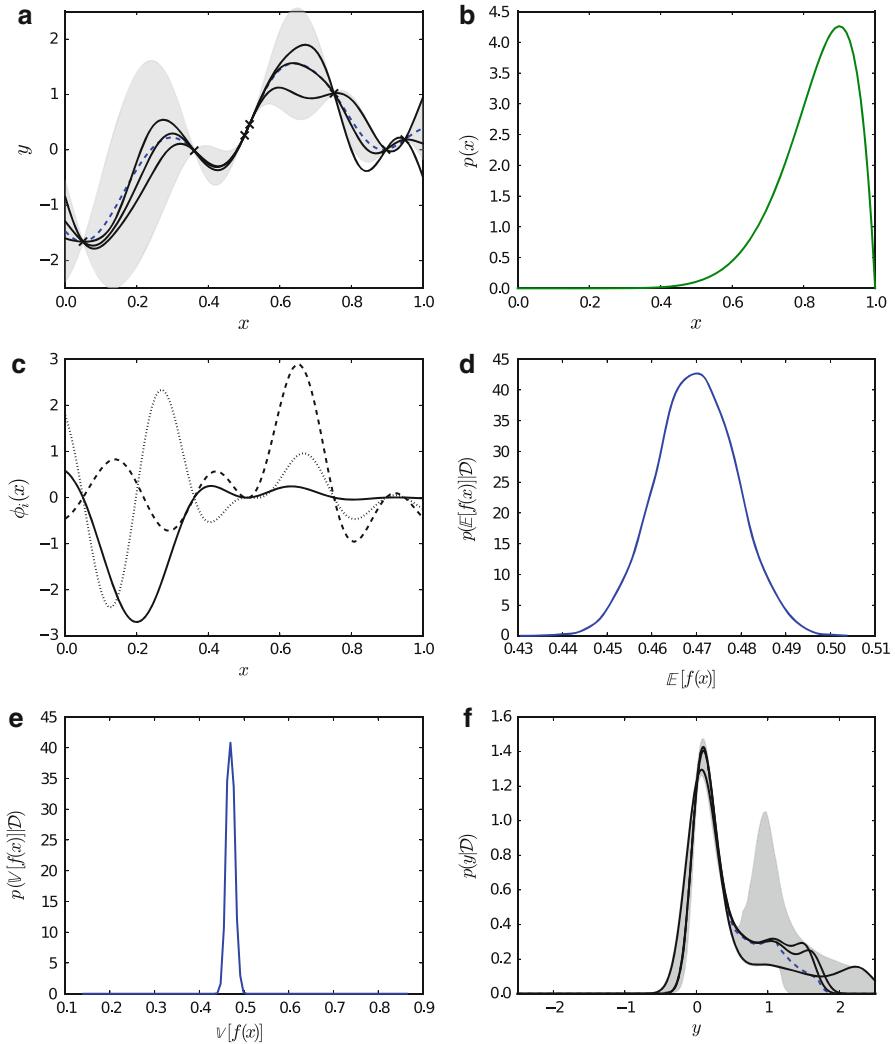


Fig. 15.1 Synthetic: Subfigure (a) shows the observed data (cross symbols), the mean (dashed blue line), the 95% predictive intervals (shaded gray area), and three samples (solid black lines) from the posterior Gaussian process conditioned on the observed data. The green line of subfigure (b) shows the probability density function imposed on the input x . The three lines in subfigure (c) correspond to the first three eigenfunctions used in the KLE of the posterior GP. Subfigures (d) and (e) depict the predictive distribution conditioned on the observations of the mean and the variance statistic of $f(x)$, respectively. Subfigure (f) shows the mean predicted probability density of $y = f(x)$ (blue dashed line) with 95% predictive intervals (shaded gray area), and three samples (solid black lines) from the posterior predictive probability measure on the space of probability densities

In order to sample possible surrogates from Equation (15.40), the Karhunen-Loëve approach for constructing $\hat{f}(\cdot; \theta)$ is followed (see Equations (15.74) and (15.83)), retaining $d_\omega = 3$ eigenfunctions (see Equations (15.81) and (15.91)) of the posterior covariance which account for more than $\alpha = 90\%$ of the energy of the posterior GP (see Equation (15.92)). These eigenfunctions are shown in Fig. 15.1c. Using the constructed $\hat{f}(\cdot; \theta)$, one can sample candidate surrogates. Three such samples are shown as solid black lines in Fig. 15.1a.

Having constructed a finite dimensional representation of the posterior GP, one is in a position to characterize one's state of knowledge about arbitrary statistics of the response, which is captured by Equation (15.17). Here the suggested two-step procedure is followed. That is, candidate surrogates are repeatedly sampled and then the statistic of interest are computed for each sample. In the results presented, 1,000 sampled candidate surrogates are used. Figure 15.1d shows the predictive probability density for the mean of the response $p(\mathbb{E}[f(\cdot)]|\mathcal{D})$. Note that this result can also be obtained semi-analytically using Equation (15.104). Figure 15.1e shows the predictive probability density for the variance of the response $p(\mathbb{V}[f(\cdot)]|\mathcal{D})$, which cannot be approximated analytically. Finally, subfigure (f) of the same figure characterizes the predictive distribution of the PDF of the response $p(y)$. Specifically, the blue dashed line corresponds to the median of the PDFs of each one of the 1,000 sampled candidate surrogates, while the gray shaded area corresponds to a 95% predictive interval around the median. The solid black lines of the same figure are the PDFs of three arbitrary sampled candidate surrogates.

3.2 Dynamical System Example

In this example, the Bayesian approach to uncertainty propagation is applied to a dynamical system with random initial conditions. In particular, the dynamical system [49]:

$$\begin{aligned}\frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2,\end{aligned}$$

subject to random uncertain conditions at $t = 0$:

$$y_1(0) = 1, \quad y_2(0) = 0.1\xi_1, \quad y_3(0) = \xi_2,$$

where

$$\xi_i \sim \mathcal{U}([-1, 1]), \quad i = 1, 2,$$

is considered. To make the connection with the notation of this chapter, note that $d_s = 0$, $d_t = 1$, $d_\xi = 2$, and $d_y = 3$. For each choice of ξ , the computer

emulator, $\mathbf{f}_c(\xi)$ of Equation (15.7), reports the response at $n_t = 20$ equidistant time steps in $[0, 10]$, \mathbf{X}_t of Equation (15.5). The result of n_ξ randomly picked simulations is observed, and one wants to characterize his state of knowledge about the statistics of the response. Consider the case of $n_\xi = 70, 100$, and 150 . Note that propagating uncertainty through this dynamical system is not trivial since there exists a discontinuity in the response surface as ξ_1 crosses zero.

The prior GP is picked to be a multi-output GP with linearly correlated outputs, Equation (15.67), with a constant mean function, $h(t, \xi) = 1$, and a separable covariance function, Equation (15.46), with both the time and stochastic covariance functions being SE, Equation (15.37), with nuggets, Equation (15.35). Denote the hyper-parameters of the time and stochastic part of the covariance by $\psi_t = \{\ell_t, \sigma_t\}$ and $\psi_\xi = \{\ell_{\xi,1}, \ell_{\xi,2}, \sigma_\xi\}$, respectively. An exponential prior is assigned to all of them, albeit with different rate parameters. Specifically, the rate of ℓ_t is 2, the rate of $\ell_{\xi,i}, i = 1, 2$ is 20, and the rate of the nuggets σ_t and σ_ξ is 10^6 . This assignment corresponds to the vague prior knowledge that the a priori mean of the time scale is about 0.5 of the time unit, the scale of ξ is about 0.05 of its unit, and the nuggets expected to be around 10^{-6} . According to the comment below Equation (15.70), the signal strength can be picked to be identically equal to one since it is absorbed by the covariance matrix Σ . For the hyper-parameters of the mean function, i.e., the constant number, a flat uninformative prior is assigned. As already discussed, with this choice it is possible to integrate it out of the model analytically.

The model is trained by sampling the posterior of $\psi = \{\psi_t, \psi_\xi\}$ (see Equation (15.43)) using a mixed MCMC-Gibbs scheme (see [10] for a discussion on the scheme and evidence of convergence). After the MCMC chain sufficiently mixed (it takes about 500 iterations), a particle approximation of the posterior state of knowledge about the response surface is constructed. This is done this as follows. For every 100-th step of the MCMC chain (the intermediate 99 steps are dropped to reduce the correlations), 100 candidate surrogates are drawn using the O'Hagan procedure with a tolerance of $\epsilon = 10^{-2}$.

In all plots, the blue solid lines and the shaded gray areas depict the predictive mean and 95% intervals of the corresponding statistics, respectively. The prediction about the time evolution of the mean response, $p(\mathbb{E}[y_i(t)] | \mathcal{D}), i = 1, 3$ for the case of $n_\xi = 100$ observations is shown in the first row of Fig. 15.2. Note that there is very little residual epistemic uncertainty for this prediction. The time evolution of the variance of the response, $p(\mathbb{V}[y_i(t)] | \mathcal{D})$, is shown on the second and third rows of the same figure for $n_\xi = 100$ and $n_\xi = 150$, respectively. Notice how the width of the predictive interval decreases with increasing n_ξ . In Fig. 15.3, the time evolution of the probability density of $y_2(t)$ is summarized. Specifically, the four rows correspond to four different time instants, $t = 4, 6, 8$, and 10 , and the columns refer to different sample sizes of $n_\xi = 70, 100$, and 150 , counting from the left.

3.3 Partial Differential Equation Example

In this example, it is shown how the Bayesian approach to uncertainty propagation can be applied to a partial differential equation. In particular, a two-dimensional

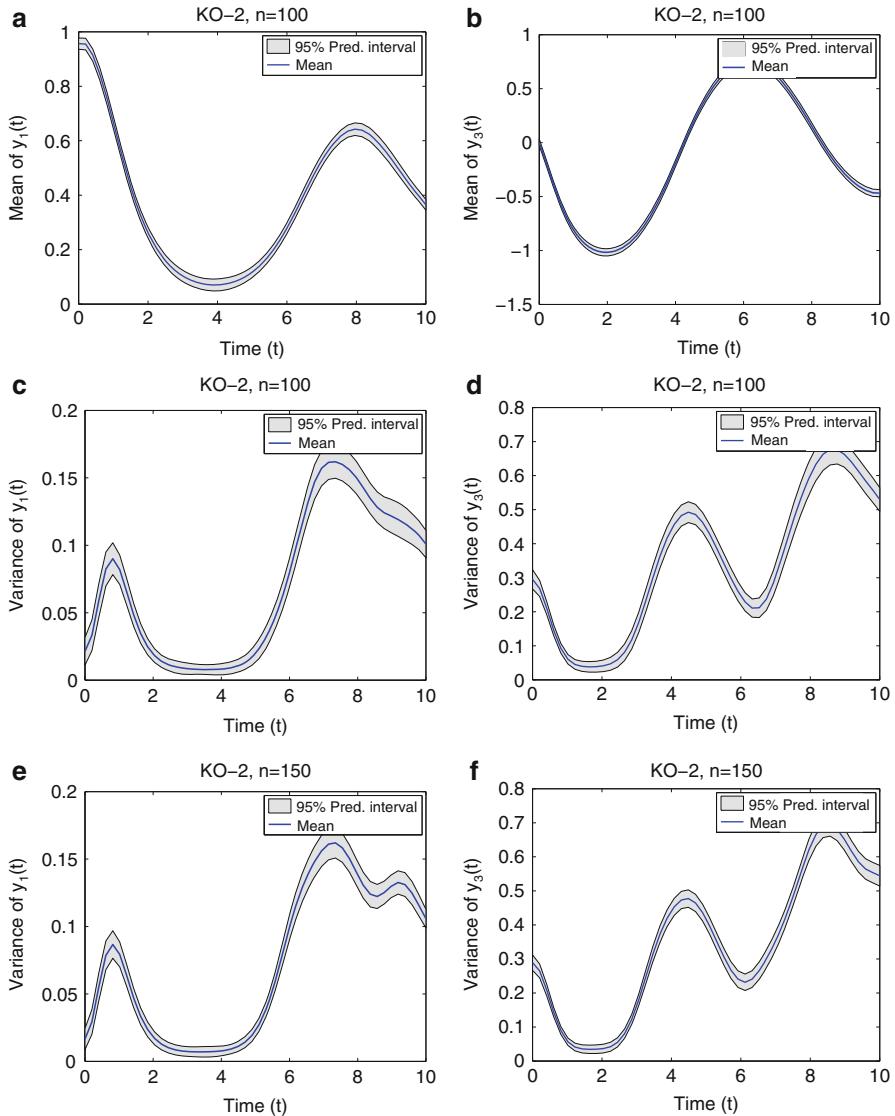


Fig. 15.2 Dynamical system: Subfigures (a) and (b) correspond to the predictions about the mean of $y_1(t)$ and $y_3(t)$, respectively, using $n_\xi = 100$ simulations. Subfigures (c) and (d) (e) and (f) show the predictions about the variance of the same quantities for $n_\xi = 100$ ($n_\xi = 150$) simulations (Reproduced with permission from [10])

($\mathcal{X}_s = [0, 1]^2$ and $d_s = 2$), single-phase, steady-state ($d_t = 0$) flow through an uncertain permeability field is studied; see Aarnes et al. [1] for a review of the underlying physics and solution methodologies. The uncertainty in the permeability is represented by a truncated KLE of an exponentiated Gaussian random field with

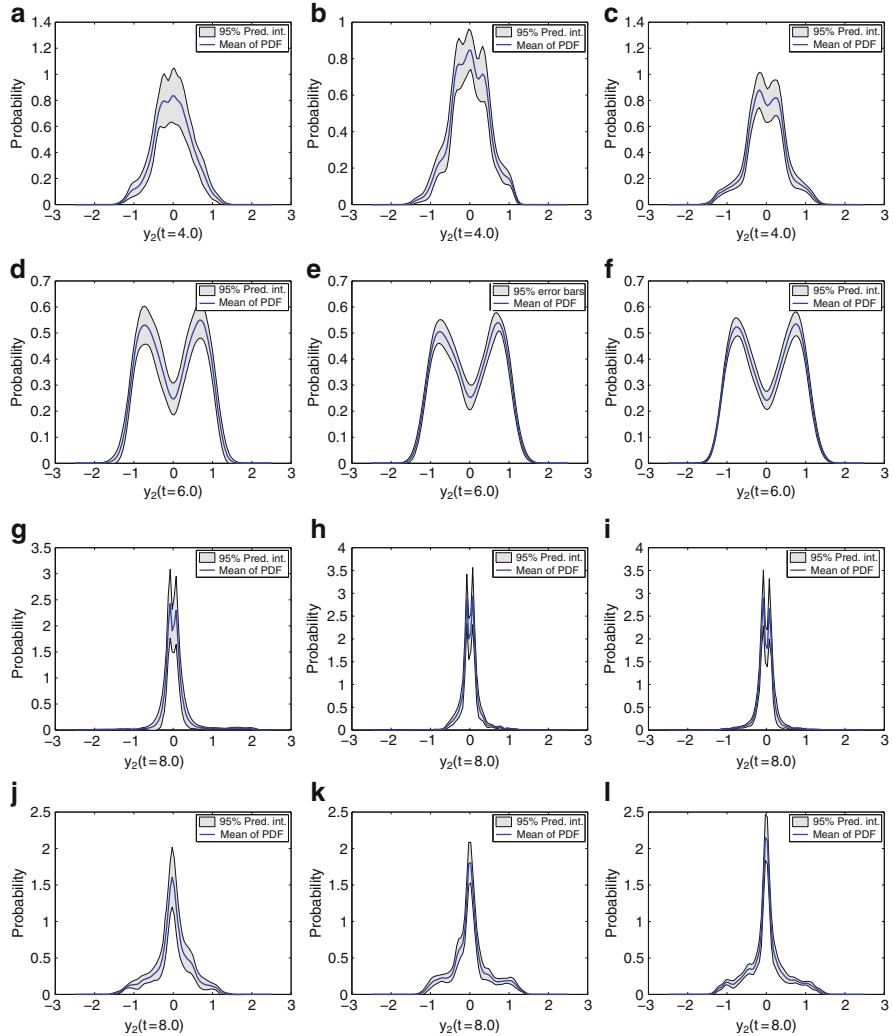


Fig. 15.3 Dynamical system: Columns correspond to results using $n_\xi = 70, 100$, and 150 simulations counting from the left. Counting from the top, rows one, two, three, and four show the predictions about the PDF of $y_2(t)$ at times $t = 4, 6, 8, 10$, respectively (Reproduced with permission from [10])

exponential covariance function of signal strength equal to one and correlation length equal to 0.1 and a zero mean. The total number of stochastic variables corresponds to the truncation order of the KLE, and it is chosen to be $n_\xi = 50$. Three outputs, $d_y = 3$, are considered: the pressure, $p(\mathbf{x}_s; \xi)$, and the horizontal and vertical components of the velocity field, $u(\mathbf{x}_s; \xi)$ and $v(\mathbf{x}_s; \xi)$, respectively. The emulator, $\mathbf{f}_c(\xi)$ of Equation (15.7), is based on the finite element method and is

described in detail in [10], and it reports the response on a regular 32×32 spatial grid, i.e., $n_s = 32^2 = 1,024$. The objective is to quantify the statistics of the response using a limited number of $n_\xi = 24, 64$, and 120 simulations. The results are validated by comparing against a plain vanilla MC estimate of the statistics using 108,000 samples.

As in the previous example, the prior state of knowledge is represented using a multi-output GP with linearly correlated outputs, Equation (15.67); a constant mean function, $h(t, \xi) = 1$; and a separable covariance function, Equation (15.46), with both the space and stochastic covariance functions being SE, Equation (15.37), with nuggets, Equation (15.35). Denote the hyper-parameters of the spatial and stochastic part of the covariance by $\psi_s = \{\ell_{s,1}, \ell_{s,2}, \sigma_s\}$ and $\psi_\xi = \{\ell_{\xi,1}, \dots, \ell_{\xi,50}, \sigma_\xi\}$, respectively. Note that the fact that the spatial component is also separable is exploited to significantly reduce the computational cost of the calculations. Again, exponential priors are assigned. The rate parameters of the spatial length scales are 100 corresponding to an a priori expectation of 0.01 spatial units. The rates of $\ell_{xi,i}, \sigma_s$, and σ_ξ are 3, 100, and 100, respectively.

The posterior of the hyper-parameters, Equation (15.43), is sampled using 100,000 iterations of the same MCMC-Gibbs procedure as in the previous example. However, in order to reduce the computational burden, a single-particle MAP approximation to the posterior, Equation (15.59), is constructed, by searching for the MAP over the 100,000 MCMC-Gibbs samples collected. Then, 100 candidate surrogate surfaces are sampled following the O'Hagan procedure with a tolerance of $\epsilon = 10^{-2}$. For each sampled surrogate, the statistics of interest are calculated and compared to MC estimates.

In Fig. 15.4 the mean prediction is compared to the mean of the horizontal component of the velocity, $u(\mathbf{x}_s; \xi)$ as a function of the spatial coordinates, $\mathbb{E}_\xi[u(\mathbf{x}_s; \xi)]$, conditioned on $n_\xi = 24, 64$, and 120 simulations, subfigures (a), (b), and (c), respectively, to the MC estimate, subfigure (e). The error bars shown in subfigure (d) of the same figure correspond to two standard deviations of the predictive $p(\mathbb{E}_\xi[u(\mathbf{x}_s; \xi)] | \mathcal{D})$ for the case of $n_\xi = 120$ simulations. Figures 15.5 and 15.6 report the same statistic for the y-component of the velocity, $v(\mathbf{x}_s; \xi)$, and the pressure, $p(\mathbf{x}_s; \xi)$, respectively. Similarly, in Figs. 15.7, 15.8, and 15.9, the predictive distributions of the variances of the horizontal component of the velocity, $p(\mathbb{V}[u(\mathbf{x}_s; \xi_s)] | \mathcal{D})$; the vertical component of the velocity, $p(\mathbb{V}[v(\mathbf{x}_s; \xi_s)] | \mathcal{D})$; and the pressure, $p(\mathbb{V}[p(\mathbf{x}_s; \xi_s)] | \mathcal{D})$, respectively, are characterized. Even though one observes an underestimation of the variance, which is more pronounced for the limited simulation cases, the truth is well covered by the predicted error bars.

In Fig. 15.10 the solid blue line and the shaded gray area correspond to the mean of the predictive probability density of the PDF of the horizontal component of the velocity $u(\mathbf{x}_s = (0.5, 0.5); \xi)$, respectively, conditioned on 24 (a), 64 (b), and 120 (c) simulations, and compares it to an MC estimate (d). Notice that the ground truth, i.e., the MC estimate, always falls within the shaded areas. Finally, Figs. 15.11, 15.12, and 15.13 show the predictive distribution of the PDF of $u(\mathbf{x}_s = (0.25, 0.25); \xi)$ and $p(\mathbf{x}_s = (0.25, 0.25); \xi)$, respectively.

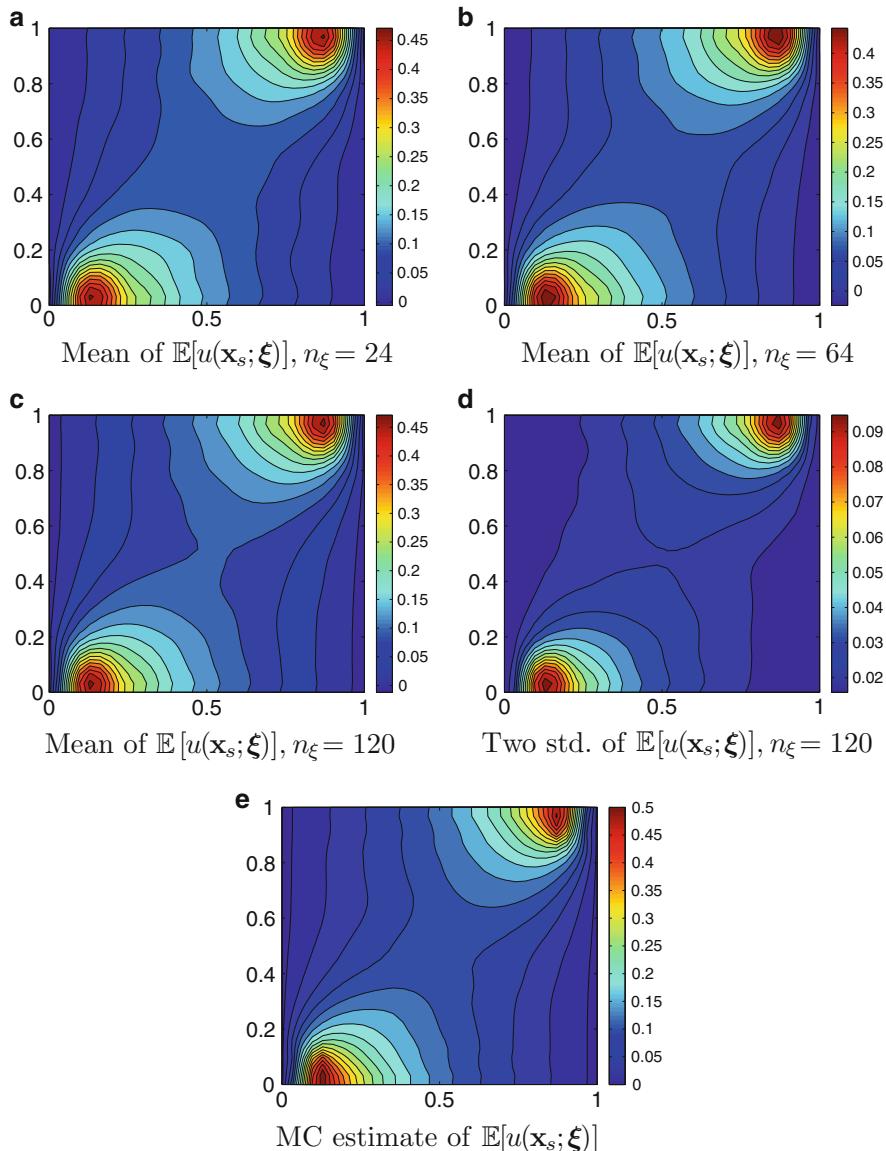


Fig. 15.4 Partial differential equation: Mean of $\mathbb{E}[u(\mathbf{x}_s; \xi)]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{E}[u(\mathbf{x}_s; \xi)]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations $\mathbb{E}[u(\mathbf{x}_s; \xi)]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

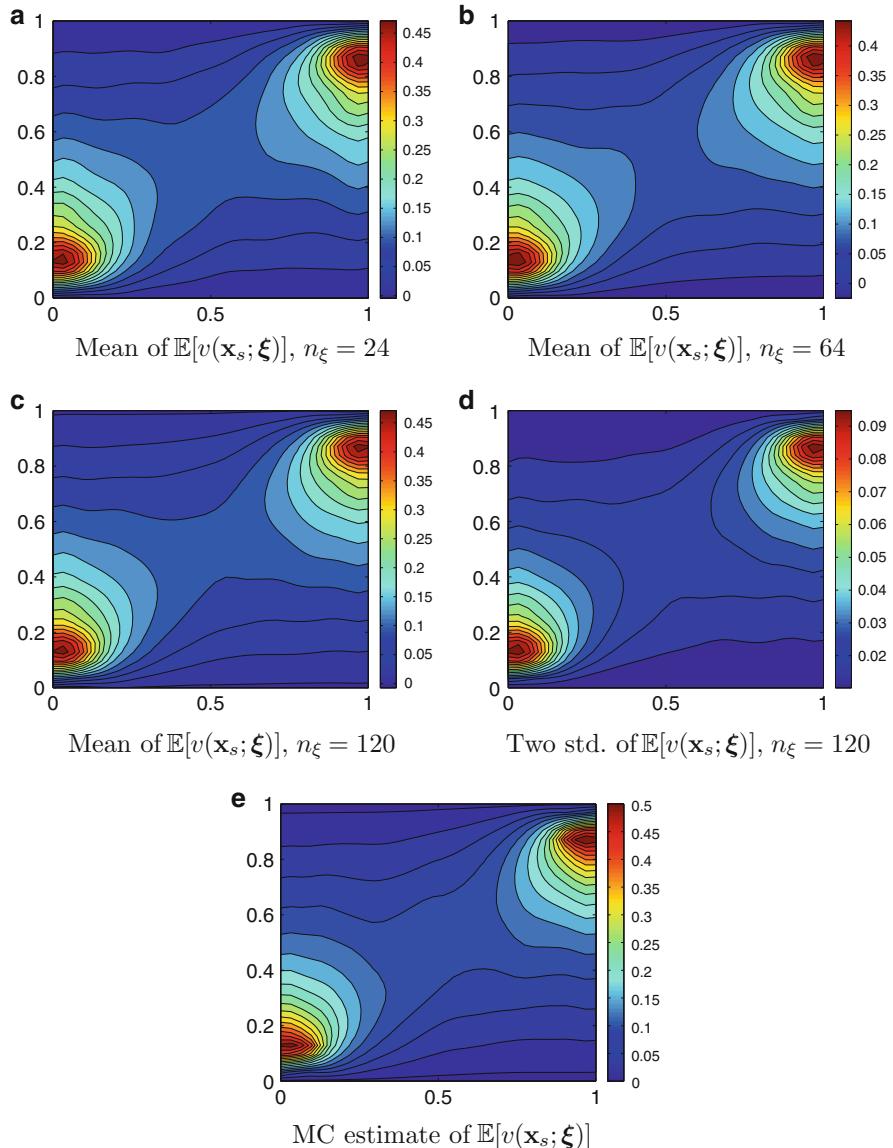


Fig. 15.5 Partial differential equation: Mean of $\mathbb{E}[v(\mathbf{x}_s; \boldsymbol{\xi})]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{E}[v(\mathbf{x}_s; \boldsymbol{\xi})]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations $\mathbb{E}[v(\mathbf{x}_s; \boldsymbol{\xi})]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

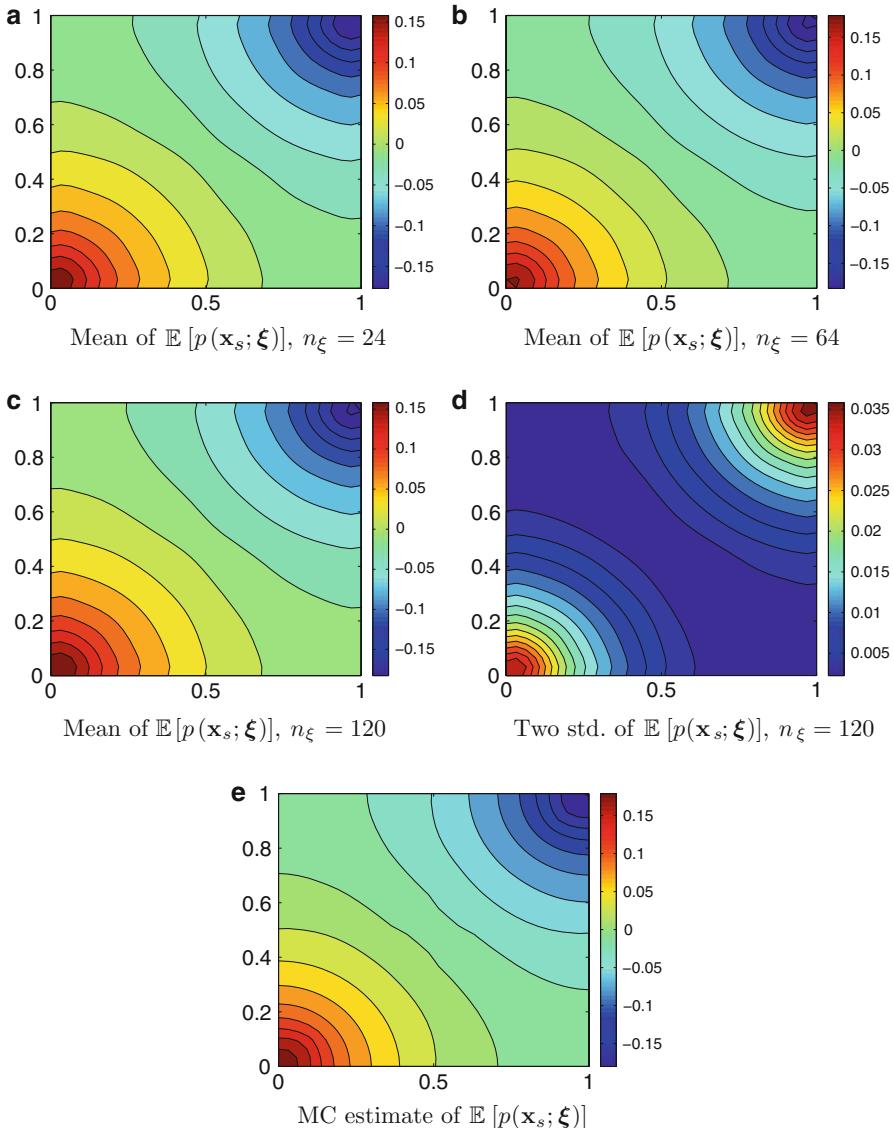


Fig. 15.6 Partial differential equation: Mean of $\mathbb{E}[p(\mathbf{x}_s; \boldsymbol{\xi})]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{E}[p(\mathbf{x}_s; \boldsymbol{\xi})]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations of $\mathbb{E}[p(\mathbf{x}_s; \boldsymbol{\xi})]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

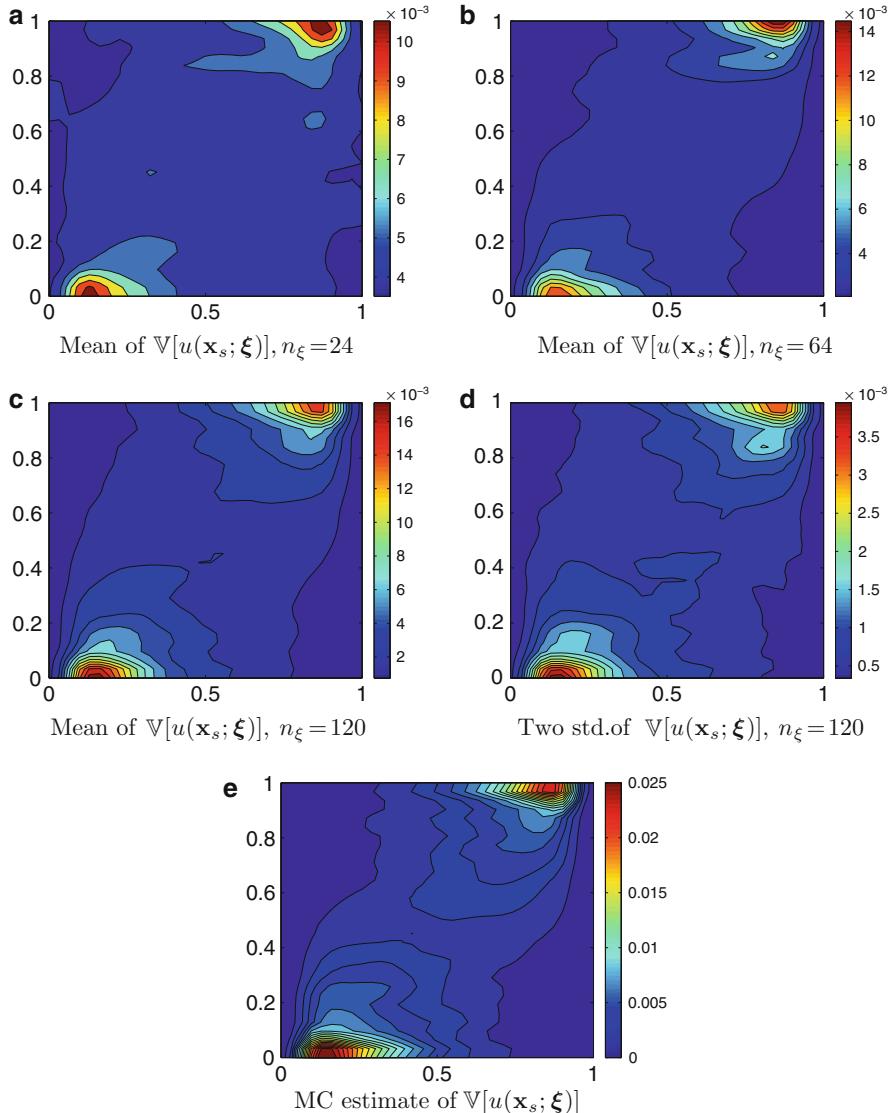


Fig. 15.7 Partial differential equation: Mean of $\mathbb{V}[u(\mathbf{x}_s; \boldsymbol{\xi})]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{V}[u(\mathbf{x}_s; \boldsymbol{\xi})]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations of $\mathbb{V}[u(\mathbf{x}_s; \boldsymbol{\xi})]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

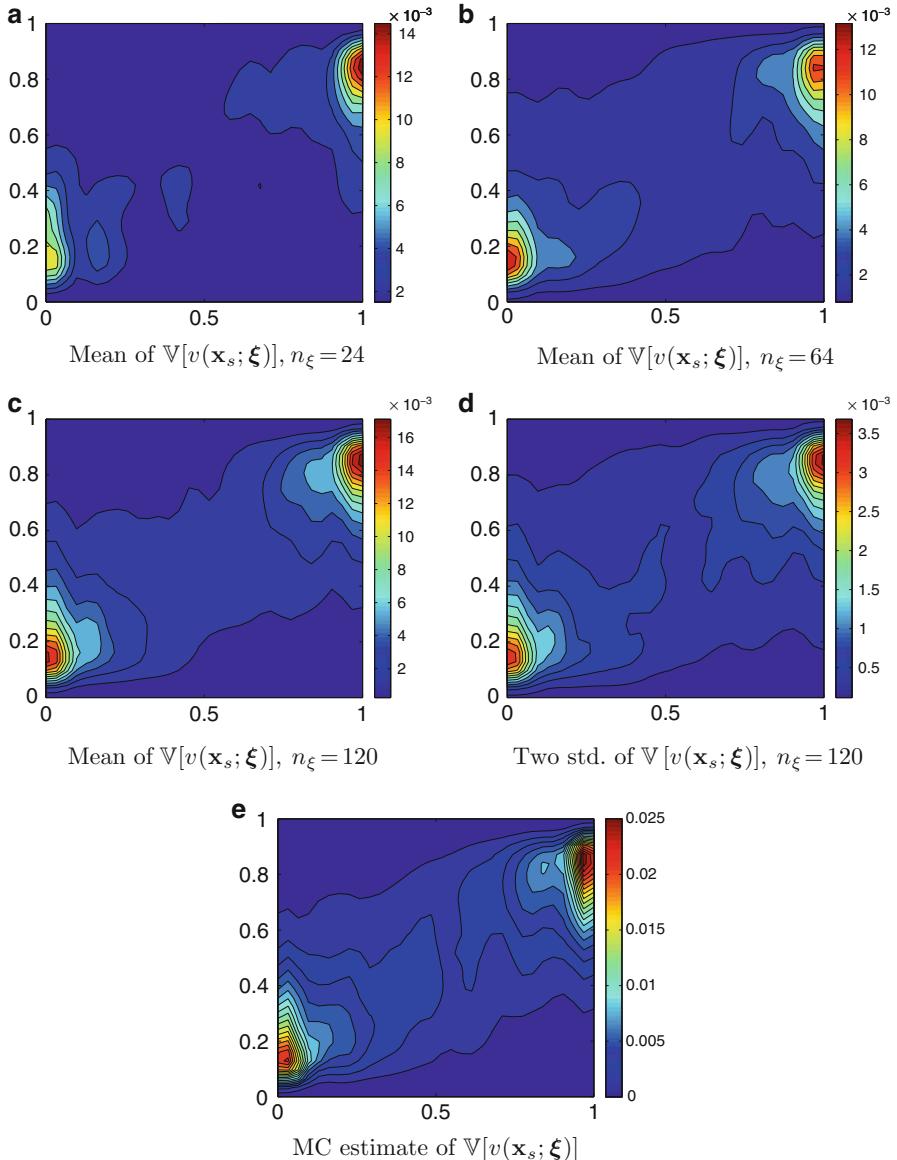


Fig. 15.8 Partial differential equation: Mean of $\mathbb{V}[v(\mathbf{x}_s; \xi)]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{V}[v(\mathbf{x}_s; \xi)]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations of $\mathbb{V}[v(\mathbf{x}_s; \xi)]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

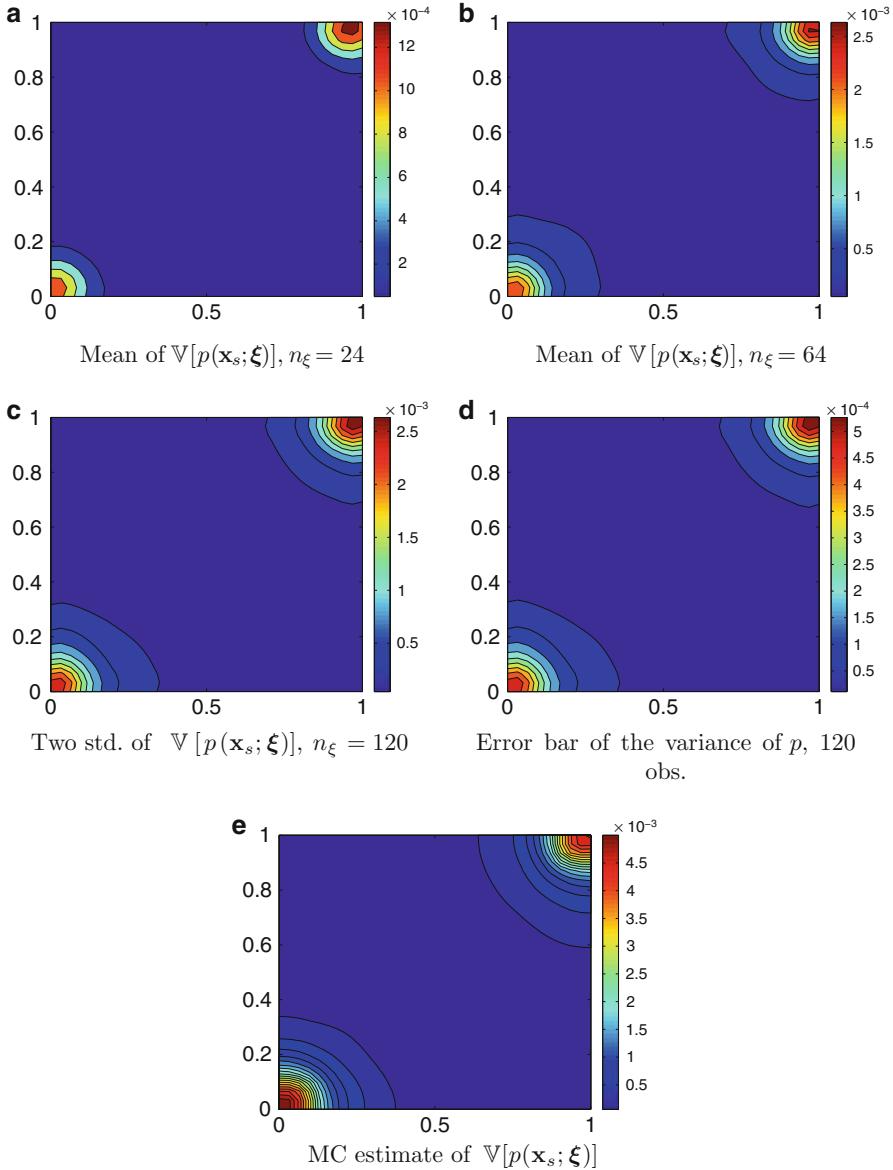


Fig. 15.9 Partial differential equation: Mean of $\mathbb{V}[p(\mathbf{x}_s; \boldsymbol{\xi})]$. Subfigures (a), (b), and (c) show the predictive mean of $\mathbb{V}[p(\mathbf{x}_s; \boldsymbol{\xi})]$ as a function of \mathbf{x}_s conditioned on 24, 64, and 120 simulations, respectively. Subfigure (d) plots two standard deviations of $\mathbb{V}[p(\mathbf{x}_s; \boldsymbol{\xi})]$ conditioned on 120 observations. Subfigure (e) shows the MC estimate of the same quantity (Reproduced with permission from [10])

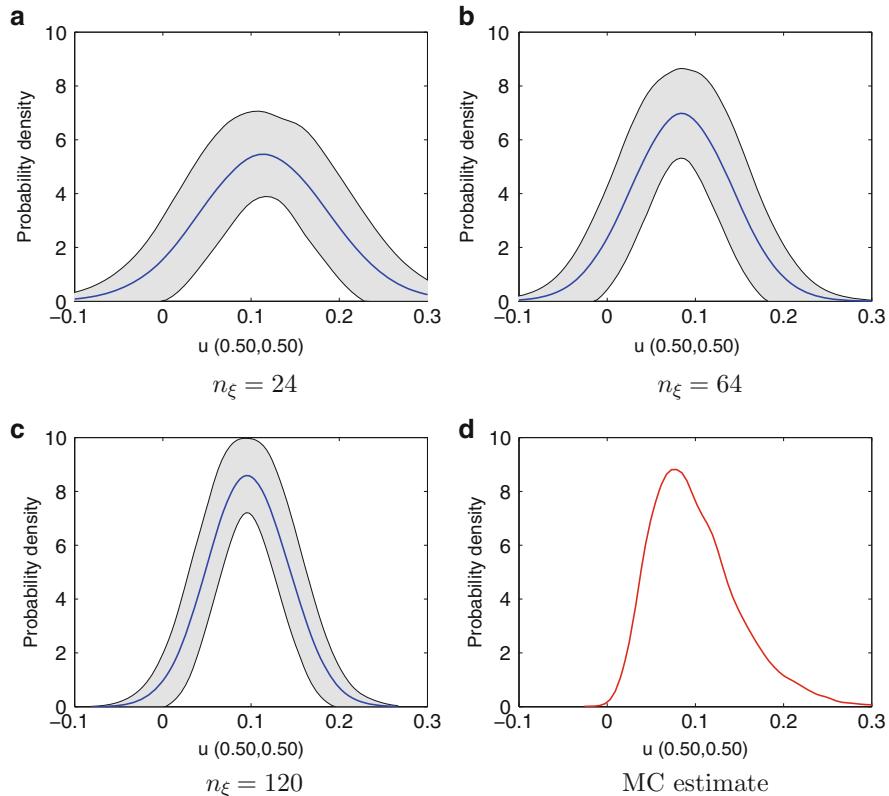


Fig. 15.10 Partial differential equation: The prediction for the PDF of $u(\mathbf{x}_s = (0.5, 0.5); \boldsymbol{\xi})$. The solid blue line shows the mean predictive distribution of the PDF conditioned on 24 (a), 64 (b), and 120 (c) simulations. The filled gray area depicts two standard deviations of the predictive distribution PDFs about the predictive mean of PDF. The solid red line of (d) shows the MC estimate for comparison (Reproduced with permission from [10])

4 Conclusions

In this chapter we presented a comprehensive review of the Bayesian approach to the UP problem that is able to quantify the epistemic uncertainty induced by limited number of simulations. The core idea was to interpret a GP as a probability measure on the space of surrogates which characterizes our prior state of knowledge about the response surface. We focused on practical aspects of GPs such as the treatment of spatiotemporal variation and multi-output responses. We showed how the prior GP can be conditioned on the observed simulations to obtain a posterior GP, whose probability mass corresponds to the epistemic uncertainty introduced by the limited number of simulations, and we introduced sampling-based techniques that allow for its quantification.

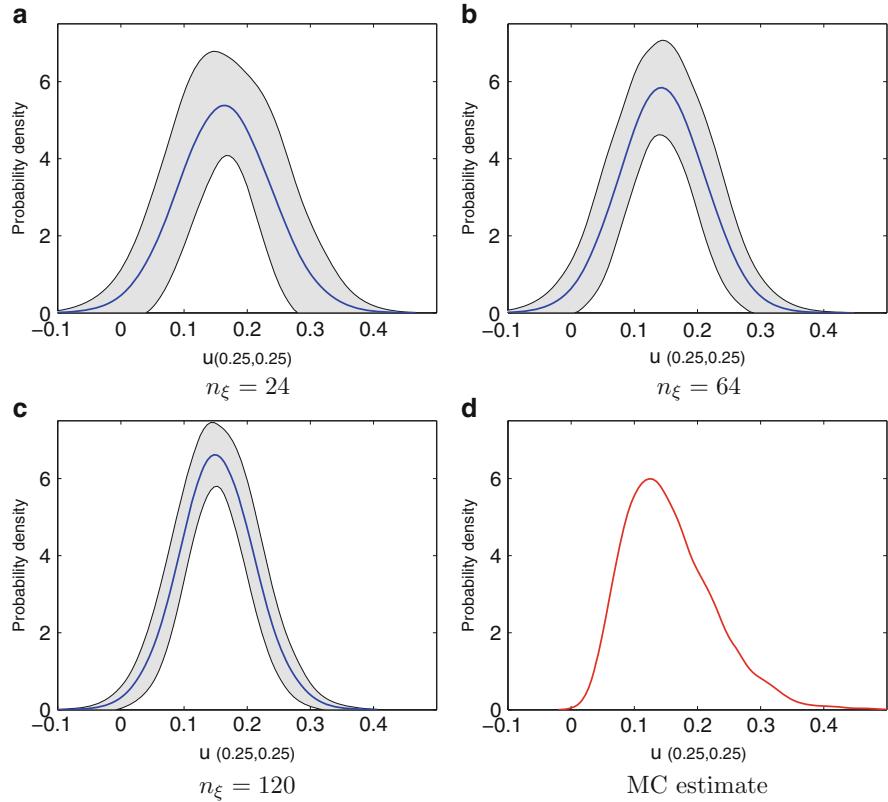


Fig. 15.11 Partial differential equation: The prediction for the PDF of $u(\mathbf{x}_s = (0.25, 0.25); \xi)$. The *solid blue line* shows the mean predictive distribution of the PDF conditioned on 24 (a), 64 (b), and 120 (c) simulations. The *filled gray area* depicts two standard deviations of the predictive distribution PDFs about the predictive mean of PDF. The *solid red line* of (d) shows the MC estimate for comparison (Reproduced with permission from [10])

Despite the successes of the current state of the Bayesian approach to the UP problem, there is still a wealth of open research questions. First, carrying out GP regression in high dimensions is not a trivial problem since it requires the development of application-specific covariance functions. The study of covariance functions that automatically perform some kind of internal dimensionality reduction seems to be a promising step forward. Second, in order to capture sharp variations in the response surface, such as localized bumps or even discontinuities, there is a need for flexible nonstationary covariance functions or alternative approaches based on mixtures of GPs, e.g., see [14]. Third, there is a need for computationally efficient ways of treating nonlinear correlations between distinct model outputs, since this is expected to squeeze more information out of the simulations. Fourth, as a semi-

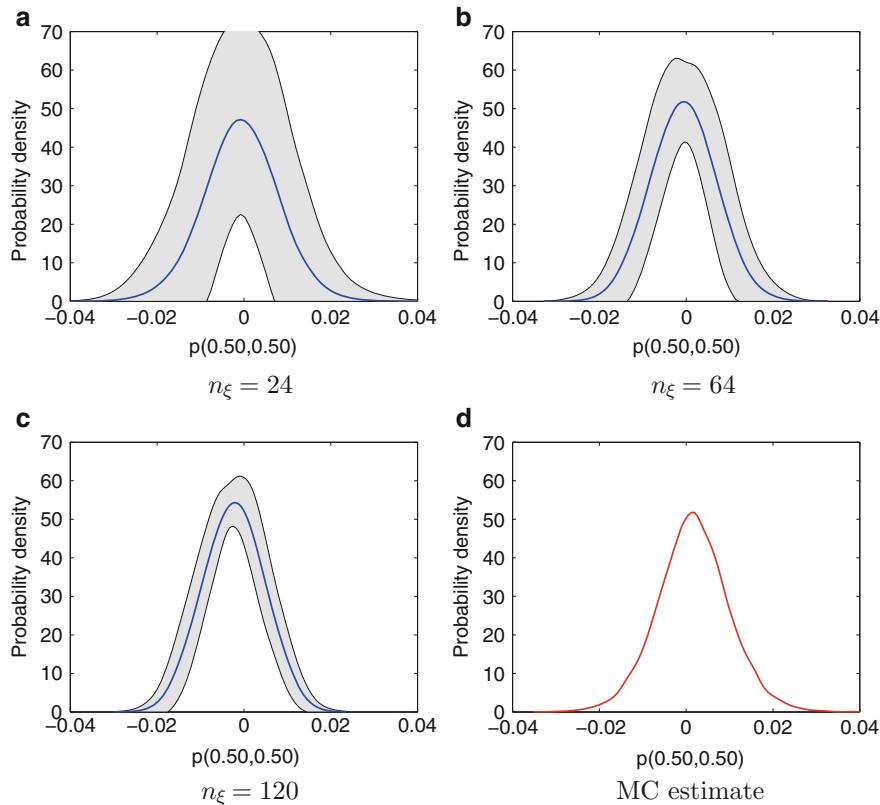


Fig. 15.12 Partial differential equation: The prediction for the PDF of $p(\mathbf{x}_s = (0.5, 0.5); \xi)$. The solid blue line shows the mean predictive distribution of the PDF conditioned on 24 (a), 64 (b), and 120 (c) simulations. The filled gray area depicts two standard deviations of the predictive distribution PDFs about the predictive mean of PDF. The solid red line of (d) shows the MC estimate for comparison (Reproduced with permission from [10])

intrusive approach, the mathematical models describing the physics of the problem could be used to derive physics-constrained covariance functions that would, presumably, force the prior GP probability measure to be compatible with known response properties, such as mass conservation. That is, such an approach would put more effort on better representing our prior state of knowledge about the response. Fifth, there is an evident need for developing simulation selection policies which are specifically designed to gather information about the uncertainty propagation task. Finally, note that the Bayesian approach can also be applied to other important contexts such as model calibration and design optimization under uncertainty. As a result, all the open research questions have the potential to also revolutionize these fields.

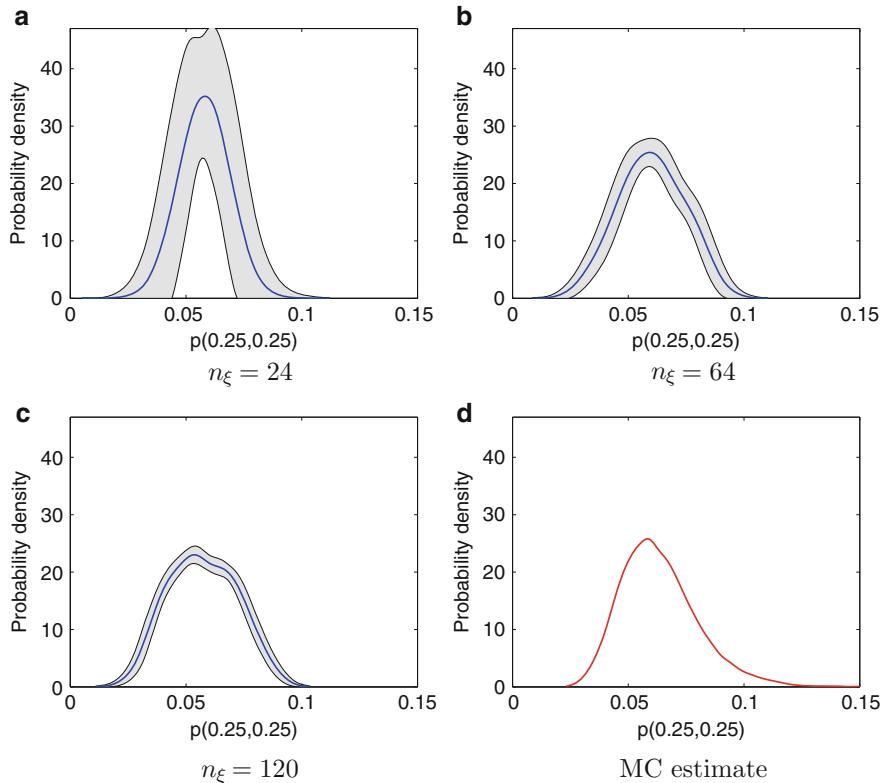


Fig. 15.13 Partial differential equation: The prediction for the PDF of $p(\mathbf{x}_s = (0.25, 0.25); \xi)$. The *solid blue line* shows the mean predictive distribution of the PDF conditioned on 24 (a), 64 (b), and 120 (c) simulations. The *filled gray area* depicts two standard deviations of the predictive distribution PDFs about the predictive mean of PDF. The *solid red line* of (d) shows the MC estimate for comparison (Reproduced with permission from [10])

References

1. Aarnes, J.E., Kippe, V., Lie, K.A., Rustad, A.B.: Modelling of multiscale structures in flow simulations for petroleum reservoirs. In: Hasle, G., Lie, K.A., Quak, E. (eds.): Geometric Modelling, Numerical Simulation, and Optimization, chap. 10, pp. 307–360. Springer, Berlin/Heidelberg (2007). doi:[10.1007/978-3-540-68783-2_10](https://doi.org/10.1007/978-3-540-68783-2_10)
2. Alvarez, M., Lawrence, N.D.: Sparse convolved Gaussian processes for multi-output regression. In: Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.): Advances in Neural Information Processing Systems 21 (NIPS 2008), Vancouver, B.C., Canada (2008)
3. Alvarez, M., Luengo-Garcia, D., Titsias, M., Lawrence, N.: Efficient multioutput Gaussian processes through variational inducing kernels. In: Ft. Lauderdale, FL, USA (2011)
4. Babuska, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM J. Numer. Anal. **45**(3), 1005–1034 (2007)

5. Betz, W., Papaioannou, I., Straub, D.: Numerical methods for the discretization of random fields by means of the Karhunen-Loeve expansion. *Comput. Methods Appl. Mech. Eng.* **271**, 109–129 (2014). doi:[10.1016/j.cma.2013.12.010](https://doi.org/10.1016/j.cma.2013.12.010)
6. Bilionis, I.: py-orthpol: Construct orthogonal polynomials in python. <https://github.com/PredictiveScienceLab/py-orthpol> (2013)
7. Bilionis, I., Zabaras, N.: Multi-output local Gaussian process regression: applications to uncertainty quantification. *J. Comput. Phys.* **231**(17), 5718–5746 (2012) doi:[10.1016/J.Jcp.2012.04.047](https://doi.org/10.1016/J.Jcp.2012.04.047)
8. Bilionis, I., Zabaras, N.: Multidimensional adaptive relevance vector machines for uncertainty quantification. *SIAM J. Sci. Comput.* **34**(6), B881–B908 (2012). doi:[10.1137/120861345](https://doi.org/10.1137/120861345)
9. Bilionis, I., Zabaras, N.: Solution of inverse problems with limited forward solver evaluations: a Bayesian perspective. *Inverse Probl.* **30**(1), Artn 015004 (2014). doi:[10.1088/0266-5611/30/1/015004](https://doi.org/10.1088/0266-5611/30/1/015004)
10. Bilionis, I., Zabaras, N., Konomi, B.A., Lin, G.: Multi-output separable Gaussian process: towards an efficient, fully Bayesian paradigm for uncertainty quantification. *J. Comput. Phys.* **241**, 212–239 (2013). doi:[10.1016/J.Jcp.2013.01.011](https://doi.org/10.1016/J.Jcp.2013.01.011)
11. Bilionis, I., Drewniak, B.A., Constantinescu, E.M.: Crop physiology calibration in the CLM. *Geoscientific Model Dev.* **8**(4), 1071–1083 (2015). doi:[10.5194/gmd-8-1071-2015](https://doi.org/10.5194/gmd-8-1071-2015), <http://www.geosci-model-dev.net/8/1071/2015> <http://www.geosci-model-dev.net/8/1071/2015/gmd-8-1071-2015.pdf>
12. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York (2006)
13. Boyle, P., Frean, M.: Dependent Gaussian processes. In: Saul, L.K., Weiss, Y., and Bottou L. (eds.): Advances in Neural Information Processing Systems 17 (NIPS 2004), Whistler, B.C., Canada (2004)
14. Chen, P., Zabaras, N., Bilionis, I.: Uncertainty propagation using infinite mixture of Gaussian processes and variational Bayesian inference. *J. Comput. Phys.* **284**, 291–333 (2015)
15. Conti, S., O'Hagan, A.: Bayesian emulation of complex multi-output and dynamic computer models. *J. Stat. Plan. Inference* **140**(3), 640–651 (2010). doi:[10.1016/J.Jspi.2009.08.006](https://doi.org/10.1016/J.Jspi.2009.08.006)
16. Currin, C., Mitchell, T., Morris, M., Ylvisaker, D.: A Bayesian approach to the design and analysis of computer experiments. Report, Oak Ridge Laboratory (1988)
17. Currin, C., Mitchell, T., Morris, M., Ylvisaker, D.: Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Stat. Assoc.* **86**(416), 953–963 (1991). doi:[10.2307/2290511](https://doi.org/10.2307/2290511)
18. Dawid, A.P.: Some matrix-variate distribution theory – notational considerations and a Bayesian application. *Biometrika* **68**(1), 265–274 (1981)
19. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B. (Stat. Methodol.)* **68**(3), 411–436 (2006)
20. Delves, L.M., Walsh, J.E., of Manchester Department of Mathematics, U., of Computational LUD, Science, S.: Numerical Solution of Integral Equations. Clarendon Press, Oxford (1974)
21. Doucet, A., De Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science). Springer, New York (2001)
22. Durrande, N., Ginsbourger, D., Roustant, O.: Additive covariance kernels for high-dimensional Gaussian process modeling. arXiv:11116233 (2011)
23. Duvenaud, D., Nickisch, H., Rasmussen, C.E.: Additive Gaussian processes. In: Advances in Neural Information Processing Systems, vol. 24, pp. 226–234 (2011)
24. Gautschi, W.: On generating orthogonal polynomials. *SIAM J. Sci. Stat. Comput.* **3**(3), 289–317 (1982). doi:[10.1137/0903018](https://doi.org/10.1137/0903018)
25. Gautschi, W.: Algorithm-726 – ORTHPOL – a package of routines for generating orthogonal polynomials and Gauss-type quadrature rules. *ACM Trans. Math. Softw.* **20**(1), 21–62 (1994) doi:[10.1145/174603.174605](https://doi.org/10.1145/174603.174605)
26. Ghanem, R., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach, rev. edn. Dover Publications, Mineola (2003)

27. Gramacy, R.B., Lee, H.K.H.: Cases for the nugget in modeling computer experiments. *Stat. Comput.* **22**(3), 713–722 (2012) doi:[10.1007/s11222-010-9224-x](https://doi.org/10.1007/s11222-010-9224-x)
28. Haff, L.: An identity for the Wishart distribution with applications. *J. Multivar. Anal.* **9**(4), 531–544 (1979). doi:[http://dx.doi.org/10.1016/0047-259X\(79\)90056-3](http://dx.doi.org/10.1016/0047-259X(79)90056-3)
29. Hastings, W.K.: Monte-Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970). doi:[10.2307/2334940](https://doi.org/10.2307/2334940)
30. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
31. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics. Springer, New York (2001)
32. Loève, M.: Probability Theory, 4th edn. Graduate Texts in Mathematics. Springer, New York (1977)
33. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953). doi:[10.1063/1.1699114](https://doi.org/10.1063/1.1699114)
34. Oakley, J., O'Hagan, A.: Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika* **89**(4), 769–784 (2002)
35. Oakley, J.E., O'Hagan, A.: Probabilistic sensitivity analysis of complex models: a Bayesian approach. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **66**, 751–769 (2004). doi:[10.1111/j.1467-9868.2004.05304.x](https://doi.org/10.1111/j.1467-9868.2004.05304.x)
36. O'Hagan, A.: Bayes-Hermite quadrature. *J. Stat. Plan. Inference* **29**(3), 245–260 (1991)
37. O'Hagan, A., Kennedy, M.: Gaussian emulation machine for sensitivity analysis (GEM-SA) (2015). <http://www.tonyohagan.co.uk/academic/GEM/>
38. O'Hagan, A., Kennedy, M.C., Oakley, J.E.: Uncertainty analysis and other inference tools for complex computer codes. *Bayesian Stat.* **6**, 503–524 (1999)
39. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
40. Reinhardt, H.J.: Analysis of Approximation Methods for Differential and Integral Equations. Applied Mathematical Sciences. Springer, New York (1985)
41. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer Texts in Statistics. Springer, New York (2004)
42. Sacks, J., Welch, W.J., Mitchell, T., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
43. Seeger, M.: Low rank updates for the Cholesky decomposition. Report, University of California at Berkeley (2007)
44. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Sov. Math. Dokl.* **4**, 240–243 (1963)
45. Stark, H., Woods, J.W., Stark, H.: Probability and Random Processes with Applications to Signal Processing, 3rd edn. Prentice Hall, Upper Saddle River (2002)
46. Stegle, O., Lippert, C., Mooij, J.M., Lawrence, N.D., Borgwardt, K.M.: Efficient inference in matrix-variate Gaussian models with *backslash* iid observation noise. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger K.Q. (eds.): Advances in Neural Information Processing Systems 24 (NIPS 2011), Granada, Spain (2011)
47. Van Loan, C.F.: The ubiquitous Kronecker product. *J. Comput. Appl. Math.* **123**(1–2), 85–100 (2000)
48. Wan, J., Zabaras, N.: A Bayesian approach to multiscale inverse problems using the sequential Monte Carlo method. *Inverse Probl.* **27**(10), 105004 (2011)
49. Wan, X.L., Karniadakis, G.E.: An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.* **209**(2), 617–642 (2005). doi:[10.1016/j.jcp.2005.03.023](https://doi.org/10.1016/j.jcp.2005.03.023), <Go to ISI>://WOS:000230736700011
50. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D.: Screening, predicting, and computer experiments. *Technometrics* **34**(1), 15–25 (1992)
51. Xiu, D.B.: Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys.* **2**(2), 293–309 (2007)

52. Xiu, D.B., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
53. Xiu, D.B., Karniadakis, G.E.: The wiener-askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)

Solution Algorithms for Stochastic Galerkin Discretizations of Differential Equations with Random Data

16

Howard Elman

Abstract

This chapter discusses algorithms for solving systems of algebraic equations arising from stochastic Galerkin discretization of partial differential equations with random data, using the stochastic diffusion equation as a model problem. For problems in which uncertain coefficients in the differential operator are linear functions of random parameters, a variety of efficient algorithms of multigrid and multilevel type are presented, and, where possible, analytic bounds on convergence of these methods are derived. Some limitations of these approaches for problems that have nonlinear dependence on parameters are outlined, but for one example of such a problem, the diffusion equation with a diffusion coefficient that has exponential structure, a strategy is described for which the reformulated problem is also amenable to efficient solution by multigrid methods.

Keywords

Convergence analysis • Iterative methods • Multigrid • Stochastic Galerkin

Contents

1	Introduction: The Stochastic Finite Element Method	602
2	Solution Algorithms	606
2.1	Multigrid Methods I	606
2.2	Multigrid Methods II: Mean-Based Preconditioning	608
2.3	Hierarchical Methods	610
3	Approaches for Other Formulations	612
4	Conclusion	615
	References	615

H. Elman (✉)

Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA

e-mail: elman@cs.umd.edu

1 Introduction: The Stochastic Finite Element Method

This chapter is concerned with algorithms for solving the systems of algebraic equations arising from stochastic Galerkin discretization of partial differential equations (PDEs) with random data. Consider a PDE of generic form $\mathcal{L}u = f$ on a spatial domain \mathcal{D} , subject to boundary conditions $\mathcal{B}u = g$ on $\partial\mathcal{D}$, where one or more of the operators \mathcal{L} , \mathcal{B} , or functions f , g depend on random data. The most challenging scenario is when the dependence is in the differential operator \mathcal{L} and the discussion will be concentrated on this case. An example is the diffusion equation, where $\mathcal{L}u \equiv -\nabla \cdot (a\nabla u)$, subject to boundary conditions $u = g_D$ on $\partial\mathcal{D}_D$, $a \frac{\partial u}{\partial n} = 0$ on $\partial\mathcal{D}_N$. The diffusion coefficient $a = a(\mathbf{x}, \omega)$ is a random field: given a probability space (Ω, \mathcal{F}, P) , for each $\omega \in \Omega$, the realization $a(\cdot, \omega)$ is a function defined on \mathcal{D} , and for each $\mathbf{x} \in \mathcal{D}$, $a(\mathbf{x}, \cdot)$ is a random variable. Although the focus will be on the diffusion equation, the solution algorithms described are easily generalized to other problems with random coefficients. (See, e.g., [22] for a study of the Navier-Stokes equations.) In discussion of the diffusion problem, it will be assumed that $a(\mathbf{x}, \omega)$ is uniformly bounded, i.e., $0 < \alpha_1 \leq a(\mathbf{x}, \omega) \leq \alpha_2 < \infty$ a.e., which ensures well posedness. The Galerkin formulation of the stochastic diffusion problem augments the standard (spatial) Galerkin methodology using averaging with respect to expected value in the probability measure.

To begin, this section contains a concise introduction to the stochastic Galerkin methodology. Comprehensive treatments of this approach can be found in [10, 17, 18, 33]. The next section contains a description and analysis of two efficient solution algorithms that use a multigrid structure in the spatial component of the problem, together with a description of an algorithm that can be viewed as having a multilevel structure in the stochastic component. These methods are designed for problems that depend linearly on a set of random parameters. This is followed by a section discussing some limitations of these ideas for problems with nonlinear dependence on random parameters, coupled with the presentation of an effective strategy to handle one specific version of such problems, where the diffusion coefficient is of exponential form. The final section contains a brief recapitulation of the chapter.

Let $H^1(\mathcal{D})$ denote the Sobolev space of functions on \mathcal{D} with square integrable first derivatives, let $H_E^1(\mathcal{D}) \equiv \{u \in H^1(\mathcal{D}) \mid u = g_D \text{ on } \partial\mathcal{D}_D\}$, let $H_0^1(\mathcal{D}) \equiv \{u \in H^1(\mathcal{D}) \mid u = 0 \text{ on } \partial\mathcal{D}_D\}$, and let $L^2(\Omega)$ denote the Hilbert space with inner product defined by expected value on Ω ,

$$\langle v, w \rangle \equiv \int_{\Omega} v(\omega)w(\omega)dP(\Omega).$$

The weak formulation of the stochastic diffusion problem is then to find $u \in H_E^1(\mathcal{D}) \otimes L^2(\Omega)$ such that

$$\int_{\Omega} \int_{\mathcal{D}} a \nabla u \cdot \nabla v d\mathbf{x} dP(\Omega) = \int_{\Omega} \int_{\mathcal{D}} f v d\mathbf{x} dP(\Omega) \quad (16.1)$$

for all $v \in H_0^1(\mathcal{D}) \otimes L^2(\Omega)$.

For the abstract formulation (16.1) to be suitable for computation, the random field $a(\cdot, \cdot)$ must be expressed in terms of a finite number of random variables. Galerkin methods are most useful when this dependence is *linear*:

$$a(\mathbf{x}, \omega) = a_0(\mathbf{x}) + \sum_{r=1}^m a_r(\mathbf{x}) \xi_r(\omega), \quad (16.2)$$

where $\{\xi_r\}_{r=1}^m$ is a set of m random variables. A typical example comes from a Karhunen-Loëve (KL) expansion derived from a covariance function. In this case, $\{a_r\}_{r=1}^m$ correspond to discrete versions of the m eigenfunctions of the covariance operator associated with the m largest eigenvalues, and $\{\xi_r\}$ are uncorrelated zero-mean random variables defined on Ω . (It is assumed for (16.2) that the associated eigenvalues are incorporated into $\{a_r\}$ as factors.) The covariance function is positive semi-definite and $\{a_r\}_{r=1}^m$ can be listed in nonincreasing order of magnitude, according to the nonincreasing sizes of the eigenvalues.

Let $\boldsymbol{\xi} \equiv (\xi_1, \dots, \xi_m)^T$. For fixed \mathbf{x} , the diffusion coefficient can be viewed as a function of $\boldsymbol{\xi}$, and it will be written using the same symbol, as $a(\mathbf{x}, \boldsymbol{\xi})$. It can then be shown that the solution $u(\mathbf{x}, \cdot)$ is also a function of $\boldsymbol{\xi}$ [18, Ch. 9], and if the joint density $\rho(\boldsymbol{\xi})$ is known, then (16.1) can be rewritten as

$$\int_{\Gamma} \int_{\mathcal{D}} a(\mathbf{x}, \boldsymbol{\xi}) \nabla u(\mathbf{x}, \boldsymbol{\xi}) \cdot \nabla v(\mathbf{x}, \boldsymbol{\xi}) d\mathbf{x} \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int_{\Gamma} \int_{\mathcal{D}} f(\mathbf{x}, \boldsymbol{\xi}) v(\mathbf{x}, \boldsymbol{\xi}) d\mathbf{x} \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (16.3)$$

where $\Gamma = \boldsymbol{\xi}(\Omega)$. Note that in (16.3), the vector of random variables is playing a role analogous to spatial Cartesian coordinates in the physical domain. For a d -dimensional physical domain, (16.3) is similar in form to the weak formulation of a $(d+m)$ -dimensional continuous deterministic problem.

Numerical approximation is done on a finite-dimensional subset of $H_E^1(\mathcal{D}) \otimes L^2(\Omega)$. For this, let $\mathcal{S}^{(h)}$ denote a finite-dimensional subspace of $H_0^1(\mathcal{D})$ with basis $\{\phi_j\}_{j=1}^{N_x}$, let $\mathcal{S}_E^{(h)}$ be an extended version of this space with additional functions $\{\phi_j\}_{j=N_x+1}^{N_x+N_\partial}$, so that $\sum_{j=N_x+1}^{N_x+N_\partial} u_j \phi_j$ interpolates the boundary data g_D on $\partial\mathcal{D}$, and let $\mathcal{T}^{(p)}$ denote a finite-dimensional subspace of $L^2(\Gamma)$ with basis $\{\psi_q\}_{q=1}^{N_\xi}$. The discrete weak problem is then to find $u^{(hp)} \in \mathcal{S}_E^{(h)} \otimes \mathcal{T}^{(p)}$ such that

$$\int_{\Gamma} \int_{\mathcal{D}} a \nabla u^{(hp)} \cdot \nabla v^{(hp)} d\mathbf{x} \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int_{\Gamma} \int_{\mathcal{D}} f v^{(hp)} d\mathbf{x} \rho(\boldsymbol{\xi}) d\boldsymbol{\xi},$$

for all $v^{(hp)} \in \mathcal{S}^{(h)} \otimes \mathcal{T}^{(p)}$. The discrete solution has the form

$$u^{(hp)}(\mathbf{x}, \boldsymbol{\xi}) = \sum_{q=1}^{N_\xi} \sum_{j=1}^{N_x} u_{jq} \phi_j(\mathbf{x}) \psi_q(\boldsymbol{\xi}) + \sum_{j=N_x+1}^{N_x+N_\partial} u_j \phi_j(\mathbf{x}). \quad (16.4)$$

Computing the solution entails solving a linear system of equations

$$A^{(hp)} \mathbf{u}^{(hp)} = \mathbf{f}^{(hp)} \quad (16.5)$$

for the coefficients $\{u_{jq}\}_{j=1:N_x, q=1:N_\xi}$. (From the assumption of deterministic boundary data, the known coefficients of the spatial basis functions on the Dirichlet boundary $\partial\Omega_D$ can be incorporated into the right-hand side of (16.4).) If the vector $\mathbf{u}^{(hp)}$ of these coefficients is ordered by grouping the spatial indices together, as

$$u_{11}, u_{21}, \dots, u_{N_x 1}, u_{12}, u_{22}, \dots, u_{N_x 2}, u_{1N_\xi}, u_{2N_\xi}, \dots, u_{N_x N_\xi},$$

and the equations are ordered in an analogous way, then the coefficient matrix is a sum of matrices of Kronecker-product structure,

$$A^{(hp)} = G_0^{(p)} \otimes A_0^{(h)} + \sum_{r=1}^m G_r^{(p)} \otimes A_r^{(h)}, \quad (16.6)$$

where

$$\begin{aligned} [A_r^{(h)}]_{jk} &= \int_{\mathcal{D}} a_r(\mathbf{x}) \nabla \phi_k(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x}, \\ [G_0^{(p)}]_{lq} &= \int_{\Gamma} \psi_q(\boldsymbol{\xi}) \psi_l(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad [G_r^{(p)}]_{lq} = \int_{\Gamma} \xi_r \psi_q(\boldsymbol{\xi}) \psi_l(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \end{aligned} \quad (16.7)$$

and the Kronecker product of matrices G (of order N_ξ) and A (of order N_x) is the matrix of order $N_x N_\xi$ given by

$$G \otimes A = \begin{bmatrix} g_{11}A & g_{12}A & \cdots & g_{1N_\xi}A \\ g_{21}A & g_{22}A & \cdots & g_{2N_\xi}A \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_\xi 1}A & g_{N_\xi 2}A & \cdots & g_{N_\xi N_\xi}A \end{bmatrix}.$$

The joint density function $\rho(\boldsymbol{\xi})$ may not be known, and an additional assumption typically made is that the random variables $\{\xi_r\}$ are *independent* with known marginal density functions $\{\rho_r\}$ [34] (cf. [12] for an analysis of this assumption). The joint density function is then the product of marginal density functions, $\rho(\boldsymbol{\xi}) = \rho_1(\xi_1)\rho_2(\xi_2)\cdots\rho_m(\xi_m)$. This also enables a simple choice of basis functions for $\mathcal{T}^{(p)}$. Let $\{p_j^{(r)}\}_{j \geq 0}$ be the set of polynomials orthogonal with respect to the density function ρ_r , normalized so that $\langle p_j^{(r)}, p_j^{(r)} \rangle = 1$. Then, each basis function ψ_q can be taken to be a product of univariate polynomials [10, 34]

$$\psi_q(\boldsymbol{\xi}) = p_{j_1}^{(1)}(\xi_1) p_{j_2}^{(2)}(\xi_2) \cdots p_{j_m}^{(m)}(\xi_m) \quad (16.8)$$

where the index q is determined by a mapping between multi-indices $\{j_1 j_2 \cdots j_m\}$ of polynomial degrees and integers in $\{1, 2, \dots, N_\xi\}$. This construction is known as the *generalized polynomial chaos*. A common choice of basis is the set of m -variate

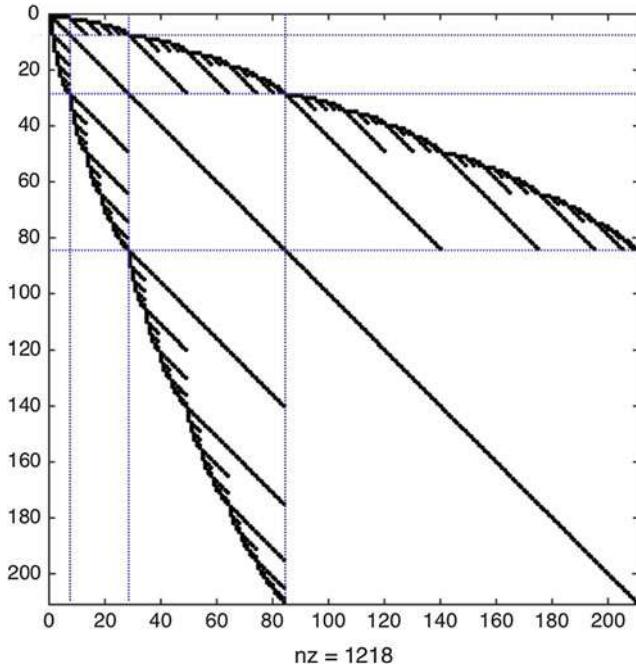


Fig. 16.1 Representative sparsity pattern of coefficient matrix obtained from stochastic Galerkin discretization with $m = 6$, $p = 4$, giving block order 210

polynomials of total degree at most p , i.e., $j_1 + \dots + j_m \leq p$. Then, $N_\xi = \binom{m+p}{p}$, and it follows from properties of orthogonal polynomials that the matrices $\{G_r^{(p)}\}$ are sparse, with at most two nonzeros per row. $A^{(hp)}$ is also sparse; a representative example of its sparsity pattern, for $m = 4$ and $p = 6$, is shown in Fig. 16.1. Each pixel in the figure corresponds to a matrix of order N_x with nonzero structure like that obtained from discretization of a deterministic PDE. The blocking of the matrix will be explained below.

The following lemma will be of use for analyzing the convergence properties of iterative solution algorithms for the linear system.

Lemma 1. *If $\mathcal{T}^{(p)}$ consists of polynomials of total degree p specified above and each density function ρ_r is an even function of ξ_r , then for $r \geq 1$, the eigenvalues of $G_r^{(p)}$ are contained in the symmetric interval $[-\eta_p, +\eta_p]$ where η_p is the maximal positive root of the polynomial $p_{p+1}^{(r)}$ of degree $p+1$.*

See [8] or [21] for a proof. It is shown in the latter reference that η_p is bounded by $\sqrt{p} + \sqrt{p-1}$ in the case of standard Gaussian random variables and by $\sqrt{3}$ for uniform variables with mean 0 and variance 1. The assumption concerning normalized polynomials implies that $G_0^{(p)} = I$, the identity matrix of order N_ξ .

2 Solution Algorithms

This section presents a collection of efficient iterative solution algorithms for the coupled linear system (16.5) that arises from stochastic finite element methods. The emphasis is on approaches whose convergence behavior has been shown to be insensitive to the parameters such as spatial mesh size or polynomial order that determine the discretization; they include multigrid and multilevel approaches and techniques that take advantage of the hierarchical structure of the problem. Other methods, mostly developed earlier than the ones described here, can be viewed as progenitors of these techniques; they include methods based on incomplete factorization [11, 20] or block splitting methods [15, 24]. The discussion will be limited to primal formulations of the problem; see [5, 9] for treatment of the stochastic diffusion equation using mixed finite elements in space.

2.1 Multigrid Methods I

One way to apply multigrid to the stochastic Galerkin system (16.5) is by generalizing methods devised for deterministic problems. This idea was developed in [16], and convergence analysis was presented in [4]; see also [26]. Assume there is a set of spatial grids of mesh width $h, 2h, 4h$, etc., and let the discrete space $\mathcal{T}^{(p)}$ associated with the stochastic component of the problem be fixed. The coefficient matrices for fine-grid and coarse-grid spatial discretizations are

$$A^{(hp)} = G_0^{(p)} \otimes A_0^{(h)} + \sum_{r=1}^m G_r^{(p)} \otimes A_r^{(h)}, \quad A^{(2h,p)} = G_0^{(p)} \otimes A_0^{(2h)} + \sum_{r=1}^m G_r^{(p)} \otimes A_r^{(2h)}.$$

Let P_{2h}^h denote a prolongation operator mapping (spatial) coarse-grid vectors to fine-grid vectors, and let R_h^{2h} denote a restriction operator mapping fine to coarse. Typically P_{2h}^h is specified using interpolation and $R_h^{2h} = [P_{2h}^h]^T$ [7]. Then, $\mathcal{P} = I \otimes R_h^{2h}$ is a restriction operator on the tensor product space that leaves the discrete stochastic space intact, and $\mathcal{P} = I \otimes P_{2h}^h$ is an analogous prolongation operator. In addition, let Q represent a smoothing operator on the fine grid; a more precise specification of Q is given below. One step of a *two-grid algorithm* for (16.5) to update an approximate solution $\mathbf{u}^{(hp)}$ is defined as follows:

Recall that any positive-definite matrix M induces a norm $\|\mathbf{v}\|_M \equiv (\mathbf{v}, M\mathbf{v})^{1/2}$. Convergence of Algorithm 1 is established by the following result [4]. It is stated in terms of generic constants c_1 and c_2 whose properties will be discussed below.

Theorem 1. *Let $\mathbf{u}_i^{(hp)}$ denote the approximate solution to (16.5) obtained after i steps of Algorithm 1, with error $\mathbf{e}_i = \mathbf{u}^{(hp)} - \mathbf{u}_i^{(hp)}$. If the smoothing property*

$$\|A^{(hp)}(I - Q^{-1}A^{(hp)})^k \mathbf{y}\|_2 \leq \eta(k) \|\mathbf{y}\|_{A^{(hp)}} \quad \text{for all } \mathbf{y} \in \mathbb{R}^{N_x N_\xi} \quad (16.9)$$

Algorithm 1 One step of a two-grid method for the stochastic Galerkin system, given an estimate $\mathbf{u}^{(hp)}$ for the solution.

```

for  $j = 1 : k$ 
     $\mathbf{u}^{(hp)} \leftarrow \mathbf{u}^{(hp)} + Q^{-1}(\mathbf{f}^{(hp)} - A^{(hp)}\mathbf{u}^{(hp)})$        $k$  smoothing steps
end
 $\mathbf{r}^{(2h,p)} = \mathcal{R}(\mathbf{f}^{(hp)} - A^{(hp)}\mathbf{u}^{(hp)})$           Restriction,  $\mathcal{R} = I \otimes R_h^{2h}$ 
Solve  $A^{(2h,p)}\mathbf{c}^{(2h,p)} = \mathbf{r}^{(2h,p)}$           Compute coarse-grid correction
 $\mathbf{u}^{(hp)} \leftarrow \mathbf{u}^{(hp)} + \mathcal{P}\mathbf{c}^{(2h,p)}$           Prolongation,  $\mathcal{P} = I \otimes P_{2h}^h$ 

```

holds with $\eta(k) \rightarrow 0$ as $k \rightarrow \infty$, and the approximation property,

$$\|[(A^{(hp)})^{-1} - \mathcal{P}(A^{(2h,p)})^{-1}\mathcal{R}]\mathbf{y}\|_{A^{(hp)}} \leq c_1 \|\mathbf{y}\|_2 \quad \text{for all } \mathbf{y} \in \mathbb{R}^{N_x N_\xi}, \quad (16.10)$$

holds, then the error satisfies $\|\mathbf{e}_i\|_A \leq c_2 \|\mathbf{e}_{i-1}\|_{A^{(hp)}}$.

Proof. The errors associated with Algorithm 1 satisfy the recursive relationship

$$\mathbf{e}_i = [(A^{(hp)})^{-1} - \mathcal{P}(A^{(2h,p)})^{-1}\mathcal{R}][A^{(hp)}(I - Q^{-1}A^{(hp)})^k]\mathbf{e}_{i-1}.$$

Application of the approximation property (16.10) and smoothing property (16.9) gives

$$\begin{aligned} \|\mathbf{e}_i\|_{A^{(hp)}} &= \|[(A^{(hp)})^{-1} - \mathcal{P}(A^{(2h,p)})^{-1}\mathcal{R}][A^{(hp)}(I - Q^{-1}A^{(hp)})^k]\mathbf{e}_{i-1}\|_{A^{(hp)}} \\ &\leq c_1 \|[A^{(hp)}(I - Q^{-1}A^{(hp)})^k]\mathbf{e}_{i-1}\|_2 \\ &\leq c_1 \eta(k) \|\mathbf{e}_{i-1}\|_{A^{(hp)}} \leq c_2 \|\mathbf{e}_{i-1}\|_{A^{(hp)}} \end{aligned}$$

for all large enough k .

An outline of what is required to establish (16.9)–(16.10) is as follows. For the smoothing property (16.9), a simple splitting operator is $Q = \theta I$ for constant θ . With this choice, each smoothing step in Algorithm 1 consists of a damped Richardson iteration with damping parameter $1/\theta$. A standard analysis of multigrid [2, Ch. V] [7, Ch. 2] shows that the smoothing property holds for $\theta \geq \lambda_{\max}(A^{(hp)})$. Thus, it suffices to have an upper bound for the maximal eigenvalue of the symmetric positive-definite matrix $A^{(hp)}$, which can be obtained using properties of Kronecker products [14] and sums of symmetric matrices [13]:

$$\lambda_{\max}(A^{(hp)}) \leq \sum_{r=0}^m \lambda_{\max}(G_r^{(p)} \otimes A_r^{(h)}) = \sum_{r=0}^m \lambda_{\max}(G_r^{(p)}) |\lambda_{\max}(A_r^{(h)})|. \quad (16.11)$$

Bounds on the eigenvalues of $G_r^{(p)}$ come from Lemma 1. The eigenvalues of $A_r^{(h)}$ can be bounded using Rayleigh quotients; see (16.15) below.

To see what is needed to establish the approximation property (16.10), first recall some results for deterministic problems. Let $\mathcal{S}^{(h)}$ and $\mathcal{S}_E^{(h)}$ be above. Given $\mathbf{y} \in \mathbb{R}^{N_x}$, let $y = y^{(h)} = \sum_{j=1}^{N_x} y_j \phi_j \in \mathcal{S}^{(h)}$, and consider the deterministic diffusion equation $-\nabla \cdot (a \nabla u) = y$ on \mathcal{D} , for which no components of the problem depend on random data. Let $u \in H_E^1(\mathcal{D})$ denote the (weak) solution. Then, $\mathbf{u} = (A^{(h)})^{-1} \mathbf{y}$ corresponds to a discrete weak solution $u^{(h)} \in \mathcal{S}^{(h)}$ (which can be extended to $\mathcal{S}_E^{(h)}$ as in (16.4)). Similarly, $\mathbf{u} = (A^{(2h)})^{-1} R_h^{2h} \mathbf{y}$ corresponds to a coarse-grid solution $u^{(2h)}$. The approximation property follows from the relations

$$\begin{aligned} \|[(A^{(h)})^{-1} - P_{2h}^h (A^{(2h)})^{-1} R_h^{2h}] \mathbf{y}\|_{A^{(h)}} &= \|\nabla(u^{(h)} - u^{(2h)})\|_0 \\ &\leq \|\nabla(u^{(h)} - u)\|_0 + \|\nabla(u - u^{(2h)})\|_0 \\ &\leq c_1 \|\mathbf{y}\|_2. \end{aligned} \quad (16.12)$$

The last inequality depends on an assumption of regularity of the solution that $u \in H^2(\mathcal{D})$.

The key to this analysis is that the difference between the fine-grid and coarse-grid solutions is bounded by the sum of the fine-grid and coarse-grid errors. Generalization to the stochastic problem requires an analogue of the deterministic solution u . This is obtained using a semi-discrete space $H^1(\mathcal{D}) \otimes \mathcal{T}^{(p)}$. The weak solution in this semi-discrete setting is $u^{(p)} \in H_E^1(\mathcal{D}) \otimes \mathcal{T}^{(p)}$ for which (16.1) holds for all $v^{(p)} \in H_0^1(\mathcal{D}) \otimes \mathcal{T}^{(p)}$. The analysis then proceeds as in (16.12) using $u^{(hp)}$, $u^{(2h,p)}$ and $u^{(p)}$ in place of the deterministic quantities. Complete details can be found in [4]. Extension from two grid to multigrid follows arguments for deterministic problems.

Theorem 1 establishes “textbook” multigrid convergence, i.e., it shows that the convergence factor is independent of the discretization parameter h . The constant c_1 in the approximation property does not depend on the number of terms m in the representation of the diffusion coefficient (16.2) or the polynomial degree p used to discretize the probability space. The smoothing property and constant c_2 depend on bounds on $\lambda_{\max}(A^{(hp)})$ from (16.11); these bounds may depend on m . (They are independent of p if the support of the density function ρ is bounded.) Computational results in [4] and [16] showed no dependence on this parameter. The number of nonzero entries in $A^{(hp)}$ is $O(N_x N_\xi)$, so that the cost of the matrix-vector product is linear in the problem size. The coarse-grid solves require solutions of systems of equations of order proportional to $|\mathcal{T}^{(p)}|$, the dimension of $\mathcal{T}^{(p)}$. If this is not too large, then direct methods can be used for these computations. Modulo this computation, the cost per step of the multigrid algorithm is also of order $N_x N_\xi$.

2.2 Multigrid Methods II: Mean-Based Preconditioning

A different way to apply multigrid to the system (16.5) comes from use of $\mathcal{Q}_0^{(hp)} \equiv G_0^{(p)} \otimes A_0^{(h)}$ as a preconditioner for the coefficient matrix $A^{(hp)}$ [21]. The preconditioning operator derives from the mean a_0 of the diffusion coefficient.

If $a(\cdot, \cdot)$ is not too large a perturbation of a_0 , then $Q_0^{(hp)}$ will be a reasonable approximation of $A^{(hp)}$ and can be used as a preconditioner in combination with Krylov subspace methods such as the conjugate gradient method (CG). Moreover, since $G_0^{(p)} = I_{N_\xi}$, the identity matrix, $Q_0^{(hp)}$, is a block-diagonal matrix consisting of N_ξ decoupled copies of $A_0^{(h)}$. Thus, the overhead of using it with CG entails applying the action of the inverse of N_ξ decoupled copies of $A^{(h)}$ at each step.

Assume that $a_0(\mathbf{x})$ is uniformly bounded below by $\alpha_1 > 0$. The following result establishes the effectiveness of $Q_0^{(hp)}$ as a preconditioner.

Theorem 2. *The eigenvalues of the generalized eigenvalue problem $A^{(hp)}\mathbf{v} = \mu Q_0^{(hp)}\mathbf{v}$ are contained in an interval $[1 - \tau, 1 + \tau]$, where*

$$\tau \leq \sum_{r=1}^m |\eta_{\max}(G_r^{(p)})| \left(\frac{\|a_r\|_\infty}{\alpha_1} \right), \quad (16.13)$$

and $|\eta_{\max}(G_r^{(p)})|$ is the modulus of the eigenvalue of $G_r^{(p)}$ of maximal modulus.

If $\tau < 1$, the condition number of the preconditioned operator $(Q_0^{(hp)})^{-1}A^{(hp)}$ is bounded by $\frac{1+\tau}{1-\tau}$.

Proof. The proof is done by establishing bounds on the Rayleigh quotient

$$\frac{(\mathbf{v}, A^{(hp)}\mathbf{v})}{(\mathbf{v}, Q_0^{(hp)}\mathbf{v})} = 1 + \sum_{r=1}^m \frac{\left(\mathbf{v}, \left(G_r^{(p)} \otimes A_r^{(h)} \right) \mathbf{v} \right)}{\left(\mathbf{v}, \left(G_0^{(p)} \otimes A_0^{(h)} \right) \mathbf{v} \right)}.$$

Each of the fractions in the sum is bounded by the product of the maximal eigenvalue of $G_r^{(p)}$ times the modulus of the maximal eigenvalue of $(A_0^{(h)})^{-1}A_r^{(h)}$. The latter quantities are the extrema of the Rayleigh quotient

$$\frac{|(\mathbf{v}^{(h)}, A_r^{(h)}\mathbf{v}^{(h)})|}{(\mathbf{v}^{(h)}, A_0^{(h)}\mathbf{v}^{(h)})}, \quad (16.14)$$

where $\mathbf{v}^{(h)}$ is a vector of length N_x that corresponds to $v_h \in \mathcal{S}^{(h)}$. The terms in (16.14) are bounded as

$$\begin{aligned} |(\mathbf{v}^{(h)}, A_r^{(h)}\mathbf{v}^{(h)})| &= \left| \int_{\mathcal{D}} a_r(\mathbf{x}) \nabla v_h(\mathbf{x}) \cdot \nabla v_h(\mathbf{x}) d\mathbf{x} \right| \leq \|a_r\|_\infty \int_{\mathcal{D}} |\nabla v_h(\mathbf{x})| |\nabla v_h(\mathbf{x})| d\mathbf{x}, \\ (\mathbf{v}^{(h)}, A_0^{(h)}\mathbf{v}^{(h)}) &= \int_{\mathcal{D}} a_0(\mathbf{x}) \nabla v_h(\mathbf{x}) \cdot \nabla v_h(\mathbf{x}) d\mathbf{x} \geq \alpha_1 \int_{\mathcal{D}} |\nabla v_h(\mathbf{x})| |\nabla v_h(\mathbf{x})| d\mathbf{x}. \end{aligned} \quad (16.15)$$

The assertion follows. Bounds on the eigenvalues of $G_r^{(p)}$ again come from Lemma 1.

It is generally preferable to *approximate* the action of $(A_0^{(h)})^{-1}$, which can be done using multigrid. In particular, suppose $(Q_{0,MG}^{(h)})^{-1}$ represents a spectrally equivalent approximation to $(A_0^{(h)})^{-1}$ resulting from a fixed number of multigrid steps. That is,

$$\beta_1 \leq \frac{(\mathbf{v}^{(h)}, A_0^{(h)} \mathbf{v}^{(h)})}{(\mathbf{v}^{(h)}, Q_{0,MG}^{(h)} \mathbf{v}^{(h)})} \leq \beta_2$$

for constants β_1, β_2 independent of the spatial discretization parameter h . With $Q_{0,MG}^{(hp)} \equiv G_0^{(p)} \otimes Q_{0,MG}^{(h)}$, this leads to the following result.

Corollary 1. *The condition number of the multigrid preconditioned operator $(Q_{0,MG}^{(hp)})^{-1} A^{(hp)}$ is bounded by $(\frac{1+\tau}{1-\tau}) (\frac{\beta_2}{\beta_1})$.*

As is well known, the number of steps required for convergence of the conjugate gradient method is bounded in terms of the condition number of the coefficient matrix [7]. Thus, Corollary 1 establishes the optimality of this method. The requirement $\tau < 1$ implies that the method is useful only if $A^{(hp)}$ is not too large, a perturbation of $Q_{0,MG}^{(hp)}$. However, this method is significantly simpler to implement than Algorithm 1. In particular, it is straightforward to use algebraic multigrid (e.g., [25]) to handle the approximate action of $(A_0^{(h)})^{-1}$, and there are no coarse-grid operations with matrices of order $|\mathcal{T}^{(p)}|$. A comparison of these two versions of multigrid, showing some advantages of each, is given in [24]. It is also possible to improve performance of the mean-based approach using a variant of the form $\widehat{Q}_p^{(hp)} \equiv \widehat{G}^{(p)} \otimes A_0^{(h)}$, where some $\widehat{G}^{(p)}$ replaces the identity in the Kronecker product [29]; a good choice of $\widehat{G}^{(p)}$ is determined by minimizing the Frobenius norm $\|A^{(hp)} - \widehat{G}^{(p)} \otimes A_0^{(h)}\|_F$ [31].

2.3 Hierarchical Methods

The blocking of the matrix shown in Fig. 16.1 reflects the hierarchical structure of the discrete problem. The coefficient matrix arising from the subspace $\mathcal{T}^{(p)} \subset L^2(\Gamma)$, consisting of polynomials of total degree p , has the form

$$A^{(hp)} = \begin{bmatrix} A^{(h,p-1)} & B^{(hp)} \\ C^{(hp)} & D^{(hp)} \end{bmatrix}$$

where the $(1, 1)$ -subblock comes from $\mathcal{S}^{(h)} \otimes \mathcal{T}^{(p-1)}$. In the figure, $p = 4$ and (with $m = 6$), the block corresponding to $\mathcal{T}^{(3)}$ has order $\binom{6+3}{3} = 84$. It follows from the orthogonality of the basis for $\mathcal{T}^{(p)}$ that the $(2, 2)$ -block $D^{(hp)}$ is a block-diagonal

matrix, each of whose blocks is A_0^h (see, e.g., [24]). The number of such blocks is the number of basis functions of total degree exactly p .

This structure has been used in [28], building on ideas in [11], to develop a hierarchical preconditioning strategy. The approach can also be viewed as a multilevel method in the stochastic dimension, and it bears some resemblance to iterative substructuring methods as described, for example, in [27]. To make the description more readable, the dependence on the spatial mesh size h will be suppressed from the notation. The coefficient matrix then has a block factorization

$$A^{(p)} = \begin{bmatrix} A^{(p-1)} & B^{(p)} \\ C^{(p)} & D^{(p)} \end{bmatrix} = \begin{bmatrix} I & B^{(p)}(D^{(p)})^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} S^{(p-1)} & 0 \\ 0 & D^{(p)} \end{bmatrix} \begin{bmatrix} I & 0 \\ (D^{(p)})^{-1}B^{(p)} & I \end{bmatrix},$$

where $S^{(p-1)} \equiv A^{(p-1)} - B^{(p)}(D^{(p)})^{-1}C^{(p)}$ is a Schur complement. Equivalently,

$$(A^{(p)})^{-1} = \begin{bmatrix} I & 0 \\ -(D^{(p)})^{-1}B^{(p)} & I \end{bmatrix} \begin{bmatrix} (S^{(p-1)})^{-1} & 0 \\ 0 & (D^{(p)})^{-1} \end{bmatrix} \begin{bmatrix} I & -B^{(p)}(D^{(p)})^{-1} \\ 0 & I \end{bmatrix}.$$

The Schur complement is expensive to work with, and the idea in [28] is to replace $S^{(p-1)}$ with $A^{(p-1)}$, giving the approximate factorization

$$(A^{(p)})^{-1} \approx \begin{bmatrix} I & 0 \\ -(D^{(p)})^{-1}B^{(p)} & I \end{bmatrix} \begin{bmatrix} (A^{(p-1)})^{-1} & 0 \\ 0 & (D^{(p)})^{-1} \end{bmatrix} \begin{bmatrix} I & -B^{(p)}(D^{(p)})^{-1} \\ 0 & I \end{bmatrix}. \quad (16.16)$$

The preconditioning operator is then defined by applying this strategy recursively to $(A^{(p-1)})^{-1}$; implementation details are given in [28]. A key point is that the only (subsidiary) system solves required entail independent computations of the action of $(A_0^{(h)})^{-1}$. These are analogous to the coarse-grid solves required by Algorithm 1, and they can be replaced by approximate solves.

The following convergence result is given in [28].

Theorem 3. *Let $Q^{(hp)}$ denote the preconditioning operator defined by recursive application of the approximation (16.16). Then, the condition number of the preconditioned operator $(Q^{(hp)})^{-1}A^{(hp)}$ is bounded by $1/\left(\prod_{q=1}^{p-1} \beta_{1q}\right)$, where*

$$\beta_{1q}(\mathbf{u}, A^{(q)}\mathbf{u})^{1/2} \leq (\mathbf{u}, S^{(q)}\mathbf{u})^{1/2} \leq (\mathbf{u}, A^{(q)}\mathbf{u})^{1/2}, \quad 1 \leq q \leq p-1.$$

Bounds on the terms $\{\beta_{1q}\}$ appearing in this expression have not been established. However, experimental results in [28] suggest that for a diffusion equation with coefficient as in (16.2) (with uniformly distributed random variables), the condition number of the preconditioned system $(Q^{(hp)})^{-1}A^{(hp)}$ is very close to 1 and largely insensitive to the problem parameters, i.e., spatial mesh size h , stochastic dimension m , and polynomial degree p .

This hierarchical strategy is more effective than a pure multilevel method. A two-level strategy would have the form of Algorithm 1 with the “lower-level” space $\mathcal{S}^{(h)} \otimes \mathcal{T}^{(p-1)}$ used in place of the coarse physical space $\mathcal{S}^{(2h)} \otimes \mathcal{T}^{(p)}$. The latter steps of the computation, corresponding to the analogous steps in Algorithm 1, are:

$$\begin{array}{ll} \mathbf{r}^{(2,p-1)} = \mathcal{R}_p^{p-1}(\mathbf{f}^{(hp)} - A^{(hp)}\mathbf{u}^{(hp)}) & \text{Restriction} \\ \text{Solve } A^{(h,p-1)}\mathbf{c}^{(h,p-1)} = \mathbf{r}^{(h,p-1)} & \text{Compute coarse-grid correction} \\ \mathbf{u}^{(hp)} \leftarrow \mathbf{u}^{(hp)} + \mathcal{P}_{p-1}^p \mathbf{c}^{(h,p-1)} & \text{Prolongation} \end{array}$$

Results for such a strategy, for restriction and prolongation operators $\mathcal{R}_p^{p-1} = [I, 0]$, $\mathcal{P} = \mathcal{R}^T$ (injection), are discussed in [15, Ch. 4] and [24, Sect. 4], where it is shown that this method *does not* display costs that grow linearly with the number of unknowns.

3 Approaches for Other Formulations

As noted above, the stochastic Galerkin method is most effective when the dependence of the problem on random data is a linear function of parameters, as in (16.2). This section expands on this point, showing what issues arise for problems depending on random fields that are nonlinear functions of the data. In addition, one method is presented that avoids some of these issues for one important class of problems, the particular case of the diffusion equation in which the diffusion coefficient has an exponential structure. A commonly used model of this type is where the diffusion coefficient has a log-normal distribution, see [1, 35].

Suppose now that $a(\mathbf{x}, \xi)$ is a nonlinear function of ξ , where ξ is of length m , and that this function has a series expansion in the polynomial chaos basis

$$a(\mathbf{x}, \xi) = \sum_r a_r(\mathbf{x}) \psi_r(\xi),$$

where, as in (16.8), each index r is associated with a multi-index of m -tuples of nonnegative integers. Assume that this series can be approximated well using a finite-term sum whose length will temporarily be denoted by \hat{m} . Then, the formalism of the stochastic Galerkin discretization giving rise to equation (16.5) carries over essentially unchanged, where

$$A^{(hp)} = \sum_{r=1}^{\hat{m}} G_r^{(p)} \otimes A_r^{(h)}. \quad (16.17)$$

The main difference lies in the definition of the matrices $\{G_r^{(p)}\}$ associated with the stochastic terms, where ξ_r in (16.7) is replaced by $\psi_r(\xi)$:

$$[G_r^{(p)}]_{\ell q} = \int_{\Gamma} \psi_r(\xi) \psi_q(\xi) \psi_\ell(\xi) \rho(\xi) d\xi. \quad (16.18)$$

(There is also a slight difference in indexing conventions, in that the lowest index in (16.17) is $r = 1$.) If the finite-dimensional subspace $\mathcal{T}^{(p)}$ is defined by the generalized polynomial chaos of maximal degree p , then this determines the length of the finite-term approximation to $a(\cdot, \cdot)$. In particular, it follows from properties of orthogonal polynomials that the triple product $\langle \psi_r \psi_q \psi_\ell \rangle = 0$ for all indices r corresponding to polynomials of degree greater than $2p$ [19]. Thus, the number of terms in (16.17) is $\hat{m} = \binom{m+2p}{2p}$.

It is possible to develop preconditioning strategies that are effective with respect to iteration counts to solve the analogue of (16.5) in this setting. In particular, let $a_0(\mathbf{x}) \equiv \langle a(\mathbf{x}, \cdot) \rangle$ denote the mean of $a(\mathbf{x}, \cdot)$, and let $A_0^{(h)}$ denote the matrix obtained from finite element discretization of the diffusion operator with diffusion coefficient a_0 . Then, the analogue of the mean-based preconditioner described above is $Q_0^{(hp)} \equiv I \otimes A_0^{(h)}$, and the analysis of [21] establishes mesh-independent conditioning of the preconditioned operator $(Q_0^{(hp)})^{-1} A^{(hp)}$.

Unfortunately, there are serious drawbacks to this approach. The coefficient matrices of (16.18) are considerably more dense than those of (16.7). Indeed, $A^{(hp)}$ of (16.17) is *block dense*; that is, most blocks of order N_x are nonzero [19]. Although each individual block has sparsity structure like that arising from finite element discretization in space, most blocks are nonzero. Equivalently, the analogue of Fig. 16.1 would be dense. As a result, the cost of a matrix-vector product required by a Krylov subspace method is of order $O(N_x N_\xi^2)$. Even with an optimally small number of iterations, the costs of a preconditioned iterative solver cannot scale linearly with the number of unknowns, $N_x N_\xi$. The mean-based preconditioned operator is also ill conditioned with respect to the polynomial degree p [23]. Thus, the stochastic Galerkin method is less effective for problems whose dependence on ξ is nonlinear than in the linear case.

There is a way around this in the particular setting where the diffusion coefficient is the exponential of a (smooth enough) random field. Our description follows [30]. Consider the equation

$$-\nabla \cdot (\exp(c) \nabla u) = f.$$

Premultiplying by $\exp(-c)$ and applying the product rule to the divergence operator transforms this problem to

$$-\Delta u + \mathbf{w} \cdot \nabla = f \exp(-c) \quad (16.19)$$

where $\mathbf{w} = -\nabla c$. Now suppose that $c = c(\mathbf{x}, \xi)$ is a random field that is well approximated by a finite-term KL expansion, i.e.,

$$c(\mathbf{x}, \xi) = c_0(x) + \sum_{r=1}^m c_r(x) \xi_r. \quad (16.20)$$

A typical example is where c is a finite-term approximation to a normally distributed random field, so that $a = \exp(c(\mathbf{x}, \xi))$ is essentially of log-normal form. Then, (16.19) is a convection-diffusion equation with convection coefficient

$$\mathbf{w} = \nabla c(\mathbf{x}, \xi) = \nabla c_0(\mathbf{x}) + \sum_{r=1}^m \nabla c_r(x) \xi_r, \quad (16.21)$$

a random field that is linear in the parameters $\{\xi_r\}$. The stochastic Galerkin method can be applied in the identical way it was used for the diffusion equation. The coefficient matrix has the form

$$A^{(hp)} = G_0^{(p)} \otimes (L^{(h)} + N_0^{(h)}) + \sum_{r=1}^m G_r^{(p)} \otimes N_r^{(h)},$$

where

$$[L^{(h)}]_{jk} = \int_{\mathcal{D}} \nabla \phi_k(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x}, \quad [N_r^{(h)}]_{jk} = - \int_{\mathcal{D}} \phi_j(\mathbf{x}) \nabla c_r \cdot \nabla \phi_k(\mathbf{x}) d\mathbf{x},$$

and, most importantly, $\{G_r^{(p)}\}$ are as in (16.7).

Remark 1. Although this method avoids the difficulties associated with nonlinear dependence on parameters, it is somewhat less general than a more direct approach. In particular, if (16.20) is a truncated approximation to a convergent series, then the series expansion of the gradient ∇c , for which (16.21) is an approximation, must also be convergent. This is true if and only if ∇c_0 and the second derivatives of the covariance function for c are continuous; see, e.g., [3, Sect. 4.3]. It holds, for example, for covariance functions of the form $\exp(-\|x - y\|_2^2)$ but not for those of the form $\exp(-\|x - y\|_2)$.

Any of the solution methods developed for problems that depend linearly on ξ are applicable in this setting, with slight modifications. For example, the mean-based preconditioner is $Q_0^{(hp)} \equiv G_0^{(p)} \otimes (L^{(h)} + N_0^{(h)})$. Since $A^{(hp)}$ is nonsymmetric, this preconditioner must be used with a Krylov subspace method such as GMRES designed for nonsymmetric systems. Moreover, $L^{(h)} + N_0^{(h)}$ is a discrete convection-diffusion operator, so that applying the preconditioner requires the (approximate) solution of N_ξ decoupled such operators. There are robust methods for solving the discrete convection-diffusion equation, using multigrid (see, e.g., [7, 32]), so the presence of this operator is not a significant drawback of this approach.

Convergence analysis of the GMRES method typically takes the form of bounds on the eigenvalues of the preconditioned operator $(Q_0^{(hp)})^{-1} A^{(hp)}$, which determines bounds on the asymptotic convergence factor for GMRES. The proof and additional details can be found in [30].

Theorem 4. *If the matrix N_0 has positive semi-definite symmetric part, then the eigenvalues of the mean-based preconditioned operator $(Q_0^{(hp)})^{-1} A^{(hp)}$ are contained in the circle*

$$\left\{ z \in \mathbb{C} : |z - 1| \leq 2c \eta_p \left(\sum_{r=1}^m \|\nabla a_m\|_\infty, \right) \right\},$$

where η_p is as in Lemma 1 and $c > 0$ is a constant independent of h and p .

4 Conclusion

This chapter contains a description of some of the main approaches for solving the coupled system of equations associated with stochastic Galerkin discretization of parameter-dependent partial differential equations. This “intrusive” discretization strategy gives rise to large algebraic systems of linear equations, but there is a variety of computational algorithms available that, especially for problems that depend linearly on the random parameters, offer the prospect of rapid solution. Indeed, through the use of such solvers, Galerkin methods are competitive with other ways of handling stochastic components of such models, such as stochastic collocation methods [6]. Their utility for more complex models remains an open question.

References

1. Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **45**(3), 1005–1034 (2007)
2. Braess, D.: *Finite Elements*. Cambridge University Press, London (1997)
3. Christakos, G.: *Random Field Models in Earth Sciences*. Academic, New York (1992)
4. Elman, H., Furnival, D.: Solving the stochastic steady-state diffusion problem using multigrid. *IMA J. Numer. Anal.* **27**, 675–688 (2007)
5. Elman, H.C., Furnival, D.G., Powell, C.E.: H(div) preconditioning for a mixed finite element formulation of the diffusion problem with random data. *Math. Comput.* **79**, 733–760 (2009)
6. Elman, H.C., Miller, C.W., Phipps, E.T., Tuminaro, R.S.: Assessment of collocation and Galerkin approaches to linear diffusion equations with random data. *Int J. Uncertain. Quantif.* **1**, 19–33 (2011)
7. Elman, H.C., Silvester, D.J., Wathen, A.J.: *Finite Elements and Fast Iterative Solvers*, 2nd edn. Oxford University Press, Oxford (2014)
8. Ernst, O.G., Ullmann, E.: Stochastic Galerkin matrices. *SIAM J. Matrix Anal. Appl.* **31**, 1818–1872 (2010)
9. Ernst, O.G., Powell, C.E., Silvester, D.J., Ullmann, E.: Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data. *SIAM J. Sci. Comput.* **31**, 1424–1447 (2009)
10. Ghanem, R., Spanos, P.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
11. Ghanem, R.G., Kruger, R.M.: Numerical solution of spectral stochastic finite element systems. *Comput. Methods Appl. Mech. Eng.* **129**, 289–303 (1996)

12. Grigoriu, M.: Probabilistic models for stochastic elliptic partial differential equations. *J. Comput. Phys.* **229**, 8406–8429 (2010)
13. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, New York (1991)
14. Horn, R.A., Johnson, C.R.: *Topics in Matrix Analysis*. Cambridge University Press, New York (1991)
15. Keese, A.: Numerical solution of systems with stochastic uncertainties. PhD thesis, Universität Braunschweig, Braunschweig (2004)
16. Le Maître, O.P., Knio, O.M., Debusschere, B.J., Najm, H.N., Ghanem, R.G.: A multigrid solver for two-dimensional stochastic diffusion equations. *Comput. Methods Appl. Mech. Eng.* **192**, 4723–4744 (2003)
17. Le Maître, O.P., Knio, O.M.: *Spectral Methods for Uncertainty Quantification*. Springer, New York (2010)
18. Lord, G.J., Powell, C.E., Shardlow, T.: *An Introduction to Computational Stochastic PDEs*. Cambridge University Press, London (2014)
19. Matthies, H.G., Keese, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**, 1295–1331 (2005)
20. Pellissetti, M.F., Ghanem, R.G.: Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Adv. Eng. Softw.* **31**, 607–616 (2000)
21. Powell, C.E., Elman, H.C.: Block-diagonal preconditioning for spectral stochastic finite element systems. *IMA J. Numer. Anal.* **29**, 350–375 (2009)
22. Powell, C.E., Silvester, D.J.: Preconditioning steady-state Navier-Stokes equations with random data. *SIAM J. Sci. Comput.* **34**, A2482–A2506 (2012)
23. Powell, C.E., Ullmann, E.: Preconditioning stochastic Galerkin saddle point matrices. *SIAM J. Matrix Anal. Appl.* **31**, 2813–2840 (2010)
24. Rosseel, E., Vandewalle, S.: Iterative methods for the stochastic finite element method. *SIAM J. Sci. Comput.* **32**, 372–397 (2010)
25. Ruge, J.W., Stüben, K.: Algebraic multigrid (AMG). In: McCormick, S.F. (ed.) *Multigrid Methods, Frontiers in Applied Mathematics*, pp. 73–130. SIAM, Philadelphia (1987)
26. Saynaeve, B., Rosseel, E., b Nicolai, Vandewalle, S.: Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *J. Comput. Phys.* **224**, 132–149 (2007)
27. Smith, B., Bjørstad, P., Gropp, W.: *Domain Decomposition*. Cambridge University Press, Cambridge (1996)
28. Sousedík, B., Ghanem, R.G., Phipps, E.T.: Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods. *Numer. Linear Algebra Appl.* **21**, 136–151 (2014)
29. Ullmann, E.: A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. *SIAM J. Sci. Comput.* **32**, 923–946 (2010)
30. Ullmann, E., Elman, H.C., Ernst, O.G.: Efficient iterative solvers for stochastic Galerkin discretizations of log-transformed random diffusion problems. *SIAM J. Sci. Comput.* **34**, A659–A682 (2012)
31. Van Loan CF, Pitsianis, N.: Approximation with Kronecker products. In: Moonen, M.S., Golub, G.H., de Moor, B.L.R. (eds.) *Linear Algebra for Large Scale and Real-time Applications*, pp. 293–314. Kluwer, Dordrecht (1993)
32. Wesseling, P.: *An Introduction to Multigrid Methods*. John Wiley & Sons, New York (1992)
33. Xiu, D.: *Numerical Methods for Stochastic Computations*. Princeton University Press, Princeton (2010)
34. Xiu, D., Karniadakis, G.E.: Modeling uncertainty in steady-state diffusion problems using generalized polynomial chaos. *Comput. Methods Appl. Mech. Eng.* **191**, 4927–4948 (2002)
35. Zhang, D.: *Stochastic Methods for Flow in Porous Media. Coping with Uncertainties*. Academic, San Diego (2002)

Bert Debusschere

Abstract

Polynomial chaos (PC)-based intrusive methods for uncertainty quantification reformulate the original deterministic model equations to obtain a system of equations for the PC coefficients of the model outputs. This system of equations is larger than the original model equations, but solving it once yields the uncertainty information for all quantities in the model. This chapter gives an overview of the literature on intrusive methods, outlines the approach on a general level, and then applies it to a system of three ordinary differential equations that model a surface reaction system. Common challenges and opportunities for intrusive methods are also highlighted.

Keywords

Galerkin projection • Intrusive spectral projection • Polynomial chaos

Contents

1	Introduction	618
2	Theory and Algorithms	619
3	Example: Intrusive Propagation of Uncertainty Through ODEs	625
4	Challenges and Opportunities	631
5	Software for Intrusive UQ	632
6	Conclusions	632
	Cross-References	633
	References	633

B. Debusschere (✉)

Mechanical Engineering, Sandia National Laboratories, Livermore, CA, USA

Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA

e-mail: bjdebus@sandia.gov

1 Introduction

Polynomial chaos expansions (PCEs) are spectral representations of random variables in terms of a set of polynomial basis functions that are orthogonal with respect to the density of a reference random variable. By treating uncertain quantities as random variables, PCEs provide a convenient way to represent uncertainty in model parameters or other inputs such as boundary and initial conditions. The current chapter focuses on propagating such input uncertainty through a forward model to obtain a PCE for the quantities of interest (QoIs) that are predicted by the model.

There are many approaches for the forward propagation of uncertainty, but largely they can be divided into collocation methods, which match a PCE to the forward model output for sampled values of the input uncertainty, and Galerkin projection methods, which obtain a PCE by projecting the model output onto the space covered by the PC basis functions. Within the family of Galerkin projection methods, there are Nonintrusive spectral projection (NISP) methods, which perform the Galerkin projection using samples of the model output, for specific values of the uncertain inputs. The focus of this section, however, is on intrusive spectral projection (ISP) methods, which apply the Galerkin projection to the forward model as a whole, leading to a reformulated system of equations for the coefficients of the spectral PCE for all uncertain model variables. The need to reformulate the system of equations is what gives this method the label of *intrusive*. All other sampling-based approaches can rely on the original deterministic implementation of the forward model to generate samples, thereby making them *non-intrusive* to the forward model.

The foundations for spectral representations of random variables were laid out in a 1938 paper by Wiener [56] and brought into the engineering community for the purpose of intrusive uncertainty quantification in the early 1990s [13]. In the early 2000s, PCEs were generalized to a wider family of basis types [57], and ISP was extended to a wider range of functions including transcedentals [8]. This allowed application to a wider class of problems, including fluid flow [15, 16, 29, 58], multi-physics microfluidics [7], and chemical reactions [29, 51], as well as many others.

Challenges encountered in these applications led to many advances in the use of PCEs for representing random variables. One development is the use of decompositions of the stochastic space, with local PCEs defined in each region, rather than global PCEs, which are defined over the full stochastic domain. This stochastic space decomposition allows the effective representation of random variables with very challenging distributions (e.g., multimodal distributions) [18, 19, 21, 38, 49, 53–55]. Other developments involve the use of variable transformations [22, 37], custom solvers, and preconditioners [10, 17, 24, 40, 41, 47, 48] to make the Galerkin-projected systems of equations easier or faster to solve, as well as methods to automate the construction of the Galerkin-projected systems [32, 33].

Clearly, intrusive uncertainty propagation is a very active research field with rich mathematical, algorithmic, and computational developments. The aim of this chapter is to give an introduction to the foundations of ISP. As such, rather than

trying to cover all of the material in this research field, it focuses on the Galerkin projection of governing equations in the context of global PCEs only.

The next section outlines the main theoretical and algorithmic concepts for ISP methods, followed by a specific example of propagation of parametric uncertainty through a set of ordinary differential equations (ODEs). After that, some of the challenges as well as opportunities for intrusive spectral propagation of uncertainty are covered. Finally, pointers are provided to some software packages that are set up for ISP uncertainty quantification.

2 Theory and Algorithms

Consider spectral PCEs following Wiener [56]. Let u be a real second-order random variable, i.e., a Lebesgue-measurable mapping from a probability space (Ω, Θ, P) into \mathbb{R} , where Ω is the set of elementary events, $\Theta(\xi)$ is a σ -algebra on Ω generated by the germ ξ , and P is a probability measure on (Ω, Θ) [13]. Such a random variable can be represented by a PCE as follows [11, 13, 56]:

$$u = \sum_{k=0}^{\infty} u_k \psi_k(\xi) \quad (17.1)$$

where the functions ψ_k are orthogonal polynomials of a set of random variables ξ and the u_k are deterministic coefficients, referred to as PC coefficients in this section. The random variables ξ in the germ of the PCE have zero mean and a unit standard deviation. The polynomials ψ_k are orthogonal with respect to the density $w(\xi)$ of their germ:

$$\langle \psi_i \psi_j \rangle \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi_i(\xi) \psi_j(\xi) w(\xi) d\xi = \delta_{ij} \langle \psi_i^2 \rangle \quad (17.2)$$

In the original Gauss-Hermite PCE type, the basis functions are Hermite polynomials as a function of Gaussian random variables ξ . Other commonly used choices are Legendre polynomials as a function of uniform random variables, referred to as Legendre-Uniform PCEs.

In practice, the choice of the germ and the associated basis functions affects the ease of representing the random variables of interest. For example, if a random variable can possibly take on any value on the real line, then a Gauss-Hermite expansion is often preferred since the basis functions have infinite support. However, if a random variable is subject to physical bounds on its possible values, then a basis with compact support, such as the Legendre-Uniform PCEs, is more convenient. A compact support basis makes it much easier to enforce hard bounds on the values that a PCE can take on. Sometimes quantities have a lower bound, but no upper bound, such as the temperature T in a chemically reacting system. In this case, a gamma-Laguerre PCE may be the most appropriate as the gamma distribution has support from 0 to $+\infty$. The gamma distribution can also be tailored to have a tail

that extends to higher values of ξ in case random variables with long tails need to be represented. Many other choices of random variables and associated polynomials are available, as detailed in [57].

A related question is how many basis terms are needed to accurately represent a random variable with a PCE. In general, the more similar the germ is to the random variable that needs to be represented, the fewer terms will be needed. For example, to represent a normally distributed random variable u with mean μ and standard deviation σ , a Gauss-Hermite expansion only needs two terms

$$u = \mu + \sigma \xi \quad (17.3)$$

while a Legendre-Uniform expansion would need many more terms. Even a 10th order Legendre-Uniform expansion offers only a very crude approximation [43].

While this rule of thumb is quite intuitive, the formal study of the convergence of PCEs is not as straightforward. For Gauss-Hermite expansions, a lot of convergence results are available, but this is not the case for most other types of bases. A careful analysis is presented by Ernst *et al.* [11]. One of the key findings in this chapter is that the distribution of a random variable needs to be uniquely determined by its moments in order for this random variable to be a valid germ for a PCE. One notable example that fails this test is a lognormal random variable. Further, for a given ξ , a proof is provided that shows that any random variable with finite variance that resides in the probability space with a Sigma algebra that is generated by ξ can be represented with a PCE that has ξ as its germ. While these conditions are very informative, it is clear however that the formal convergence of PCEs is a topic of ongoing research.

In practice, the choice of the PCE basis type and the order of the representation are often an engineering choice that depends on the amount of information provided about the random variable that needs to be represented.

To determine the PC coefficients for an arbitrary random variable u , the inner product (17.2) can be used as follows: First, multiply both sides of Eq. (17.1) with the basis function ψ_k , dropping the dependence on ξ for clarity of notation:

$$\psi_k u = \psi_k \sum_{i=0}^{\infty} u_i \psi_i \quad (17.4)$$

Then, move ψ_k inside the summation on the right-hand side, and take the expectation:

$$\langle \psi_k u \rangle = \left\langle \sum_{i=0}^{\infty} u_i \psi_i \psi_k \right\rangle \quad (17.5)$$

After rearranging some terms and recognizing that u_i does not depend on ξ , it can be factored outside of the expectation operator:

$$\langle \psi_k u \rangle = \sum_{i=0}^{\infty} u_i \langle \psi_i \psi_k \rangle \quad (17.6)$$

Taking into account the orthogonality of Eq. (17.2):

$$u_k = \frac{\langle u \psi_k \rangle}{\langle \psi_k^2 \rangle} \quad (17.7)$$

In other words, the PC coefficients u_k can be obtained by projecting the random variable u onto the space spanned by the PC basis functions, which is commonly referred to as a Galerkin projection.

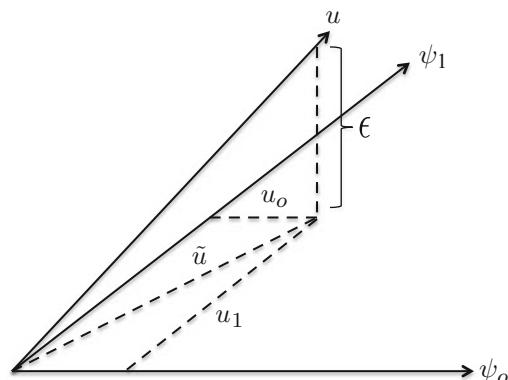
Figure 17.1 graphically illustrates the Galerkin projection for a two-term PCE. In practice, the PCE summation is generally truncated at some suitable high order, such that

$$u \approx \tilde{u} = \sum_{k=0}^P u_k \psi_k(\xi) \quad (17.8)$$

One nice property of the Galerkin projection is that the residual ϵ between the random variable u and its PC representation \tilde{u} is orthogonal to the space covered by the basis functions. In essence, \tilde{u} is the best possible representation of u with the given basis set (in the corresponding ℓ_2 norm on that same space).

Note that so far, the dimensionality of the PCE germ ξ has not been specified. In general, the dimensionality of ξ corresponds to the number of degrees of freedom in the random variable that is represented by the PCE. For example, if a quantity depends on two other uncertain inputs, then its PCE would depend on $\xi = \{\xi_1, \xi_2\}$. In a more general context, consider a random field, which has infinitely many degrees of freedom. To properly capture random fields with finite dimensional representations, a commonly used tool is the Karhunen-Loève (KL) expansion [13, 26]. The KL expansion represents a random field in terms of the eigenfunctions of its covariance matrix, multiplied with random variables that are uncorrelated. If the spectrum of the eigenvalues of the covariance matrix decays rapidly, then only a few eigenfunctions are needed to properly represent the random field with a truncated KL expansion. This generally occurs when the random field

Fig. 17.1 Galerkin projection of a random variable u onto the space covered by the PC basis functions



has long correlation lengths. However, when the random field has short correlation lengths, its realizations will have a lot of small-scale variability, which results in a slowly decaying eigenspectrum. In this case, many more terms are needed in the KL expansion to properly capture the random field, and a high-dimensional spectral representation is required.

As far as ISP is concerned, however, all methods described here apply naturally to PCEs of any dimension. The only thing that changes is the number of terms $P + 1$ in the PCE. As such, in this work, all PCEs will be expressed as a function of the germ ξ without specifying its dimensionality. In many cases, ξ will even be omitted from the notations for clarity.

In intrusive spectral uncertainty propagation, the Galerkin projection (17.7) is applied to the model equations in order to obtain a set of reformulated equations in the PCE coefficients. Consider, e.g., a simple ODE with an uncertain parameter λ .

$$\frac{du}{dt} = \lambda u \quad (17.9)$$

Assume a PCE is specified for λ :

$$\lambda = \sum_{i=0}^P \lambda_i \Psi_i(\xi) \quad (17.10)$$

Since the uncertainty in λ will create uncertainty in u , a PCE for u is specified:

$$u(t) = \sum_{j=0}^P u_j(t) \Psi_j(\xi) \quad (17.11)$$

with unknown coefficients $u_j(t)$. The task of forward propagation of uncertainty then is to determine the unknown PC coefficients $u_i(t)$, given the known coefficients λ_i and the governing Equation (17.9).

To accomplish this, substitute the PCEs for u and λ into the governing Equation (17.9):

$$\frac{d}{dt} \left(\sum_{j=0}^P u_j(t) \Psi_j(\xi) \right) = \left(\sum_{i=0}^P \lambda_i \Psi_i(\xi) \right) \left(\sum_{j=0}^P u_j(t) \Psi_j(\xi) \right) \quad (17.12)$$

and rearrange terms to get

$$\sum_{j=0}^P \frac{du_j(t)}{dt} \Psi_j(\xi) = \sum_{i=0}^P \sum_{j=0}^P \lambda_i u_j(t) \Psi_i(\xi) \Psi_j(\xi) \quad (17.13)$$

The next step applies a Galerkin projection onto the spectral basis functions by multiplying both sides of the equation with Ψ_k and taking the expectation with respect to the basis germ ξ :

$$\left\langle \sum_{j=0}^P \frac{du_j(t)}{dt} \Psi_j(\xi) \Psi_k(\xi) \right\rangle = \left\langle \sum_{i=0}^P \sum_{j=0}^P \lambda_i u_j(t) \Psi_i(\xi) \Psi_j(\xi) \Psi_k(\xi) \right\rangle \quad (17.14)$$

Since only the basis functions depend on ξ , the expectation operator can be moved inside the summations and products to yield

$$\sum_{j=0}^P \frac{du_j(t)}{dt} \langle \Psi_j(\xi) \Psi_k(\xi) \rangle = \sum_{i=0}^P \sum_{j=0}^P \lambda_i u_j(t) \langle \Psi_i(\xi) \Psi_j(\xi) \Psi_k(\xi) \rangle \quad (17.15)$$

Given the orthogonality of the basis functions, Equation (17.2), and dropping the dependence on ξ for notational clarity, this can be simplified to

$$\frac{du_k}{dt} \langle \Psi_k^2 \rangle = \sum_{i=0}^P \sum_{j=0}^P \lambda_i u_j \langle \Psi_i \Psi_j \Psi_k \rangle \quad (17.16)$$

After dividing by $\langle \Psi_k^2 \rangle$, this becomes

$$\frac{du_k}{dt} = \sum_{i=0}^P \sum_{j=0}^P \lambda_i u_j C_{ijk} \quad (17.17)$$

where

$$C_{ijk} = \frac{\langle \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k^2 \rangle} \quad (17.18)$$

Note that Equation (17.17), when written for $k = 0, \dots, P$ represents a system of $P + 1$ deterministic equations in the coefficients u_k of the PCE for u . The triple products C_{ijk} in these equations do not depend on either u or ξ ; so they can be precomputed at the beginning of a simulation based on the knowledge of the basis functions.

The procedure outlined above for the reformulation of the original governing equation into a set of equations for the PC coefficients can be applied to any model for which a governing equation is available in order to perform intrusive uncertainty quantification. However, the nature of the equations that result from this reformulation depends on the complexity of the original governing equation. Summations and subtractions are straightforward. Products between two uncertain variables can be handled similarly to the example above. For example, for the product of u and v , one gets

$$w_k = \sum_{i=0}^P \sum_{j=0}^P u_i v_j C_{ijk} \quad k = 0, \dots, P \quad (17.19)$$

which can be written symbolically as $w = u * v$ with “ $*$ ” representing a product between two PCEs. However, when three or more variables are multiplied with each other, e.g., in u^3 , the most practical way to proceed is often to perform the operation in pairs, i.e., compute u^3 as $(u*u)*u$. This is often referred to as a *pseudo-spectral* operation, since at each stage in the computation, the intermediate results are projected back onto a PCE of the same order as the basis functions, before the next operation is executed. Alternatively, a fully spectral multiplication can be done,

which multiplies all factors in the product at once and keeps all higher-order terms in the process. However, this leads to the need to compute the norms of the products of many basis functions, which becomes unwieldy and expensive as the number of factors in a product increases [8]. In practice, the pseudo-spectral approach is used most often for its ease of use. However, it is important to choose a PC expansion with a high-enough order to minimize the errors associated with the loss of information when intermediate results with higher-order terms are truncated in the process.

Divisions of variables represented by PC expansions can be handled by constructing an appropriate system of equations. Consider, e.g., $w = u/v$, which can be rewritten as $u = v * w$. By writing out this product using Equation (17.19), and substituting in the known PC expansions for u and v , a system of $P + 1$ equations in the unknown PC coefficients of w is obtained.

Through the sum, product, and division, polynomial expressions of PC expansions as well as ratios of such expressions can be evaluated. However, non-polynomial expressions can be more challenging to compute intrusively. One approach is to expand non-polynomial functions as a Taylor series around the mean of the random variable that is the function argument. This approach is often very effective when the function has a Taylor series that converges rapidly and when the uncertainty in the function argument is low. In the present context, low uncertainty means that realizations of the random variable that is represented with the PCE fall within the radius of convergence of the Taylor series around the mean. For functions that have a very large or infinite radius of convergence, such as the exponential function, this is not an issue, and Taylor series tend to be quite robust. However, other functions, such as the logarithm, have a narrow radius of convergence, which can make convergence tricky. In particular, when Gauss-Hermite expansions are used, the corresponding random variable can in theory take on any value on the real line. In practice, the Taylor series for the logarithm of Gauss-Hermite PCEs will only converge if the uncertainty is very low so that the probability of realizations falling outside the radius of convergence is very low [8].

For some specific functions, namely, those whose derivative can be written as a rational expression of the function outcome or its argument, the PC expansion of the function evaluation can be found through an integration approach. Consider, e.g., the `exp()` function. If $v = e^u$, then $dv = e^u du = v du$. As such,

$$e^b - e^a = \int_a^b v \, du \quad (17.20)$$

where the integration is performed along a suitable path from a to b . A convenient choice is often to choose the mean of b as the starting point for the integration. In essence, $a = b_0$, such that e^a is trivial to compute as the PC coefficient b_0 is a deterministic, scalar number. For more details on the integration approach, as well as illustrations of its applicability, see [8].

The key takeaway of this section is that for many operations, the stochastic reformulation is quite straightforward, allowing the application of the intrusive spectral projection approach to the governing equations of a wide variety of problems. ISP has been applied to elliptic equations [2, 3, 6, 17, 27, 44, 46],

hyperbolic equations [5, 34–36, 52], the Navier-Stokes equations [14–16, 20, 31, 58], systems with oscillatory behavior [4, 23, 28, 60], flow through porous media [12], the heat equation [59], systems with complex multiphysics coupling [7], chemically reacting systems [42], and many other fields. In the next section, it will be applied to a set of ODEs that govern a chemically reacting system.

3 Example: Intrusive Propagation of Uncertainty Through ODEs

To illustrate the intrusive UQ approach on a simple but non-trivial example, consider the system of three ODEs below:

$$\frac{du}{dt} = az - cu - 4duv \quad (17.21)$$

$$\frac{dv}{dt} = 2bz^2 - 4duv \quad (17.22)$$

$$\frac{dw}{dt} = ez - fw \quad (17.23)$$

$$z = 1 - u - v - w \quad (17.24)$$

with initial conditions:

$$u(0) = v(0) = w(0) = 0.0 \quad (17.25)$$

and the following nominal values for the six parameters:

$$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad (17.26)$$

$$d = 1.0 \quad e = 0.36 \quad f = 0.016 \quad (17.27)$$

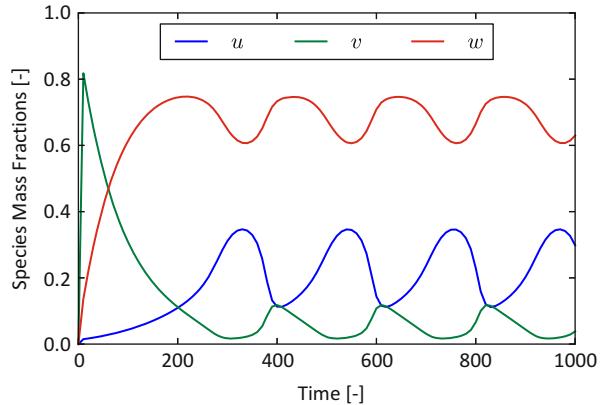
This set of equations models a heterogeneous surface reaction involving a monomer and a dimer that can react with each other after adsorbing onto a surface out of a gas phase. The reaction product is released back into the gas phase. An inert species competes for vacancies on the surface [25, 50]. In the set of equations above, u represents the coverage fraction on the surface of the monomer, v represents the coverage fraction of the dimer, w represents the coverage fraction of the inert species, and z represents the vacant fraction.

While this system of equations is relatively simple, it describes a rich set of dynamics, including oscillatory behavior for $b \in [20.2, 21.2]$. For the nominal values of the parameters, the system behavior is shown in Fig. 17.2.

For values of b outside of this interval, either damped oscillations or asymptotic approaches to steady-state values are observed.

Assuming b is an uncertain input to this system, its uncertainty can be propagated into the model outputs u , v , and w using the intrusive approach. As outlined above,

Fig. 17.2 System dynamics for nominal parameter values



PCEs for the model variables are first postulated, which are then substituted into the governing equations.

$$b = \sum_{i=0}^P b_i \Psi_i(\xi) \quad (17.28)$$

$$u(t) = \sum_{i=0}^P u_i(t) \Psi_i(\xi) \quad v(t) = \sum_{i=0}^P v_i(t) \Psi_i(\xi) \quad (17.29)$$

$$w(t) = \sum_{i=0}^P w_i(t) \Psi_i(\xi) \quad z(t) = \sum_{i=0}^P z_i(t) \Psi_i(\xi) \quad (17.30)$$

Substituting those PCEs first into Equation (17.21), multiplying with Ψ_k , and taking the expectation w.r.t. ξ results in

$$\frac{du}{dt} = az - cu - 4duv \quad (17.31)$$

$$\frac{d}{dt} \sum_{i=0}^P u_i \Psi_i = a \sum_{i=0}^P z_i \Psi_i - c \sum_{i=0}^P u_i \Psi_i - 4d \sum_{i=0}^P u_i \Psi_i \sum_{j=0}^P v_j \Psi_j \quad (17.32)$$

$$\begin{aligned} \left\langle \Psi_k \frac{d}{dt} \sum_{i=0}^P u_i \Psi_i \right\rangle &= \left\langle a \Psi_k \sum_{i=0}^P z_i \Psi_i \right\rangle - \left\langle c \Psi_k \sum_{i=0}^P u_i \Psi_i \right\rangle \\ &\quad - \left\langle 4d \Psi_k \sum_{i=0}^P u_i \Psi_i \sum_{j=0}^P v_j \Psi_j \right\rangle \end{aligned} \quad (17.33)$$

after rearranging terms and invoking the orthogonality of the basis functions, this results in

$$\frac{d}{dt} u_k \langle \Psi_k^2 \rangle = az_k \langle \Psi_k^2 \rangle - cu_k \langle \Psi_k^2 \rangle - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j \langle \Psi_i \Psi_j \Psi_k \rangle \quad (17.34)$$

$$\frac{d}{dt} u_k = az_k - cu_k - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j C_{ijk} \quad (17.35)$$

where Equation (17.35) is solved for $k = 0, \dots, P$, using the same C_{ijk} constants as defined before in Equation (17.18).

For the dimer coverage fraction, the same procedure is applied to Equation (17.22) to get

$$\frac{dv}{dt} = 2bz^2 - 4duv \quad (17.36)$$

$$\frac{d}{dt} \sum_{i=0}^P v_i \Psi_i = 2 \sum_{h=0}^P b_h \Psi_h \sum_{i=0}^P z_i \Psi_i \sum_{j=0}^P z_j \Psi_j - 4d \sum_{i=0}^P u_i \Psi_i \sum_{j=0}^P v_j \Psi_j \quad (17.37)$$

$$\begin{aligned} \left\langle \Psi_k \frac{d}{dt} \sum_{i=0}^P v_i \Psi_i \right\rangle &= \left\langle 2\Psi_k \sum_{h=0}^P b_h \Psi_h \sum_{i=0}^P z_i \Psi_i \sum_{j=0}^P z_j \Psi_j \right\rangle \\ &\quad - \left\langle 4d \Psi_k \sum_{i=0}^P u_i \Psi_i \sum_{j=0}^P v_j \Psi_j \right\rangle \end{aligned} \quad (17.38)$$

which, after rearranging terms and accounting for orthogonality, results in

$$\frac{d}{dt} v_k \langle \Psi_k^2 \rangle = 2 \sum_{h=0}^P \sum_{i=0}^P \sum_{j=0}^P b_h z_i z_j \langle \Psi_h \Psi_i \Psi_j \Psi_k \rangle - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j \langle \Psi_i \Psi_j \Psi_k \rangle \quad (17.39)$$

$$\frac{d}{dt} v_k = 2 \sum_{h=0}^P \sum_{i=0}^P \sum_{j=0}^P b_h z_i z_j \frac{\langle \Psi_h \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k^2 \rangle} - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j \frac{\langle \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k^2 \rangle} \quad (17.40)$$

$$\frac{d}{dt} v_k = 2 \sum_{h=0}^P \sum_{i=0}^P \sum_{j=0}^P b_h z_i z_j D_{hijk} - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j C_{ijk} \quad (17.41)$$

In Equation (17.41), the factors D_{hijk} result from the fact that the original equation contains a product of three uncertain variables, and all terms in the product are retained (full spectral product). However, as discussed in the previous section, precomputing and storing the D_{hijk} can get pretty tedious. Further, even though

the product is performed fully spectrally, only the first $(P + 1)$ terms are retained, since Equation (17.41) is only solved for $k = 0, \dots, P$.

To avoid the need for computing the D_{hijk} factors, a pseudo-spectral approach is used by rewriting Equation (17.22) using an auxiliary variable g as follows:

$$g = z^2 \quad (17.42)$$

$$\frac{dv}{dt} = 2bg - 4d\bar{uv} \quad (17.43)$$

Using this transformation, the product of three uncertain variables has been removed, and the Galerkin-projected equations for the PC coefficients become

$$g_k = \sum_{i=0}^P \sum_{j=0}^P z_i z_j C_{ijk} \quad (17.44)$$

$$\frac{d}{dt} v_k = 2 \sum_{i=0}^P \sum_{j=0}^P b_i g_j C_{ijk} - 4d \sum_{i=0}^P \sum_{j=0}^P u_i v_j C_{ijk} \quad (17.45)$$

which is solved for $k = 0, \dots, P$.

Equation (17.23) for w is similarly reformulated to

$$\frac{d}{dt} w_k = ez_k - fw_k \quad k = 0, \dots, P \quad (17.46)$$

For z , Equation (17.25) becomes

$$z_0 = 1 - u_0 - v_0 - w_0 \quad (17.47)$$

$$z_k = -u_k - v_k - w_k \quad k = 1, \dots, P \quad (17.48)$$

The combined set of reformulated Equations (17.35), (17.44), (17.45), (17.46), (17.47), and (17.48) results in a system of $4(P + 1)$ equations in the unknowns u_k , v_k , w_k , and z_k .

As an illustration, this system of equations is solved here for the case where the parameter b is uniformly distributed with an uncertainty of 0.5% (mean over standard deviation). All other parameters are kept at their nominal values. As b is uniformly distributed and the model has some nonlinearities, a 5th order Legendre-Uniform PC basis set is used to represent the uncertain variables in the system.

Figure 17.3 shows the means and standard deviations for the main quantities of interest. Starting from a deterministic initial condition, the uncertainty in each quantity is initially very small, but it grows as time goes on. This is also visible in Fig. 17.4, which shows the PC coefficients of u as a function of time. The coefficient for the mean, u_0 , shows a steady oscillatory behavior. The first-order coefficient, u_1 , grows as a function of time, and the higher-order coefficients also become nonzero

Fig. 17.3 Mean and standard deviation for u , v , and w up to $t = 1000$

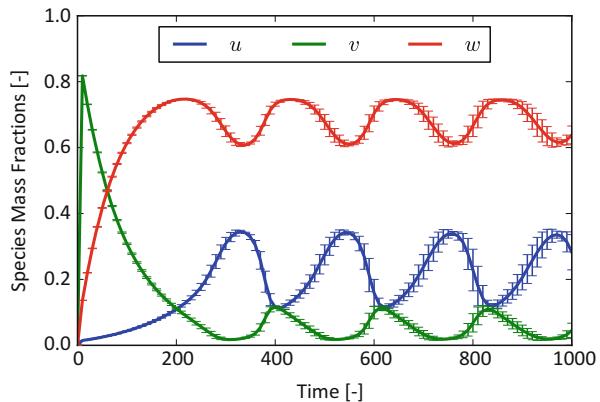
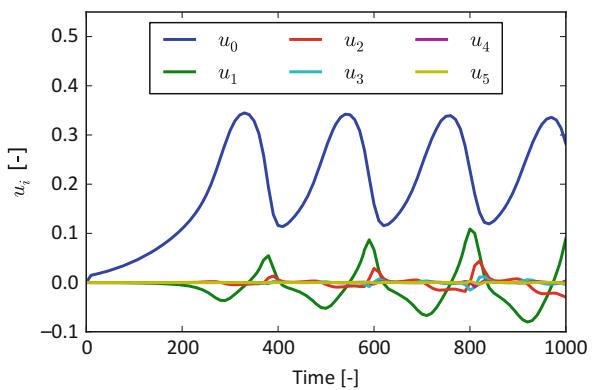


Fig. 17.4 PC coefficients for u as a function of time



later on in the simulation, suggesting that both the uncertainty and skewness of the distribution of u are changing in time. Note that the higher-order terms are the smallest in magnitude when the mean value goes through a local maximum or minimum and the largest in magnitude when the mean value has the largest gradients in time. This indicates that a lot of the uncertainty in u is due to phase shifts in the oscillation due to uncertainty in b .

To get a feel for the distribution of u , Fig. 17.5 shows PDFs of u at selected instances in time. To generate these PDFs, the PCEs of u at the corresponding points in time were evaluated for 100,000 samples of the PCE germ, and the resulting u samples were used as data to get the PDFs with kernel density estimation (KDE). For the points in time where the mean of u has a high value (left plot in Fig. 17.5), the PDFs show a distinct peak, but there is tail toward lower values that gets broader in time. For the cases where the standard deviation has a local maximum (right plot in Fig. 17.5), the PDFs of u show a more and more distinctive bimodal behavior as time goes on.

Figure 17.6 shows a comparison between PDFs for u , generated with intrusive spectral projection (ISP), nonintrusive spectral projection (NISP), and Monte Carlo

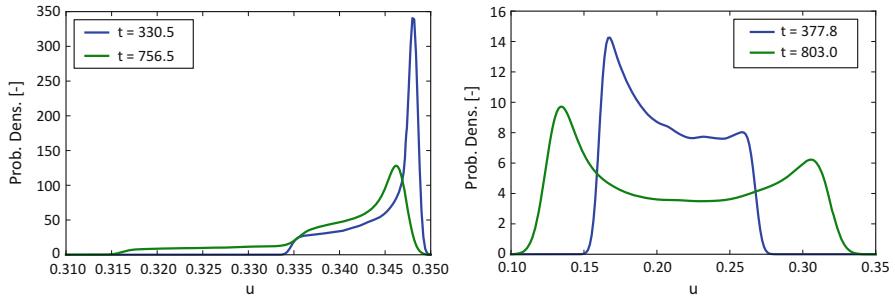
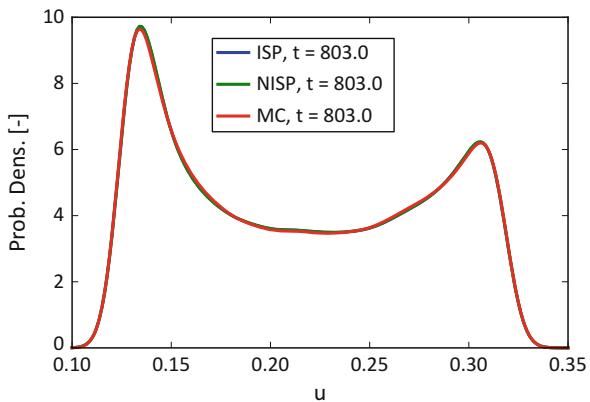


Fig. 17.5 Probability density functions (PDFs) of u at select points in time. *Left:* two instances in time where the mean value of u reaches a maximum. *Right:* two instances in time where the standard deviation of u reaches a maximum

Fig. 17.6 Comparison between the PDFs for u as obtained with intrusive spectral projection (ISP), nonintrusive spectral projection (NISP), and Monte Carlo sampling (MC). There is very good agreement in the results from all three approaches



sampling (MC). For the NISP results, the deterministic system of ODEs was evaluated for six values of b , sampled at the Gauss-Legendre quadrature points. The resulting u samples were then used to project u onto the PC basis function with quadrature integration. For more information on this approach, see ▶ Chap. 21, “Sparse Collocation Methods for Stochastic Interpolation and Quadrature.” In the Monte Carlo approach, the deterministic ODE system was evaluated for 50,000 randomly sampled values of b . The resulting u values were not used to construct a PCE, but instead fed directly into KDE to generate a PDF. All three approaches show very good agreement with each other, which validates the implementation of the ISP and NISP methods and also confirms that the choice of a 5th-order PCE was appropriate for this example.

A more detailed comparison between the ISP and NISP results is shown in Fig. 17.7, which compares the 4th- and 5th-order terms in the PCE of u as a function of time.

The lower-order terms in the PCE of u are indistinguishable (not shown) between the ISP and NISP approaches. The 4th- and 5th-order terms are also nearly indistinguishable, except at late time, when very minor differences become

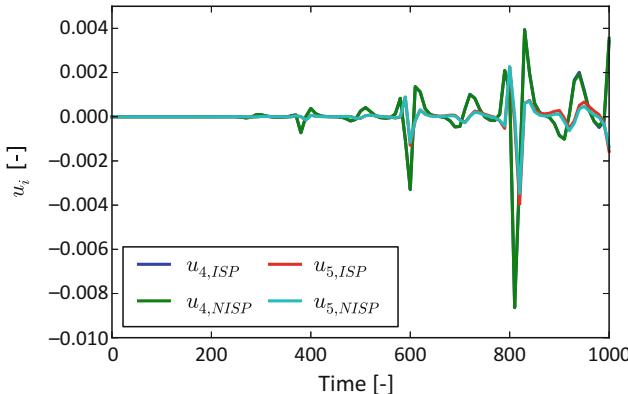


Fig. 17.7 Comparison of the 4th- and 5th-order terms in the PCE of u as a function of time, generated with intrusive (ISP) and nonintrusive (NISP) spectral projection

visible. If the integration time horizon were further extended, those differences would become larger, due to the need to use higher-order representations to properly capture all intermediate variables in the computations and avoid phase errors in the oscillatory dynamics [38]. This touches on some of the challenges with the ISP approach, which are discussed in the next section.

4 Challenges and Opportunities

As illustrated in the previous sections, the intrusive spectral projection approach offers an elegant way to solve for the PC coefficients of uncertain quantities in a computational model. However, in practice, some challenges may emerge.

A first drawback of the intrusive approach is that a governing equation needs to be present for the quantities of interest. This is usually not an issue, but for some applications it is a concern. Consider, e.g., the period of oscillation in the surface reaction example in the previous section. While governing equations are present for u , v , w , and z , there is no governing equation for the period of oscillation. With a sampling-based approach, on the other hand, the periodicity can be extracted from the time trace of each realization, making NISP readily applicable.

A second concern is that the basis and the order of the PCEs in ISP need to be chosen so that they can represent not only the output quantities of interest but all intermediate variables that are needed to compute the outputs as well. This is especially a concern if the model has strong nonlinearities. For example, if any of the computations involve $v = u^{12}$, then the PCE needs to be able to represent the high-order information that will show up in v , even if the outputs of the model only contain lower-order information [8]. Another example would be chemically reacting systems, where quantities such as temperature and species concentrations need to be strictly positive and highly nonlinear reaction dynamics create a lot of higher-order

information, with both of these conditions placing a lot of demands on the spectral representation of uncertain quantities.

The reformulation of the governing equations actually alters the nature of the equations to be solved. Not only is the system of equations larger than the original deterministic system of equations, but its dynamics can change, e.g., through the emergence of spurious eigenvalues [45]. All of this requires careful construction of solvers for the systems of equations resulting from intrusive Galerkin projection [1, 30].

Due to these challenges, intrusive uncertainty quantification approaches have mostly found use in problems with near linear behavior. Also, the fact that intrusive methods require a code rewrite remains a significant barrier to their adoption. Instead, the various nonintrusive approaches are much more widely used, thanks to their relative ease of implementation and the fact that they rely on the same solvers that are used for the original deterministic equations. As of recently, however, intrusive UQ methods are gathering increased attention, for their potential to make better use of extreme scale computing architectures. Using proper preconditioners and solvers, the large systems of equations resulting from the intrusive Galerkin reformulation are often well suited to take advantage of intra-node concurrency, e.g., via multi-core nodes or GPUs [40]. As such, research in intrusive methods for uncertainty propagation is alive and well.

5 Software for Intrusive UQ

Given the complexities of creating and solving the systems of equations for intrusive UQ, and its status more of a research topic rather than a mainstream application method, relatively few software packages are available for intrusive UQ. Two open-source packages are mentioned here: UQTk and Stokhos.

UQTk stands for *UQ Toolkit* [9], containing libraries and tools for the quantification of uncertainty in computational models. It includes tools for intrusive Galerkin projection of many numerical operations. It is geared toward algorithm prototyping, tutorial, and academic use.

Stokhos [39] is part of the Trilinos project. It provides methods for computing well-known intrusive stochastic Galerkin projections such as polynomial chaos and generalized polynomial chaos, interfaces for forming the resulting nonlinear systems, and linear solver methods for solving block stochastic Galerkin linear systems. Stokhos is geared toward handling large systems of equations and, through its connection with other Trilinos libraries, is set up for high performance computing platforms.

6 Conclusions

Intrusive methods for uncertainty quantification provide a powerful and elegant way to propagate uncertainties through computational models. Through the Galerkin projection of the original governing equations of a deterministic problem with

uncertain inputs, a new system of equations is obtained for the PC coefficients of the quantities of interest in the problem. Solving this system once provides the PCEs for all quantities of interest. For most problems, this reformulation of the governing equations is straightforward, and software is available to automate part(s) of this process. There are some challenges, however, for intrusive methods, as the Galerkin projection for some operations is not straightforward, e.g., some transcendental functions, and the resulting systems of equations can be difficult to solve. In recent years, intrusive methods have been garnering renewed attention as the large systems of equations that they generate are often well suited for extreme scale computing hardware.

Cross-References

- ▶ [Embedded Uncertainty Quantification Methods via Stokhos](#)
 - ▶ [Polynomial Chaos: Modeling, Estimation, and Approximation](#)
 - ▶ [Uncertainty Quantification Toolkit \(UQTk\)](#)
-

References

1. Augustin, F., Rentrop, P.: Stochastic Galerkin techniques for random ordinary differential equations. *Numer. Math.* **122**(3), 399–419 (2012)
2. Babuška, I., Tempone, R., Zouraris, G.: Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.* **42**(2), 800–825 (2004)
3. Babuška, I., Tempone, R., Zouraris, G.: Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Eng.* **194**, 1251–1294 (2005)
4. Beran, P.S., Pettit, C.L., Millman, D.R.: Uncertainty quantification of limit-cycle oscillations. *J. Comput. Phys.* **217**(1), 217–47 (2006). doi:10.1016/j.jcp.2006.03.038
5. Chen, Q.Y., Gottlieb, D., Hesthaven, J.: Uncertainty analysis for the steady-state flows in a dual throat nozzle. *J. Comput. Phys.* **204**, 378–398 (2005)
6. Deb, M.K., Babuška, I., Oden, J.: Solution of stochastic partial differential equations using Galerkin finite element techniques. *Comput. Methods Appl. Mech. Eng.* **190**, 6359–6372 (2001)
7. Debusschere, B., Najm, H., Matta, A., Knio, O., Ghanem, R., Le Maître, O.: Protein labeling reactions in electrochemical microchannel flow: numerical simulation and uncertainty propagation. *Phys. Fluids* **15**(8), 2238–2250 (2003)
8. Debusschere, B., Najm, H., Pébay, P., Knio, O., Ghanem, R., Le Maître, O.: Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.* **26**(2), 698–719 (2004)
9. Debusschere, B., Sargsyan, K., Safta, C., Chowdhary, K.: UQ Toolkit. <http://www.sandia.gov/UQToolkit> (2015)
10. Elman, H.C., Miller, C.W., Phipps, E.T., Tuminaro, R.S.: Assessment of collocation and Galerkin approaches to linear diffusion equations with random data. *Int. J. Uncertain. Quantif.* **1**(1), 19–33 (2011)
11. Ernst, O., Mugler, A., Starkloff, H.J., Ullmann, E.: On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.* **46**, 317–339 (2012)

12. Ghanem, R., Dham, S.: Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transp. Porous Media* **32**, 239–262 (1998)
13. Ghanem, R., Spanos, P.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
14. Knio, O., Le Maître, O.: Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dyn. Res.* **38**(9), 616–40 (2006)
15. Le Maître, O., Knio, O., Najm, H., Ghanem, R.: A stochastic projection method for fluid flow I. Basic formulation. *J. Comput. Phys.* **173**, 481–511 (2001)
16. Le Maître, O., Reagan, M., Najm, H., Ghanem, R., Knio, O.: A stochastic projection method for fluid flow II. Random process. *J. Comput. Phys.* **181**, 9–44 (2002)
17. Le Maître, O., Knio, O., Debusschere, B., Najm, H., Ghanem, R.: A multigrid solver for two-dimensional stochastic diffusion equations. *Comput. Methods Appl Mech. Eng.* **192**, 4723–4744 (2003)
18. Le Maître, O., Ghanem, R., Knio, O., Najm, H.: Uncertainty propagation using Wiener-Haar expansions. *J. Comput. Phys.* **197**(1), 28–57 (2004)
19. Le Maître, O., Najm, H., Ghanem, R., Knio, O.: Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Phys.* **197**, 502–531 (2004)
20. Le Maître, O., Reagan, M., Debusschere, B., Najm, H., Ghanem, R., Knio, O.: Natural convection in a closed cavity under stochastic, non-Boussinesq conditions. *SIAM J. Sci. Comput.* **26**(2), 375–394 (2004)
21. Le Maître, O., Najm, H., Pébay P., Ghanem, R., Knio, O.: Multi-resolution analysis scheme for uncertainty quantification in chemical systems. *SIAM J. Sci. Comput.* **29**(2), 864–889 (2007)
22. Le Maître, O.P., Mathelin, L., Knio, O.M., Hussaini, M.Y.: Asynchronous time integration for polynomial chaos expansion of uncertain periodic dynamics. *Discret. Contin. Dyn. Syst.* **28**(1), 199–226 (2010)
23. Lucor, D., Karniadakis, G.: Noisy inflows cause a shedding-mode switching in flow past an oscillating cylinder. *Phys. Rev. Lett.* **92**(15), 154501 (2004)
24. Ma, X., Zabaras, N.: A stabilized stochastic finite element second-order projection method for modeling natural convection in random porous media. *J. Comput. Phys.* **227**(18), 8448–8471 (2008)
25. Makeev, A.G., Maroudas, D., Kevrekidis, I.G.: “Coarse” stability and bifurcation analysis using stochastic simulators: kinetic Monte Carlo examples. *J. Chem. Phys.* **116**(23), 10,083 (2002)
26. Marzouk, Y.M., Najm, H.N.: Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *J. Comput. Phys.* **228**(6), 1862–1902 (2009)
27. Matthies, H.G., Keese, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**, 1295–1331 (2005)
28. Millman, D., King, P., Maple, R., Beran, P., Chilton, L.: Uncertainty quantification with a B-spline stochastic projection. *AIAA J.* **44**(8), 1845–1853 (2006)
29. Najm, H.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Ann. Rev. Fluid Mech.* **41**(1), 35–52 (2009). doi:10.1146/annurev.fluid.010908.165248
30. Najm, H., Valorani, M.: Enforcing positivity in intrusive PC-UQ methods for reactive ODE systems. *J. Comput. Phys.* **270**, 544–569 (2014)
31. Narayanan, V., Zabaras, N.: Variational multiscale stabilized FEM formulations for transport equations: stochastic advection-diffusion and incompressible stochastic Navier-Stokes equations. *J. Comput. Phys.* **202**(1), 94–133 (2005)
32. Pawłowski, R.P., Phipps, E.T., Salinger, A.G.: Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: Template-based generic programming. *Sci. Program.* **20**(2), 197–219 (2012). doi:10.3233/SPR-2012-0350, arXiv:1205.3952v1
33. Pawłowski, R.P., Phipps, E.T., Salinger, A.G., Owen, S.J., Siefert, C.M., Staten, M.L.: Automating embedded analysis capabilities and managing software complexity in multi-

- physics simulation part II: application to partial differential equations. *Sci. Program.* **20**(3), 327–345 (2012). doi:10.3233/SPR-2012-0351, arXiv:1205.3952v1
- 34. Perez, R., Walters, R.: An implicit polynomial chaos formulation for the euler equations. In: Paper AIAA 2005-1406, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno (2005)
 - 35. Pettersson, M.P., Iaccarino, G., Nordström, J.: Polynomial Chaos Methods for Hyperbolic Partial Differential Equations. *Numerical Techniques for Fluid Dynamics Problems in the Presence of Uncertainties*. Springer, Cham (2015)
 - 36. Pettersson, P., Nordström, J., Iaccarino, G.: Boundary procedures for the time-dependent Burgers' equation under uncertainty. *Acta Math. Sci.* **30**(2), 539–550 (2010). doi:10.1016/S0252-9602(10)60061-6
 - 37. Pettersson, P., Iaccarino, G., Nordström, J.: A stochastic Galerkin method for the Euler equations with Roe variable transformation. *J. Comput. Phys.* **257**(PA), 481–500 (2014)
 - 38. Pettit, C.L., Beran, P.S.: Spectral and multiresolution wiener expansions of oscillatory stochastic processes. *J. Sound Vib.* **294**(4/5):752–779 (2006). doi:10.1016/j.jsv.2005.12.043
 - 39. Phipps, E.: Stokhos. <https://trilinos.org/packages/stokhos/> (2015). Accessed 9 Sept 2015
 - 40. Phipps, E., Hu, J., Ostien, J.: Exploring emerging manycore architectures for uncertainty quantification through embedded stochastic Galerkin methods. *Int. J. Comput. Math.* 1–23 (2013). doi:10.1080/00207160.2013.840722
 - 41. Powell, C.E., Elman, H.C.: Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA J. Numer. Anal.* **29**(2), 350–375 (2009)
 - 42. Reagan, M., Najm, H., Debusschere, B., Le Maître O., Knio, O., Ghanem, R.: Spectral stochastic uncertainty quantification in chemical systems. *Combust. Theory Model.* **8**, 607–632 (2004)
 - 43. Sargsyan, K., Debusschere, B., Najm, H., Marzouk, Y.: Bayesian inference of spectral expansions for predictability assessment in stochastic reaction networks. *J. Comput. Theor. Nanosci.* **6**(10), 2283–2297 (2009)
 - 44. Schwab, C., Todor, R.: Sparse finite elements for stochastic elliptic problems. *Numer. Math.* **95**, 707–734 (2003)
 - 45. Sonday, B., Berry, R., Najm, H., Debusschere, B.: Eigenvalues of the Jacobian of a Galerkin-projected uncertain ODE system. *SIAM J. Sci. Comput.* **33**, 1212–1233 (2011)
 - 46. Todor, R., Schwab, C.: Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients. *IMA J. Numer. Anal.* **27**, 232–261 (2007)
 - 47. Tryoen, J., Le Maître, O., Ndjinga, M., Ern, A.: Intrusive Galerkin methods with upwinding for uncertain nonlinear hyperbolic systems. *J. Comput. Phys.* **229**(18), 6485–6511 (2010)
 - 48. Tryoen, J., Le Maître, O., Ndjinga, M., Ern, A.: Roe solver with entropy corrector for uncertain hyperbolic systems. *J. Comput. Appl. Math.* **235**(2), 491–506 (2010)
 - 49. Tryoen, J., Maître, O.L., Ern, A.: Adaptive anisotropic spectral stochastic methods for uncertain scalar conservation laws. *SIAM J. Sci. Comput.* **34**(5), A2459–A2481 (2012)
 - 50. Vigil, R., Willmore, F.: Oscillatory dynamics in a heterogeneous surface reaction: Breakdown of the mean-field approximation. *Phys. Rev. E. Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics* **54**(2), 1225–1231 (1996)
 - 51. Villegas, M., Augustin, F., Gilg, A., Hmaidi, A., Wever, U.: Application of the Polynomial Chaos Expansion to the simulation of chemical reactors with uncertainties. *Math. Comput. Simul.* **82**(5), 805–817 (2012). doi:10.1016/j.matcom.2011.12.001
 - 52. Wan, X., Karniadakis, G.: Long-term behavior of polynomial chaos in stochastic flow simulations. *Comput. Methods Appl. Mech. Eng.* **195**(2006), 5582–5596 (2006)
 - 53. Wan, X., Karniadakis, G.: Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.* **28**(3), 901–928 (2006)
 - 54. Wan, X., Karniadakis, G.E.: An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.* **209**, 617–642 (2005)
 - 55. Wan, X., Xiu, D., Karniadakis, G.: Stochastic solutions for the two-dimensional advection-diffusion equation. *SIAM J. Sci. Comput.* **26**(2), 578–590 (2004)
 - 56. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**, 897–936. doi:10.2307/2371268 (1938)

-
57. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002). doi:10.1137/S1064827501387826
 58. Xiu, D., Karniadakis, G.: Modeling uncertainty in flow simulations via generalized polynomial chaos. *J. Comput. Phys.* **187**, 137–167 (2003)
 59. Xiu, D., Karniadakis, G.: A new stochastic approach to transient heat conduction modeling with uncertainty. *Int. J. Heat Mass Transf.* **46**(24), 4681–4693 (2003)
 60. Xiu, D., Lucor, D., Su, C.H., Karniadakis, G.: Stochastic modeling of flow-structure interactions using generalized polynomial chaos. *ASME J. Fluids Eng.* **124**, 51–59 (2002)

Olivier P. Le Maître and Omar M. Knio

Abstract

We survey the application of multiresolution analysis (MRA) methods in uncertainty propagation and quantification problems. The methods are based on the representation of uncertain quantities in terms of a series of orthogonal multiwavelet basis functions. The unknown coefficients in this expansion are then determined through a Galerkin formalism. This is achieved by injecting the multiwavelet representations into the governing system of equations and exploiting the orthogonality of the basis in order to derive suitable evolution equations for the coefficients. Solution of this system of equations yields the evolution of the uncertain solution, expressed in a format that readily affords the extraction of various properties.

One of the main features in using multiresolution representations is their natural ability to accommodate steep or discontinuous dependence of the solution on the random inputs, combined with the ability to dynamically adapt the resolution, including basis enrichment and reduction, namely, following the evolution of the surfaces of steep variation or discontinuity. These capabilities are illustrated in light of simulations of simple dynamical system exhibiting a bifurcation and more complex applications to a traffic problem and wave propagation in gas dynamics.

Keywords

Multiresolution analysis • Multiwavelet basis • Stochastic refinement • Stochastic bifurcation

O.P. Le Maître (✉)
LIMSI-CNRS, Orsay, France
e-mail: olm@limsi.fr

O.M. Knio
Pratt School of Engineering, Mechanical Engineering and Materials Science, Duke University,
Durham, NC, USA
e-mail: omar.knio@duke.edu

Contents

1	Introduction	638
2	One-Dimensional Multiresolution System	639
2.1	Multiresolution Analysis and Multiresolution Space	640
2.2	Stochastic Element Basis	641
2.3	Multiwavelet Basis	643
3	Multidimensional Extension and Multiscale Operators	646
3.1	Binary-Tree Representation	646
3.2	Multidimensional Extension	648
3.3	Multiscale Operators	652
4	Adaptivity	653
4.1	Coarsening	654
4.2	Anisotropic Enrichment	655
5	Illustrations	659
5.1	Simple ODE Problem	659
5.2	Scalar Conservation Law	662
6	Conclusions	670
	References	671

1 Introduction

A severe difficulty arising in the analysis of stochastic systems concerns the situation where the solution exhibits steep or discontinuous dependence on the random variables that are used to parametrize the uncertain inputs. Well-known examples where these situations arise include complex systems involving shock formation or an energy cascade [1, 2], bifurcations [3, 4], and chemical system ignition [5].

It is well known that these complex settings result in severe difficulties in representing the solution in terms of smooth global functionals. For instance, Gaussian processes frequently exhibit large errors when the solution is discontinuous and generally become impractical. When global polynomial basis functions are used [6], the representation generally suffers from low convergence rate when the solution varies steeply with the random inputs and requires an excessively large basis when discontinuities arise.

In order to address and overcome these difficulties, it is generally desirable to develop methodologies that can provide appropriate resolution in regions of (smooth) steep variation and in the extreme cases isolate the regions where discontinuities occur. When discontinuities occur along hypersurfaces of well-defined regions, one can generally decompose the random parameter domain into regions over which the solution varies smoothly and consequently build a global representation as a collection of smooth representations defined over subdomains.

The domain decomposition approach [4] is relatively straightforward in situations where the hypersurfaces across which the solution varies discontinuously have fixed positions and when the latter are known. When this is not the case, the hypersurfaces must be first localized. Various approaches have been recently developed to address this problem, including detection approaches [7].

In many cases, however, one is faced with the additional complexity that the hypersurfaces of discontinuity can evolve according to the dynamics of the system as well as the uncertain inputs. This necessitates a methodology that can dynamically adapt the representation of the solution response in such a way as to accommodate such behavior. The focus of this chapter is to provide a brief outline of a class of adaptive multiresolution analysis (MRA) methods that can achieve this capability.

In contrast to methods based on global polynomials [6], experiences with the application of MRA methods to uncertain systems have been more limited [3–5, 8–14]. Our goals in the present discussion are twofold: (1) to outline the basic construction of multiresolution representations in a probabilistic setting and their implementation in the context of a Galerkin formalism for uncertainty propagation [15] and (2) to illustrate applications in which dynamic adaptation of the representation is particularly essential.

This chapter is organized as follows. Section 2 provides an introduction to the multiresolution system (MRS) for the case of a single dimension. Section 3 details the construction of multiresolution spaces in higher dimension, relying on binary trees, and introduces essential multiscale operators. Section 4 discusses adaptivity based on the multiresolution framework, detailing crucial criteria for deciding the coarsening and enrichment of the approximation spaces. Illustrations of simulations using multiresolution schemes are provided in Sect. 5, in light of applications to an elementary dynamical system and to a traffic flow problem. Concluding remarks are given in Sect. 6.

2 One-Dimensional Multiresolution System

We are interested in the output of a (generally nonlinear) model involving uncertain input quantities. For simplicity, we shall assume that the uncertain inputs can be parametrized by a finite number of N independent real-valued random variables $\xi := \{\xi_1 \dots \xi_N\}$ with known distributions. We further restrict ourselves to situations where each random variable ξ_i can be mapped to a random variable x_i with uniform distribution on the unit interval $\mathcal{U} \doteq [0, 1]$. In other words, the parameters can be expressed in terms of the random vector $\mathbf{X} \in \mathbb{R}^N$ having independent components and uniform density $p_{\mathbf{X}}$ over the unit hypercube $\Xi \doteq \mathcal{U}^N$:

$$p_{\mathbf{X}}(\mathbf{x} \in \mathbb{R}^N) = \begin{cases} 1 & x_{i=1,\dots,N} \in \mathcal{U} \\ 0 & \text{otherwise} \end{cases}. \quad (18.1)$$

Let $L_2(\Xi)$ be the space of second-order random variables defined on the probability space $\mathcal{P}_{\Xi} := (\Xi, \mathcal{B}_{\Xi}, p_{\mathbf{X}})$, where \mathcal{B}_{Ξ} is the Borel set of Ξ . The

space $L_2(\Xi)$ is equipped with the inner product denoted $\langle \cdot, \cdot \rangle_{\Xi}$ and norm $\|\cdot\|_2$ given by

$$\langle U, V \rangle_{\Xi} \doteq \int_{\Xi} U(x) V(x) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \quad \|U\|_2^2 = \langle U, U \rangle. \quad (18.2)$$

We recall that

$$\|U\|_2 < +\infty \Leftrightarrow U \in L_2(\Xi). \quad (18.3)$$

The objective is then to construct an approximation of the model output, say $U \in \mathbb{R}$, which is a functional of the random input \mathbf{X} . We shall assume that $U : \mathbf{X} \in \Xi \mapsto U(\mathbf{X}) \in L_2(\Xi)$. In practice, the approximation is obtained by projecting U on a finite dimensional subspace of $L_2(\Xi)$. To this end, Hilbertian bases of random functionals in \mathbf{X} spanning $L_2(\Xi)$ are often considered. Upon truncation of the Hilbertian basis, the problem reduces to the determination of a finite number of coordinates defining the approximation of U in the truncated basis. It is obvious that the approximation error depends heavily on the subspace of $L_2(\Xi)$ spanned by the random functionals retained in the approximation. When sufficient smoothness in the mapping $\mathbf{X} \mapsto f(\mathbf{X})$ is expected, it is well known that using smooth spectral functionals in \mathbf{X} , for instance, multivariate polynomials, is an effective approach to achieve high convergence of the approximation error as the dimension of the subspace increases. However, as discussed in the introduction, steep or even non-smooth dependences of $U(\mathbf{X})$ may significantly delay or even compromise the convergence of the approximation. The multiresolution analysis provides a convenient framework to construct an alternative approximation space spanned by piecewise smooth random functionals, and eventually to adapt the approximation subspace to the random function U , therefore minimizing the computational complexity of determining its approximation.

In this section, we restrict ourselves to the case of a single random parameter, $X = X_1$, so $\Xi = \mathcal{U}$. In Sect. 2.1, we introduce the sequence of one-dimensional multiresolution spaces. We then discuss in Sects. 2.2 and 2.3 two different bases for the multiresolution spaces. These two bases have different hierarchical structures that are outlined. The extension of these concepts to higher-dimensional spaces is addressed in Sect. 3.

2.1 Multiresolution Analysis and Multiresolution Space

Let \mathcal{U} be the unit interval. For given integer $k \geq 0$, called the resolution level, we define the partition $\mathcal{P}^{(k)}$ of \mathcal{U} into 2^k non-overlapping subintervals $\mathcal{U}_l^{(k)}$ having equal size,

$$\mathcal{P}^{(k)} = \left\{ \mathcal{U}_l^{(k)}, l = 1, \dots, 2^k \right\}, \quad \mathcal{U}_l^{(k)} \doteq \left[\frac{l-1}{2^k}, \frac{l}{2^k} \right],$$

so we have

$$\bigcup_{l=1}^{2^k} \mathcal{U}_l^{(k)} = \mathcal{U}, \quad \mathcal{U}_l^{(k)} \cap \mathcal{U}_{l'}^{(k)} = \emptyset \text{ for } l \neq l'.$$

The partition $\mathcal{P}^{(k)}$ can also be defined recursively, through successive dyadic partitioning starting from $\mathcal{P}^{(0)} = \mathcal{U}$.

For $No = 0, 1, \dots$ and $k = 0, 1, 2, \dots$, we consider the space $\mathbf{V}_{No}^{(k)}$ of piecewise polynomial functions associated to the partition $\mathcal{P}^{(k)}$ according to:

$$\mathbf{V}_{No}^{(k)} = \left\{ U : x \in \mathcal{U} \mapsto \mathbb{R}; U|_{\mathcal{U}_l^{(k)}} \in \pi_{No} \left(\mathcal{U}_l^{(k)} \right), l = 1, \dots, 2^k \right\}, \quad (18.4)$$

where we have denoted by $\pi_{No}(\mathcal{J})$ the space of polynomials with degree less or equal to No defined over \mathcal{J} .

In other words, the space $\mathbf{V}_{No}^{(k)}$ consists of the functions that are polynomials of degree at most No over each subintervals $\mathcal{U}_l^{(k)}$ of the partition $\mathcal{P}^{(k)}$. Observe that $\dim(\mathbf{V}_{No}^{(k)}) = (No + 1) \times 2^k$. In addition, these spaces have a nested structure in the sense that

$$\mathbf{V}_{No}^{(0)} \subset \mathbf{V}_{No}^{(1)} \subset \mathbf{V}_{No}^{(2)} \subset \dots \quad \text{and} \quad \mathbf{V}_0^{(k)} \subset \mathbf{V}_1^{(k)} \subset \mathbf{V}_2^{(k)} \subset \dots$$

Denoting \mathbf{V}_{No} and $\mathbf{V}^{(k)}$ the union of all spaces,

$$\mathbf{V}_{No} = \overline{\bigcup_{k \geq 0} \mathbf{V}_{No}^{(k)}} \quad \text{and} \quad \mathbf{V}^{(k)} = \overline{\bigcup_{No \geq 0} \mathbf{V}_{No}^{(k)}},$$

it is remarked [16] that these union spaces are dense in $L_2(\mathcal{U})$, the space of square integrable functions equipped with the inner product denoted $\langle \cdot, \cdot \rangle_{\mathcal{U}}$.

We now construct two distinct orthonormal bases for $\mathbf{V}_{No}^{(k)}$.

2.2 Stochastic Element Basis

For the stochastic element (SE) bases, we start from the Legendre basis for $\mathbf{V}_{No}^{(0)}$, that is, the space at resolution level $k = 0$,

$$\mathbf{V}_{No}^{(0)} = \text{span} \{ \mathcal{L}_\alpha, \alpha = 0, \dots, No \}. \quad (18.5)$$

Here, we denoted $\mathcal{L}_\alpha(x)$ the normalized Legendre polynomial of degree α , rescaled to \mathcal{U} , such that

$$\langle \mathcal{L}_\alpha, \mathcal{L}_\beta \rangle_{\mathcal{U}} = \begin{cases} 1, & \alpha = \beta \\ 0, & \alpha \neq \beta \end{cases}, \quad \|\mathcal{L}_\alpha\| = 1. \quad (18.6)$$

Introducing the affine mapping $M_l^{(k)} : \mathcal{U}_l^{(k)} \mapsto \mathcal{U}$ through

$$M_l^{(k)}(x) \doteq (2^k x - l + 1), \quad (18.7)$$

we construct the associated shifted and scaled versions of the Legendre polynomials \mathcal{L}_α as follows:

$$\Phi_{\alpha,l}^{(k)}(x) = \begin{cases} 2^{k/2} \mathcal{L}_\alpha(M_l^{(k)} x), & x \in \mathcal{U}_l^{(k)}, \\ 0, & \text{otherwise.} \end{cases} \quad (18.8)$$

The resulting functions $\Phi_{\alpha,l}^{(k)}(x)$ obviously belong to $\mathbf{V}_{\text{No} \geq \alpha}^{(k)}$, and, for given resolution level k , one can easily verify that they form an orthonormal set

$$\langle \Phi_{\alpha,l}^{(k)}, \Phi_{\beta,l'}^{(k)} \rangle_{\mathcal{U}} = \begin{cases} 1, & \alpha = \beta \text{ and } l = l', \\ 0, & \text{otherwise.} \end{cases} \quad (18.9)$$

Thus, $\{\Phi_{\alpha,l}^{(k)}; \alpha = 0, \dots, \text{No}; l = 1, \dots, 2^k\}$ is an orthonormal basis of $\mathbf{V}_{\text{No}}^{(k)}$ since $\dim(\mathbf{V}_{\text{No}}^{(k)}) = 2^k (\text{No} + 1)$.

Therefore, any function $U \in \mathbf{V}_{\text{No}}^{(k)}$ can be expanded as

$$\mathbf{V}_{\text{No}}^{(k)} \ni U(x) = \sum_{\alpha=0}^{\text{No}} \sum_{l=1}^{2^k} u_{\alpha,l}^{(k)} \Phi_{\alpha,l}^{(k)}(x), \quad (18.10)$$

where the coefficients $u_{\alpha,l}^{(k)}$ are the SE coefficients of U . Also, the orthogonal projection of $U \in L_2(\mathcal{U})$ in $\mathbf{V}_{\text{No}}^{(k)}$, denoted hereafter $\mathbb{P}_{\text{No}}^{(k)} U$, minimizes $\|U - V\|_{\mathcal{U}}$ over all $V \in \mathbf{V}_{\text{No}}^{(k)}$ and has for expression

$$\mathbb{P}_{\text{No}}^{(k)} U(x) = \sum_{\alpha=0}^{\text{No}} \sum_{l=1}^{2^k} u_{\alpha,l}^{(k)} \Phi_{\alpha,l}^{(k)}(x), \quad u_{\alpha,l}^{(k)} = \langle U, \Phi_{\alpha,l}^{(k)} \rangle_{\mathcal{U}}. \quad (18.11)$$

The projection error $U - \mathbb{P}_{\text{No}}^{(k)} U$ can be reduced by considering richer projection spaces, that is, by refining the partition $\mathcal{P}^{(k)}$ (increasing the resolution level k), or the polynomial degree (increasing No), or both.

From Eq. (18.11), we remark that for a fixed resolution level k , the projection coefficient $u_{\alpha,l}^{(k)}$ associated to polynomial degrees $\alpha = 1, \dots, \text{No}$ are not affected when considering spaces with higher polynomial degrees. In other words, increasing the polynomial degree amounts to complement the SE expansion with additional (higher degree), leaving unchanged the previous ones. On the contrary, the whole

SE expansion is affected when one changes the resolution level k , for instance, considering a finer partition $\mathcal{P}^{(k)}$, even if the polynomial degree No is kept constant. This is due to the structure of the SE basis functions which are not orthogonal (in general) for different resolution levels. The absence of orthogonality between SE basis functions at different resolution levels makes it difficult to control and reduce the projection error through partition refinement. This motivates the introduction of hierarchical enrichments having the *orthogonal* properties between resolution levels. Stochastic multiwavelet (MW) expansions exactly achieve this task.

2.3 Multiwavelet Basis

2.3.1 Detail Spaces

Let us denote by $\mathbf{W}_{No}^{(k)}$, $k = 0, 1, 2, \dots$, the subspace corresponding to the orthogonal complement of $\mathbf{V}_{No}^{(k)}$ in $\mathbf{V}_{No}^{(k+1)}$; that is,

$$\mathbf{V}_{No}^{(0)} \oplus \mathbf{W}_{No}^{(k)} = \mathbf{V}_{No}^{(k+1)}, \quad \mathbf{W}_{No}^{(k)} \perp \mathbf{V}_{No}^{(k)}. \quad (18.12)$$

The spaces $\mathbf{W}_{No}^{(k)}$ are called detail spaces; we have

$$\overline{\mathbf{V}_{No}^{(0)} \bigoplus_{k \geq 0} \mathbf{W}_{No}^{(k)}} = L_2(\mathcal{U}). \quad (18.13)$$

The objective is now to construct an orthonormal basis for these detail spaces $\mathbf{W}_{No}^{(k)}$. To this end, we first observe that

$$\dim(\mathbf{W}_{No}^{(k)}) = \dim(\mathbf{V}_{No}^{(k)}) = 2^k (No + 1),$$

since $\dim(\mathbf{V}_{No}^{(k+1)}) = 2^{k+1}(No + 1)$. Focusing on the case $k = 0$, we have to determine orthonormal functions Ψ_α , $\alpha \in [0, No + 1]$, spanning $\mathbf{W}_{No}^{(0)}$. Clearly, $\Psi_\alpha \in \mathbf{V}_{No}^{(1)}$ so each function has an SE expansion of the form

$$\Psi_\alpha = \sum_{\beta=0}^{No} \sum_{l=1}^2 a_{\beta,l}^\alpha \Phi_{\beta,l}^{(1)}. \quad (18.14)$$

As a result, a total of $2(No + 1)^2$ coordinates define the $No + 1$ basis functions spanning the detail space $\mathbf{W}_{No}^{(0)}$.

To derive equations for these coefficients, we enforce the orthonormality conditions for the Ψ_α , that is,

$$\langle \Psi_\alpha, \Psi_\beta \rangle_{\mathcal{U}} = \delta_{\alpha,\beta}, \quad 0 \leq \alpha, \beta \leq No. \quad (18.15)$$

Equation (18.15) then provides $(No + 2)(No + 1)/2$ distinct conditions on the coefficients. Accounting for the orthogonality $\mathbf{W}_{No}^{(0)} \perp \mathbf{V}_{No}^{(0)}$, one obtains another set of $(No + 1)^2$, vanishing moment conditions for the functions Ψ_α :

$$\langle \Psi_\alpha, x^\beta \rangle_{\mathcal{U}} = 0, \quad 0 \leq \alpha, \beta \leq No. \quad (18.16)$$

Orthonormal and vanishing moment properties are the only conditions needed to be enforced for the Ψ_α to obtain an orthonormal basis of $\mathbf{W}_{No}^{(0)}$. Having less conditions than coefficients determining all the Ψ_α reflects the fact that there is an infinite number of orthonormal bases for $\mathbf{W}_{No}^{(0)}$. Higher moment vanishing conditions could then be introduced on some of the functions Ψ_α . This could be particularly useful for the compression of the MW expansions (see the thresholding procedure described below). However, improved vanishing moments properties will not be exploited here, and the orthogonality between functions at different resolution levels is really what matters; therefore, we shall rely in Sect. 2.3.2 on a simple but systematic construction procedure to obtain a set of functions Ψ_α satisfying conditions (18.15) and (18.16).

When the so-called mother functions Ψ_α are obtained, bases for the whole hierarchy of subspaces $\mathbf{W}_{No}^{(k)}$ can readily be defined by means of simple transformations. Using the affine mapping in (18.7), we define the MW basis functions $\Psi_{\alpha,l}^{(k)}$ through

$$\Psi_{\alpha,l}^{(k)}(x) = \begin{cases} 2^{k/2} \Psi_\alpha(M_l^{(k)} x), & x \in \mathcal{U}_l^{(k)}, \\ 0, & \text{otherwise.} \end{cases} \quad (18.17)$$

It is easily shown that

$$\langle \Psi_{\alpha,l}^{(k)}, \Psi_{\beta,l'}^{(k')} \rangle_{\mathcal{U}} = \begin{cases} 1, & \text{if } k = k', \alpha = \beta, l = l' \\ 0, & \text{otherwise.} \end{cases} \quad (18.18)$$

In addition, by construction, we have

$$\langle \Psi_{\alpha,l}^{(k)}, \Phi_{\beta,l'}^{(k')} \rangle_{\mathcal{U}} = 0 \quad \forall k' \leq k \text{ and } l' \leq l. \quad (18.19)$$

As a result, the projection of $U \in L_2(\mathcal{U})$ in $\mathbf{V}_{No}^{(k)}$ can be expressed in the MW basis as

$$\mathbb{P}_{No}^{(k)} U(x) = \sum_{\alpha=0}^{No} u_{\alpha,1}^{(0)} \Phi_{\alpha,1}^{(0)}(x) + \sum_{i=0}^{k-1} \sum_{l=1}^{2^i} \sum_{\alpha=0}^{No} \tilde{u}_{\alpha,l}^{(i)} \Psi_{\alpha,l}^{(i)}(x). \quad (18.20)$$

The first sum in (18.20) corresponds to the projection of U on $\mathbf{V}_{No}^{(0)}$, whereas the remaining contributions are orthogonal details at successive resolution levels $i = 0$

to $i = k - 1$. Owing to the properties of the MW basis functions, the detail coefficients are given by

$$\tilde{u}_{\alpha,l}^{(i)} = \left\langle U, \Psi_{\alpha,l}^{(i)} \right\rangle_{\mathcal{U}} \quad i \geq 0, \alpha = 0, \dots, \text{No} \text{ and } l = 1, \dots, 2^i. \quad (18.21)$$

2.3.2 Construction of the MW Mother Functions

In this section, we briefly summarize a generation procedure proposed in [16] to produce a set of $\text{No} + 1$ orthonormal functions $\Psi_{\alpha} \in \mathbf{V}_{\text{No}}^{(1)}$, satisfying (18.15) and (18.16).

The procedure is initiated with two sets of $\text{No} + 1$ polynomial functions $p_{\alpha}(x) \in \mathbf{V}_{\text{No}}^{(0)}$ and $\tilde{q}_{\alpha}(x) \in \mathbf{V}_{\text{No}}^{(1)}$, for $\alpha = 0, \dots, \text{No}$, respectively defined as:

$$p_{\alpha}(x) = x^{\alpha}, \quad \tilde{q}_{\alpha}(x) = \begin{cases} x^{\alpha}, & 0 \leq x < 1/2 \\ -x^{\alpha}, & \text{otherwise.} \end{cases} \quad (18.22)$$

In a first step, each function \tilde{q}_{α} is orthogonalized with respect to all the functions $p_{\beta}(x)$ for $\beta = 0, \dots, \text{No}$. That is, we determine a new set of functions $q_{\alpha}(x)$ as

$$q_{\alpha}(x) \doteq \tilde{q}_{\alpha}(x) + \sum_{\beta=0}^{\text{No}} c_{\alpha\beta} p_{\beta}(x), \quad (18.23)$$

so that

$$\langle q_{\alpha}, p_{\beta} \rangle_{\mathcal{U}} = 0, \quad \text{for } \alpha, \beta = 0, 1, \dots, \text{No}. \quad (18.24)$$

The resulting functions $\{q_{\alpha}, \alpha = 0, \dots, \text{No}\}$ have clearly the required vanishing moment properties (18.16). In a second step, the functions q_{α} are orthonormalized, for instance, through a standard Gram-Schmidt procedure [17]. Because the orthonormalization involves linear combinations of the initial functions q_i , which all have first $\text{No} + 1$ vanishing moments, the final orthonormal functions also have the desired moment properties. These functions are then the sought functions Ψ_{α} . This constructive procedure can be carried out numerically, using an exact quadrature to compute scalar products between polynomial functions. Since the procedure involves and manipulates functions in $\mathbf{V}_{\text{No}}^{(1)}$, the scalar products are broken into sub-integrals over the two subintervals of $\mathcal{P}^{(1)}$. For instance, Gauss-type quadrature rules can be used over $\mathcal{U}_1^{(1)}$ and $\mathcal{U}_2^{(1)}$ to ensure sufficient polynomial exactness.

Figure 18.1 depicts the MW mother functions Ψ_{α} , $\alpha = 0, \dots, \text{No}$, for different polynomial orders No . The case $\text{No} = 0$ corresponding to the Haar system (piecewise constant approximation) is not shown but corresponds to a single mother function which is the step function (see [3] for more details). The case $\text{No} = 1$ (top left panel) has two functions Ψ_0 and Ψ_1 which are seen to be both piece-polynomials of degree $\text{No} = 1$, with discontinuities of order 1 and 0, respectively, at the center

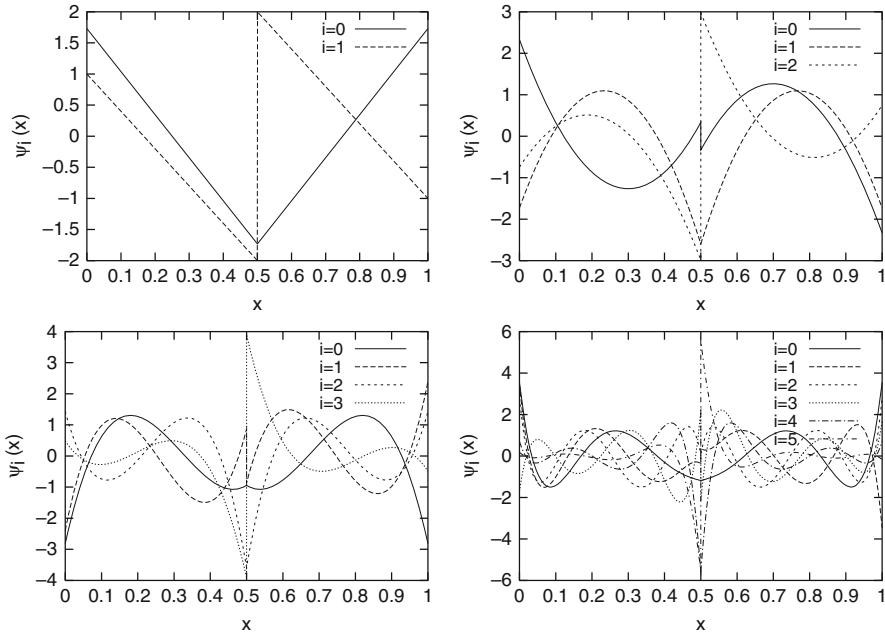


Fig. 18.1 Mother MW functions $\Psi_\alpha(x)$ for polynomial degrees $No = 1, 2, 3$ and 5 (Adapted from [6])

point of \mathcal{U} . For higher polynomial degrees, the functions are similarly piecewise No-degree polynomials with discontinuities at different orders ranging from 0 to No localized at the center of \mathcal{U} .

3 Multidimensional Extension and Multiscale Operators

3.1 Binary-Tree Representation

The partitions $\mathcal{P}^{(k)}$ and their associated approximation spaces can be conveniently represented in terms of binary trees. In a binary tree T , every node has either zero or two children, and every node, except the root node denoted by n_0 , has a unique parent. Nodes are collected in the set $\mathcal{N}(T)$. Nodes with no children are called leaves and are collected in the set $\mathcal{L}(T)$, while nodes with children are collected in the set $\widehat{\mathcal{N}}(T) := \mathcal{N}(T) \setminus \mathcal{L}(T)$. The two children of a node $n \in \widehat{\mathcal{N}}(T)$ are called left and right children (and also sisters) and are denoted by $c^-(n)$ and $c^+(n)$, respectively. The parent of a node $n \in \mathcal{N}(T) \setminus \{n_0\}$ is denoted by $p(n)$. To each node $n \in \mathcal{N}(T)$, we assign an element of a partition $S(n) = \mathcal{U}_i^{(k)}$ as follows. The support of a node is denoted $S(n) = [x_n^-, x_n^+] \subseteq \mathcal{U}$. We set $S(n_0) = \mathcal{U}$. The supports of the other nodes are defined recursively by a dyadic partition of the support of the parent node. Then,

the supports of the left and right children are, respectively, $S(c^-(n)) = [x_n^-, (x_n^- + x_n^+)/2]$ and $S(c^+(n)) = [(x_n^- + x_n^+)/2, x_{n,d}^+]$. This construction leads to a partition of \mathcal{U} in the form

$$\mathcal{U} = \bigcup_{l \in \mathcal{L}(T)} S(l). \quad (18.25)$$

For a node $n \in \mathcal{N}(T)$, its depth $|n|$ is defined as the number of generations it takes to reach n from the root node n_0 . It is readily seen that the support of node n has measure $|S(n)| := 2^{-|n|}$. Finally, for any node $n \in \mathcal{N}(T)$, M_n denotes the affine map from $S(n)$ onto the reference stochastic domain \mathcal{U} .

In practice, we consider binary trees T with a fixed maximum number of successive partitions allowed. This quantity is called the resolution level and is denoted by N_r in the following. As a result, there holds, for all $n \in \mathcal{N}(T)$, $|S(n)| \geq 2^{-N_r}$. A particular case of a binary tree is the complete binary tree where $|S(l)| = 2^{-N_r}$ for all the leaves. As there are 2^{N_r} leaves in a complete binary tree, each of the leaves is identified as an element of the partition \mathcal{P}^{N_r} . An example of a complete binary tree with $N_r = 4$ is provided in the top plot of Fig. 18.2. However, trees need not be complete, and in view of adaptivity, we consider general binary trees with leaves having different depth $|n|$. An example of such incomplete binary tree is reported in the bottom plot of Fig. 18.2.

Even if the binary tree is incomplete, one can identify with each leaf an element $U_i^{(k)}$ of a certain partition \mathcal{P}^k with $k \leq N_r$. As a result, we can associate to a binary tree T an approximation space $V(T)$ which is the space of functions in $L_2(\mathcal{U})$ being polynomials of degree at most N_0 over the support of each leaf of T . Note that the notation does not explicitly mention the polynomial degree as it will be considered

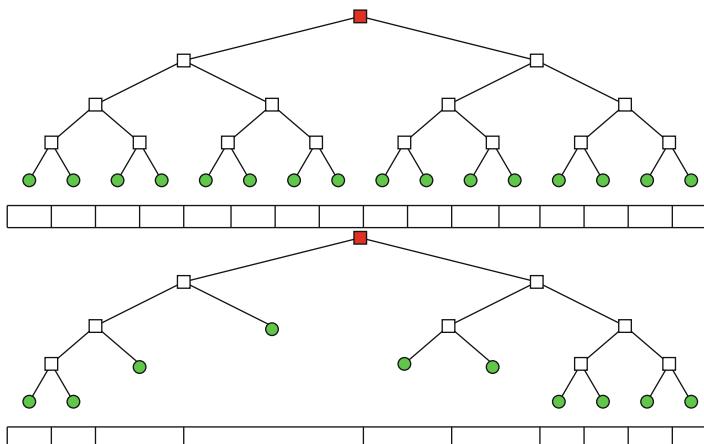


Fig. 18.2 Complete binary tree (top) and incomplete binary tree (bottom) in one dimension; the corresponding partitions of $\mathcal{U} = [0, 1]$ are shown below the trees

and kept constant. On the contrary, we aim at adapting the tree to the functional to be approximated. The SE expansion of $U \in \mathbf{V}(\mathbb{T})$ can further be expressed as

$$\mathbf{V}(\mathbb{T}) \ni U(x) = \sum_{n \in \mathcal{L}(\mathbb{T})} \sum_{\alpha=0}^{N_0} u_\alpha^n \Phi_\alpha^n(x), \quad (18.26)$$

where $u_\alpha^n \doteq \langle U, \Phi_\alpha^n \rangle_{\mathcal{U}}$ for $\alpha = 0, \dots, N_0$ are the SE coefficients associated to the leaf $n \in \mathcal{L}(\mathbb{T})$. Similarly, the MW expansion can be expressed as

$$\mathbf{V}(\mathbb{T}) \ni U(x) = \sum_{\alpha=0}^{N_0} u_\alpha^{n_0} \Phi_\alpha^{n_0}(x) + \sum_{n \in \widehat{\mathcal{N}}(\mathbb{T})} \sum_{\alpha=0}^{N_0} \tilde{u}_\alpha^n \Psi_\alpha^n(x), \quad (18.27)$$

where the detail coefficients of a node, which is not a leaf, are given by $\tilde{u}_\alpha^n \doteq \langle U, \Psi_\alpha^n \rangle_{\mathcal{U}}$ for $\alpha = 0, \dots, N_0$.

3.2 Multidimensional Extension

Using the binary tree structure to represent partitions and define approximation spaces, the extension of the one-dimensional multiresolution framework to higher number of dimensions becomes quite straightforward. We shall now consider the case of an N -dimensional parameter space consisting of the N dimensional hypercube $\mathcal{E} \doteq \mathcal{U}^N$. Regarding the partitioning of \mathcal{E} , instead of extending the previous approach to general N -ary trees, which would become quickly intractable as N increases, we require to keep the binary structure (each node has zero or two children), starting from the root node $n_0 = \mathcal{E}$. This requires the introduction of an indicator for each node $n \in \widehat{\mathcal{N}}(\mathbb{T})$ which keeps track of the direction along which n is partitioned dyadically to produce its two children. Denoted by $d(n) \in \{1 \dots N\}$ the direction along which the dyadic partition of the support $S(n)$ is performed, the left and right children have for respective support

$$S(c^-(n)) = [x_{n,1}^-, x_{n,1}^+] \times \dots \times [x_{n,d(n)}^-, (x_{n,d(n)}^- + x_{n,d(n)}^+)/2] \times \dots \times [x_{n,N}^-, x_N^+]$$

and

$$S(c^+(n)) = [x_{n,1}^-, x_{n,1}^+] \times \dots \times [(x_{n,d(n)}^- + x_{n,d(n)}^+)/2, x_{n,d(n)}^+] \times \dots \times [x_{n,N}^-, x_N^+].$$

The process is illustrated in Fig. 18.3.

There is however an essential difference between one-dimensional and multidimensional binary trees: for $N > 1$, there are in general more than one tree with the same set of leaves, i.e., yielding the same partition of \mathcal{E} . This is illustrated in Fig. 18.4 for $N = 2$. Consequently, we say that two trees T and T' are equivalent if

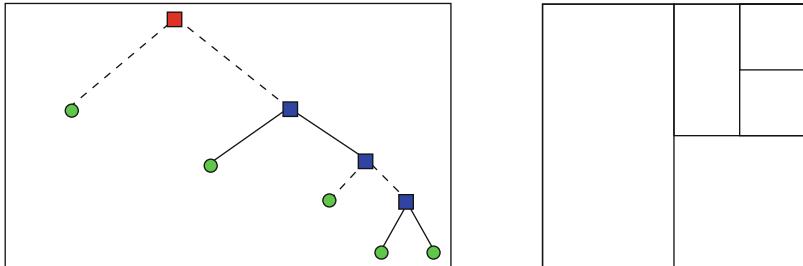


Fig. 18.3 Multidimensional binary tree for $N = 2$ (left). Dash (resp. full) segments represent a partition along the first (resp. second) direction. Corresponding partition of $\mathcal{E} = [0, 1]^2$ (right)

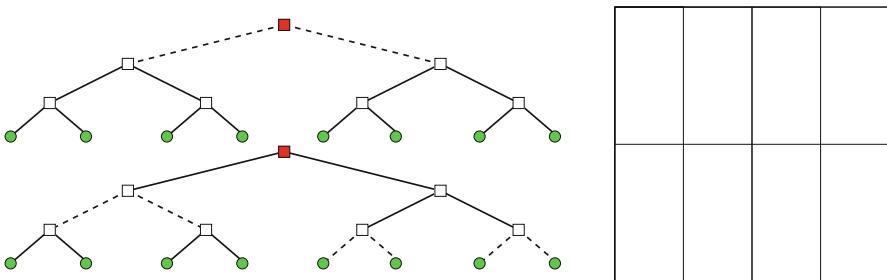


Fig. 18.4 Example of two equivalent trees for $N = 2$. The solid (resp. dash) segments represent a partition along the first (resp. second) direction. The partition of \mathcal{E} is shown on the right

they share the same set of leaves:

$$T \equiv T' \Leftrightarrow \mathcal{L}(T) = \mathcal{L}(T'). \quad (18.28)$$

The notion of equivalent trees is needed in the coarsening and enrichment procedures of Sect. 4.

In practice, we shall consider binary trees T with a fixed maximum number of successive partitions allowed in each direction $d \in \{1 \dots N\}$. As for the one-dimensional case, this quantity is called the resolution level and is denoted by N_r . Thus, there are 2^{N_r} leaves in a complete binary tree in N dimensions, showing that adaptive strategies are mandatory to apply these multiresolution schemes in high dimensions.

The construction of the SE basis in the multidimensional case can proceed as in the one-dimensional case. Let us denote as Π a prescribed polynomial space over \mathcal{E} , P the dimension of the polynomial space, and $\{\Phi_1, \dots, \Phi_P\}$ an orthonormal basis of Π . For instance, one may consider to construct Π from a tensorization of the normalized Legendre basis of $\pi_{N_0}(\mathcal{U})$, resulting in $P = (N_0 + 1)^N$ for a full tensorization of the Legendre polynomials or $P + 1 = (N_0 + N!)/(N_0!N!)$ for a total degree truncation strategy. In any case, the SE basis functions Φ_α^n associated to a leaf n are defined from the N -variate polynomials Φ_α using the mapping M_n

between $S(n)$ and \mathcal{E} and a scaling factor $|S(n)|^{-1/2}$. We are then in position to define the multiresolution space $\mathbf{V}(T)$, as the space of functions whose restrictions to each leaves of T are polynomials belonging to Π . Clearly, equivalent trees are associated to the same multiresolution space.

For $U \in L_2(\mathcal{E})$, we shall denote U^T its approximation in $\mathbf{V}(T)$. The SE expansion U^T has a general structure similar to Eq. (18.26), specifically

$$\mathbf{V}(T) \ni U^T(\mathbf{x}) = \sum_{n \in \mathcal{L}(T)} \sum_{\alpha=1}^P u_\alpha^n \Phi_\alpha^n(\mathbf{x}), \quad u_\alpha^n = \langle U, \Phi_\alpha^n \rangle_{\mathcal{E}}. \quad (18.29)$$

Extending to the multidimensional case, the MW expansion in (18.27) is slightly more complicated than for the SE expansion. The main reason is that, because of the binary structure of the dyadic partition, we have to consider MW detail functions that depend on the direction $d(n)$ along which the support of the node $S(n)$ is split. In fact, when a split direction $d(n_0)$ is provided, the procedure provided in Sect. 2.3.2 to generate the one-dimensional detail mother functions Ψ_α can be extended to the multidimensional case. It amounts to construct, for every direction, P , mother functions Ψ_α^d having SE expansions of the form

$$\Psi_\alpha^d = \sum_{\beta=1}^P \left[a_{\alpha,\beta}^d \Phi_\beta^{c^-(n_0)} + b_{\alpha,\beta}^d \Phi_\beta^{c^+(n_0)} \right]. \quad (18.30)$$

It is required that the directional MW mother functions are all orthogonal to any function of Π (*i.e.*, they have vanishing moments) or equivalently

$$\langle \Psi_\alpha^d, \Phi_\beta^{n_0} \rangle_{\mathcal{E}} = 0, \quad 1 \leq \alpha, \beta \leq P \text{ and } d = 1, \dots, N. \quad (18.31)$$

For a direction d , the corresponding P mother functions should also form an orthonormal set:

$$\langle \Psi_\alpha^d, \Psi_\beta^d \rangle_{\mathcal{E}} = \delta_{\alpha,\beta}, \quad 1 \leq \alpha, \beta \leq P, \text{ for } d = 1, \dots, N. \quad (18.32)$$

Once the N sets of mother functions are determined, the affine mapping M_n can be used to derive the MW functions of any node $n \in \widehat{\mathcal{N}}(T)$, through

$$\Psi_\alpha^n(\mathbf{x}) = \begin{cases} |S(n)|^{-1/2} \Psi_\alpha^{d(n)}(M_n \mathbf{x}), & \mathbf{x} \in S(n) \\ 0, & \text{otherwise.} \end{cases} \quad (18.33)$$

It can be easily verified that

$$\text{span}_\alpha \{ \Phi_\alpha^{c^-(n)}, \Phi_\alpha^{c^+(n)} \} = \text{span}_\alpha \{ \Phi_\alpha^n \} \oplus \text{span}_\alpha \{ \Psi_\alpha^n \}. \quad (18.34)$$

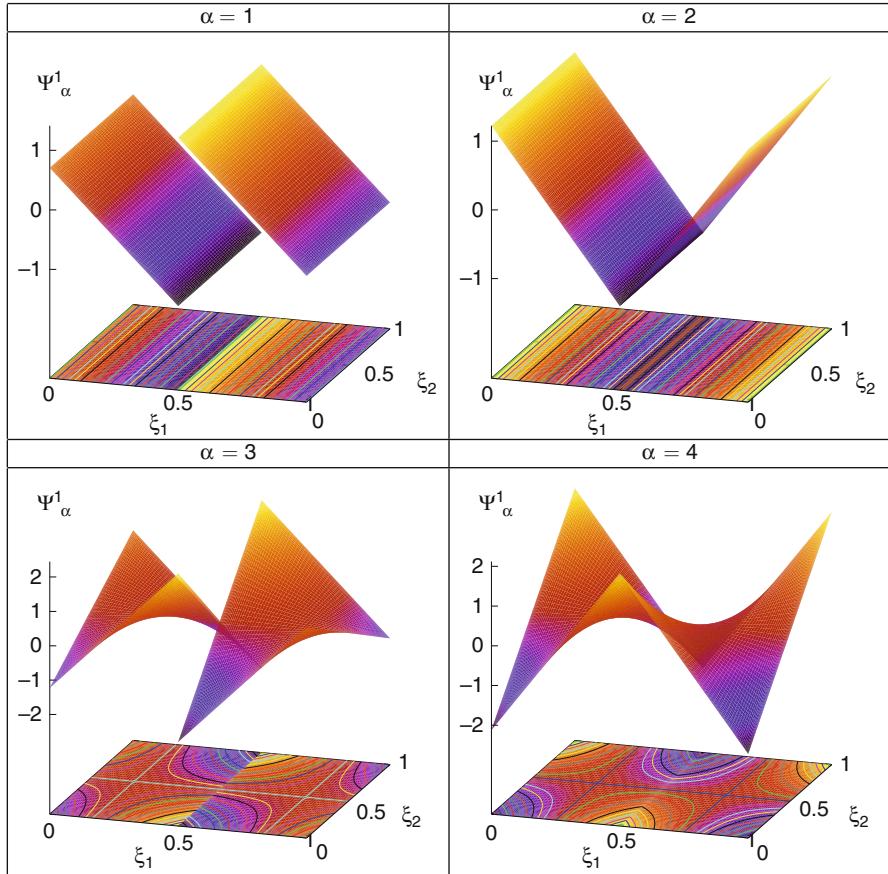


Fig. 18.5 Mother multiwavelets Ψ_α^d for $N = 2$, $No = 1$ in direction $d = 1$

As an illustration, the mother functions $\Psi_\alpha^{d=1}$ are reported in Fig. 18.5 for the two-dimensional case ($N = 2$) and a polynomial space Π consisting of the full tensorization of the degree one polynomials, so $P = (No + 1)^2 = 4$. The plots of the MW mother functions show the different types of singularity across the line $x_{d=1} = 1/2$ corresponding to the split into the two children.

Then, the approximation of $f \in L_2(\mathcal{E})$ in $\mathbf{V}(T)$ has the hierarchical expansion in the MW basis given by

$$\mathbf{V}(T) \ni U^T(\mathbf{x}) = \sum_{\alpha=1}^P u_\alpha^{n_0} \Phi_\alpha^{n_0}(\mathbf{x}) + \sum_{n \in \widehat{\mathcal{N}}(T)} \sum_{\alpha=1}^P \tilde{u}_\alpha^n \Psi_\alpha^n(\mathbf{x}), \quad \tilde{u}_\alpha^n = \langle U, \Psi_\alpha^n \rangle_{\mathcal{E}}. \quad (18.35)$$

3.3 Multiscale Operators

In this section, we introduce two essential multiscale operators, the restriction and prediction operators, that are useful tools in the adaptive context. We start by introducing the notion of inclusion over trees. Let T_1 and T_2 be two binary trees. We say that $T_1 \subset T_2$ if

$$\forall l_2 \in \mathcal{L}(T_2), \exists ! l_1 \in \mathcal{L}(T_1) \text{ s.t. } S(l_1) \subset S(l_2). \quad (18.36)$$

Clearly, if $T_1 \subset T_2$, then $\mathbf{V}(T_1) \subset \mathbf{V}(T_2)$.

3.3.1 Restriction Operator

Let T_1 and T_2 be two binary trees such that $T_1 \subset T_2$. Given $U^{T_2} \in \mathbf{V}(T_2)$, we define the restriction of U^{T_2} to $\mathbf{V}(T_1)$, denoted $\mathcal{R}_{\downarrow T_1} U^{T_2}$, as the orthogonal $L^2(\mathcal{E})$ -projection of U^{T_2} onto $\mathbf{V}(T_1)$, i.e., $(U^{T_2} - \mathcal{R}_{\downarrow T_1} U^{T_2}) \perp \mathbf{V}(T_1)$. In terms of MW coefficients, the restriction operation is straightforward. Letting \tilde{u}_α^n be the MW coefficients of U^{T_2} and using the orthonormality of the MW basis yields, for all $n \in \widehat{\mathcal{N}}(T_1)$ and all $\alpha \in \mathcal{P}$,

$$\widetilde{(\mathcal{R}_{\downarrow T_1} U^{T_2})}_\alpha^n = \tilde{u}_\alpha^n. \quad (18.37)$$

(The large tilde over the left-hand-side stresses that we express a *MW coefficient* of the restriction of U^{T_2} to $\mathbf{V}(T_1)$). It shows that the restriction of the approximation space, from $\mathbf{V}(T_2)$ to $\mathbf{V}(T_1)$, preserves the MW coefficients, but reduces the set of nodes supporting these coefficients: $\widehat{\mathcal{N}}(T_1) \subset \widehat{\mathcal{N}}(T_2)$. The computation of the SE coefficients of the restriction is not as immediate. Assuming that the SE expansion of U^{T_2} is known, we construct a sequence of trees $T^{(i)}$ such that $T_2 = T_{(0)} \supset \dots \supset T_{(i)} \supset \dots \supset T_{(l)} = T_1$, where two consecutive trees differ by one generation only, i.e., a leaf of $T_{(i+1)}$ is either a leaf or a node with leaf children in $T_{(i)}$. Therefore, the transition from $T^{(i)}$ to $T^{(i+1)}$ consists in removing pairs of sister leaves. The process is illustrated in the left part of Fig. 18.6 for the removal of a single pair of sister leaves. Focusing on the removal of a (left-right ordered) pair of sister leaves

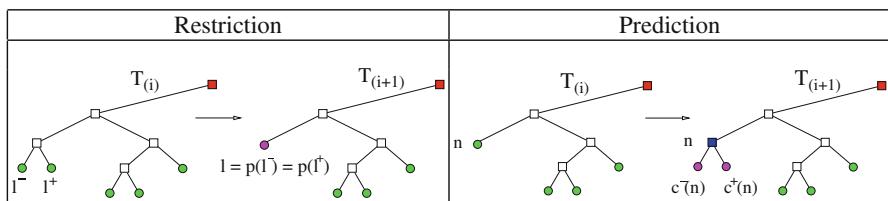


Fig. 18.6 Schematic representation of the elementary restriction (*left*) and prediction (*right*) operators through the removal and creation respectively of the (leaves) children of a node

$\{l^-, l^+\}$, the SE coefficients of the restriction of $U^{T(i)}$ associated with the new leaf $l = p(l^-) = p(l^+) \in \mathcal{L}(T_{(i+1)})$ in direction $d(l)$ are

$$u_\alpha^l = \sum_{\beta \in \mathcal{P}} \left[R_{\alpha,\beta}^{-,d(l)} u_\beta^{l^-} + R_{\alpha,\beta}^{+,d(l)} u_\beta^{l^+} \right], \quad (18.38)$$

where, for all $d \in \{1 \dots N\}$, the transition matrices $R^{\pm,d}$ of order P have entries given by $R_{\alpha,\beta}^{\pm,d} = \left\langle \Phi_\alpha^{n_0}, \Phi_\beta^{c_d^{\pm(n_0)}} \right\rangle$.

3.3.2 Prediction Operator

Let T_1 and T_2 be two binary trees such that $T_1 \subset T_2$. The prediction operation consists of extending $U^{T_1} \in \mathbf{V}(T_1)$ to the larger stochastic space $\mathbf{V}(T_2)$. We denote by $\mathcal{P}_{\uparrow T_2} U^{T_1}$ this prediction. Different predictions can be used (see [18, 19]); here we have considered the simplest one, where no information is generated by the prediction. As for the restriction operation, the MW expansion of the prediction is immediately obtained from the MW coefficients of U^{T_1} . We obtain, for all $n \in \widehat{\mathcal{N}}(T_2)$ and all $\alpha \in \mathcal{P}$,

$$\left(\widetilde{\mathcal{P}_{\uparrow T_2} U^{T_1}} \right)_\alpha^n = \begin{cases} \tilde{u}_\alpha^n, & n \in \widehat{\mathcal{N}}(T_2), \\ 0, & \text{otherwise.} \end{cases} \quad (18.39)$$

For the SE coefficients of $\mathcal{P}_{\uparrow T_2} U^{T_1}$, we can again proceed iteratively, starting from the SE expansion over T_1 , using a series of increasing intermediate trees, differing by only one generation from one to the other. This time, the elementary operation consists in adding to some node n , being a leaf of the current tree, children in a prescribed direction $d(n)$. The process is illustrated in the right part of Fig. 18.6. The SE coefficients associated to the new leaves of a node n are given by

$$u_\alpha^{c^-(n)} = \sum_{\beta \in \mathcal{P}} R_{\alpha,\beta}^{-,d(n)} u_\beta^n, \quad u_\alpha^{c^+(n)} = \sum_{\beta \in \mathcal{P}} R_{\alpha,\beta}^{+,d(n)} u_\beta^n, \quad (18.40)$$

with the same transition coefficients as those used in (18.38). For two trees $T_1 \subset T_2$, we observe that $\mathcal{R}_{\downarrow T_1} \circ \mathcal{P}_{\uparrow T_2} = \mathcal{I}_{T_1}$, while in general $\mathcal{P}_{\uparrow T_2} \circ \mathcal{R}_{\downarrow T_1} \neq \mathcal{I}_{T_2}$ (\mathcal{I} denoting the identity).

4 Adaptivity

In this section, we detail the essential adaptivity tools needed for the control of the local stochastic resolution, with the objective of efficiently reducing the complexity of the computations. There are two essential ingredients: the coarsening and enrichment procedures. Below we detail the criteria needed to perform these

two operations. These thresholding and enrichment criteria were initially proposed in [11].

Recall that for $n \in \mathcal{N}(T)$, $|n|$ is the distance of n from the root node n_0 so the measure of its support is $|S(n)| := 2^{-|n|}$. We shall also need the measure of $S(n)$ in specific direction $d \in [1, N]$, that is $|S(n)|_d := x_{n,d}^+ - x_{n,d}^-$, its diameter $\text{diam}(S(n)) := \max_d |S(n)|_d$, and its volume in all directions except d as $|S(n)|_{\sim d} := |S(n)| / |S(n)|_d$.

4.1 Coarsening

Let T be a binary tree and let $U^T \in \mathbf{V}(T)$. The coarsening procedure aims at constructing a subtree $T^- \subset T$ (or, equivalently, a stochastic approximation subspace $\mathbf{V}(T^-) \subset \mathbf{V}(T)$) through a thresholding of the MW expansion coefficients of U^T .

4.1.1 Thresholding Error

Let $\eta > 0$ be a tolerance and recall that Nr denotes the resolution level. Let \tilde{u}_α^n denote the MW expansion coefficients of U^T ; see (18.35). We define $\mathcal{D}(\eta, Nr)$ as the subset of $\widehat{\mathcal{N}}(T)$ such that

$$\mathcal{D}(\eta, Nr) := \left\{ n \in \widehat{\mathcal{N}}(T); \|\tilde{\mathbf{u}}^n\|_{\ell^2} \leq 2^{-|n|/2} (NNr)^{-1/2} \eta \right\}, \quad (18.41)$$

where $\tilde{\mathbf{u}}^n := (\tilde{u}_\alpha^n)_{\alpha \in P}$ and $\|\tilde{\mathbf{u}}^n\|_{\ell^2}^2 = \sum_{\alpha \in P} (\tilde{u}_\alpha^n)^2$. The motivation for (18.41) is that, letting \widehat{U}^T be the thresholded version of U^T obtained by omitting in the second sum of (18.35) the nodes $n \in \mathcal{D}(\eta, Nr)$, there holds

$$\|\widehat{U}^T - U^T\|_{L^2(\mathcal{E})}^2 = \sum_{n \in \mathcal{D}(\eta, Nr)} \|\tilde{\mathbf{u}}^n\|_{\ell^2}^2 \leq \sum_{n \in \mathcal{D}(\eta, Nr)} 2^{-|n|} (NNr)^{-1} \eta^2 \leq \eta^2 \quad (18.42)$$

since

$$\sum_{n \in \mathcal{D}(\eta, Nr)} 2^{-|n|} = \sum_{j=0}^{NNr-1} \#\{n \in \mathcal{D}(\eta, Nr); |n| = j\} 2^{-j} \leq \sum_{j=0}^{NNr-1} 1 = NNr.$$

Therefore, using the definition in (18.41) for the thresholding the MW expansion of a function in $\mathbf{V}(T)$, we can guaranty an L_2 error less than η .

4.1.2 Coarsening Procedure

Two points deserve particular attention. The first one is that $\mathcal{N}(T) \setminus \mathcal{D}(\eta, Nr)$ does not have a binary tree structure in general, so that a procedure is needed to maintain this structure when removing nodes of T . Here, we choose a conservative approach where the resulting subtree T^- may still contain some nodes in the set $\mathcal{D}(\eta, Nr)$.

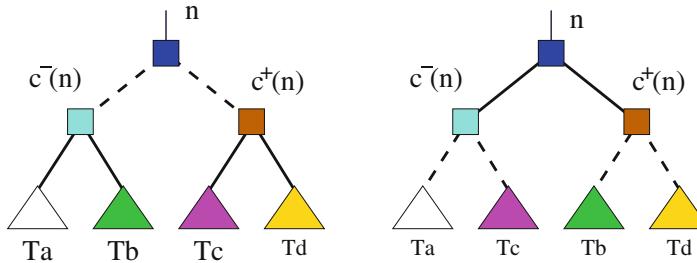


Fig. 18.7 Illustration of the elementary operation to generate equivalent trees: the pattern of a node with its children divided along the same direction (*left*) is replaced by the same pattern but with an exchange of the partition directions (*right*) plus the corresponding permutation of the descendants of the children

Specifically, we construct a sequence of nested trees, obtained through the removal of pairs of sister leaves from one tree to the next: a couple of sister leaves having node n for parent is removed if $n \in \mathcal{D}(\eta, N_r)$. The coarsening sequence is stopped whenever no couple of sister leaves can be removed, and this yields the desired subtree T^- while ensuring the binary structure. In addition, the procedure preserves the error bounds.

The second point is that the above algorithm only generates trees such that, along the sequence, the successive (coarser and coarser) partitions of \mathcal{E} follow, in backward order, the partition directions $d(n)$ prescribed by T . This is unsatisfying because for $N > 1$, there are many trees equivalent to T , and we would like the coarsened tree to be independent of any particular choice in this equivalence class. To avoid arbitrariness, the trees of the sequence are periodically substituted by equivalent ones, generated by searching in the current tree for the pattern of a node n whose children $c^-(n)$ and $c^+(n)$ are not leaves and are subsequently partitioned along the same direction $d(c^+(n)) = d(c^-(n))$ which differs from $d(n)$; when such a pattern is found, the two successive partition directions are exchanged, $d(n) \leftrightarrow d(c^-(n)) = d(c^+(n))$, together with the corresponding permutation of the descendants of the children nodes. This operation, illustrated in Fig. 18.7, is applied periodically and randomly during the coarsening procedure.

4.2 Anisotropic Enrichment

Let T be a binary tree and let $U^T \in \mathbf{V}(T)$. The purpose of the enrichment is to increase the dimension of $\mathbf{V}(T)$, by adding descendants to some of its leaves. Enrichment of the stochastic space is required to adaptively construct approximations of particular quantities. Another typical situation where enrichment is necessary is in dynamical problems, with the possible emergence in time of new features in the stochastic solution, such as shocks, that require more resolution. Classically, the enrichment of a tree T is restricted to at most one partition along each dimension in

dynamical problems. For steady problems, the enrichment procedure can be simply applied multiple times to end up with sufficiently refined approximations.

The simplest enrichment procedure consists in systematically partitioning *all* the leaves $\mathfrak{l} \in \mathcal{L}(T)$ once for all $d \in \{1 \dots N\}$ provided $|S(\mathfrak{l})|_d > 2^{-Nr}$. This procedure generates a tree T^+ that typically has $2^N \text{card}(\mathcal{L}(T))$ leaves, which is only practical when N is small.

More economical strategies are based on the analysis of the MW coefficients in U^T to decide which leaves of T need be partitioned and along which direction (see, for instance, [4, 5]). We derive below two directional enrichment criteria in the context of N -dimensional binary trees.

4.2.1 Multidimensional Enrichment Criterion

Classically, the theoretical decay rate of the MW coefficients with resolution level is used to decide the partition of a leaf from the norm of MW coefficients of its parent (see, for instance, [18, 20] in the deterministic case).

We first recall some background in the 1D case ($N = 1$). Let $U \in L^2(\mathcal{U})$. Let T_{1D} be a 1D binary tree and let $U^{T_{1D}}$ be the $L^2(\mathcal{U})$ -orthogonal projection of U onto $\mathbf{V}(T_{1D})$. Let \tilde{u}_α^n denote the MW coefficients of $U^{T_{1D}}$. Then, if U is locally smooth enough, the magnitude of the MW coefficients \tilde{u}_α^n of a generic node $n \in \widehat{\mathcal{N}}(T_{1D})$ can be bounded as

$$|\tilde{u}_\alpha^n| = \inf_{P \in \pi_{No}(\mathcal{U})} |((U - P), \Psi_\alpha^n)_U| \leq C |S(n)|^{No+1} \|U\|_{H^{No+1}(S(n))}, \quad (18.43)$$

where $H^{No+1}(S(n))$ is the usual Sobolev space of order $(No + 1)$ on $S(n)$. Recalling that $|S(n)| = 2^{-|n|}$, the bound (18.43) shows that the norm of the MW coefficients decays roughly as $\mathcal{O}(2^{-|n|(No+1)})$ for smooth U . Therefore, the norm of the (unknown) MW coefficients of a leaf $\mathfrak{l} \in \mathcal{L}(T_{1D})$ can be estimated from the norm of the (known) MW coefficients of its parent as

$$\|\tilde{\mathbf{u}}^\mathfrak{l}\|_{\ell^2} \sim 2^{-(No+1)} \|\tilde{\mathbf{u}}^{\mathfrak{p}(\mathfrak{l})}\|_{\ell^2}.$$

This estimate can, in turn, be used to derive an enrichment criterion; specifically, a leaf \mathfrak{l} is partitioned if the estimate of $\|\tilde{\mathbf{u}}^\mathfrak{l}\|_{\ell^2}$ exceeds the thresholding criterion (18.41), that is, if

$$\|\tilde{\mathbf{u}}^{\mathfrak{p}(\mathfrak{l})}\|_{\ell^2} \geq 2^{No+1} 2^{-|\mathfrak{l}|/2} Nr^{-1/2} \eta \quad \text{and} \quad |S(\mathfrak{l})| > 2^{-Nr}. \quad (18.44)$$

The extension to $N > 1$ of the enrichment criterion (18.44) is not straightforward in the context of binary trees. Indeed, the MW coefficients associated with a node n carry information essentially related to the splitting direction $d(n)$. Thus, for a leaf $\mathfrak{l} \in \mathcal{L}(T)$, they cannot be used for an enrichment criterion in a direction $d \neq d(\mathfrak{p}(\mathfrak{l}))$. To address this issue, we define, for any leaf $\mathfrak{l} \in \mathcal{L}(T)$ and any direction $d \in \{1 \dots N\}$, its virtual parent $\mathfrak{p}^d(\mathfrak{l})$ as the (virtual) node that would have \mathfrak{l} as a child after a dyadic partition along the d th direction. Consistently, $\mathfrak{s}^d(\mathfrak{l})$ denotes the

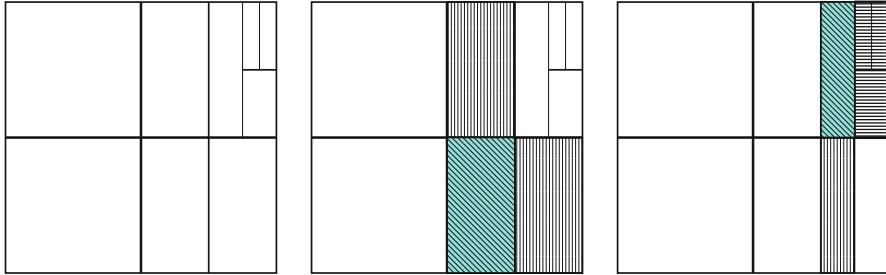


Fig. 18.8 Illustration of the virtual sisters of a leaf l of a tree T whose partition is shown in the *left plot*. In the *center plot*, the leaf l is hatched diagonally in *blue* and its two virtual sisters for $d = 1$ and 2 (hatched horizontally and vertically respectively) are leaves of T , both being $c^+(\mathbf{p}^d(l))$. In the *right plot*, a different leaf l is considered (still hatched diagonally in *blue*) with virtual sisters $\mathbf{s}^d(l)$ which for $d = 1$ (hatched horizontally) is a node of T but not a leaf and which for $d = 2$ (hatched vertically) is not a node of T

virtual sister of l along direction d . Note that $\mathbf{p}^d(l) \in \mathcal{N}(T)$ only for $d = d(\mathbf{p}(l))$; moreover, in general, $\mathbf{s}^d(l) \notin \mathcal{N}(T)$. These definitions are illustrated in Fig. 18.8 which shows for $N = 2$ the partition associated with a tree T (left plot) and the virtual sisters of two leaves.

The SE coefficients of the virtual sisters,

$$u_\alpha^{s^d(l)} := \left\langle U^\top, \Phi_\alpha^{s^d(l)} \right\rangle_{\mathcal{E}}, \quad \alpha = 1, \dots, P, \quad (18.45)$$

are efficiently computed by exploiting the binary structure of T and relying on the elementary restriction and prediction operators defined in Sect. 3.3. Without going into too many details, let us mention that the computation of the SE coefficients of $\mathbf{s}^d(l)$ amounts to (i) finding the subset of leaves in $\mathcal{L}(T)$ whose supports overlap with $S(\mathbf{s}^d(l))$, (ii) constructing the subtree having for leaves this subset, and (iii) restricting the solution over this subtree up to \mathbf{s}^d . In practice, one can reuse the restriction operator defined in Sect. 3.3 to compute the usual details in the $\{\Psi_\alpha^{n,d}\}_{\alpha=1,\dots,P}$ basis for a chosen direction d .

We now return to the design of a multidimensional enrichment criterion. Assuming an underlying isotropic polynomial space Π , with degree No in all directions, a natural extension of (18.44) is that a leaf l is partitioned in the direction d if

$$\|\tilde{\mathbf{u}}^{p^d(l)}\|_{\ell^2} \geq \left(\frac{\text{diam}(S(\mathbf{p}^d(l)))}{\text{diam}(S(l))} \right)^{No+1} 2^{-|l|/2} (NNr)^{-1/2} \eta \quad \text{and} \quad |S(l)|_d > 2^{-Nr}. \quad (18.46)$$

We recall that $\text{diam}(S(n))$ is the diameter of the support of the considered node. The criterion in (18.46) is motivated by the following multidimensional extension of the bound (18.43) for the magnitude of the MW coefficients \tilde{u}_α^n in the direction d for a

generic node \mathbf{n} ,

$$|\tilde{u}_\alpha^{\mathbf{n}}| = \inf_{P \in \Pi} |\langle (U - P), \Psi_\alpha^{\mathbf{n},d} \rangle_{\Xi}| \leq C \operatorname{diam}(S(\mathbf{n}))^{N_0+1} \|U\|_{H^{N_0+1}(S(\mathbf{n}))}. \quad (18.47)$$

4.2.2 Directional Enrichment Criterion

We want to improve the criterion (18.46) since the isotropic factor

$$\rho_l \doteq \operatorname{diam}(S(\mathbf{p}^l(l)))/\operatorname{diam}(S(l))$$

can take the value 1 in the context of anisotropic refinement. An alternative is to devise a criterion with the factor 2^{N_0+1} , since this will lead to smaller enriched trees. For this purpose, we derive an alternative criterion that is fully directional. For any direction $d \in \{1 \dots N\}$ and any node $\mathbf{n} \in T$, we define the directional detail coefficients $\bar{u}_{\beta \in \{1 \dots N_0+1\}}^{\mathbf{n},d}$ through

$$\bar{u}_{\beta}^{\mathbf{n},d} := \left\langle U, \bar{\Psi}_{\beta}^{\mathbf{n},d} \right\rangle_{\Xi}, \quad \bar{\Psi}_{\beta}^{\mathbf{n},d}(\mathbf{x}) = \begin{cases} |S(\mathbf{n})|^{-1/2} \Psi_{\beta}^* \left(\frac{x_d - x_{\mathbf{n},d}^-}{x_{\mathbf{n},d}^+ - x_{\mathbf{n},d}^-} \right), & \xi \in S(\mathbf{n}), \\ 0, & \text{otherwise,} \end{cases} \quad (18.48)$$

where $\{\Psi_{\beta}^*\}_{\beta \in \{1 \dots N_0+1\}}$ is the set of 1D wavelet functions defined on $[0, 1]$. The vector of coefficients $\bar{\mathbf{u}}^{\mathbf{n},d}$ measures details in U at the scale $|S(\mathbf{n})|_d$, in direction d only, by averaging out any variability in U along the other directions.

For a direction $d \in \{1 \dots N\}$, let $\sim d$ denote all the directions except d . Because by construction $\|\bar{\Psi}_{\beta}^{\mathbf{n},d}\|_{\Xi} = 1$, it follows that

$$\begin{aligned} |\bar{u}_{\beta}^{\mathbf{n},d}| &= \inf_{P \in \pi_{N_0}} \left| \left\langle U - P, \bar{\Psi}_{\beta}^{\mathbf{n},d} \right\rangle_{\Xi} \right| \\ &= \inf_{P \in \pi_{N_0}} \left| \int_{S(\mathbf{n})} (U(\mathbf{x}_{\sim d}, x_d) - P(x_d)) \bar{\Psi}_{\beta}^{\mathbf{n},d}(x_d) d\mathbf{x} \right| \\ &= \inf_{P \in \pi_{N_0}} |S(\mathbf{n})|_{\sim d} \left| \int_{S_d(\mathbf{n})} (\bar{U}_{\sim d}^{\mathbf{n}}(x_d) - P(x_d)) \bar{\Psi}_{\beta}^{\mathbf{n},d}(x_d) d\mathbf{x} \right| \\ &\leq C |S(\mathbf{n})|_{\sim d} |S(\mathbf{n})|_d^{N_0+1} \|\bar{U}_{\sim d}^{\mathbf{n}}\|_{H^{N_0+1}(S_d(\mathbf{n}))} \|\bar{\Psi}_{\beta}^{\mathbf{n},d}\|_{L^2(S_d(\mathbf{n}))} \\ &= C |S(\mathbf{n})|_{\sim d}^{1/2} |S(\mathbf{n})|_d^{N_0+1} \|\bar{U}_{\sim d}^{\mathbf{n}}\|_{H^{N_0+1}(S_d(\mathbf{n}))}, \end{aligned} \quad (18.49)$$

where $\bar{U}_{\sim d}^{\mathbf{n}}(x_d) = |S(\mathbf{n})|_{\sim d}^{-1} \int_{S_d(\mathbf{n})} U(\mathbf{x}_{\sim d}, x_d) d\mathbf{x}_{\sim d}$ is the marginalization of $U(\mathbf{x})$ over the support $S(\mathbf{n})$ in all the directions $\sim d$. Furthermore, (omitting the reference to the node \mathbf{n}),

$$\begin{aligned}\|\bar{U}_{\sim d}\|_{H^{\text{No}+1}(S_d)}^2 &= \int_{S_d} \left| \frac{\partial^{\text{No}+1}}{\partial x_d} \frac{1}{|S|_{\sim d}} \left(\int_{S_{\sim d}} U(\mathbf{x}_{\sim d}, x_d) d\mathbf{x}_{\sim d} \right) \right|^2 dx_d \\ &= \frac{1}{|S|_{\sim d}^2} \int_{S_d} \left| \int_{S_{\sim d}} \frac{\partial^{\text{No}+1}}{\partial x_d} U(\mathbf{x}_{\sim d}, x_d) d\mathbf{x}_{\sim d} \right|^2 dx_d \leq |S|_{\sim d}^{-1} \\ &\quad \int_S \left| \frac{\partial^{\text{No}+1}}{\partial x_d} U \right|^2 dx,\end{aligned}$$

whence we deduce

$$\|\bar{U}_{\sim d}\|_{H^{\text{No}+1}(S_d)} \leq |S|_{\sim d}^{-1/2} \|U\|_{L^2(S_{\sim d}, H^{\text{No}+1}(S_d))}, \quad (18.50)$$

with anisotropic Sobolev norm $\|U\|_{L^2(S_{\sim d}, H^{\text{No}+1}(S_d))}^2 = \int_{S_{\sim d}} \|U(\mathbf{x}_{\sim d}, \cdot)\|_{H^{\text{No}+1}(S_d)}^2 d\mathbf{x}_{\sim d}$.

Combining (18.49) with (18.50), the estimate for the directional details magnitude is now

$$|\bar{u}_\beta^{\text{n},d}| = \inf_{P(\xi_d) \in \pi_{\text{No}}(\mathcal{U})} \left| \langle (U - P), \bar{\Psi}_\beta^{\text{n},d} \rangle_{\Xi} \right| \leq C |S(\mathbf{n})|_d^{\text{No}+1} \|U\|_{L_2(S_{\sim d}(\mathbf{n}), H^{\text{No}+1}(S_d(\mathbf{n})))}. \quad (18.51)$$

Proceeding as previously, the enrichment criterion states that a leaf \mathfrak{l} is partitioned along direction d if

$$\|\bar{\mathbf{u}}^{\mathfrak{p}^d(\mathfrak{l})}\|_{\ell^2} \geq 2^{\text{No}+1} 2^{-|\mathfrak{l}|/2} (\text{NNr})^{-1/2} \eta \quad \text{and} \quad |S(\mathfrak{l})|_d > 2^{-\text{Nr}}. \quad (18.52)$$

The details norm associated with the basis $\{\bar{\Psi}_\beta^{\text{n},d}\}_{\beta \in \mathcal{P}}$ can be obtained explicitly from the vector of MW coefficients $\tilde{\mathbf{u}}^{\text{n},d}$ by averaging it in all but the d th direction.

5 Illustrations

We illustrate the effectiveness of the multiresolution approach for parametric uncertainty problems with increasing complexity.

5.1 Simple ODE Problem

We start by considering a simple ordinary differential equation (ODE) involving a single random parameter; the ODE solution $U(t, \xi)$ satisfies the governing equation

$$\frac{d^2 U}{dt^2} + f \frac{dU}{dt} = F(U), \quad F(U) = -\frac{35}{2} U^3 + \frac{15}{2} U, \quad (18.53)$$

which describes the motion of a particle in a deterministic potential field F , with damping governed by the friction factor $f > 0$. The problem is completed with an initial condition at $t = 0$, assumed uncertain and given by a function $U^0(\xi)$. Because of the dissipative dynamics, the particle asymptotically reaches a fixed location

$$U^\infty(\xi) = \lim_{t \rightarrow \infty} U(t, \xi), \quad (18.54)$$

which depends on the initial position and so is uncertain. However, the mapping $\xi \mapsto U^\infty$ is not necessarily smooth; indeed $F(u) = 0$ has two (stable) roots $r_1 = -r_2 = \pm\sqrt{15/35}$ so the system has two stable steady points. For simplicity, we shall consider the uncertain initial condition $U^0(\xi) = 0.05 + 0.2\xi$, where ξ has a uniform distribution in $[-1, 1]$; the multiresolution scheme detailed above can then be applied with $x = (\xi + 1)/2$.

The problem is solved relying on a stochastic Galerkin projection method [15, 21], seeking an expansion of $U(t, \xi)$ according to

$$U(t, \xi) = \sum_{\alpha} u_{\alpha}(t) \theta_{\alpha}(\xi), \quad (18.55)$$

for selected functionals θ_{α} forming a Hilbertian basis. Introducing this expansion of U into the governing equation, and requiring the orthogonality of the equation

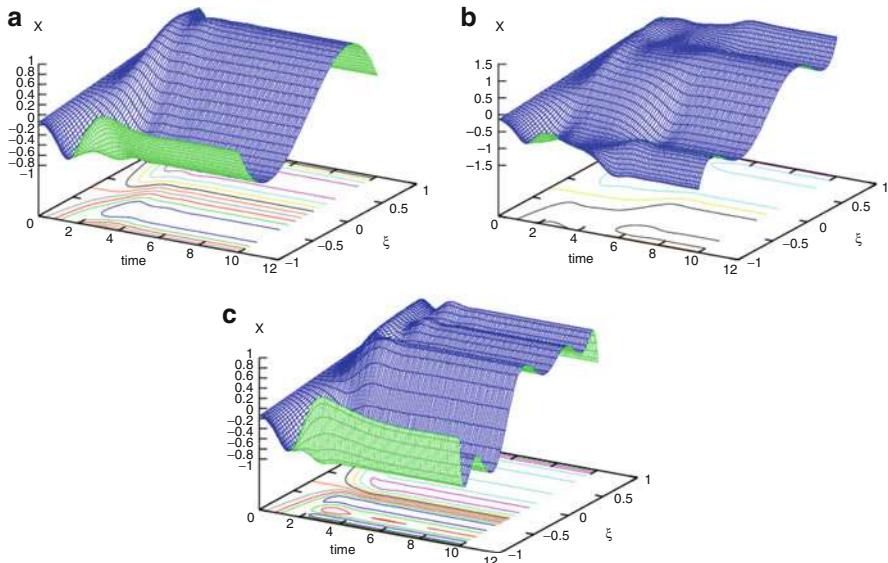


Fig. 18.9 Time evolution of the Galerkin solution of (18.53) discretized in spectral space $\mathbf{V}_{No}^{(0)}$ (scaled Legendre polynomials) with $No = 3, 5$, and 9 (from left to right)

residual with respect to the stochastic approximation space, one obtains the governing equations for the expansion coefficients; specifically this results in

$$\frac{du_\alpha}{dt} = \left\langle F \left(\sum_\beta u_\beta \theta_\beta \right), \theta_\alpha \right\rangle. \quad (18.56)$$

Upon truncation of the expansion, one has to solve a set of coupled nonlinear ODEs for the expansion coefficients. The initial conditions for these expansion coefficients are also obtained by projecting the uncertain initial data, yielding $u_\alpha(t = 0) = \langle U^0, \theta_\alpha \rangle$.

Figure 18.9 shows the time evolutions of the solution from the uniformly distributed initial condition, for different one-dimensional multiresolution spaces $\mathbf{V}_{N_0}^{(0)}$ as defined in Sect. 2.1. Computations for different polynomial orders $No = 3, 5$, and 9 are reported. It is seen that the smooth approximations at level 0, corresponding to spectral expansions with the functionals θ_α being the scaled Legendre polynomials, are not able to properly represent the evolutions and asymptotic behavior of U . Specifically, when increasing the polynomial order No , the computed asymptotic steady state is plagued with Gibb's oscillations.

In contrast, piecewise constant approximations in $\mathbf{V}_0^{(Nr)}$ quickly converge with the resolution level Nr , as can be appreciated from the plots of Fig. 18.10. The plots depict the solutions for increasing resolution levels $Nr = 2, 3$, and 5 . In particular, it is seen that the exact asymptotic steady solution is recovered for most values of ξ , except over the subinterval containing the discontinuity. When the resolution level increases, a finer definition of the discontinuity location is achieved, resulting in a highly accurate approximation.

Note that in the case of an approximation in $\mathbf{V}_{N_0}^{(Nr)}$, one can decide to formulate Galerkin problem in the SE or MW basis. From the computational point of view, using one or the other basis can dramatically affect the complexity of the computations. Specifically, for the SE element basis, the nonlinear systems of $2^{Nr} \times (No + 1)$ equations form in fact a set of 2^{Nr} uncoupled nonlinear systems with size $(No + 1)$, because of the limited number of overlapping supports. In contrast, formulating the Galerkin problem in the MW basis maintains the nonlinear coupling between all the MW coefficients of the solution, requiring the resolution of a significantly larger problem. As a general rule, determining approximations in multiresolution spaces is generally more conveniently performed for the SE expansions (i.e., computing local expansions over the set of leaves), while performing the multiresolution analysis to decide on enrichment/coarsening uses mostly the MW coefficients as shown in Sect. 4. The multiscale operators and the tree representation provide convenient means to efficiently translate one set of coefficients into another.

This example also illustrates the need for adaptivity tools that allow the tuning of the stochastic discretization effort, i.e., the resolution level, dynamically in time and locally in the stochastic space. For instance, it is seen that an essentially uniform resolution level is needed at early times while, asymptotically, one needs only detail

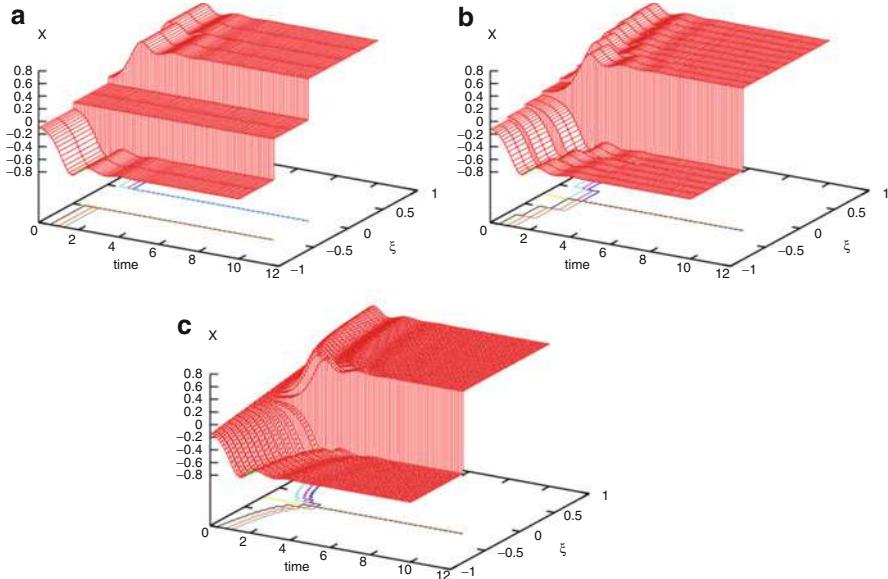


Fig. 18.10 Time evolution of the Galerkin solution of (18.53) discretized in spectral space $\mathbf{V}_0^{(\text{Nr})}$, that is, piecewise constant approximations, with $\text{Nr} = 2, 3$, and 5 (from left to right)

functions around the region of discontinuity, as the solution tends to an actual piecewise constant solution.

5.2 Scalar Conservation Law

In [11], a fully adaptive strategy of the stochastic discretization was proposed for the resolution of scalar conservation equations.

5.2.1 Test Problem

The test problem consists of the one-dimensional conservation equation:

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x} F(U; \xi) = 0, \quad x \in [0, 1], \quad (18.57)$$

with periodic boundary conditions (here x denotes the spatial variable). This equation models the evolution of a normalized density of vehicles, U , on a road (closed track, in this case); the traffic model corresponds to a flux function having the form

$$F(U; \xi) = A(\xi)U(\xi)(1 - U(\xi)), \quad (18.58)$$

where $A(\xi)$ is almost surely positive and represents an (uncertain) reference velocity. The solution is then uncertain because the initial condition $U^{IC}(x, \xi)$ is uncertain, and because uncertainty in the characteristic velocity results in an uncertain flux function $F(\cdot; \xi)$. Specifically, the initial condition consists of four piecewise constant uncertain states in x , parametrized using four independent random variables ξ_1, ξ_2, ξ_3 , and ξ_4 , with uniform distributions in $[0, 1]$:

$$U^{IC}(x, \xi) = \bar{U}(\xi_1) - U^-(\xi_2)\mathbb{I}_{[0.1, 0.3]}(x) + U^+(\xi_3)\mathbb{I}_{[0.3, 0.5]}(x) - U^-(\xi_4)\mathbb{I}_{[0.5, 0.7]}(x), \quad (18.59)$$

where

$$\begin{aligned}\bar{U}(\xi_1) &= 0.25 + 0.01\xi_1 \sim \mathcal{U}[0.25, 0.26], \\ U^-(\xi_{2,4}) &= 0.2 + 0.015\xi_{2,4} \sim \mathcal{U}[0.2, 0.215],\end{aligned}$$

and

$$U^+(\xi_3) = 0.1 + 0.015\xi_3 \sim \mathcal{U}[0.1, 0.115].$$

In (18.59), \mathbb{I}_Z denotes the characteristic function of the set Z ,

$$\mathbb{I}_Z(x) = \begin{cases} 1, & x \in Z, \\ 0, & \text{otherwise.} \end{cases}$$

Because the uncertain reference velocity is independent of the initial conditions, it is parametrized with another independent random variable ξ_5 with uniform distribution in $[0, 1]$:

$$A(\xi_5) = 1 + 0.05(2\xi_5 - 1) \sim \mathcal{U}[0.95, 1.05]. \quad (18.60)$$

The problem has therefore five stochastic dimensions ($N = 5$) and the selected polynomial space Π is spanned by the partially tensorized Legendre polynomials

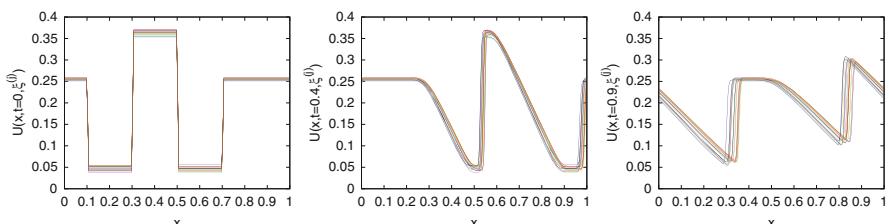


Fig. 18.11 Stochastic traffic equation: sample set of 20 realizations of the initial condition (left) and computed solution at $t = 0.4$ (middle) and $t = 0.9$ (right) (Adapted from [11])

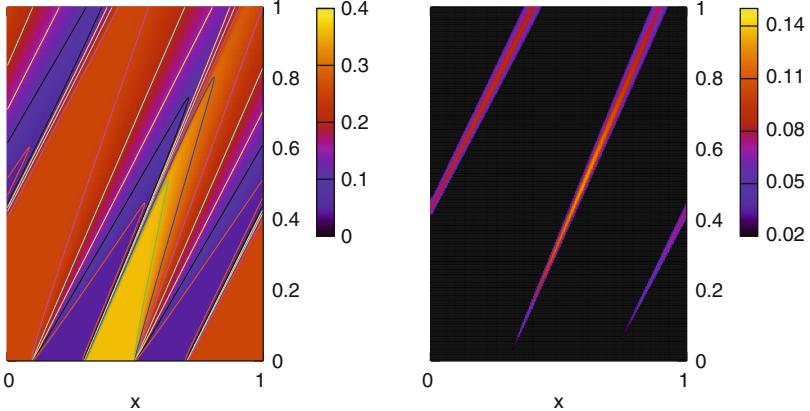


Fig. 18.12 Space-time diagrams of the solution expectation (left) and standard deviation (right) (Adapted from [11])

with degree $\leq N_o$, so that $P = \frac{(N+N_o)!}{N!N_o!}$. Regarding the spatial discretization, a uniform mesh of 200 cells is used to discretize the stochastic conservation law relying on a classical finite volume approach. However, the computation of the numerical fluxes between the finite volumes needs be carefully designed in the stochastic case. The stochastic Galerkin Roe solver proposed in [22, 23] is used in the simulations below.

The initial condition is illustrated in the left panel of Fig. 18.11, which depicts 20 realizations of $U^{IC}(\xi)$ drawn at random. The middle and right panels of Fig. 18.11 report the 20 realizations of the solution at times $t = 0.4$ and 0.9 , respectively. The generation of two expansion waves from $x = 0.1$ to $x = 0.5$ and of two shock waves from $x = 0.3$ to $x = 0.7$ is observed. As time evolves, the first expansion wave reaches the first shock, while the second expansion wave reaches the second shock. Because of the uncertainties in the wave velocities, the instants where the waves catch up are uncertain as well. The dynamics and the impact of uncertainties can be better appreciated on the space-time diagram of the solution expectation and standard deviation plotted for $t \in [0, 1]$ in the left and right panels of Fig. 18.12. The plots highlight the smooth nature of the solution expectation and the steep variations in the solution standard deviation, with maxima reached along the paths of the two shocks.

Clearly, one can expect the approximation of the solution to require a higher stochastic discretization effort along the path of these structures, particularly the shocks with uncertain locations. In addition, the uncertain solution $U(x, t, \xi)$ typically depends on a subset of the uncertain parameters in ξ for a given couple (x, t) . This is evidenced in Fig. 18.13 which illustrates the space-time diagram of the first-order sensitivity indices S_d of the solution (see [11, 24] for a complete definition and detailed procedure for the evaluation of the sensitivity indices). It is observed that, before the merging of the expansion and shock waves ($t < 0.4$), significant

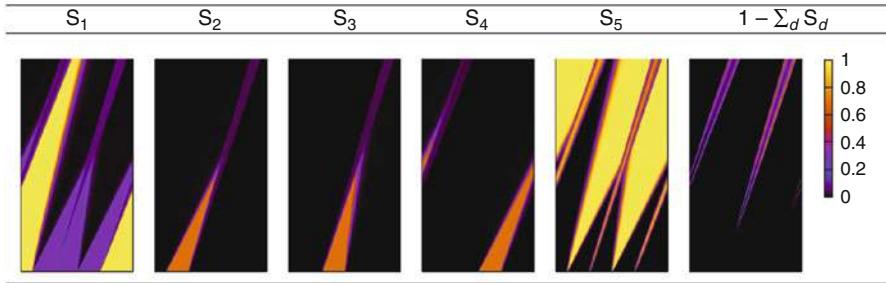


Fig. 18.13 Space-time diagrams $(x, t) \in [0, 1] \times [0, 1]$ of the first-order sensitivity indices and the contribution of sensitivity indices of higher order (Adapted from [11])

values are observed for S_1 , S_2 , S_3 , and S_4 over portions of the computational domain corresponding to the three dependence cones between the waves, where the solution takes one of the three initial uncertain states. The portions of the spatial domain where S_{1-4} take significant values decrease as time increases, indicating the emergence of more and more interactions between the random parameters. On the contrary, because ξ_5 parametrizes the uncertain velocity, A , the significant values of S_5 appear along paths of the different waves and affect a portion of the spatial domain that increases with time. The emergence of interactions between parameters can be appreciated from the rightmost panel of Fig. 18.13, where the quantity $1 - \sum_{d=1}^N S_d$, i.e., the fraction of the variance due to higher-order sensitivity indices, is plotted. This figure shows that interactions primarily take place along the shock paths. We also present the total sensitivity indices T_d which measure the total sensitivity of the solution with respect to the parameter ξ_d . These total sensitivity indices are displayed in Fig. 18.14 as functions of x at the same times as in Fig. 18.11. We recall that $T_d \leq 1$, while $\sum_d T_d > 1$ in general. We observe that T_2 and T_3 (resp. T_4) take significant values over supports that are compact in the neighborhood of the first (resp. second) shock wave and that their magnitude tends to decay in time. On the contrary, the portion of the spatial domain where T_5 reaches a value close to 1 becomes larger as time increases, indicating the extension of the domain of influence of the uncertainty in A . For instance, for $t = 0.9$, the set $\{T_5 \approx 0\}$ is included in $x \in [0.4, 0.5]$, that is, in the only remaining part of the domain where the stochastic solution is spatially constant (see the right plot of Fig. 18.11). Finally, the dynamics of T_1 , which is related to an uncertainty in the initial data that is nonlocal, is much more complex. Specifically, T_1 continues to be significant in areas where the stochastic solution is piecewise constant in space and along the shocks, while in rarefaction waves T_1 becomes quickly insignificant.

5.2.2 Adaptive Computations

Based on the understanding of the solution structure, it is clear that the stochastic discretization should be adapted in both time and space, anisotropically in the parameter domain. In the implementation proposed in [11], each cell of the spatial

mesh supports its own binary tree for its stochastic solution space, which is adapted at each time step of the simulation, by relying on the coarsening and refinement procedures detailed above.

The two enrichment criteria (multidimensional (18.46) and directional (18.52)) were tested for different values for η and No and a fixed maximal resolution $Nr = 6$. For fixed η and No , the multidimensional criterion leads to more refined stochastic discretizations but with only a marginal reduction of the approximation error (as measured by the stochastic approximation error ε_{sd} defined in (18.61) below) compared to the directional criterion. This is illustrated in Fig. 18.15 which shows the time evolution of the total number of SE for the two enrichment criteria, different values of η , and $No = 3$. The rightmost plot shows the corresponding error ε_{sd} as a function to the total number of SE at $t = 0.5$. Because the two enrichment criteria have similar computational complexity, the directional criterion (18.52) is generally preferred.

The dependence on space and time of the adapted trees can be appreciated from Fig. 18.16 which displays the averaged depths of the trees measured as $\log_2 \text{card}(\mathcal{L}(T_i^n))$ in the computation with $\eta = 10^{-4}$ and $No = 3$. This plot shows the adaptation to the local stochastic smoothness; as expected, a finer stochastic discretization along the path of the shock waves is necessary, while a coarser discretization suffices in the expansion waves and in the regions where the solution is spatially constant. The right plot in Fig. 18.16 shows the time evolution of the total number of leaves in the stochastic discretization. We observe a monotonic increase in the number of leaves, with higher rates when additional wave interactions occur and, subsequently, with a roughly constant rate since the stochastic shocks, which dominate the discretization need, affect a portion of the spatial domain growing linearly in time.

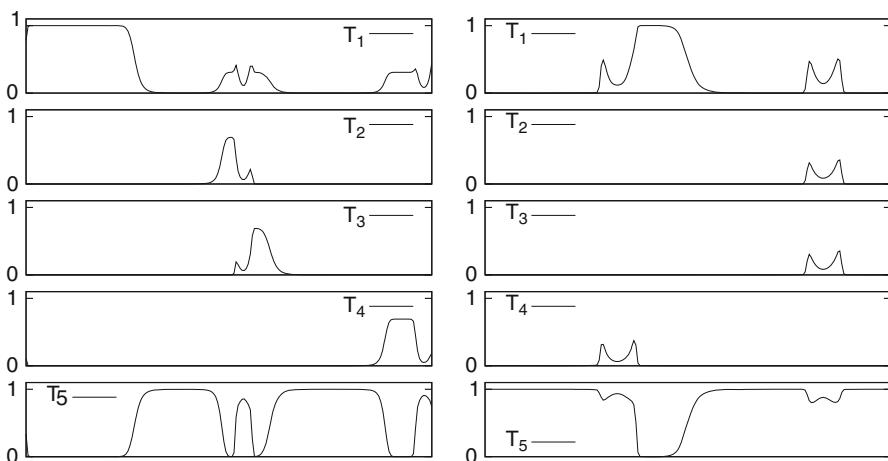


Fig. 18.14 Total sensitivity indices as a function of $x \in [0, 1]$ at $t = 0.4$ (left) and $t = 0.9$ (right) (Adapted from [11])

The anisotropy of the refinement procedure is illustrated in Fig. 18.17, which shows the space-time diagrams of the averaged directional depths defined for $d \in \{1 \dots 5\}$ by $D_d := -\log_2(\sum_{l \in \mathcal{L}(T_i^n)} |S(l)|_d / \text{card}(\mathcal{L}(T_i^n)))$ and the aspect ratio $\rho := \max_{l \in \mathcal{L}(T_i^n)} (\max_d |S(l)|_d / \min_d |S(l)|_d)$ in the rightmost panel. Because ξ_1 parametrizes the uncertain initial condition on the whole domain, this variable affects the velocity of the two shock waves, so that the discretization is finer in the neighborhood of the two shocks. Then, ξ_2 and ξ_3 (resp. ξ_4) affect the velocity of the first shock wave (resp. the second), so that the discretization is finer in the neighborhood of the first (resp. the second) shock. Finally, ξ_5 , which parametrizes the velocity A and therefore affects the velocity of the two shocks, is observed to be the most influential parameter, so that the trees are deeper in the fifth direction; this explains the high values of the aspect ratio near the shocks.

5.2.3 Convergence and Computational Time Analysis

The convergence of the adaptive stochastic method is numerically investigated for a fixed spatial discretization, by estimating the stochastic error at time $t = 0.5$ for different values of η and different polynomial order No. The error is then defined with respect to the semi-discrete solution using the following measure:

$$\varepsilon_{sd}^2 = \Delta x \sum_{i=1}^{Nc} \int_{\mathcal{E}} (U_i^n(\xi) - U_{ex,i}^n(\xi))^2 d\xi, \quad (18.61)$$

where $U_{ex,i}^n$ denotes the exact stochastic semi-discrete solution and $Nc = 200$ is the number of spatial cells. In practice, the error is approximated by means of a Monte Carlo simulation from a uniform sampling of \mathcal{E} . For each element of

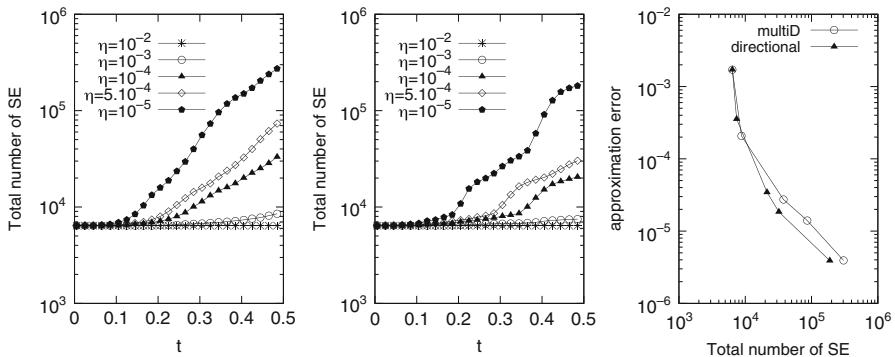


Fig. 18.15 Comparison of the two enrichment criteria for $No = 3$ and different values of η as indicated. Evolution in time of the total number of stochastic elements in the discretization for the multiD criterion (18.46) (left plot) and the directional criterion (18.52) (center plot). Right plot: corresponding error measures ε_{sd} at $t = 0.5$ as a function of the total number of SE for the two enrichment criteria (Adapted from [11])

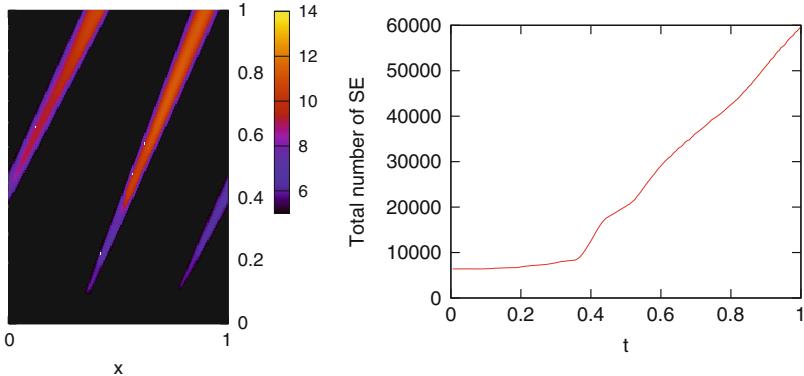


Fig. 18.16 Space-time diagrams of the averaged depth of local trees in \log_2 scale (left) and evolution in time of the total number of stochastic elements (right) (Adapted from [11])

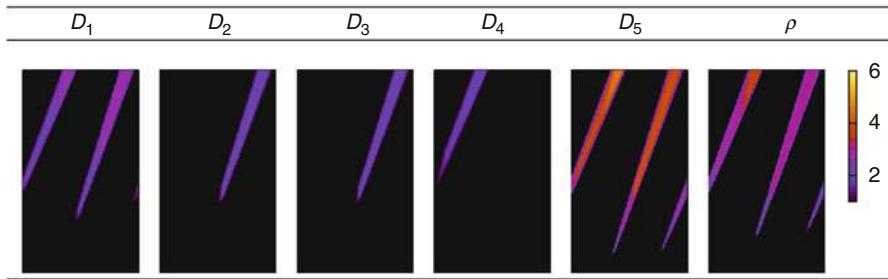


Fig. 18.17 Space-time diagrams $(x, t) \in [0, 1] \times [0, 1]$ of the averaged directional depths and of the aspect ratio (Adapted from [11])

the MC sample, the corresponding discrete deterministic problem is solved with a deterministic Roe solver and the difference with the computed adapted solution is obtained. A total of 10,000 MC samples were used to obtain a well-converged error measure.

Figure 18.18 shows the decay of error $\varepsilon_{\text{sd}}^2$ when the tolerance η in the adaptive algorithm is decreased. The different curves correspond to polynomial degrees $\text{No} \in \{2 \dots 5\}$. The left plot depicts the error measure as a function of the total number of elements (leaves) in the adaptive stochastic discretization at $t^n = 0.5$, namely, the sum over all spatial cells i of $\text{card}(\mathcal{L}(T_i^n))$. The convergence of the semi-discrete solution as η is lowered is first observed for all polynomial degrees tested. In fact, the higher is No , the lower is the error and the faster is the convergence rate, owing to richer approximation spaces for an equivalent number of stochastic elements. However, plotting the error measure ε_{sd} as a function of the total number of degrees of freedom (expansion coefficients), i.e., the total number of leaves times P , as shown in the right plot of Fig. 18.18, we observe that for low resolution (largest η), low polynomial degrees are more efficient than larger ones. On the

contrary, for highly resolved computations (lowest values of η), high polynomial degrees achieve a more accurate approximation for a lower number of degrees of freedom. Such behavior is typical of multiresolution schemes; when numerical diffusion slightly smoothes out the discontinuity, at high-enough resolution, high-degree approximations recover their effectiveness.

To complete this example on stochastic adaptation, computational efficiency is briefly discussed. The purpose is limited here to demonstrate that the overhead arising from adapting the stochastic discretization in space and time is limited. It is first recalled that when considering SE expansions, the determination of the solution is independent from a leaf to another. This characteristic offers opportunities for designing efficient parallel solvers, with different processors dealing with independent subsets of leaves. For the present example, for instance, the computation of the stochastic (Roe) flux over different leaves was performed in parallel, and the only remaining issue concerns the load balancing in the case of complicated trees, particularly for trees evolving dynamically in time. Regarding the other parts of the adaptive procedure, namely, the enrichment and coarsening steps, it is important that they do not consume too much computational resources; otherwise, the gain in adapting the approximation space would be lost. For the present example, numerical experiments demonstrate that the computational times of the enrichment and coarsening steps roughly scale with the number of leaves. This is shown in the plots of Fig. 18.19 which report the CPU times (in arbitrary units) for the advancement of the solution over a single time step as a function of the number of leaves, when using the discretization parameters $No = 2$, $\eta = 10^{-3}$, and $No = 3$, $\eta = 10^{-4}$, respectively. The global CPU times reported are in fact split into different contributions, including flux computation times and coarsening and enrichment

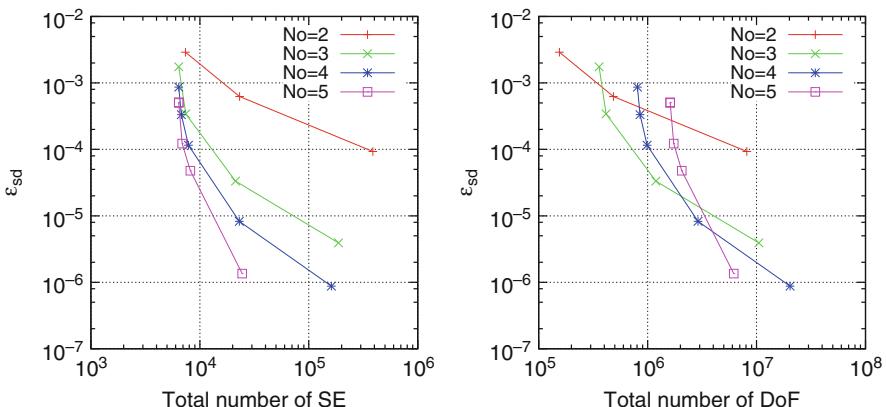


Fig. 18.18 Convergence of the semi-discrete error ε_{sd} at time $t^n = 0.5$ for different values of $\eta \in [10^{-2}, 10^{-5}]$ and different polynomial degrees $No \in \{2 \dots 5\}$. The *left plot* reports the error as a function of the total number of stochastic elements, while the *right plot* shows the error as a function of the total number of degrees of freedom in the stochastic approximation space (Adapted from [11])

computational times. These numerical experiments demonstrate that for the present problem, owing to the representation of the stochastic approximation spaces using binary tree structures, an asymptotically linear computational time in the number of leaves is achieved for the adaptation specific steps of the computation. Further, in this example, the computational times dedicated to adaptivity management (the coarsening and enrichment procedures) are not significant compared to the rest of the computations.

6 Conclusions

This chapter has presented a multiresolution approach for propagating input uncertainties in a model output. The multiresolution is designed to handle situations where the uncertain model solution has complex dependences with respect to the parameters. These include steep variations and discontinuities, a situation that is extremely challenging for classical spectral approaches based on smooth global basis functions.

The key feature of the multiresolution spaces is their piecewise polynomial nature that can, with sufficient resolution, accommodate singularities of various kinds. However, this capability comes at the cost of a potentially significant increase in the dimensionality of approximation spaces, making it essential to consider effective adaptive strategies that are able to tune the local resolution level according to the local complexity of the model solution. We have shown that this can be efficiently achieved by relying on a suitable data structure, namely, binary trees. One significant advantage of the binary tree structure is that, not only does it scale reasonably with the dimensionality of the input parameter space, but it also facilitates the conversion of local expansions (in the SE basis) to detail expansions (in the MW basis), namely, through the recursive application of scale-independent operators. As a result, fast enrichment and coarsening procedures can be implemented without slowing down significantly the computation and so taking full advantage of having a stochastic discretization which is adapted to the solution

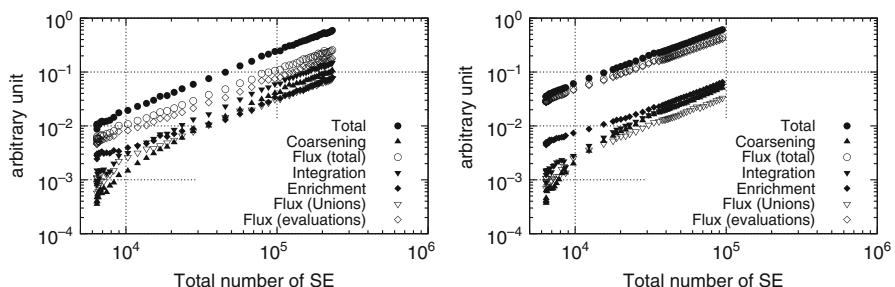


Fig. 18.19 Dependence of the CPU time (per time iteration) on the stochastic discretization measured by the total number of leaves; *left*: $No = 2$ and $\eta = 10^{-3}$; *right*: $No = 3$ and $\eta = 10^{-4}$. The contributions of the various steps of the adaptive algorithm are also shown (Adapted from [11])

needs. All these characteristics are crucial to make the problem tractable from the efficiency perspective.

Multiresolution frameworks and algorithms have enabled the resolution of engineering uncertainty quantification problems that could not be solved with classical spectral approaches, such as global PC representations. These successes include application to conservation law models, compressible flows, stiff chemical systems, and dynamical systems with uncertain parametric bifurcations. MRA schemes, however, are complex to implement, especially in high-dimensional problems. So far, applications have only considered a limited set of uncertain input variables. Another aspect preventing a wider spread of the MRA approach is the absence of available software libraries automating the adaptive refinement procedures. It is expected that such numerical tools will be made available in the coming years. Another area that could also result in wider adoption of MRA schemes concerns the application of associated MW representations in a nonintrusive context. Recent advances in compressive sensing, sparse approximations, and low rank approximation methodologies offer promising avenues toward the emergence of new, highly attractive capabilities.

Acknowledgements The authors are thankful to Dr. Alexandre Ern and Dr. Julie Tryoen for their helpful discussions and for their contributions to the work presented in this chapter.

References

1. Chorin, A.J.: Gaussian fields and random flow. *J. Fluid Mech.* **63**, 21–32 (1974)
2. Meecham, W.C., Jeng, D.T.: Use of the Wiener-Hermite expansion for nearly normal turbulence. *J. Fluid Mech.* **32**, 225 (1968)
3. Le Maître, O., Knio, O., Najm, H., Ghanem, R.: Uncertainty propagation using Wiener-Haar expansions. *J. Comput. Phys.* **197**(1), 28–57 (2004)
4. Le Maître, O.P., Najm, H.N., Ghanem, R.G., Knio, O.M.: Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Phys.* **197**(2), 502–531 (2004)
5. Le Maître, O.P., Najm, H.N., Pébay, P.P., Ghanem, R.G., Knio, O.M.: Multi-resolution-analysis scheme for uncertainty quantification in chemical systems. *SIAM J. Sci. Comput.* **29**(2), 864–889 (2007)
6. Le Maître, O., Knio, O.: Spectral Methods for Uncertainty Quantification. Scientific Computation. Springer, Dordrecht/New York (2010)
7. Gorodetsky, A., Marzouk, Y.: Efficient localization of discontinuities in complex computational simulations. *SIAM J. Sci. Comput.* **36**, A2584–A2610 (2014)
8. Beran, P.S., Pettit, C.L., Millman, D.R.: Uncertainty quantification of limit-cycle oscillations. *J. Comput. Phys.* **217**, 217–247 (2006)
9. Pettit, C.L., Beran, P.S.: Spectral and multiresolution Wiener expansions of oscillatory stochastic processes. *J. Sound Vib.* **294**, 752–779 (2006)
10. Tryoen, J., Le Maître, O., Ndjinga, M., Ern, A.: Multi-resolution analysis and upwinding for uncertain nonlinear hyperbolic systems. *J. Comput. Phys.* **228**, 6485–6511 (2010)
11. Tryoen, J., Le Maître, O., Ern, A.: Adaptive anisotropic spectral stochastic methods for uncertain scalar conservation laws. *SIAM J. Sci. Comput.* **34**, 2459–2481 (2012)
12. Ren, X., Wu, W., Xanthis, L.S.: A dynamically adaptive wavelet approach to stochastic computations based on polynomial chaos – capturing all scales of random modes on independent grids. *J. Comput. Phys.* **230**, 7332–7346 (2011)

13. Sahai, T., Pasini, J.M.: Uncertainty quantification in hybrid dynamical systems. *J. Comput. Phys.* **237**, 411–427 (2013)
14. Pettersson, P., Iaccarino, G., Nordström, J.: A stochastic galerkin method for the Euler equations with roe variable transformation. *J. Comput. Phys.* **257**, 481–500 (2014)
15. Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach. Dover, Minneola (2003)
16. Alpert, B.K.: A class of bases in L_2 for the sparse representation of integral operators. *J. Math. Anal.* **24**, 246–262 (1993)
17. Strang, G.: Introduction to Applied Mathematics. Wellesley-Cambridge Press, Wellesley (1986)
18. Cohen, A., Müller, S., Postel, M., Kaber, S.: Fully adaptive multiresolution schemes for conservation laws. *Math. Comput.* **72**, 183–225 (2002)
19. Cohen, A., Dahmen, W., DeVore, R.: Adaptive wavelet techniques in numerical simulation. In: Stein, E., de Borst, R., Hughes, T.J.R. (eds.) Encyclopedia of Computational Mechanics, vol. 1, pp. 157–197. Wiley, Chichester (2004)
20. Harten, A.: Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun. Pure Appl. Math.* **48**(12), 1305–1342 (1995)
21. Le Maître, O.P., Knio, O.M.: Spectral Methods for Uncertainty Quantification. Springer, Dordrecht/New York (2010)
22. Tryoen, J., Le Maître, O., Ndjinga M., Ern, A.: Intrusive projection methods with upwinding for uncertain nonlinear hyperbolic systems. *J. Comput. Phys.* **228**(18), 6485–6511 (2010)
23. Tryoen, J., Le Maître, O., Ndjinga, M., Ern, A.: Roe solver with entropy corrector for uncertain nonlinear hyperbolic systems. *J. Comput. Appl. Math.* **235**(2), 491–506 (2010)
24. Crestaux, T., Le Maître, O.P., Martinez, J.M.: Polynomial chaos expansion for sensitivity analysis. *Reliab. Eng. Syst. Saf.* **94**(7), 1161–1172 (2009)

Surrogate Models for Uncertainty Propagation and Sensitivity Analysis

19

Khachik Sargsyan

Abstract

For computationally intensive tasks such as design optimization, global sensitivity analysis, or parameter estimation, a model of interest needs to be evaluated multiple times exploring potential parameter ranges or design conditions. If a single simulation of the computational model is expensive, it is common to employ a precomputed surrogate approximation instead. The construction of an appropriate surrogate does still require a number of training evaluations of the original model. Typically, more function evaluations lead to more accurate surrogates, and therefore a careful accuracy-vs-efficiency tradeoff needs to take place for a given computational task. This chapter specifically focuses on polynomial chaos surrogates that are well suited for forward uncertainty propagation tasks, discusses a few construction mechanisms for such surrogates, and demonstrates the computational gain on select test functions.

Keywords

Bayesian inference • Global sensitivity analysis • Polynomial chaos • Regression • Surrogate modeling

Contents

1	Introduction	674
2	Surrogate Modeling for Forward Propagation	675
3	Polynomial Chaos Surrogate	677
3.1	Input PC Specification	677
3.2	PC Surrogate Construction	679
3.3	Surrogate Construction Challenges	683
3.4	Moment Evaluation	688
3.5	Global Sensitivity Analysis	690

K. Sargsyan (✉)

Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA
e-mail: ksargsy@sandia.gov

4 Conclusions	693
Appendix	695
References	695

1 Introduction

Over the last decade, improved computing capabilities have enabled computationally intensive model studies that seemed infeasible before. In particular, models nowadays are being explored in a wide range of condition targeting, e.g., optimal input parameter settings or predictions with uncertainties that correspond to the variability of input conditions. In turn, the algorithmic improvements of such inverse and forward modeling studies have pushed the boundaries of the state-of-the-art computational resources *if* a single simulation of the complex model at hand is expensive enough. In this regard, fast-to-evaluate surrogate models serve to alleviate the computational expense of having to simulate a complex model prohibitively many times. In various applications, while leading to major computational efficiency improvements, such inexpensive approximations of full models across a range of conditions have also been called response surface models [25, 55], metamodels [39, 78], fully equivalent operational models (FEOM) [38, 49], or emulators [5, 8]. Such synthetic, surrogate approximations have been utilized in various computationally intensive studies, such as design optimization [48], parameter estimation [41], global sensitivity analysis [62], uncertainty propagation [25], and reliability analysis [70]. It is useful to differentiate surrogate models as completely unphysical functional approximations unlike, say, reduced order models that tend to preserve certain physical or application-specific mechanisms. Furthermore, parametric surrogates have a predefined form thus formulating surrogate construction as a parameter estimation problem. Polynomial chaos (PC) surrogates, due to orthogonality of the underlying polynomial bases, offer closed-form formulae for the output moment and sensitivity computations. They also allow orthogonal projection formulae with quadrature integration for PC coefficient estimation. However, the projection may not scale well into high-dimensional settings and is generally inaccurate if the function evaluations are noisy. Regression methods – in particular, Bayesian regression – suffer from these difficulties in a much more controllable fashion. Besides reviewing the PC regression approach in general and in a Bayesian setting, the focus of this chapter is on two key forward uncertainty quantification tasks, uncertainty propagation and sensitivity analysis, and the assessment of how surrogate models help accelerate such studies. The proof-of-concept demonstrations will be based on synthetic models that are not expensive to simulate but will help demonstrate the efficiency gains in terms of the number of times the “expensive” model is evaluated.

In this chapter, general principles of surrogate modeling are presented, with specific focus on PC surrogates that offer convenient means for forward uncertainty propagation and sensitivity analysis. Various methods of constructing PC surrogates are presented with an emphasis on Bayesian regression methods that are well positioned to work with noisy, sparse function evaluations and provide efficient

tools for uncertain model prediction, as well as model comparison and selection. The chapter will also highlight key challenges of surrogate modeling, such as model selection, high dimensionality, and nonsmoothness, and review the available methods for tackling these challenges.

2 Surrogate Modeling for Forward Propagation

Consider a model of interest $f(\lambda)$ that depends on d parameters $\lambda = (\lambda_1, \dots, \lambda_d)$ and is defined on a hypercube $[-1, 1]^d$, without loss of generality. The underlying premise is that the model is expensive to simulate at any given value of λ , suggesting potential acceleration of any sampling-intensive study by employing a surrogate approximation $g(\lambda) \approx f(\lambda)$.

Typical surrogate construction requires a set of *training* simulations. In other words, one selects a set $\mathcal{T} = \{\lambda^{(i)}\}_{i=1}^N$ of N training samples and evaluates the model at these parameter settings to obtain $f^{(i)} = f(\lambda^{(i)})$ and arrive at model simulation data pairs $(\lambda^{(i)}, f^{(i)})_{i=1}^N$. Two general classes for surrogates are listed below.

- *Parametric*, where the surrogate has a predefined form with a set of parameters that need to be determined. A parametric surrogate is convenient since one simply needs to store a vector of parameters to fully describe it. However, it typically entails an underlying assumption about some features of the function, e.g., smoothness. Most often, parametric surrogates also are equipped with a mechanism to increase the “resolution” or accuracy within the same functional form, e.g., polynomial order in polynomial chaos (PC) surrogates [44] or Padé approximations [12], or the resolution level for wavelet-based approximations [35].
- *Nonparametric*, where underlying assumptions help derive rules to construct the surrogate for any given λ , without a predefined functional form. Thus, nonparametric surrogates often provide more flexibility, albeit at an extra computational cost associated with their construction and evaluation. It is worth noting that nonparametric surrogates also rely on some structural assumptions about the function and may involve a controlling parameter set, such as correlation length in Gaussian process [51] surrogates, regularization parameter in radial basis function construction [45], or knot locations for spline-based approximations [64].

A useful categorization in surrogate construction is distinction between *interpolation* methods, which require function evaluations to match exactly with the surrogate at the training simulation points, and *regression* approaches that generally aim to “fit” a surrogate function without necessarily matching exactly at the training locations. Interpolation methods are usually nonparametric as they typically do not presume a parameterized functional form. For example, Lagrange polynomial interpolation has a polynomial expansion form but is nonparametric as the specific polynomial bases depend on the training set of input locations [44, 76]. Another

important characterization in surrogate modeling is the distinction between *local* and *global* surrogates. Global surrogates seek for an approximation $g(\boldsymbol{\lambda}) \approx f(\boldsymbol{\lambda})$ that is valid across the full domain, in this case, $\boldsymbol{\lambda} \in [-1, 1]^d$, while local surrogates typically entail a splitting of the domain adaptively or according to a predefined rule, with each partition being associated with its own surrogate construction. As such, local surrogates are essentially nonparametric according to the definition of the latter given above. This chapter primarily focuses on global, parametric surrogates denoted by $g_c(\boldsymbol{\lambda})$, where c is a vector of K parameters describing the surrogate model. Note that for a well-defined parameter estimation problem, one needs $K < N$, i.e., fewer surrogate parameters than the number of training data points. This setting implies that interpolation, i.e., guaranteeing $g_c(\boldsymbol{\lambda}^{(i)}) = f(\boldsymbol{\lambda}^{(i)})$ for all $i = 1, \dots, N$, is essentially infeasible, and one has to operate in the regression setting, i.e., seeking the best set of surrogate parameters c such that $g_c(\boldsymbol{\lambda}) \approx f(\boldsymbol{\lambda})$ with respect to some distance measure.

Uncertainty propagation through the model $f(\boldsymbol{\lambda})$ then generally refers to the representation and analysis of the probability distribution for $f(\boldsymbol{\lambda})$ or summaries of it, given a probability distribution on $\boldsymbol{\lambda}$. Assuming $f(\boldsymbol{\lambda})$ is an expensive function to evaluate, the goal of efficient surrogate construction is to obtain an approximate function $g_c(\boldsymbol{\lambda}) \approx f(\boldsymbol{\lambda})$ with as few training samples $\{f(\boldsymbol{\lambda}^{(i)})\}_{i=1}^N$ as possible and then evaluate $g_c(\boldsymbol{\lambda})$ instead of $f(\boldsymbol{\lambda})$ for the estimation of any statistics of $f(\boldsymbol{\lambda})$. Further in this chapter, the effect of surrogate construction – in terms of both accuracy and efficiency – will be demonstrated on two major uncertainty propagation tasks:

- *Moment evaluation:* This work will specifically focus on the first two moments and demonstrate the computation of model mean $\mathbb{E}f(\boldsymbol{\lambda})$ and model variance $\mathbb{V}f(\boldsymbol{\lambda})$ with surrogates.
- *Sensitivity analysis:* This forward propagation task is otherwise called global sensitivity analysis or variance-based decomposition which helps attribute the output variance to the input dimensions, thus enabling parameter importance ranking and dimensionality reduction studies. The main effect sensitivity of dimension i refers to the fractional contribution of the i -th parameter toward the total variance via $S_i = \frac{\mathbb{V}_{\lambda_i}[\mathbb{E}_{\lambda_{-i}}f(\boldsymbol{\lambda}|\lambda_i)]}{\mathbb{V}f(\boldsymbol{\lambda})}$, where \mathbb{V}_{λ_i} and $\mathbb{E}_{\lambda_{-i}}$ refer to variance with respect to the i -th parameter and mean with respect to the rest of the parameters, respectively.

In lieu of a surrogate, Monte Carlo (MC) methods are often employed for such uncertainty propagation tasks. Further in this chapter, such MC estimators for moments and sensitivities, used for comparison, will be described. With the underlying premise of computationally expensive forward model $f(\boldsymbol{\lambda})$, the measure of success of the surrogate construction will be the estimation of moments and sensitivity indices that requires a smaller number of training samples to achieve an accuracy comparable to the MC method.

Polynomial chaos (PC) expansions are chosen here for the illustration of the surrogate-based acceleration of computationally intensive studies. In fact, they offer

the additional convenience for the selected uncertainty propagation tasks since the first two moments and the sensitivity indices are analytically computable from the PC surrogate [14, 69]. Generally, if analytical expressions are not available, assuming surrogate evaluation is much cheaper than the model evaluation itself, one can employ MC estimates for moments and sensitivities of the surrogate with a much larger number of model evaluations otherwise infeasible with the original function $f(\lambda)$.

3 Polynomial Chaos Surrogate

3.1 Input PC Specification

Polynomial chaos expansions serve as convenient means for propagating input uncertainties to outputs of interest for general computational models [20, 33, 44, 73, 77]. They have been successfully employed in a wide range of applications, from reacting systems [53, 54] to climate science [62, 72], both as means of accelerating forward problems [75] and inverse problems [40, 41]. Consider a vector λ of a finite number of model inputs. In general, a model can possess inputs that are functions, e.g., boundary conditions. Such inputs typically allow some parameterization with a finite number of parameters stemming from, e.g., principal component analysis or discretization. Without losing generality, it is assumed that λ encapsulates all such parameters. Additionally, assume that these parameters are uncertain and are equipped with a joint continuous probability density function (PDF) $\pi_\lambda(\cdot)$, i.e., λ is a random vector. Then, for any random vector $\xi = \{(\xi_1, \dots, \xi_L)\}$ with finite marginal moments and a continuous PDF, one can represent components of λ as polynomial expansions

$$\lambda_i = \sum_{k=0}^{\infty} \lambda_{ik} \Psi_k(\xi) \quad (19.1)$$

that converge in an L_2 sense, *provided* that the moment problem for each ξ_m is uniquely solvable. The latter condition means that the moments of ξ_m uniquely determine the probability density function of ξ_m . This is a key condition and a special case of more general results developed in Ernst et al. [16]. The polynomials in the expansion (19.1) are constructed to be orthogonal with respect to the PDF of ξ , $\pi_\xi(\cdot)$, i.e.,

$$\langle \Psi_k \Psi_j \rangle = \int \Psi_k(\xi) \Psi_j(\xi) \pi_\xi(\xi) d\xi = \delta_{kj} \|\Psi_k\|^2, \quad (19.2)$$

where δ_{kj} is the Kronecker delta, and the polynomial norms are defined as

$$\|\Psi_k\| = \left(\int \Psi_k^2(\xi) \pi_\xi(\xi) d\xi \right)^{1/2}. \quad (19.3)$$

Selection of the best underlying random variable ξ and its dimensionality is generally a nontrivial task. However, for the convenience of the PC construction, and given a generic expectation that a model with d uncertain inputs be represented by a d -dimensional random vector, one selects $L = d$, i.e., λ and ξ to have the same dimensionality. Moreover, it is common to use standard, independent variables as components of ξ , such as uniform on $[-1, 1]$ or standard normal, leading to Legendre or Hermite polynomials, respectively. Typically, as a rule of “rule of thumb,” one chooses the Legendre-Uniform polynomial-variable pair, if the corresponding input parameter has a compact support, and the Hermite-Gauss PC, in case the underlying input parameter has infinite support. More generally, the standard polynomial-variable pairs can be chosen from the Wiener-Askey generalized PC scheme [77] depending on the form of the PDFs $\pi_\lambda(\cdot)$.

The PDF $\pi_\lambda(\cdot)$ or samples from it can be obtained from expert opinion or preliminary calibration studies. With such a PDF available, one can use the inverse of the Rosenblatt transformation that maps the random vector λ to a standard random vector ξ , creating a function $\lambda(\xi)$ and subsequently building a polynomial approximation (19.1) for such a map via projection formula

$$\lambda_{ik} = \frac{1}{||\Psi_k||^2} \langle \lambda_i \Psi_k \rangle = \frac{1}{||\Psi_k||^2} \int \lambda_i(\xi) \Psi_k(\xi) \pi_\xi(\xi) d\xi. \quad (19.4)$$

In practice, one truncates the infinite sum (19.1) to include only K_{in} terms

$$\lambda_i = \sum_{k=0}^{K_{in}-1} \lambda_{ik} \Psi_k(\xi) \quad (19.5)$$

according to a predefined truncation rule. The indexing $k = k(\alpha)$ in the polynomial expansion (19.5) can be selected, say, according to the graded lexicographic ordering [13] of the multi-indices $\alpha = (\alpha_1, \dots, \alpha_d)$ that comprise the dimension-specific orders of each polynomial term $\Psi_{k(\alpha)}(\xi_1, \dots, \xi_d) = \psi_{\alpha_1}(\xi_1) \dots \psi_{\alpha_d}(\xi_d)$ for standard univariate polynomials $\psi_i(\xi)$. A common truncation rule is according to the total degree of the retained polynomials, i.e., $\alpha_1 + \dots + \alpha_d \leq p_{in}$ for some degree p_{in} , leading to $K_{in} = (d + p_{in})! / (d! p_{in}!)$ basis terms. For a description and analysis of various truncation options, see Blatman and Sudret [7] and Sargsyan et al. [62].

In cases when input components λ_i are independent, one arrives at a much simpler, univariate PC expansion for each component, up to order p_i in the i -th dimension,

$$\lambda_i = \sum_{k=0}^{p_i} \lambda_{ik} \psi_k(\xi_i). \quad (19.6)$$

Frequently only the mean and standard deviation, or bounds, of λ_i are reported in the literature or extracted by expert elicitation. In such cases, the maximum entropy considerations lead to Gaussian and uniform random inputs, respectively [29]. The former case corresponds to a univariate, first-order Gauss-Hermite expansion

$\lambda_i = \mu_i + \sigma_i \xi_i$, where $\xi_i \sim N(0, 1)$, while the latter is a univariate, first-order Legendre-Uniform expansion of form

$$\lambda_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} \xi_i, \quad (19.7)$$

where ξ_i are *i.i.d.* uniform random variables on $[-1, 1]$. The classical task of forward propagation of uncertainties for a black-box model $y = f(\boldsymbol{\lambda})$ is then to find the coefficients of the PC representation of the output y :

$$y \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}). \quad (19.8)$$

The implicit relationship between $\boldsymbol{\lambda}$ and y encoded in Eqs. (19.5) and (19.8) serves as a surrogate for the function $f(\boldsymbol{\lambda})$. While the general PC forms (19.5) and (19.6) should be used for uncertainty propagation from input $\boldsymbol{\lambda}$ to the output $f(\boldsymbol{\lambda})$, the PC surrogate in particular is constructed within the same framework using a first-order input (19.7) for all parameters. In such cases, the polynomial function with respect to scaled inputs

$$y = f(\boldsymbol{\lambda}) \simeq g_c(\boldsymbol{\lambda}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}(\boldsymbol{\lambda})), \quad (19.9)$$

where $\boldsymbol{\xi}(\boldsymbol{\lambda})$ is the simple scaling relationship, in this case, derived from the Legendre-Uniform linear PC in Eq. (19.7), i.e., $\xi_i = -\frac{a_i+b_i}{b_i-a_i} + \frac{2}{b_i-a_i} \lambda_i$ for $i = 1, \dots, d$. Similar to the general input PC case (19.5), one typically truncates the polynomial expansion (19.9) according to a total degree p , such that the number of terms is $K = (p+d)!/p!d!$.

3.2 PC Surrogate Construction

Two commonly used approaches for finding PC coefficients c_k , *projection* and *regression*, are highlighted below.

- Projection relies on the orthogonality of the basis functions enabling the formula

$$c_k^{Proj} = \frac{1}{||\Psi_k||^2} \int_{\boldsymbol{\xi}} f(\boldsymbol{\lambda}(\boldsymbol{\xi})) \Psi_k(\boldsymbol{\xi}) \pi_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (19.10)$$

and minimizes the L_2 -distance between the function f and its surrogate

$$\mathbf{c}^{Proj} = \arg \min_{\mathbf{c}} \int_{\boldsymbol{\xi}} \left(f(\boldsymbol{\lambda}(\boldsymbol{\xi})) - \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}) \right)^2 \pi_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (19.11)$$

The projection integral in (19.10) is typically computed by quadrature integration

$$c_k^{Proj} \approx \frac{1}{||\Psi_k||^2} \sum_{q=1}^Q f(\lambda(\xi^{(q)})) \Psi_k(\xi^{(q)}) w_q, \quad (19.12)$$

where quadrature point-weight pairs $\{(\xi^{(q)}, w_q)\}_{q=1}^Q$ are usually chosen such that the integration of the highest-degree polynomial coefficient is sufficiently accurate. For high-dimensional problems, one can use sparse quadrature [4, 19, 23, 76] in order to reduce the number of required function evaluations for a given level of integration accuracy.

- Regression-based approaches directly minimize a distance measure given any set of evaluations of the function $f = \{f(\lambda(\xi^{(n)}))\}_{n=1}^N$ and the surrogate that can be written in a matrix form

$$\mathbf{g}_c = \left\{ \sum_{k=0}^{K-1} c_k \Psi_k(\xi^{(n)}) \right\}_{n=1}^N = \mathbf{G} \mathbf{c} \quad (19.13)$$

denoting the *measurement* matrix by $\mathbf{G}_{nk} = \Psi_k(\xi^{(n)})$. The minimization problem can then generally be written as

$$\mathbf{c}^{Regr} = \arg \min_{\mathbf{c}} \rho(f, \mathbf{g}_c), \quad (19.14)$$

where $\rho(\mathbf{u}, \mathbf{v})$ is a distance measure between two vectors, \mathbf{u} and \mathbf{v} . Most commonly, one chooses an ℓ_2 distance $\rho(f, \mathbf{g}_c) = ||f - \mathbf{g}_c||_2$, leading to a least-squares estimate

$$\mathbf{c}^{LSQ} = \arg \min_{\mathbf{c}} \sum_{n=1}^N \left(f(\lambda(\xi^{(n)})) - \sum_{k=0}^{K-1} c_k \Psi_k(\xi^{(n)}) \right)^2 = \arg \min_{\mathbf{c}} ||f - \mathbf{G} \mathbf{c}||_2 \quad (19.15)$$

that has a closed-form solution

$$\mathbf{c}^{LSQ} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T f. \quad (19.16)$$

Both projection and regression fall into the category of *collocation* approaches in which the surrogate is constructed using a finite set of evaluations of $f(\lambda)$ [40, 74, 76]. While the projection typically requires function evaluations at predefined parameter values $\lambda(\xi^{(q)})$ corresponding to quadrature points, regression has the additional flexibility that the function evaluations or training samples can be located arbitrarily, although the accuracy of the resulting surrogate and numerical stability of the coefficient computation will depend on the distribution of the input parameter samples. Furthermore, if function evaluations are corrupted by noise or occasional faults, the projection loses its attractive properties – in particular, sparse quadrature

is extremely unstable due to negative weights [19]. Regression is much more stable in such noisy training data scenarios and is flexible enough to use various objective functions. For example, if the function is fairly smooth but one expects rare outliers in function evaluations due to, e.g., soft faults in computing environment, then an ℓ_1 objective function is more appropriate instead of the ℓ_2 objective function shown in (19.15) as it leads to a surrogate with the fewest number of “outlier” residuals, inspired by compressed sensing techniques [9, 10, 43, 63]. Besides, the regression approach allows direct extension to a *Bayesian* framework, detailed in the next subsection.

3.2.1 Bayesian Regression

Bayesian methods [6, 11, 65] are well positioned to deal with noisy function evaluations, allow a construction of an *uncertain* surrogate with any number of samples via posterior probability distributions on PC coefficient vector \mathbf{c} , and are efficient in sequential scenarios where the surrogate is updated *online*, i.e., as new evaluations of $f(\lambda)$ arrive [60]. Besides, Bayesian machinery, while computationally more expensive than the simple minimization (19.14), puts the construction of the objective function $\rho(f, g_c)$ within a formal probabilistic context where the objective function can be interpreted as a Bayesian log-likelihood and, say, ℓ_2 or least-squares objective function corresponds to an *i.i.d.* Gaussian assumption for the misfit random variable $f(\lambda) - g_c(\lambda)$. The probabilistic interpretation is particularly attractive when training function evaluations are corrupted by noise or the complex model $f(\lambda)$ itself is stochastic. In these cases projection methods suffer due to instabilities in quadrature integration, while deterministic regression lacks mechanisms to appropriately incorporate the data noise into predictions.

Bayes’ formula in the regression context reads as

$$\overbrace{p(\mathbf{c}|\mathcal{D})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}|\mathbf{c})}^{\text{Likelihood}} \overbrace{p(\mathbf{c})}^{\text{Prior}}}{\underbrace{p(\mathcal{D})}_{\text{Evidence}}}, \quad (19.17)$$

relating a prior probability distribution on surrogate parameters \mathbf{c} to the posterior distribution via the likelihood function

$$\mathcal{L}_{\mathcal{D}}(\mathbf{c}) = p(\mathcal{D}|\mathbf{c}), \quad (19.18)$$

which essentially measures the goodness of fit of the model training data $\mathcal{D} = \{f\}$ to the surrogate model evaluations \mathbf{g}_c for a parameter set \mathbf{c} . As far as the estimation of \mathbf{c} is concerned, the evidence $p(\mathcal{D})$ is simply a normalizing factor. Nevertheless, it plays a crucial role in model comparison and model selection studies, discussed further in this chapter. Since for general likelihoods and high-dimensional parameter vector \mathbf{c} the posterior distribution in (19.17) is hard to compute, one often employs Markov chain Monte Carlo (MCMC) approaches to sample from it. MCMC

methods perform search in the parameter space exploring potential values for \mathbf{c} and, via an accept-reject mechanism, generate a Markov chain that has the posterior PDF as its stationary distribution [17, 22]. MCMC methods require many evaluations of the surrogate model. However, such construction is performed once only, and the surrogates are typically inexpensive to evaluate. Therefore, the main computational burden is in simulating the complex model $f(\cdot)$ at training input locations in order to generate the dataset \mathcal{D} for Bayesian inference via MCMC. As such, one should try to find the most accurate surrogate with as few model evaluations as possible.

The posterior distribution reaches its maximum at the maximum a posteriori (MAP) value. Working with logarithms of the prior and posterior distributions as well as the likelihood, the MAP value solves the optimization problem

$$\mathbf{c}^{MAP} = \arg \max_{\mathbf{c}} \log p(\mathbf{c} | \mathcal{D}) = \arg \max_{\mathbf{c}} [\log L_{\mathcal{D}}(\mathbf{c}) + \log p(\mathbf{c})]. \quad (19.19)$$

Clearly, this problem is equivalent to the deterministic regression with the negative log-likelihood $-\log L_{\mathcal{D}}(\mathbf{c})$ playing the role of an objective function augmented by the regularization term that is the negative log-prior $-\log p(\mathbf{c})$. In principle, the Bayesian framework also allows inclusion of nuisance parameters, e.g., parameters of the prior or the likelihood, that are inferred together with \mathbf{c} and subsequently integrated out to lead to marginal posterior distributions on \mathbf{c} . In a classical case, assuming a uniform prior $p(\mathbf{c})$ and an *i.i.d* Gaussian likelihood with, say, constant variance σ^2 ,

$$-\log L_{\mathcal{D}}(\mathbf{c}) = \frac{N}{2} \log 2\pi + N \log \sigma + \frac{1}{2\sigma^2} \|\mathbf{f} - \mathbf{G}\mathbf{c}\|^2, \quad (19.20)$$

one arrives at a multivariate normal posterior distribution for

$$\mathbf{c} \sim \mathcal{MVN}(\underbrace{(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{f}}_{\mu_c}, \underbrace{\sigma^2 (\mathbf{G}^T \mathbf{G})^{-1}}_{\Sigma_c}). \quad (19.21)$$

Clearly, the posterior mean value is equal to \mathbf{c}^{MAP} and also coincides with the least-squares estimate (19.16). With the probabilistic description of \mathbf{c} , the PC surrogate is *uncertain* and is in fact a Gaussian process with analytically computable mean and covariance functions

$$g_c(\lambda(\xi)) \sim \mathcal{GP}(\Psi(\xi)\mu_c, \Psi(\xi)\Sigma_c\Psi(\xi')^T), \quad (19.22)$$

where $\Psi(\xi)$ is the basis measurement vector at parameter value ξ , i.e., its k -th entry is $\Psi(\xi)_k = \Psi_k(\xi)$. Such an uncertain surrogate is particularly useful when there is a low number of training simulations, and one would like to quantify the epistemic uncertainty due to lack of information. This is the key strength of Bayesian regression – in the presence of noisy or sparse training data, it allows useful surrogate results with an uncertainty estimate that goes with it.

3.3 Surrogate Construction Challenges

3.3.1 Model Selection and Validation

The choice of the degree p of the PC surrogate is a *model selection* problem. It may be selected by the modeler according to prior beliefs about the degree of smoothness of the function $f(\lambda)$ or via a *regularization* term in addition to the objective function (19.14) which constrains the surrogate according to some a priori knowledge about the function $f(\cdot)$, such as smoothness or sparsity. In the Bayesian setting, such a regularization term is equivalent to the log-prior distribution on parameters of the surrogate. Nevertheless, typically, higher-degree surrogates allow more degrees of freedom and correspondingly are closer to the true function evaluated at the training points, i.e., the training error

$$e_{\mathcal{T}}(f, g_c) = \sqrt{\frac{1}{N} \sum_{n=1}^N \left(f(\lambda^{(n)}) - g_c(\lambda^{(n)}) \right)^2} \quad (19.23)$$

reduces with increasing p while the surrogate itself is becoming less and less accurate across the full range of values of λ . Such *overfitting* can be avoided using a separate *validation* set of samples $\mathcal{V} = \{\lambda^{(r)}\}_{r=1}^R$ that is held out, measuring the quality of the surrogate via an error measure

$$e_{\mathcal{V}}(f, g_c) = \sqrt{\frac{1}{R} \sum_{r=1}^R \left(f(\lambda^{(r)}) - g_c(\lambda^{(r)}) \right)^2}. \quad (19.24)$$

Often validation set is split from within the training set in a variety of ways, leading to cross-validation methods that can be used for selection of an appropriate surrogate degree. For a survey of cross-validation methods for model selection, see Arlot et al. [2]. In the absence of a computational budget for cross-validation, one can employ heuristic information criteria, e.g., Akaike information criterion (AIC) or Bayesian information criterion (BIC), that penalize models with too many parameters thus alleviating the overfitting issue to an extent [1]. Bayesian machinery generally allows for model selection strategies via the Bayes factor which has firm probabilistic grounds and is more general than AIC or BIC, albeit is often difficult to evaluate [30]. The Bayes factor is the ratio of the evidence $p(\mathcal{D})$ with one model versus another. While the evidence is difficult to compute in general, it admits a useful decomposition for any parameterized model (first integrating with respect to the posterior distribution then employing Bayes' formula),

$$\begin{aligned} \log p(\mathcal{D}) &= \int \log p(\mathcal{D}) p(c|\mathcal{D}) dc = \int \log \left[\frac{L_{\mathcal{D}}(c)p(c)}{p(c|\mathcal{D})} \right] p(c|\mathcal{D}) dc \\ &= \underbrace{\int \log [L_{\mathcal{D}}(c)] p(c|\mathcal{D}) dc}_{\text{Fit}} - \underbrace{\int \log \left[\frac{p(c|\mathcal{D})}{p(c)} \right] p(c|\mathcal{D}) dc}_{\text{Complexity}}. \end{aligned} \quad (19.25)$$

The posterior average of the log-likelihood measures the ability of the surrogate model to fit the data, while the second term in (19.25) is the relative entropy or the Kullback-Leibler divergence between the prior and posterior distributions [32]. In other words, it measures the information gain about the parameters \mathbf{c} given the training data set \mathcal{D} . The more complex model extracts more information from the data, therefore this “complexity” directly implements Ockham’s razor, which states that with everything else equal, one should choose the simpler model [28]. Most information criteria, such as AIC or BIC, enforce Ockham’s razor in a more heuristic way. For Bayesian regression described above in this chapter, the linearity of the model with respect to \mathbf{c} likelihood allow closed-form expressions for the fit and complexity scores from (19.25) and, consequently, for the evidence:

$$\text{Fit} = -\frac{N}{2} \log[2\pi\sigma^2] - \frac{1}{\sigma^2} \mathbf{f}^T [\mathbf{I} - \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T] \mathbf{f} \quad (19.26)$$

$$\text{Complexity} = \frac{K}{2} \log[2\pi\sigma^2] - \frac{1}{2} \log[\det(\mathbf{G}^T \mathbf{G})] - K \log(2\Delta), \quad (19.27)$$

assuming uniform prior on coefficients, i.e., $p(\mathbf{c}) = (2\Delta)^{-K}$ on $\mathbf{c} \in [-\Delta, \Delta]^K$.

Figure 19.1 shows Bayesian regression results on training data of size $N = 11$ that is computed using a third-order polynomial and “corrupted” by a Gaussian *i.i.d* noise of variance $\sigma^2 = 0.01$. The eighth-order PC surrogate, while having a better fit than the second- or third-order surrogates at the training points, is a clear example of overfitting. It can easily be exposed by a separate set of validation points or by computing the evidence according to (19.25), (19.26), and (19.27). This is illustrated in Fig. 19.2. On the left plot, the log-evidence and its component fit and complexity scores are plotted. The fit score keeps improving with an increasing PC surrogate order, albeit the improvement gets smaller and smaller in general. At the same time, the complexity score keeps decreasing, indicating that higher-order surrogates are more complex, in this case, due to more degrees of freedom. The log-evidence, as a sum of fit and complexity scores, reaches its maximum at surrogate order $p = 3$, which is indeed the true order of the underlying function. On the right plot, the log-evidence is shown again, now with the validation error computed on a separate set of $R = 100$ points according to (19.24). Clearly, the log-evidence accurately finds the correct model with the lowest validation error.

Model selection also strongly depends on the quality of information, i.e., the training samples. As Fig. 19.3 shows, the differentiation between various surrogates and the selection of the best surrogate model is easier with more training data and with lower noise variance.

3.3.2 High Dimensionality

A major challenge for surrogate construction – and PC surrogates in particular – is high dimensionality, i.e., when $d \gg 1$. First of all, the number of function evaluations needed to cover the d -dimensional space grows exponentially with the dimensionality. In other words, in order for the surrogate to be sufficiently

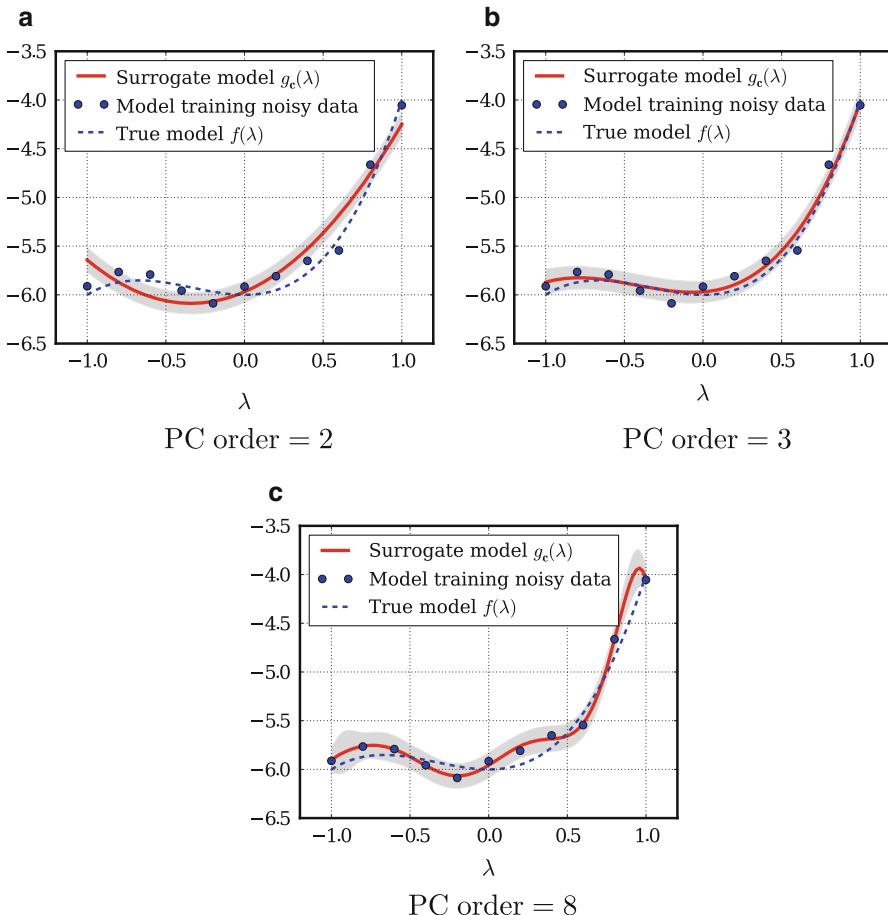


Fig. 19.1 Illustration of Bayesian regression with $N = 11$ training data extracted from noisy evaluations of a third-order polynomial $f(\lambda) = \lambda^3 + \lambda^2 - 6$, with noise variance $\sigma^2 = 0.01$. The gray region indicates the predictive variance of the resulting uncertain surrogate (19.22) together with the data noise variance. Three cases of the PC surrogate order are shown, demonstrating the high-order overfitting on the right plot. (a) PC order = 2. (b) PC order = 3. (c) PC order = 8

accurate, one needs an unfeasibly large number of function evaluations. Optimal computational design strategies can alleviate this to an extent, but they might be expensive and after all this *curse of dimensionality* remains in effect [24]. Second, the selection of multi-indices becomes a challenging task. Standard approaches, such as total-degree expansion or tensor-product expansion, lead to large multi-index sets, $K = (d + p)!/(d!p!)$ with total degree p and $K = p^d$ with degree p per dimension, correspondingly. Such large multi-index basis sets are computationally infeasible for $d \gg 1$ even with low values of p . In such cases, non-isotropic truncation rules [7] or a low-rank multi-index structure in the spirit

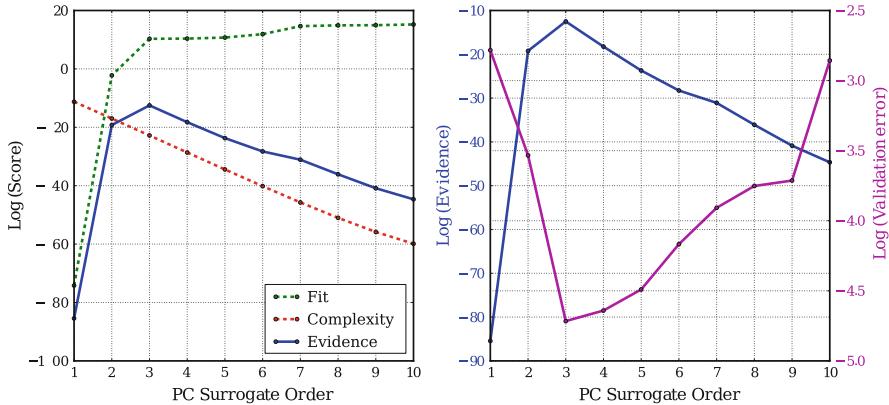


Fig. 19.2 (Left) Illustration of log-evidence together with its component scores for fit and complexity as functions of the PC surrogate order. (Right) The same log-evidence is plotted together with a validation error computed at $R = 100$ separate set of points according to (19.24)

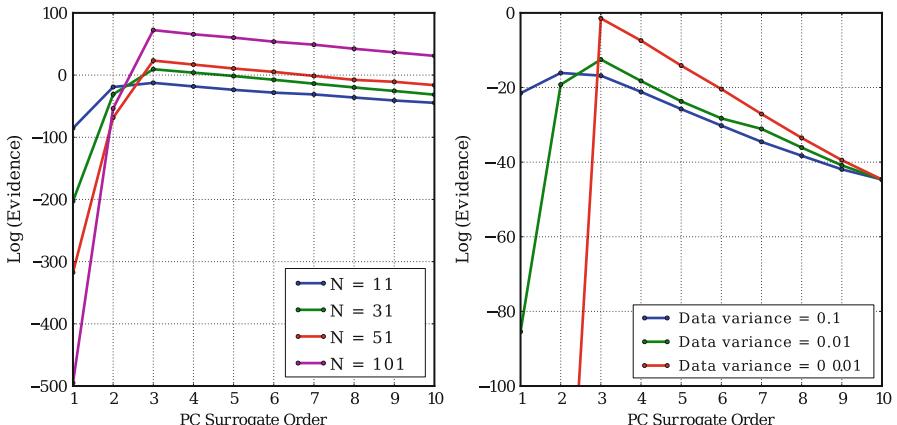


Fig. 19.3 Log-evidence as a function of the PC surrogate model for varying amount of training data (left), and for varying values of the data noise variance (right)

of high-dimensional model representation (HDMR) can be employed [50]. Further flexibility can be provided by adaptive multi-index selection methods as well as by enforcing sparsity constraints on multi-index sets in overdetermined cases, i.e., when $N < K$. In the regression setting, the latter is accomplished by adding an ℓ_1 regularization term to the objective function

$$\mathbf{c}^{Regr} = \arg \min_{\mathbf{c}} [\rho(\mathbf{f}, \mathbf{g}_c) + \mu \|\mathbf{c}\|_1], \quad (19.28)$$

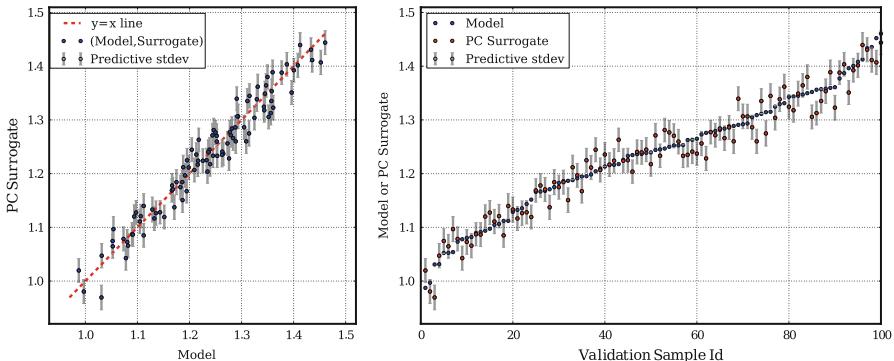


Fig. 19.4 Illustration of PC surrogate constructed via Bayesian compressive sensing, for a 50-dimensional exponential function, defined in the Appendix. The *left plot* shows PC surrogate values versus the actual model evaluations at a separate, validation set of $R = 100$ sample points. The *right plot* illustrates the same result in a different way. Namely, the validation samples are ordered according to ascending model values and plotted together with the corresponding PC surrogate values with respect to a counting index of the validation samples. The grey error bars correspond to posterior predictive standard deviations. The number of training samples used to construct the surrogate is $N = 100$, and the constructed sparse PC expansion has only $K = 11$ bases, with the highest order equal to three

in which the optimal value for μ is selected, e.g., by cross-validation. Such regularization helps find the sparsest, i.e., fewest nonzero values, parameter vector c . This approach originates from *compressed sensing* that made a breakthrough in image processing a decade ago [9, 10, 15]. There is a wide range of recent UQ studies that have benefited from such sparse reconstruction approaches specific to PC basis sets [26, 42, 47, 52, 62]. In a Bayesian context, sparse regression can be accomplished, e.g., via Bayesian compressive sensing (BCS) [3, 62] or Bayesian lasso [46]. A potential opportunity for tackling the curse of dimensionality presents itself if the function has low effective dimension, i.e., only a small set of parameters or combinations thereof impact the function in a nontrivial way. If such structure is discovered, both the computational design of training sets and PC multi-index selection should be informed appropriately, leading to computational efficiency gains. For example, Fig. 19.4 demonstrates PC surrogate construction results with BCS for a 50-dimensional exponential function, defined in the Appendix. The weight decay in the model warrants lower effective dimensionality, i.e., only a few parameters have considerable impact on the model output. A second-order PC surrogate would have $K_{\text{full}} = 52!/(50!2!) = 1326$ terms, while a third-order one would have $K_{\text{full}} = 53!/(50!3!) = 23426$ terms. With only $N = 100$ training samples, this would be a strongly underdetermined problem. Sparse regularization via BCS leads to only 11 terms in this case, producing a sparse PC surrogate – with uncertainty estimate – that is reasonably accurate for a regression in 50 dimensions with 100 training samples.

3.3.3 Nonlinear/Nonsmooth/Discontinuous Forward Model

Another major challenge for parametric surrogates arises when the function, or the forward model, $f(\lambda)$ itself is not amenable to the assumed parametric approximation. Specifically, global PC surrogates built using smooth bases have difficulty approximating functions that are nonsmooth and have discontinuities with respect to the parameters or strong nonlinearities exhibiting sharp growth in some parameter regimes. In such situations, domain decomposition or local adaptivity methods are employed allowing varying degree of resolution in different regions of parameter space and typically leading to piecewise-PC surrogates that can in principle be categorized as nonparametric. Such domain decomposition can be performed in the parameter space [34, 36, 37, 71], or in the “data” space, where the training model evaluations are clustered according to some physically meaningful criteria, followed by a surrogate construction on each cluster and employing classification to “patch” surrogates in the parameter space [59, 61, 62]. Other approaches include basis enrichment [21], or nonparametric surrogates, e.g., kriging [31, 68], that do not have a predefined form and are more flexible representing nonsmooth input-output dependences.

Having a sufficiently accurate and inexpensive surrogate in place, a forward uncertainty propagation task can then be greatly accelerated by replacing the function $f(\lambda)$ with its PC surrogate $g_c(\lambda)$ in sampling-based propagation approaches, e.g., to build an output PDF. Furthermore, because of the orthogonality of the polynomial bases in the output PC surrogate (19.9), one can obtain simple formulae for output statistics of interest that can be written as integral quantities. As discussed earlier, the main focus here will be on two basic uncertainty propagation tasks, moment evaluation and global sensitivity analysis.

3.4 Moment Evaluation

Assuming λ is distributed according to input PC expansion (19.7), i.e., uniform on $\prod_{i=1}^d [a_i, b_i]$, the moments of the function can be evaluated using standard Monte Carlo estimators with M function evaluations:

$$\mathbb{E} f(\lambda) \approx \hat{\mathbb{E}} f(\lambda) = \frac{1}{M} \sum_{m=1}^M f(\lambda(\xi^{(m)})), \quad (19.29)$$

$$\mathbb{V} f(\lambda) \approx \hat{\mathbb{V}} f(\lambda) = \frac{1}{M} \sum_{m=1}^M \left(f(\lambda(\xi^{(m)})) - \hat{\mathbb{E}} f(\lambda) \right)^2. \quad (19.30)$$

These estimates will be used to compare against surrogate-based moment evaluation. Moments of the function $f(\lambda)$ are approximated by the moments of the surrogate $g_c(\lambda)$, which can be computed exactly employing the orthogonality of the basis polynomials:

$$\mathbb{E} f(\lambda) \approx \mathbb{E} g_c(\lambda) = \int_{\lambda} g_c(\lambda) \pi_{\lambda}(\lambda) d\lambda = \int_{\xi} \sum_{k=0}^{K-1} c_k \Psi_k(\xi) \pi_{\xi}(\xi) d\xi = c_0, \quad (19.31)$$

$$\begin{aligned} \mathbb{V} f(\lambda) \approx \mathbb{V} g_c(\lambda) &= \int_{\lambda} (g_c(\lambda) - c_0)^2 \pi_{\lambda}(\lambda) d\lambda \\ &= \int_{\xi} \left(\sum_{k=1}^{K-1} c_k \Psi_k(\xi) \right)^2 \pi_{\xi}(\xi) d\xi = \sum_{k=1}^{K-1} c_k^2 \|\Psi_k\|^2. \end{aligned} \quad (19.32)$$

Note that the exact expressions for surrogate moments (19.31) and (19.32) presume that λ is distributed uniformly on $\prod_{i=1}^d [a_i, b_i]$. For more complicated PDFs $\pi_{\lambda}(\lambda)$, one can employ MC estimates with a much larger ensemble than would have been possible without using the surrogate. This is the main reason for surrogate construction – to replace the complex model in computationally intensive studies. In this chapter, however, specific moment evaluation and sensitivity computation tasks are selected that allow exact analytical formulae with PC surrogates in order to neglect errors due to finite sampling of the surrogate itself.

For the demonstration of moment estimation as well as sensitivity index computation in the next subsection, the classical least-squares regression problem (19.15) is employed for surrogate construction and estimation of PC coefficients, with a solution in (19.16). Note that if Bayesian regression is employed, the resulting uncertain surrogate can lead to an uncertainty estimate in uncertainty estimate in the evaluation of moments as well. Figure 19.5 illustrates the relative error in the estimation of mean and variance for varying number of function evaluations and PC surrogate order, for the oscillatory function, described in the Appendix, with dimensionality $d = 3$. Also, the dimensionality $d = 3$. Also, the MC estimates from (19.29) and (19.30) are shown for comparison. Clearly, for the estimate of the mean of this function, even a linear PC surrogate outperforms MC. In fact, it can be shown that the MC estimate is i.e., constant fit value with the solution of the least-squares problem (19.15). At the same time, the variance problem (19.15). At the same time, the variance estimate requires at least a second-order PC surrogate to guarantee improvement over MC error for all inspected values of the number of function evaluations.

Similar results hold across all the tested models, listed in the Appendix, as Fig. 19.6 suggests. It illustrates relative error convergence with respect to the number of function evaluations, as well as with respect to the PC surrogate order, for all the considered model functions with $d = 3$. Again, one can see that typically the mean and, to the lesser extent, the variance is much more efficient to estimate with the PC surrogate approximation than with the naïve MC estimator. In fact, continuous (model #4, discontinuous first derivative) and corner-peak (model #5, sharp peak at the domain corner) functions are less amenable to polynomial approximation, as the third column suggests, leading to moment estimates that are comparable in accuracy with MC estimates, at least for low surrogate orders.

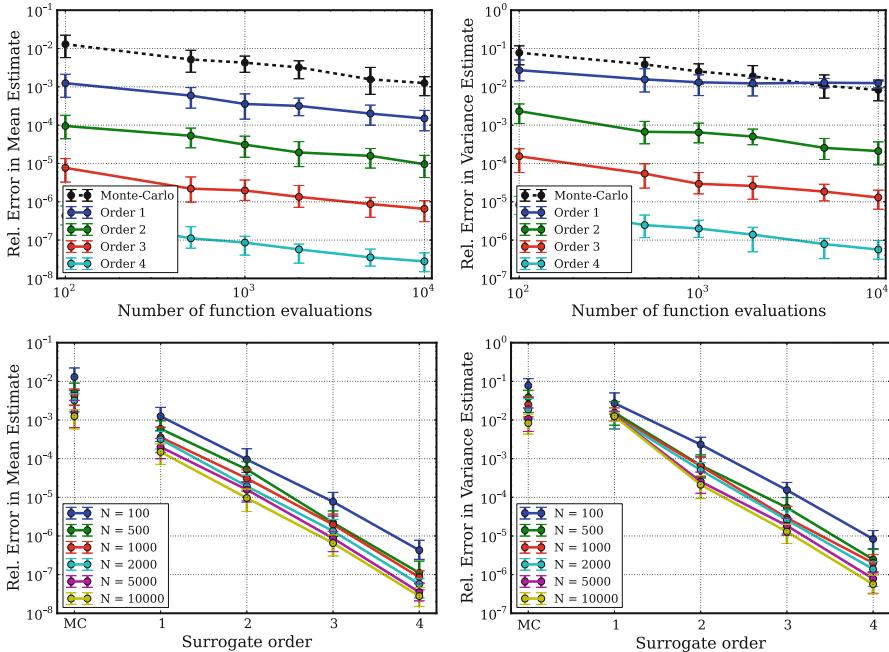


Fig. 19.5 Relative error in the estimation of mean (left column) and variance (right column) with varying number of function evaluations (top row) and PC surrogate order (bottom row) for the three-dimensional oscillatory function. The corresponding MC estimates are highlighted for comparison with dashed lines or as a separate abscissa value. Presented values of relative errors are based on medians over 100 different training sample sets, while error bars indicate 25% and 75% quantiles

3.5 Global Sensitivity Analysis

Sobol's sensitivity indices are useful statistical summaries of model output [57, 67]. They measure fractional contributions of each parameter or group of parameters toward the total output variance. In this chapter, the *main effect sensitivities* are explored, also called first-order sensitivities, that are defined as

$$S_i = \frac{\mathbb{V}_{\lambda_i} \mathbb{E}_{\lambda_{-i}}[f(\boldsymbol{\lambda}) | \lambda_i]}{\mathbb{V} f(\boldsymbol{\lambda})}, \quad (19.33)$$

where \mathbb{V}_{λ_i} and $\mathbb{E}_{\lambda_{-i}}$ indicate variance with respect to the i -th parameter and expectation with respect to the rest of the parameters, respectively. The sampling-based approach developed in Saltelli et al. [58]

$$\hat{S}_i = \frac{1}{\hat{\mathbb{V}} f(\boldsymbol{\lambda})} \left[\frac{1}{M} \sum_{m=1}^M f(\boldsymbol{\lambda}(\xi^{(m)})) \left(f(\boldsymbol{\lambda}(\bar{\xi}_{-i}^{(m)})) - f(\boldsymbol{\lambda}(\bar{\xi}^{(m)})) \right) \right] \quad (19.34)$$

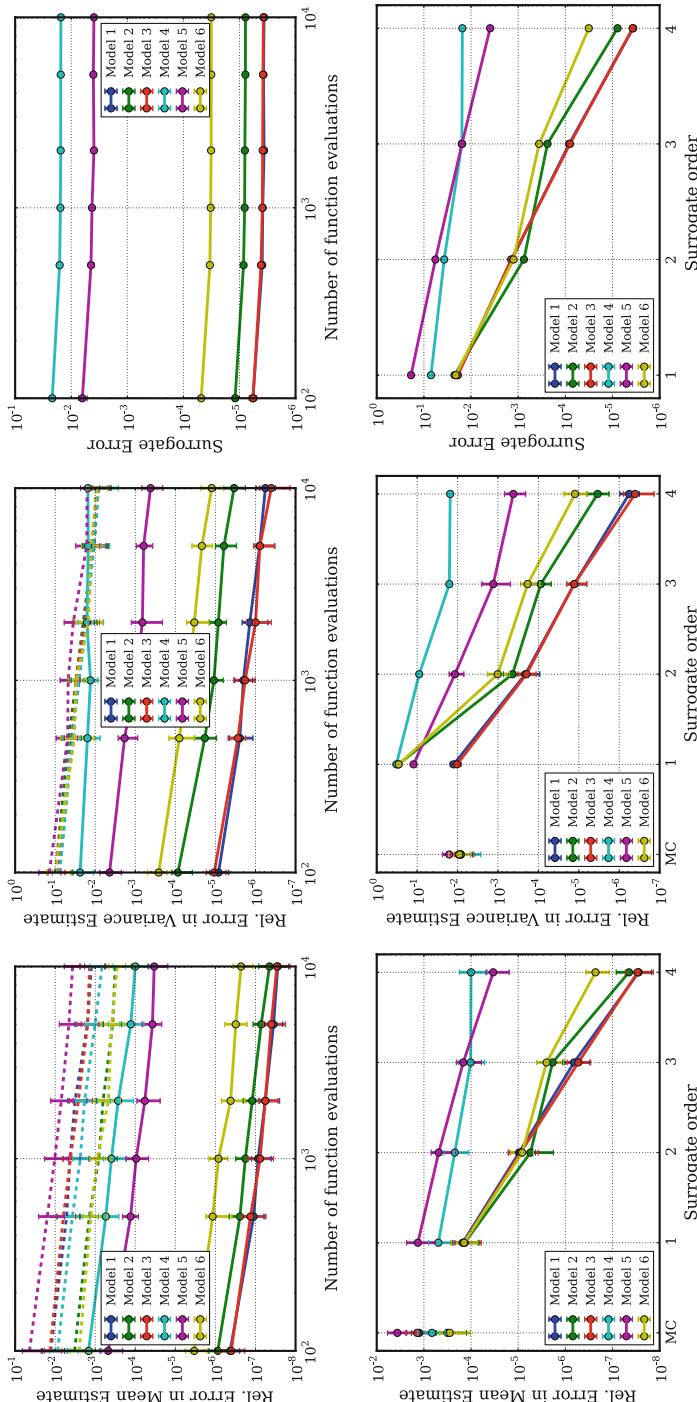


Fig. 19.6 Illustration of the relative error in the estimate of mean (*left column*) and variance (*middle column*), as well as surrogate error (*right column*), i.e., mean-square mismatch of the surrogate and the true model over a separate set of 100 validation samples. The results are illustrated in two ways, with respect to N (*top row*), the number of function evaluations for the PC surrogate order set to $p = 4$, and with respect to the PC surrogate order for $N = 10,000$ (*bottom row*). The corresponding MC estimates are highlighted for comparison with dashed lines or as a separate abscissa value. The error bars indicate 25 % and 75 % quantiles, while the dots correspond to medians over 100 replica simulations. Models are color coded as shown in the legends

is used to compute sensitivity indices in order to compare against the PC surrogate-based approach. These estimates use the sampling set of M points $\xi^{(m)}$ for $m = 1, \dots, M$, the *resampling* set $\bar{\xi}^{(m)}$, and auxiliary points $\tilde{\xi}_{-i}^{(m)}$ that coincide with resampling points except that, at the i -th dimension, the sampling point value is taken, i.e., $\tilde{\xi}_{-i}^{(m)} = (\bar{\xi}_1^{(m)}, \dots, \bar{\xi}_{i-1}^{(m)}, \xi_i^{(m)}, \bar{\xi}_{i+1}^{(m)}, \dots, \bar{\xi}_d^{(m)})$. Note that such an estimate requires $2M$ random samples and, more importantly, $(d + 2)M$ function evaluations. Other estimators for sensitivity indices are also available [27, 56, 58, 66] with similar convergence properties.

When the function $f(\lambda)$ is replaced by a PC surrogate $g_c(\lambda)$, one can compute the sensitivity indices using the orthogonality of the PC basis functions, assuming λ is uniform on the hypercube $\prod_{i=1}^d [a_i, b_i]$ per input PC expansion (19.7),

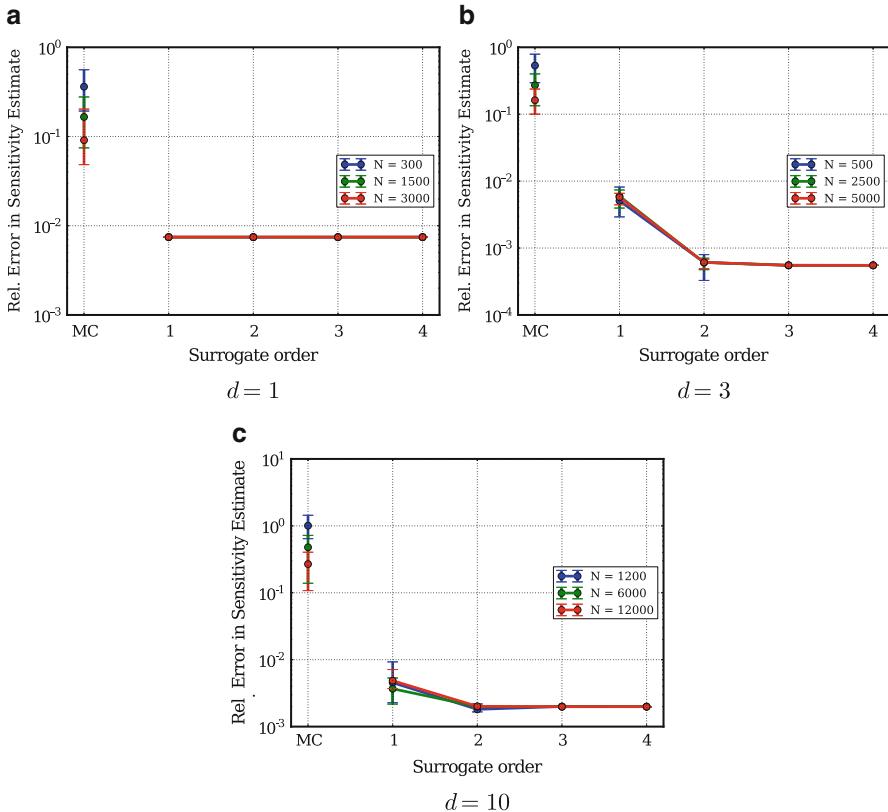


Fig. 19.7 Relative error in estimation of the first main sensitivity index, S_1 , with respect to PC surrogate order with varying number of function evaluations for the oscillatory function for three cases of dimensionality d . The corresponding MC estimates are highlighted for comparison as a separate abscissa value. Presented values of relative errors are based on medians over 100 different training sample sets, while error bars indicate 25 % and 75 % quantiles. (a) $d = 1$. (b) $d = 3$. (c) $d = 10$

$$S_i \approx \frac{\mathbb{V}_{\lambda_i} \mathbb{E}_{\lambda_{-i}} [g_c(\lambda) | \lambda_i]}{\mathbb{V} g_c(\lambda)} = \frac{\sum_{m=1}^{M_i} c_{k_m(i)}^2 \|\Psi_{k_m(i)}\|^2}{\sum_{k=1}^{K-1} c_k^2 \|\Psi_k\|^2}, \quad (19.35)$$

where $(k_1(i), \dots, k_{M_i}(i))$ is the list of indices that correspond to univariate bases in the i -th dimension only, i.e., corresponding to multi-indices of the form $(0, 0, \dots, \alpha_i, \dots, 0, 0)$ with $\alpha_i \neq 0$. Therefore, having constructed the PC surrogate, one can easily compute the sensitivity indices by computing the weighted sum of the squares of appropriately selected PC coefficients.

Figure 19.7 shows convergence with respect to the PC surrogate order for various values of the number of function evaluations, for the oscillatory function, described in the Appendix, with three different dimensionalities. Clearly, a PC surrogate, even the linear one, leads to improved errors compared to an MC-based estimate (19.34). The convergence with order does not reduce below a certain value due to the “true” sensitivity being computed only approximately via the MC formula with a large $M = 10^5$.

4 Conclusions

This chapter has focused on surrogate modeling for computationally expensive forward models. Specifically, polynomial chaos (PC) surrogates have been studied for the purposes of forward propagation and variance-based sensitivity analysis. Among various methods of PC coefficient computation, Bayesian regression has been highlighted as it offers a host of advantages. First of all, it is applicable with noisy function evaluations and allows likelihood or objective function construction stemming from formal probabilistic assumptions. Second, it provides robust answers with an uncertainty certificate with any number of and arbitrarily distributed training function evaluations, which is often very practical for complex physical models, e.g., climate models that are simulated on supercomputers with a single simulation taking hours or days. This is particularly useful for high-dimensional surrogate construction, when one necessarily operates under the condition of sparse training data. Relatedly, high dimensionality or a large number of surrogate parameters often lead to an underdetermined problem in which case Bayesian sparse learning methods such as Bayesian compressive sensing provide an uncertainty-enabled alternative to classical, deterministic regularization methods. Besides, Bayesian methods allow sequential updating of surrogates as new function evaluations arrive by encoding the state of knowledge about the surrogate parameters into a prior distribution. Finally, Bayesian regression allows an efficient solution to the surrogate model selection problem via evidence computation and Bayes factors as demonstrated on toy examples in this chapter. Irrespective of the construction methodology, a PC surrogate enables drastic computational savings when evaluating moments or sensitivity indices of complex models, as illustrated on a select class of functions with tunable dimensionality and a varying degree of smoothness.

Table 19.1 Test functions used in the studies of this section. The shift parameters are set to $u_i = 0.3$ for all dimensions $i = 1, \dots, d$, while the weight parameters are selected as $w_i = C/i$ with normalization constant $C = 1/\sum_{i=1}^d i^{-1}$ to ensure $\sum_{i=1}^d w_i = 1$. The variance formula for the product-peak function is $v(\mathbf{u}, \mathbf{w}) = \prod_{i=1}^d w_i^4 \left(\frac{1-u_i}{2(1+w_i^2(1-u_i)^2)} + \frac{u_i}{2w_i} (\arctan(w_i(1-u_i)) + \arctan(w_i u_i)) \right) - m(\mathbf{u}, \mathbf{w})^2$

Id	Function	Formula $f_{\mathbf{u}, \mathbf{w}}(\lambda)$	Exact mean	Exact variance
			$m(\mathbf{u}, \mathbf{w}) = \int_0^1 f_{\mathbf{u}, \mathbf{w}}(\lambda) d\lambda$	$v(\mathbf{u}, \mathbf{w}) = \int_0^1 (f_{\mathbf{u}, \mathbf{w}}(\lambda) - m(\mathbf{u}, \mathbf{w}))^2 d\lambda$
1	Oscillatory	$\cos \left(2\pi u_1 + \sum_{i=1}^d w_i \lambda_i \right)$	$\left(\cos 2\pi u_1 + \frac{1}{2} \sum_{i=1}^d w_i \right) \prod_{i=1}^d \frac{2 \sin(w_i/2)}{w_i}$	$\frac{1}{2} + \frac{1}{2} m(2\mathbf{u}, 2\mathbf{w}) - m(\mathbf{u}, \mathbf{w})^2$
2	Gaussian	$\exp \left(- \sum_{i=1}^d w_i^2 (\lambda_i - u_i)^2 \right)$	$\prod_{i=1}^d \frac{\sqrt{\pi}}{2w_i} (\text{erf}(w_i(1-u_i)) + \text{erf}(w_i u_i))$	$m(\mathbf{u}, \sqrt{2}\mathbf{w}) - m(\mathbf{u}, \mathbf{w})^2$
3	Exponential	$\exp \left(\sum_{i=1}^d w_i (\lambda_i - u_i) \right)$	$\prod_{i=1}^d \frac{1}{w_i} (\exp(w_i(1-u_i)) - \exp(-w_i u_i))$	$m(\mathbf{u}, 2\mathbf{w}) - m(\mathbf{u}, \mathbf{w})^2$
4	Continuous	$\exp \left(- \sum_{i=1}^d w_i \lambda_i - u_i \right)$	$\prod_{i=1}^d \frac{1}{w_i} (2 - \exp(-w_i u_i) - \exp(w_i(u_i - 1)))$	$m(\mathbf{u}, 2\mathbf{w}) - m(\mathbf{u}, \mathbf{w})^2$
5	Corner peak	$\left(1 + \sum_{i=1}^d w_i \lambda_i \right)^{-(d+1)}$	$\frac{1}{d! \prod_{i=1}^d w_i} \sum_{r \in \{0,1\}^d} \frac{(-1)^{ r _1}}{1 + \sum_{i=1}^d w_i r_i}$	Sampling estimator (19.34) with $M = 10^7$
6	Product peak	$\prod_{i=1}^d \frac{w_i^2}{1 + w_i^2 (\lambda_i - u_i)^2}$	$\prod_{i=1}^d w_i (\arctan(w_i(1-u_i)) + \arctan(w_i u_i))$	See the caption

Appendix

Table 19.1 shows six classes of functions employed in the numerical tests in this chapter. All functions, except the exponential one, are taken from the classical Genz family of test functions [18]. The weight parameters of the test functions are chosen according to a predefined “decay” rate $w_i = C/i$ and a normalization factor $C = 1/\sum_{i=1}^d i^{-1}$ to ensure $\sum_{i=1}^d w_i = 1$. The exact moments are analytically available and are used for reference to compare against, with the exception of the corner-peak function, for which the variance estimator (19.30) with sampling size $M = 10^7$ is used. The “true” reference values ‘true’ reference values for sensitivity indices S_i are also computed via Monte Carlo, using the estimates (19.34) with $M = 10^5$. The functions are defined on $\lambda \in [0, 1]^d$; assuming the inputs are *i.i.d* uniform random variables, the underlying linear input PC expansions are simple linear transformations $\lambda_i = 0.5\xi_i + 0.5$, relating the “physical” model inputs $\lambda_i \in [0, 1]$ to the PC surrogate inputs $\xi_i \in [-1, 1]$, for $i = 1, \dots, d$.

References

1. Acquah, H.: Comparison of Akaike information criterion (AIC) and Bayesian information criterion (BIC) in selection of an asymmetric price relationship. *J. Dev. Agric. Econ.* **2**(1), 001–006 (2010)
2. Arlot, S., Celisse, A., et al.: A survey of cross-validation procedures for model selection. *Stat. Surv.* **4**, 40–79 (2010)
3. Babacan, S., Molina, R., Katsaggelos, A.: Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **19**(1), 53–63 (2010)
4. Barthelmann, V., Novak, E., Ritter, K.: High-dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.* **12**, 273–288 (2000)
5. Bastos, L., O’Hagan, A.: Diagnostics for Gaussian process emulators. *Technometrics* **51**(4), 425–438 (2009)
6. Bernardo, J., Smith, A.: Bayesian Theory. Wiley Series in Probability and Statistics. Wiley, Chichester (2000)
7. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys.* **230**(6), 2345–2367 (2011)
8. Borgonovo, E., Castaings, W., Tarantola, S.: Model emulation and moment-independent sensitivity analysis: an application to environmental modelling. *Environ. Model. Softw.* **34**, 105–115 (2012)
9. Candès, E., Romberg, J.: Sparsity and incoherence in compressive sampling. *Inverse Probl.* **23**(3), 969–985 (2007)
10. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **52**(2), 489–509 (2006)
11. Carlin, B.P., Louis, T.A.: Bayesian Methods for Data Analysis. Chapman and Hall/CRC, Boca Raton (2011)
12. Chantrasmi, T., Doostan, A., Iaccarino, G.: Padé-Legendre approximants for uncertainty analysis with discontinuous response surfaces. *J. Comput. Phys.* **228**(19), 7159–7180 (2009)
13. Cox, D.A., Little, J., O’Shea, D.: Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, New York (1997)
14. Crestaux, T., Le Maître, O., Martinez, J.: Polynomial chaos expansion for sensitivity analysis. *Reliab. Eng. Syst. Saf.* **94**(7), 1161–1172 (2009)
15. Donoho, D.: Compressed sensing. *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006)

16. Ernst, O., Mugler, A., Starkloff, H.J., Ullmann, E.: On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.* **46**, 317–339 (2012)
17. Gamerman, D., Lopes, H.F.: *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and Hall/CRC, Boca Raton (2006)
18. Genz, A.: Testing multidimensional integration routines. In: *Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*. Elsevier North-Holland, Inc., pp 81–94 (1984)
19. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms* **18**, 209–232 (1998). doi:10.1023/A:1019129717644, (also as SFB 256 preprint 553, Univ. Bonn, 1998)
20. Ghanem, R., Spanos, P.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
21. Ghosh, D., Ghanem, R.: Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions. *Int. J. Numer. Method Eng.* **73**, 162–174 (2008)
22. Gilks, W.R.: *Markov Chain Monte Carlo*. Wiley Online Library (2005)
23. Griebel, M.: Sparse grids and related approximation schemes for high dimensional problems. In: *Proceedings of the Conference on Foundations of Computational Mathematics*. Santander, Spain (2005)
24. Huan, X., Marzouk, Y.: Simulation-based optimal Bayesian experimental design for nonlinear systems. *J. Comput. Phys.* **232**, 288–317 (2013)
25. Isukapalli, S., Roy, A., Georgopoulos, P.: Stochastic response surface methods (SRSMs) for uncertainty propagation: application to environmental and biological systems. *Risk Anal.* **18**(3), 351–363 (1998)
26. Jakeman, J.D., Eldred, M.S., Sargsyan, K.: Enhancing ℓ_1 -minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.* **289**, 18–34 (2015)
27. Jansen, M.J.: Analysis of variance designs for model output. *Comput. Phys. Commun.* **117**(1), 35–43 (1999)
28. Jefferys, W.H., Berger, J.O.: Ockham's razor and Bayesian analysis. *Am. Sci.* **80**, 64–72 (1992)
29. Kapur, J.N.: *Maximum-Entropy Models in Science and Engineering*. Wiley, New Delhi (1989)
30. Kass, R., Raftery, A.: Bayes factors. *J. Am. Stat. Assoc.* **90**(430), 773–795 (1995)
31. Kersaudy, P., Sudret, B., Varsier, N., Picon, O., Wiart, J.: A new surrogate modeling technique combining Kriging and polynomial chaos expansions—application to uncertainty analysis in computational dosimetry. *J. Comput. Phys.* **286**, 103–117 (2015)
32. Kullback, S., Leibler, R.: On information and sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (1951)
33. Le Maître, O., Knio, O.: *Spectral Methods for Uncertainty Quantification*. Springer, New York (2010)
34. Le Maître, O., Knio, O., Debusschere, B., Najm, H., Ghanem, R.: A multigrid solver for two-dimensional stochastic diffusion equations. *Comput. Methods Appl. Mech. Eng.* **192**, 4723–4744 (2003)
35. Le Maître, O., Ghanem, R., Knio, O., Najm, H.: Uncertainty propagation using Wiener-Haar expansions. *J. Comput. Phys.* **197**(1), 28–57 (2004a)
36. Le Maître, O., Najm, H., Ghanem, R., Knio, O.: Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Phys.* **197**, 502–531 (2004b)
37. Le Maître, O., Najm, H., Pébay, P., Ghanem, R., Knio, O.: Multi-resolution analysis scheme for uncertainty quantification in chemical systems. *SIAM J. Sci. Comput.* **29**(2), 864–889 (2007)
38. Li, G., Rosenthal, C., Rabitz, H.: High dimensional model representations. *J. Phys. Chem. A* **105**, 7765–7777 (2001)
39. Marrel, A., Iooss, B., Laurent, B., Roustant, O.: Calculations of Sobol indices for the Gaussian process metamodel. *Reliab. Eng. Syst. Saf.* **94**(3), 742–751 (2009)
40. Marzouk, Y.M., Najm, H.N.: Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *J. Comput. Phys.* **228**(6), 1862–1902 (2009)
41. Marzouk, Y.M., Najm, H.N., Rahn, L.A.: Stochastic spectral methods for efficient Bayesian solution of inverse problems. *J. Comput. Phys.* **224**(2), 560–586 (2007)

42. Mathelin, L., Gallivan, K.: A compressed sensing approach for partial differential equations with random input data. *Commun. Comput. Phys.* **12**(4), 919–954 (2012)
43. Moore, B., Natarajan, B.: A general framework for robust compressive sensing based nonlinear regression. In: 2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM), Hoboken. IEEE, pp 225–228 (2012)
44. Najm, H.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Ann. Rev. Fluid Mech.* **41**(1), 35–52 (2009). doi:10.1146/annurev.fluid.010908.165248
45. Orr, M.: Introduction to radial basis function networks. Technical Report, Center for Cognitive Science, University of Edinburgh (1996)
46. Park, T., Casella, G.: The Bayesian Lasso. *J. Am. Stat. Assoc.* **103**(482), 681–686 (2008)
47. Peng, J., Hampton, J., Doostan, A.: A weighted ℓ_1 -minimization approach for sparse polynomial chaos expansions. *J. Comput. Phys.* **267**, 92–111 (2014)
48. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. *Prog. Aerosp. Sci.* **41**(1), 1–28 (2005)
49. Rabitz, H., Alis, O.F.: General foundations of high-dimensional model representations. *J. Math. Chem.* **25**, 197–233 (1999)
50. Rabitz, H., Alis, O.F., Shorter, J., Shim, K.: Efficient input-output model representations. *Comput. Phys. Commun.* **117**, 11–20 (1999)
51. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT, Cambridge (2006)
52. Rauhut, H., Ward, R.: Sparse Legendre expansions via ℓ_1 -minimization. *J. Approx. Theory* **164**(5), 517–533 (2012)
53. Reagan, M., Najm, H., Ghanem, R., Knio, O.: Uncertainty quantification in reacting flow simulations through non-intrusive spectral projection. *Combust. Flame* **132**, 545–555 (2003)
54. Reagan, M., Najm, H., Debusschere, B., Le Maître, O., Knio, O., Ghanem, R.: Spectral stochastic uncertainty quantification in chemical systems. *Combust. Theory Model.* **8**, 607–632 (2004)
55. Rutherford, B., Swiler, L., Paez, T., Urbina, A.: Response surface (meta-model) methods and applications. In: Proceedings of 24th International Modal Analysis Conference, St. Louis, pp 184–197 (2006)
56. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002). doi:10.1016/S0010-4655(02)00280-1
57. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models. Wiley, Chichester/Hoboken (2004)
58. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**(2), 259–270 (2010)
59. Sargsyan, K., Debusschere, B., Najm, H., Le Maître, O.: Spectral representation and reduced order modeling of the dynamics of stochastic reaction networks via adaptive data partitioning. *SIAM J. Sci. Comput.* **31**(6), 4395–4421 (2010)
60. Sargsyan, K., Safta, C., Debusschere, B., Najm, H.: Multiparameter spectral representation of noise-induced competence in *Bacillus subtilis*. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **9**(6), 1709–1723 (2012a). doi:10.1109/TCBB.2012.107
61. Sargsyan, K., Safta, C., Debusschere, B., Najm, H.: Uncertainty quantification given discontinuous model response and a limited number of model runs. *SIAM J. Sci. Comput.* **34**(1), B44–B64 (2012b)
62. Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., Thornton, P.: Dimensionality reduction for complex models via Bayesian compressive sensing. *Int. J. Uncertain. Quantif.* **4**(1), 63–93 (2014). doi:10.1615/Int.J.UncertaintyQuantification.2013006821
63. Sargsyan, K., Rizzi, F., Mycek, P., Safta, C., Morris, K., Najm, H., Le Maître, O., Knio, O., Debusschere, B.: Fault resilient domain decomposition preconditioner for PDEs. *SIAM J. Sci. Comput.* **37**(5), A2317–A2345 (2015)
64. Schumaker, L.: Spline Functions: Basic Theory. Cambridge University Press, New York (2007)

65. Sivia, D.S., Skilling, J.: *Data Analysis: A Bayesian Tutorial*, 2nd edn. Oxford University Press, Oxford (2006)
66. Sobol, I.M.: Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.* **1**, 407–414 (1993)
67. Sobol, I.M.: Theorems and examples on high dimensional model representation. *Reliab. Eng. Syst. Saf.* **79**, 187–193 (2003)
68. Stein, M.L.: *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, New York (2012)
69. Sudret, B.: Global sensitivity analysis using polynomial Chaos expansions. *Reliab. Eng. Syst. Saf.* (2007). doi:10.1016/j.ress.2007.04.002
70. Sudret, B.: Meta-models for structural reliability and uncertainty quantification. In: *Asian-Pacific Symposium on Structural Reliability and its Applications*, Singapore, pp 1–24 (2012)
71. Wan, X., Karniadakis, G.E.: An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.* **209**, 617–642 (2005)
72. Webster, M., Tatang, M., McRae, G.: Application of the probabilistic collocation method for an uncertainty analysis of a simple ocean model. Technical report, MIT Joint Program on the Science and Policy of Global Change Reports Series 4, MIT (1996)
73. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**, 897–936 (1938). doi:10.2307/2371268
74. Xiu, D.: Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys.* **2**(2), 293–309 (2007)
75. Xiu, D.: Fast numerical methods for stochastic computations: a review. *J. Comput. Phys.* **5**(2–4), 242–272 (2009)
76. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
77. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002). doi:10.1137/S1064827501387826
78. Zuniga, M.M., Kucherenko, S., Shah, N.: Metamodelling with independent and dependent inputs. *Comput. Phys. Commun.* **184**(6), 1570–1580 (2013)

Stochastic Collocation Methods: A Survey

20

Dongbin Xiu

Abstract

Stochastic collocation (SC) has become one of the major computational tools for uncertainty quantification. Its primary advantage lies in its ease of implementation. To carry out SC, one needs only a reliable deterministic simulation code that can be run repetitively at different parameter values. And yet, the modern-day SC methods can retain the high-order accuracy properties enjoyed by most of other methods. This is accomplished by utilizing the large amount of literature in the classical approximation theory. Here we survey the major approaches in SC. In particular, we focus on a few well-established approaches: interpolation, regression, and pseudo projection. We present the basic formulations of these approaches and some of their major variations. Representative examples are also provided to illustrate their major properties.

Keywords

Compressed sensing • Interpolation • Least squares • Stochastic collocation

Contents

1	Introduction	700
2	Definition of Stochastic Collocation	701
3	Stochastic Collocation via Interpolation	702
3.1	Formulation	702
3.2	Interpolation SC on Structured Samples	704
3.3	Interpolation on Unstructured Samples	708
4	Stochastic Collocation via Regression	709
4.1	Over-sampled Case: Least Squares	709
4.2	Under-sampled Case: Sparse Approximations	710

D. Xiu (✉)

Department of Mathematics and Scientific Computing and Imaging Institute, University of Utah,
Salt Lake City, UT, USA
e-mail: dongbin.xiu@utah.edu

5 Stochastic Collocation via Pseudo Projection	712
6 Summary	714
References	714

1 Introduction

Stochastic collocation (SC) is a sampling-based method. The term “collocation” originates from the deterministic numerical methods for differential equations, where one seeks to satisfy the governing continuous equations discretely at a set of *collocation points*. This is to the contrary of Galerkin method, where one seeks to satisfy the governing equation in a weak form. *Stochastic collocation* was first termed in [31], although the idea and its application have existed long before that.

To illustrate the idea, let us consider, for a spatial domain D and time domain $[0, T]$ with $T > 0$, the following partial differential equation (PDE) system:

$$\begin{cases} u_t(x, t, Z) = \mathcal{L}(u), & D \times (0, T] \times I_Z, \\ \mathcal{B}(u) = 0, & \partial D \times [0, T] \times I_Z, \\ u = u_0, & D \times \{t = 0\} \times I_Z, \end{cases} \quad (20.1)$$

where $I_Z \subset \mathbb{R}^d$, $d \geq 1$ is the support of the uncertain parameters $Z = (Z_1, \dots, Z_d)$ and \mathcal{B} is the boundary condition operator. The solution is then a mapping

$$u(x, t, Z) : \bar{D} \times [0, T] \times I_Z \rightarrow \mathbb{R},$$

where for the simplicity of exposition we consider a scalar equation. In UQ computations, we are primarily interested in the solution dependence in the parameter space, that is,

$$u(\cdot, Z) : I_Z \rightarrow \mathbb{R}, \quad (20.2)$$

where the dependence on the spatial and temporal variables (x, t) is suppressed. Hereafter, all statements are made for any fixed x and t .

In stochastic collocation, the system (20.1) is solved in a discrete manner. More specifically, we seek to enforce the equation at a discrete set of nodes – “collocation points.” Let $\Theta_M = \{Z^{(j)}\}_{j=1}^M \subset I_Z$ be a set of (prescribed) nodes in the random space, where $M \geq 1$ is the number of nodes. Then in SC, we enforce (20.1) at the node $Z^{(j)}$, for all $j = 1, \dots, M$, by solving

$$\begin{cases} u_t(x, t, Z^{(j)}) = \mathcal{L}(u), & D \times (0, T], \\ \mathcal{B}(u) = 0, & \partial D \times [0, T], \\ u = u_0, & D \times \{t = 0\}. \end{cases} \quad (20.3)$$

It is easy to see that for each j , (20.3) is a deterministic problem because the value of the random parameter Z is fixed. Therefore, solving the system poses no difficulty provided one has a well-established deterministic algorithm. Let $u^{(j)} = u(\cdot, Z^{(j)})$, $j = 1, \dots, M$, be the solution of the above problem. The result of solving (20.3) is an ensemble of deterministic solutions $\{u^{(j)}\}_{j=1}^M$. And one can apply various post-processing operations to the ensemble to extract useful information about $u(Z)$.

From this point of view, all classical sampling methods belong to the class of collocation methods. For example, in *Monte Carlo sampling*, the nodal set Θ_M is generated randomly according to the distribution of Z , and the ensemble averages are used to estimate the solution statistics, e.g., mean and variance. In *deterministic sampling methods*, the nodal set is typically the nodes of a cubature rule (i.e., quadrature rule in multidimensional space) defined on I_Z such that one can use the integration rule defined by the cubature to estimate the solution statistics.

In SC, the goal is to construct an accurate approximation to the solution response function using the samples. This is a stronger goal than estimating the solution statistics and the major difference between SC and the classical sampling methods. Knowing the function response of the solution allows us to immediately derive all of the statistical information of the solution. It is not the case conversely, as knowing the solution statistics does not allow us to create the solution response. To this end, SC can be classified as *strong approximation* methods, whereas the traditional sampling methods are *weak approximation* methods. (More precise definitions of strong and weak approximations can be found in [30].)

2 Definition of Stochastic Collocation

The goal of SC is to construct a numerical approximation to the solution response (20.2) in the parameter space I_Z , using the deterministic solution ensemble $\{u(\cdot, Z^{(j)})\}$, $j = 1, \dots, M$, of (20.3). Following [30], we give the following formal definition of SC:

Definition 1 (Stochastic collocation). Let $\Theta_M = \{Z^{(j)}\}_{j=1}^M \subset I_Z$ be a set of (prescribed) nodes in the random space I_Z , where $M \geq 1$ is the number of nodes, and $\{u^{(j)}\}_{j=1}^M$ be the solution of the governing equation (20.3). Then find $w(Z) \approx u(Z)$ such that it is an approximation to the true solution $u(Z)$ in the sense that $\|w(Z) - u(Z)\|$ is sufficiently small in a strong norm defined on I_Z .

In this general definition, the norm is left unspecified. In practice, different choices of the norm lead to different SC methods. Typically, we employ L^p -norm ($p \geq 1$), with the L^2 -norm used the most in practice and leading to “mean-square” approximation.

The numerical approximation $w(Z)$ shall be chosen from a class of functions. Mathematically speaking, this implies that $w \in V$, where V is a linear space from which the approximation is sought. In SC, the most widely used choice

is polynomial space, which leads to strong ties of SC methods to generalized polynomial chaos (gPC) approximation ([16, 32]). Here, the space V is

$$\mathbb{P}_n^d = \text{span}\{z^\alpha : |\alpha| = (\alpha_1 + \dots + \alpha_d) \leq n\}, \quad (20.4)$$

where $\alpha = (\alpha_1, \dots, \alpha_d)$ is multi-index. This is the space of polynomials of degree up to n , whose cardinality is $\dim \mathbb{P}_n^d = \binom{n+d}{n}$. Other spaces of polynomials, or other classes of functions, can certainly be chosen.

The construction and properties of SC methods then critically depend on the approximation properties of w and the choice of the collocation nodal set Θ_M . Broadly speaking, the current SC methods fall into the following three categories: interpolation type, regression type, and pseudo projection type.

3 Stochastic Collocation via Interpolation

3.1 Formulation

In interpolation approach, we seek to match the numerical approximation w with the true solution u exactly at the nodal set Θ_M . More specifically, let $w \in V_N$ be constructed from a linear space V_N with cardinality $\dim V_N = N$. Let (b_1, \dots, b_N) be a basis for V_N . Then, we can express w as

$$w(Z) = \sum_{i=1}^N c_i b_i(Z), \quad (20.5)$$

where c_i are the coefficients to be determined. We then enforce the interpolation condition

$$w(Z^{(j)}) = u(Z^{(j)}), \quad \text{for all } j = 1, \dots, M. \quad (20.6)$$

This immediately leads to a linear system of equations for the unknown coefficients

$$\mathbf{A}\mathbf{c} = \mathbf{f}, \quad (20.7)$$

where

$$\mathbf{A} = (a_{ij})_{1 \leq i \leq M, 1 \leq j \leq N}, \quad a_{ij} = b_j(Z^{(i)}), \quad (20.8)$$

and

$$\mathbf{c} = (c_1, \dots, c_N)^T, \quad \mathbf{f} = (u(Z^{(1)}), \dots, u(Z^{(M)}))^T \quad (20.9)$$

are the coefficient vector and solution sample vector, respectively. For example, if one adopts the gPC expansion, then V_N is the polynomial space \mathbb{P}_n^d from (20.4), and the matrix \mathbf{A} becomes the Vandermonde-like matrix with entries

$$a_{ij} = \Phi_j(Z^{(i)}), \quad (20.10)$$

where $\Phi_j(Z)$ are the gPC orthogonal polynomials chosen based on the probability distribution of Z and satisfy

$$\int_{I_Z} \Phi_i(z) \Phi_j(z) \rho(z) dz = \delta_{ij}. \quad (20.11)$$

Here, δ_{ij} is the Kronecker delta function and the polynomials are normalized.

When the number of the collocation points is the same as the number of the basis functions, i.e., $M = N$, the matrix \mathbf{A} is square and can be inverted when it is nonsingular. One then immediately obtains

$$\mathbf{c} = \mathbf{A}^{-1} \mathbf{f}, \quad (20.12)$$

and can construct the approximation $w(Z)$ using (20.5). Although very flexible, this approach is not used widely in practice. The reason is that interpolation is often not very robust and leads to wild behavior in $w(Z)$. This is especially true in multidimensional spaces ($d > 1$). The accuracy of the interpolation is also difficult to assess and control. Even though the interpolation $w(Z)$ has no error at the nodal points, it can incur large errors between the nodes. Rigorous mathematical analysis is also lacking on this front, particularly in high dimensions. Some of these facts are well documented in texts such as [7, 23, 27].

Another approach to accomplish interpolation is to employ the *Lagrange interpolation approach*. That is, we seek

$$w(Z) = \sum_{j=1}^M u(Z^{(j)}) L_j(Z), \quad (20.13)$$

where

$$L_j(Z^{(i)}) = \delta_{ij}, \quad 1 \leq i, j \leq M, \quad (20.14)$$

are the Lagrange interpolating polynomials. By construction, the polynomial $w(Z)$ automatically satisfies the interpolation conditions. We then need to explicitly construct the Lagrange interpolation polynomials $L_j(Z)$. This can be easily done in one dimension $d = 1$, i.e.,

$$L_j(Z) = \prod_{i=1, i \neq j}^M \frac{Z - Z^{(i)}}{Z^{(j)} - Z^{(i)}}, \quad j = 1, \dots, M. \quad (20.15)$$

Much is known about polynomial interpolation in one dimension. It is widely acknowledged that interpolation on equidistance grids is unstable at higher degree

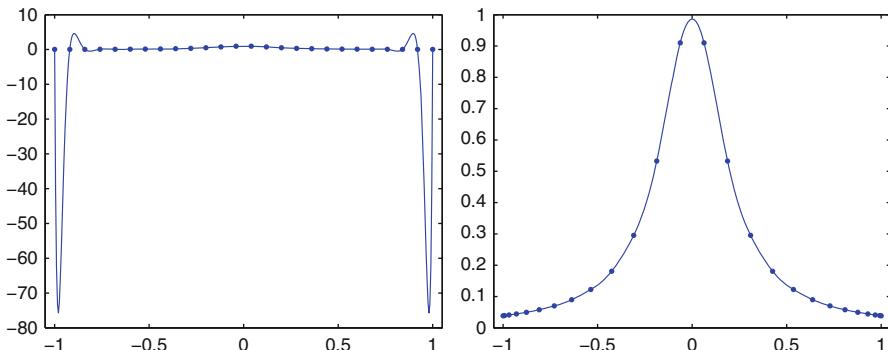


Fig. 20.1 Polynomial interpolation of $f(x) = 1/(1 + 25x^2)$ in $[-1, 1]$, the rescaled Runge function. *Left:* interpolation on uniformly distributed nodes; *Right:* interpolation on nonuniform nodes (the zeros of Chebyshev polynomials)

polynomials. To construct robust and accurate interpolations, one should employ grids that are clustered toward the boundaries of the interval. The well-known example of interpolating the Runge function clearly illustrates this property. The results are shown in Fig. 20.1. Even though both interpolations can faithfully interpolate the function data, the result by the equidistance nodes admits wild oscillations between the nodes, whereas the result by the Chebyshev nodes is well behaved and accurate.

Interpolation in multiple dimensions ($d > 1$) is usually carried out in two different approaches. The first approach is to extend the well-studied one-dimensional interpolation methods to multiple dimensions via a certain tensor product rule. This naturally results in sampling sets that are structured. An immediate consequence is that the growth of the number of samples in high dimensions can be prohibitively fast – courtesy of the “curse of dimensionality.” The second approach is to directly construct interpolations on a set of unstructured nodes. This, however, is a mathematically challenging task and leaves many open issues to study.

3.2 Interpolation SC on Structured Samples

The major difficulty in the interpolation SC is the construction of interpolation polynomials in multiple dimensions. Traditionally, this is carried out by extending the one-dimensional interpolation techniques (20.15) to higher dimensions.

3.2.1 Tensor Nodes

Since univariate interpolation is a well-studied topic, it is straightforward to employ a univariate interpolation and then fill up the multidimensional parameter space dimension by dimension. By doing so, the properties and error estimates of univariate interpolation can be retained as much as possible.

Let

$$\mathcal{Q}_{m_i}[f] = \sum_{j=1}^{m_i} f(Z_i^{(j)}) L_j(Z_i), \quad i = 1, \dots, d, \quad (20.16)$$

be the one-dimensional Lagrange interpolation in the i -th dimension, where L_j are defined in (20.15) and the number of samples is m_i . Let $\Theta_1^{m_i}$ be the interpolation nodal set in this direction. To extend this into the entire d -dimensional space I_Z , we can use tensor product approach and define the multidimensional interpolation operator as

$$\mathcal{Q}_M = \mathcal{Q}_{m_1} \otimes \cdots \otimes \mathcal{Q}_{m_d}, \quad (20.17)$$

and the nodal set is

$$\Theta_M = \Theta_1^{m_1} \times \cdots \times \Theta_1^{m_d}, \quad (20.18)$$

where the total number of nodes is $M = m_1 \times \cdots \times m_d$.

The advantage of this approach is that all the properties of the underlying one-dimensional interpolation scheme can be retained. For example, if one employs the Gauss points as $\Theta_1^{m_i}$, then the interpolation can be highly accurate and robust. The drawback is that the total number of points (20.18) grows too fast in high dimensions. And the desirable properties of the one-dimensional interpolation will be severely offset by this. For example, let us assume one uses the same number of samples in every dimensions, i.e., $m_1 = \cdots = m_d = m$. Then, the total number of points is $M = m^d$. Let us further assume that the one-dimensional interpolation error in each dimension $1 \leq i \leq d$ follows

$$(I - \mathcal{Q}_{m_i})[f] \propto m^{-\alpha},$$

where the constant $\alpha > 0$ depends on the smoothness of the function f . Then, the overall interpolation error also follows the same convergence rate

$$(I - \mathcal{Q}_M)[f] \propto m^{-\alpha}.$$

However, if we measure the convergence in terms of the total number of points, $M = m^d$ in this case, then

$$(I - \mathcal{Q}_M)[f] \propto M^{-\alpha/d}, \quad d \geq 1.$$

For large dimensions $d \gg 1$, the rate of convergence deteriorates drastically and we observe very slow convergence, if there is any, in terms of the total number of collocation points. This is the well known *curse of dimensionality*. For this reason, the tensor product construction is mostly used for low-dimensional problems with d typically less than 5. A detailed theoretical analysis for the tensor interpolation SC for stochastic diffusion equations can be found in [2].

3.2.2 Sparse Grids

An alternative approach is Smolyak sparse grids interpolation. This is based on the original work by Smolyak in [25]. It has been studied extensively in various deterministic settings (cf. the reviews in [3, 4] and the references therein) and was first used in UQ computations in [31]. The Smolyak sparse interpolation also relies on the one-dimensional interpolation (20.16). Instead of taking the full tensor product (20.17), the Smolyak interpolation takes a subset of the full tensor construction in the following manner (cf. [28]),

$$\mathcal{Q}_\ell = \sum_{\ell-d+1 \leq |\mathbf{i}| \leq \ell} (-1)^{\ell-|\mathbf{i}|} \cdot \binom{d-1}{\ell-|\mathbf{i}|} \cdot (\mathcal{Q}_{i_1} \otimes \cdots \otimes \mathcal{Q}_{i_d}), \quad (20.19)$$

where $\ell \geq d$ is an integer denoting the *level* of the construction. Though the expression is rather complex, (20.19) is nevertheless a combination of the subsets of the full tensor construction. The nodal set, the *sparse grids*, is

$$\Theta_M = \bigcup_{\ell-d+1 \leq |\mathbf{i}| \leq \ell} (\Theta_1^{i_1} \times \cdots \times \Theta_1^{i_d}). \quad (20.20)$$

Again it is clear that this is the union of a collection of subsets of the full tensor grids. Unfortunately there is usually no explicit formula to determine the total number of nodes M in terms of d and ℓ .

One popular choice of sparse grids is based on Clenshaw-Curtis nodes, which are the extrema of the Chebyshev polynomials and are defined as, for any $1 \leq i \leq d$,

$$Z_i^{(j)} = -\cos \frac{\pi(j-1)}{m_i^k - 1}, \quad j = 1, \dots, m_i^k, \quad (20.21)$$

where an additional index k is introduced to indicate the *level* of the Clenshaw-Curtis nodes. The number of points doubles with the increasing index $k > 1$, $m_i^k = 2^{k-1} + 1$, where we define $m_i^1 = 1$ and $Z_i^{(1)} = 0$. By doing so, the Clenshaw-Curtis nodes are nested, a property strongly preferred in the Smolyak construction. (For a more detailed discussion on the Clenshaw-Curtis nodes, see [14].) The total number of points satisfies the following estimate

$$M = \#\Theta_k \sim 2^k d^k / k!, \quad d \gg 1. \quad (20.22)$$

It has been shown ([3]) that the interpolation through the Clenshaw-Curtis sparse grid interpolation is exact if the function is in \mathbb{P}_k^d . (In fact, the polynomial space for which the interpolation is exact is slightly bigger than \mathbb{P}_k^d .) For large dimensions $d \gg 1$, $\dim \mathbb{P}_k^d = \binom{d+k}{d} \sim d^k / k!$. Therefore, the number of points from (20.22) is about 2^k more and the factor is *independent* of the dimension d . For this reason, the

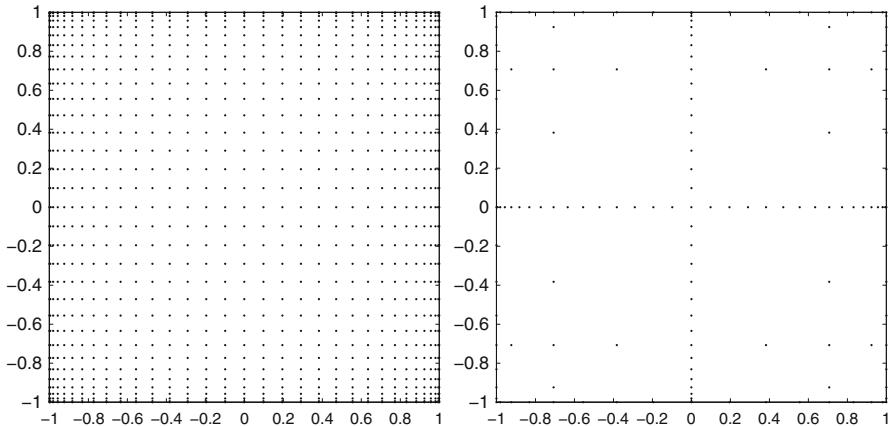


Fig. 20.2 Two-dimensional ($d = 2$) nodes based on the same one-dimensional extrema of Chebyshev polynomials at level $k = 5$. *Left:* Tensor grids. The total number of points is 1,089. *Right:* Smolyak sparse grids. The total number of nodes is 145

Table 20.1 The number of samples of the Smolyak sparse grids using Clenshaw-Curtis nodes, the cardinality of the polynomial space \mathbb{P}_k^d , and the number of samples of the full tensor grids $(k+1)^d$, for various dimensions d and polynomial order k (This is a reproduction of the Table 3.1 from [31])

d	n	Sparse grids	$\dim(\mathbb{P}_n^d)$	Tensor grids
2	1	5	3	4
	2	13	6	9
	3	29	10	16
	4	65	15	25
10	1	21	11	1,024
	2	221	66	59,049
	3	1,581	286	1,048,576
20	1	41	21	1,048,576
	2	841	231	$\approx 3.5 \times 10^9$
50	1	101	51	$\approx 1.1 \times 10^{16}$
	2	5,101	1,326	$\approx 7.2 \times 10^{23}$

Clenshaw-Curtis-based sparse grid construction is sometimes regarded as optimal in high dimensions.

An example of the two-dimensional tensor grids and sparse grids is in Fig. 20.2, where we observe significant reduction of the number of nodes in sparse grids. The reduction becomes more obvious in Table 20.1, where the number of samples of the sparse grids and tensor grids is listed for various dimensions d and polynomial degree k . We clearly see the drastic reduction in the number of samples for sparse grids, compared to that of the tensor grids. We observe the 2^k factor between the number of samples in sparse grids and the cardinality of the polynomial space \mathbb{P}_k^d in high dimensions $d \gg 1$.

Even though the sparse grids enjoy great reduction in the number of sample points, one should be aware that the total number of points can still be exceedingly large at high dimensions. The estimate for Clenshaw-Curtis (20.22) is almost the

best, as all other types of sparse grid constructions have (much) larger number of points. To this end, sparse grid interpolation has been used for moderately high dimensions, say, $d \sim O(10)$.

3.3 Interpolation on Unstructured Samples

From a practical point of view, it is highly desirable to conduct multidimensional interpolation on an arbitrary set of nodes Θ_M . To this end, however, it becomes much less clear what the best approach shall be. Arguably the only existing approach is least interpolation. This was first developed in [10, 11] and later extended to general orthogonal polynomials in [21]. The mathematical theory of this approach is rather involved and technical, although its implementation is straightforward using only numerical linear algebra. The major advantage of this approach is that one can then conduct SC interpolation on nested samples and, more importantly, samples that are arbitrarily distributed. This is especially useful from a practical point of view, for in many applications the samples are collected by practitioners at locations not following certain mathematical theory. Not to mention that in many cases there are restrictions on where one can or cannot collect the samples. Robust interpolation using this approach can be achieved by carefully designed sample sets [22]. An example of its effectiveness can be seen from simple interpolations of the following one-dimensional functions with increasing smoothness:

$$f_0(x) = \begin{cases} -1, & x < \frac{1}{2}, \\ 1, & x \geq \frac{1}{2}, \end{cases} \quad f_s(x) = \int_{-1}^x f_{s-1}(t) dx, \quad s = 1, 2, 3. \quad (20.23)$$

The results are shown in Fig. 20.3. We observe almost the same convergence rate and errors in both the least interpolation and the standard interpolation using Gauss-Legendre nodes. The fundamental difference between the two methods is that the

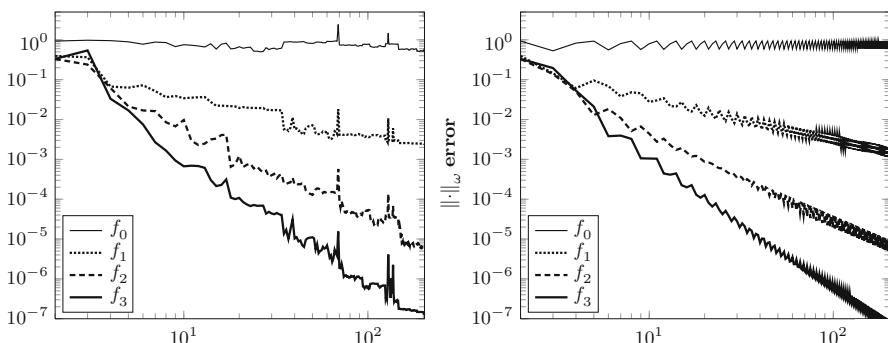


Fig. 20.3 Interpolation accuracy for functions f_0 , f_1 , f_2 , and f_3 in (20.23). Left: errors by least interpolation; Right: errors by interpolation on Legendre-Gauss nodes (Reproduction of Fig. 5.2 in [22])

least interpolation is conducted on completely nested sample nodes, allowing one to progressively add sample points for improved accuracy. On the other hand, the standard Gauss node interpolation is not nested – increasing the accuracy of the interpolation implies sampling at a new set of points.

Again, the theory of the least interpolation is quite involved. We refer the interested readers to [10, 11, 21, 22] for more details.

4 Stochastic Collocation via Regression

In regression type SC, one does not require the approximation $w(Z)$ to precisely match the solution $u(Z)$ at the collocation nodes Θ_M . Instead, one resorts to minimize the error difference

$$\|w(Z) - u(Z)\|_{\Theta}, \quad (20.24)$$

where the norm $\|\cdot\|_{\Theta}$ is a discrete norm defined over the nodal set Θ . By doing so, the numerical errors are more evenly distributed in the entire parameter space I_Z , assuming that the set Θ fills up the space I_Z in a reasonable manner. Thus, the non-robustness of interpolation can be alleviated. The regression approach also becomes a natural choice when the solution samples $u(Z^{(j)})$ are of low accuracy or contaminated by noise, in which case interpolation of the solution samples becomes an unnecessarily strong requirement.

4.1 Over-sampled Case: Least Squares

When the number of samples is larger than the cardinality of the linear space V_N , we have an over-determined system of equations (20.7) with $M > N$. Consequently, the equality cannot hold true in general. The natural approach is to use the least squares method. By doing so, the norm in (20.24) is the vector 2-norm, and we have the well-known least squares solution:

$$\mathbf{c} = \mathbf{A}^{\dagger} \mathbf{f} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{f}, \quad (20.25)$$

where \mathbf{A}^{\dagger} denotes the pseudo-inverse of \mathbf{A} .

The least squares method is an orthogonal projection onto the range of the matrix \mathbf{A} . Consequently, it is the optimal approximation (in vector 2-norm) over the Hilbert space defined by the vector inner product. Over-sampling is the key to the accuracy and efficiency of the least squares method. A rule of thumb for practical problems is that one should over-sample the system by a linear factor, i.e., $M \approx \alpha N$, where $\alpha \sim O(1)$ and is often chosen to be $\alpha = 1.5 \sim 3$. There are a variety of choices for the nodal set Θ_M . The most commonly used sets include Monte Carlo points (random sampling), quasi-Monte Carlo points, etc. It is also

worthwhile to strategically choose the points to achieve better accuracy. This is the topic of *experimental design*. Interested readers can refer to, for example, [1, 15], and the references therein.

Despite the large amount of literature on least squares methods, a series of more recent studies from the computational mathematics perspective show that the linear over-sampling of Monte Carlo and quasi-Monte Carlo points for polynomial approximation can be asymptotically unstable (cf. [8, 18–20]). Care must be taken if one intends to conduct very high-order polynomial approximations using the least squares method.

4.2 Under-sampled Case: Sparse Approximations

When the number of samples is smaller than the cardinality of the linear space V_N , we have an under-determined system of equation (20.7) with $M < N$. Equation (20.7) then admits an infinite number of solutions. In this case, one can resort to the idea of compressive sensing (CS) and seek a sparse solution:

$$\min \|\mathbf{c}\|_0 \quad \text{subject to } \mathbf{Ac} = \mathbf{f}, \quad (20.26)$$

where the $\|\mathbf{c}\|_0 = \{\#c_i : c_i \neq 0, i = 1, \dots, N\}$ is the number of nonzero entries in the vector \mathbf{c} . The solution of this constrained optimization problem leads to a sparse solution, in the sense that the number of nonzero entries in the solution is minimized. Unfortunately, this optimization is an NP-hard problem and cannot be easily solved. As a compromise, the ℓ_1 norm is often used, leading to the well-known compressive sensing formulation (cf. [5, 6, 12]):

$$\min \|\mathbf{c}\|_1 \quad \text{subject to } \mathbf{Ac} = \mathbf{f}, \quad (20.27)$$

where $\|\mathbf{c}\|_1 = |c_1| + \dots + |c_N|$. The use of the ℓ_1 norm also promotes sparsity. But the optimization problem can be cast into a linear programming and solved easily. The constraint $\mathbf{Ac} = \mathbf{f}$ effectively enforces interpolation. This does not need to be the case, especially when the samples \mathbf{f} contain errors or noises. In this case, the de-noising version of CS [5, 6, 12] can be used.

$$\min \|\mathbf{c}\|_1 \quad \text{subject to } \|\mathbf{Ac} - \mathbf{f}\| \leq \tau, \quad (20.28)$$

where $\tau > 0$ is a real number associated with the noise level in the sample data \mathbf{f} .

The idea of CS was first used in UQ computations in [13], where the gPC-type orthogonal approximations using Legendre polynomials were used. The advantage of this approach lies in the fact that it allows one to construct reliable sparse gPC approximations when the underlying system is (severely) under sampled. Most of the existing studies focus on the use of Legendre polynomials [17, 24, 33].

For example, it was proved in [24] that the Chebyshev-weighted ℓ_1 minimization algorithm using Chebyshev samples has notably higher rate of recovery and should be preferred in practice. That is, instead of (20.27), one solves

$$\min \|\mathbf{c}\|_1 \quad \text{subject to } \mathbf{WAc} = \mathbf{Wf}, \quad (20.29)$$

where \mathbf{W} is a diagonal matrix with entries $w_{j,j} = (\pi/2)^{d/2} \prod_{i=1}^d (1 - (z_i^{(j)})^2)^{1/4}$, $j = 1, \dots, M$. Note that this is the tensor product of the one-dimensional Chebyshev weights. The corresponding de-noising version for (20.28) takes a similar form. On the other hand, it was also proved that in high dimensions $d \gg 1$, the standard non-weighted version (20.27) using uniformly distributed random samples is in fact better than the weighted Chebyshev version (20.29).

The performance of the ℓ_1 minimization methods is typically measured by the recovery probability of the underlying sparse functions. For high rate of recovery, the required number of samples typically follows:

$$M \propto s \log^3(s) \log(N),$$

where s is the sparsity of the underlying function, i.e., s is the number of nonzero terms in the underlying function. A representative result can be seen in Fig. 20.4, where the ℓ_1 minimization is used to recover a $d = 3$ dimensional polynomial

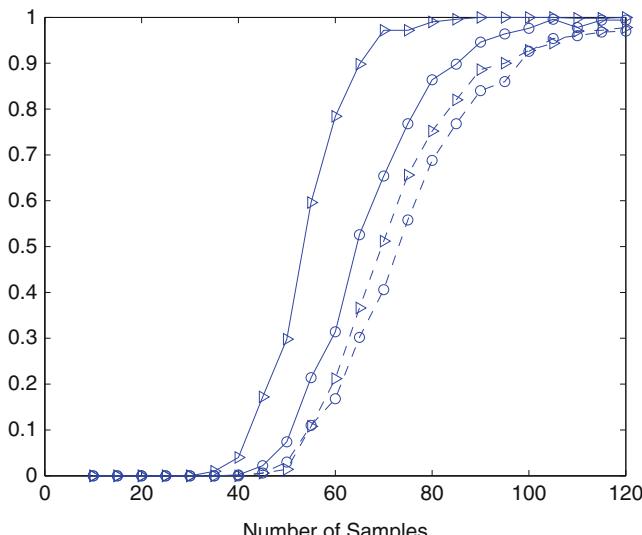


Fig. 20.4 Probability of successful function recovery vs. number of samples. ($d = 3$ and $s = 10$). The degree of polynomial space is $k = 10$ with $\dim \mathbb{P}_k^d = 286$. Line pattern: *dotted-circle*, uniform samples; *dotted-triangle*: Chebyshev samples; *solid-circle*: weighted uniform samples; *solid-triangle*: weighted Chebyshev samples (Reproduction of Fig. 1 in [33])

function with sparsity $s = 10$. Although the different implementations result in variations in the results, we can still see that the methods can recover, with very high probability, the underlying function with $M \sim 100$ samples. This is notably lower than the cardinality of the polynomial space $\dim \mathbb{P}_k^d = 286$. This clearly demonstrates the effectiveness of CS methods. We shall remark that, despite these few works mentioned here, the application of CS in UQ is still in its early stage. There exist many open issues to study.

5 Stochastic Collocation via Pseudo Projection

In the pseudo projection approach (first formally defined in [29]), one seeks to approximate the continuous orthogonal projection using an integration rule. Since the orthogonal projection is the “best approximation,” based on a properly chosen norm, the pseudo projection method allows one to obtain a “near best” approximation whenever the chosen integration is sufficiently accurate. Again, let V_N be a properly chosen linear space, from which an approximation is sought. Then, the orthogonal projection of the solution is

$$u_N := \mathcal{P}_{V_N} u, \quad (20.30)$$

where \mathcal{P} denotes the orthogonal projection operator. The projection operator is often defined via integrals. In the pseudo projection approach, one then approximates the integrals using a quadrature/cubature rule.

To better illustrate the method, let us again use the gPC-based approximation. In this case, $V_N = \mathbb{P}_n^d$, and we seek an approximation

$$w(Z) = \sum_{i=1}^N c_i \Phi_i(Z), \quad N = \dim \mathbb{P}_n^d = \binom{n+d}{n}. \quad (20.31)$$

The orthogonal projection, which is the best approximation in L_ρ^2 norm, takes the following form,

$$u_N(Z) := \sum_{i=1}^N \hat{u}_i \Phi_i(Z), \quad \hat{u}_i = \int u(z) \Phi_i(z) \rho(z) dz. \quad (20.32)$$

In the pseudo projection approach, we then use a cubature rule to approximate the coefficients \hat{u}_i . That is, in (20.31), we seek

$$c_i = \sum_{j=1}^M u(Z^{(j)}) \Phi_i(Z^{(j)}) w_j \approx \hat{u}_i, \quad i = 1, \dots, N. \quad (20.33)$$

This means the collocation nodal set Θ needs to be a quadrature such that integrals over the I_Z can be approximated by a weighted sum. That is,

$$\int_{I_Z} f(z)\rho(z)dz \approx \sum_{j=1}^M f(Z^{(j)})w_j, \quad (20.34)$$

where w_j are the weights.

The pseudo projection method turns out to be remarkably robust and accurate, provided one finds an efficient and accurate quadrature rule. Unlike the other approaches for SC, e.g., interpolation SC and regression SC, where the goal is to approximate the underlying multidimensional function u directly, in the pseudo projection approach, the challenge becomes the approximation of multidimensional integrals. The nodal set Θ should now be a good cubature rule. Depending on the problem at hand and the accuracy requirement, one can choose different cubature sets. The field of multidimensional integration is by itself a big and evolving field, with a large amount of literature. Interested readers should consult the literature on cubature rules (cf. [9, 26]).

It should also be remarked that an “easier” way to construct cubature rules is to extend the one-dimensional quadrature rule to multiple dimensions via tensor products. A construction is very much similar to the tensor product multidimensional interpolation SC, as discussed in Sect. 3.2. Quadrature rules in one dimension are well studied and understood. It is widely accepted that Gauss quadrature, particularly Chebyshev quadrature, is near optimal. One can then extend it into multiple dimensions either via full tensor products or the Smolyak construction (which is a subset of full tensor products). Upon doing so, one obtains tensor cubature or sparse grid cubature, respectively. An example of this is seen in Fig. 20.2.

When pseudo projection is used, one should pay attention to the accuracy of the cubature rule. This is because the integrals to be approximated in (20.32) become progressively more complex, when a higher degree gPC expansion is used. As a general rule of thumb, one should employ a cubature that is accurate with order at least $2n$, where n is the degree of the gPC expansion. This ensures that if the underlying unknown function is an n -degree polynomial (which is almost never the case), then the n -degree gPC expansion can be accurately constructed by the pseudo projection method. When the cubature rule is of low accuracy, then a higher order gPC expansion is pointless. This can be clearly seen in the example in Fig. 20.5. This is an example of approximating a three-dimensional ($d = 3$) nonlinear function, the example 5.1 from [29]. Here, the full tensor product cubature rule based on Gauss-Legendre quadrature is used. The number of Gauss quadrature points in each dimension is q_1 . We observe that when q_1 is small, the gPC approximation error deteriorates at higher order and the expansion fails to converge. It is only when the cubature is of sufficiently high accuracy, $q_1 = 6$ in this case, that the gPC approximation exhibits the expected exponential error convergence for up to order $n = 6$.

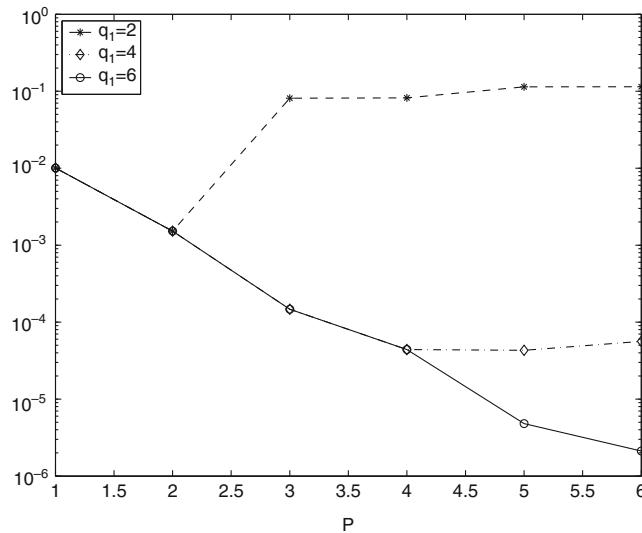


Fig. 20.5 Error convergence of pseudo projection method with different choices of cubature rule. Here the cubature rule is the full tensor quadrature rule with q_1 number of points in each dimension

6 Summary

Here, we briefly reviewed several major numerical approaches for stochastic collocation (SC). Our focus is on the fundamental approximation properties, basic formulations, and major properties of the methods. We reviewed the interpolation type SC, the regression type SC, and the pseudo projection type SC. Most, if not all, of the mainstream SC methods fall into these categories. There exist, however, a large variety of modifications and improvements over the core methods reviewed here. For example, many efforts have been devoted to the development of adaptive SC methods, particularly in conjunction of the sparse grid collocation. There also exist a large amount of works on the improvements of least squares methods. On the other hand, for under-sampled systems, the use of compressive sensing SC is still in its early stage, with many open issues to study. Overall, stochastic collocation has been under rapid development in the last decade, with many new variations emerging. Interested readers should consult the more recent literature for the latest developments.

References

1. Atkinson, A.C., Donev, A.N., Tobias, R.D.: Optimum Experimental Designs, with SAS. Oxford University Press, Oxford (2007)
2. Babuska, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM J. Numer. Anal. **45**(3), 1005–1034 (2007)

3. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grid. *Adv. Comput. Math.* **12**, 273–288 (1999)
4. Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numer.* **13**, 1–123 (2004)
5. Candès, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.* **59**(8), 1207–1223 (2006)
6. Candès, E.J., Tao, T.: Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inform. Theory* **52**(12), 5406–5425 (2006)
7. Cheney, W., Light, W.: *A Course in Approximation Theory*. Brooks/Cole Publishing Company, Pacific Grove (2000)
8. Cohen, A., Davenport, M.A., Leviatan, D.: On the stability and accuracy of least squares approximations. *Found. Comput. Math.* **13**(5), 819–834 (2013)
9. Cools, R.: Advances in multidimensional integration. *J. Comput. Appl. Math.* **149**, 1–12 (2002)
10. De Boor, C., Ron, A.: On multivariate polynomial interpolation. *Constr. Approx.* **6**, 287–302 (1990)
11. De Boor, C., Ron, A.: Computational aspects of polynomial interpolation in several variables. *Math. Comput.* **58**, 705–727 (1992)
12. Donoho, D.L.: Compressed sensing. *IEEE Trans. Inform. Theory* **52**(4), 1289–1306 (2006)
13. Doostan, A., Owhadi, H.: A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.* **230**(8), 3015–3034 (2011)
14. Engels, H.: *Numerical Quadrature and Cubature*. Academic, London/New York (1980)
15. Fedorov, V.V., Leonov, S.L.: *Optimal Design for Nonlinear Response Models*. CRC Press, Boca Raton (2014)
16. Ghanem, R.G., Spanos, P.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
17. Mathelin, L., Gallivan K.A.: A compressed sensing approach for partial differential equations with random input data. *Commun. Comput. Phys.* **12**, 919–954 (2012)
18. Migliorati, G., Nobile, F.: Analysis of discrete least squares on multivariate polynomial spaces with evaluations at low-discrepancy point sets. *J. Complex.* **31**(4), 517–542 (2015)
19. Migliorati, G., Nobile, F., E. von Schwerin, Tempone, R.: Approximation of quantities of interest in stochastic PDEs by the random discrete L^2 projection on polynomial spaces. *SIAM J. Sci. Comput.* **35**(3), A1440–A1460 (2013)
20. Migliorati, G., Nobile, F., von Schwerin, E., Tempone, R.: Analysis of the discrete L^2 projection on polynomial spaces with random evaluations. *Found. Comput. Math.* **14**(3), 419–456 (2014)
21. Narayan, A., Xiu, D.: Stochastic collocation methods on unstructured grids in high dimensions via interpolation. *SIAM J. Sci. Comput.* **34**(3), A1729–A1752 (2012)
22. Narayan, A., Xiu, D.: Constructing nested nodal sets for multivariate polynomial interpolation. *SIAM J. Sci. Comput.* **35**(5), A2293–A2315 (2013)
23. Powell, M.J.D.: *Approximation Theory and Methods*. Cambridge University Press, Cambridge (1981)
24. Rauhut, H., Ward, R.: Sparse Legendre expansions via ℓ_1 -minimization. *J. Approx. Theory* **164**, 517–533 (2012)
25. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.* **4**, 240–243 (1963)
26. Stroud, A.H.: *Approximate Calculation of Multiple Integrals*. Prentic-Hall, Englewood Cliffs (1971)
27. Trefethen, L.N.: *Approximation Theory and Approximation Practice*. SIAM, Philadelphia (2013)
28. Wasilkowski, G.W., Woźniakowski, H.: Explicit cost bounds of algorithms for multivariate tensor product problems. *J. Complex.* **11**, 1–56 (1995)
29. Xiu, D.: Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys.* **2**(2), 293–309 (2007)
30. Xiu, D.: *Numerical Methods for Stochastic Computations*. Princeton University Press, Princeton (2010)

-
31. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
 32. Xiu, D., Karniadakis, G.E.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
 33. Yan, L., Guo, L., Xiu, D.: Stochastic collocation algorithms using ℓ_1 -minimization. *Int. J. UQ* **2**(3), 279–293 (2012)

Sparse Collocation Methods for Stochastic Interpolation and Quadrature

21

Max Gunzburger, Clayton G. Webster, and Guannan Zhang

Abstract

In this chapter, the authors survey the family of sparse stochastic collocation methods (SCMs) for partial differential equations with random input data. The SCMs under consideration can be viewed as a special case of the generalized stochastic finite element method (Gunzburger et al., *Acta Numer.* 23:521–650, 2014), where the approximation of the solution dependences on the random variables is constructed using Lagrange polynomial interpolation. Relying on the “delta property” of the interpolation scheme, the physical and stochastic degrees of freedom can be decoupled, such that the SCMs have the same nonintrusive property as stochastic sampling methods but feature much faster convergence. To define the interpolation schemes or interpolatory quadrature rules, several approaches have been developed, including global sparse polynomial approximation, for which global polynomial subspaces (e.g., sparse Smolyak spaces (Nobile et al., *SIAM J Numer Anal.* 46:2309–2345, 2008) or quasi-optimal subspaces (Tran et al., *Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients. Tech. Rep. ORNL/TM-2015/341*, Oak Ridge National Laboratory, 2015)) are used to exploit the inherent regularity of the PDE solution, and local sparse approximation, for which hierarchical polynomial bases (Ma and Zabaras, *J Comput Phys.* 228:3084–3113, 2009; Bungartz and Griebel, *Acta Numer.* 13:1–123, 2004) or wavelet bases (Gunzburger et al., *Lect Notes Comput Sci Eng.* 97:137–170, Springer, 2014) are used to accurately capture irregular behaviors of the PDE solution. All these method classes are surveyed in this chapter, including

M. Gunzburger (✉)

Department of Scientific Computing, The Florida State University, Tallahassee, FL, USA
e-mail: gunzburg@fsu.edu

C.G. Webster • G. Zhang

Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN, USA
e-mail: webstercg@ornl.gov; zhangg@ornl.gov

some novel recent developments. Details about the construction of the various algorithms and about theoretical error estimates of the algorithms are provided.

Keywords

Uncertainty quantification • Stochastic partial differential equations • High-dimensional approximation • Stochastic collocation • Sparse grids • Hierarchical basis • Best approximation • Local adaptivity

Contents

1	Introduction	718
2	Problem Setting	720
3	Stochastic Finite Element Method	723
3.1	Spatial Finite Element Semi-discretization	724
3.2	Stochastic Fully Discrete Approximation	724
3.3	Stochastic Polynomial Subspaces	725
4	Global Stochastic Collocation Methods	731
4.1	Lagrange Global Polynomial Interpolation in Parameter Space	731
4.2	Generalized Sparse Grid Construction	732
4.3	Nonintrusive Sparse Interpolation in Quasi-optimal Subspaces	744
5	Local Stochastic Collocation Methods	746
5.1	Hierarchical Stochastic Collocation Methods	746
5.2	Adaptive Hierarchical Stochastic Collocation Methods	753
6	Conclusion	758
	References	759

1 Introduction

Many applications in engineering and science are affected by uncertainty in input data, including model coefficients, forcing terms, boundary condition data, media properties, source and interaction terms, as well as geometry. The presence of random input uncertainties can be incorporated into a system of partial differential equations (PDEs) by formulating the governing equations as PDEs with random inputs. In practice, such PDEs may depend on a set of distinct random parameters with the uncertainties represented by a given joint probability distribution. In other situations, the input data varies randomly from one point of the physical domain to another and/or from one time instant to another; in these cases, uncertainties in the inputs are instead described in terms of *random fields* that can be expressed as an expansion containing an infinite number of random variables. For example, for correlated random fields, one has Karhunen-Loëve (KL) expansions [49, 50], Fourier-Karhunen-Loëve expansions [48], or expansions in terms of global orthogonal polynomials [35, 70, 72]. However, in a large number of applications, it is reasonable to limit the analysis to just a finite number of random variables, either because the problem input itself can be described in that way (e.g., the random parameter case) or because the input random field can be approximated

by truncating an infinite expansion [30] (e.g., the correlated random field case). As such, a crucial, yet often complicated, ingredient that all numerical approaches to UQ must incorporate is an accurate and efficient numerical approximation technique for solving PDEs with random inputs. In some circles, although the nomenclature “stochastic partial differential equations (SPDEs)” is reserved for a specific class of PDEs having random inputs, here, for the sake of economy of notation, this terminology is used to refer to any PDE having random inputs.

Currently, there are several types of numerical methods available for solving SPDEs. Monte Carlo methods (MCMs) [29], quasi MCMs [46, 47], multilevel MCMs [4, 20, 36] stochastic Galerkin methods (SGMs) [2, 3, 35, 55], and stochastic collocation methods (SCMs) [1, 51, 57, 58, 71, 73]. In this chapter, the authors provide an overview of the family of stochastic collocation methods for SPDEs. Recently, SCMs based on either full or sparse tensor-product approximation spaces [1, 32, 54, 57, 58, 62, 71] have gained considerable attention. As shown in [1], SCMs can essentially match the fast convergence of intrusive polynomial chaos methods, even coinciding with them in particular cases. The major difference between the two approaches is that stochastic collocation methods are ensemble-based, *nonintrusive* approaches that achieve fast convergence rates by exploiting the inherent regularity of PDE solutions with respect to parameters. Compared to nonintrusive polynomial chaos methods, they also require fewer assumptions about the underlying SPDE. SCMs can also be viewed as stochastic Galerkin methods in which one employs an interpolatory basis built from the zeros of orthogonal polynomials with respect to the joint probability density function of the input random variables. For additional details about the relations between polynomial chaos methods and stochastic collocation methods, see [41], and for computational comparisons between the two approaches, see [5, 25, 28, 41].

To achieve increased rates of convergence, most SCMs described above are based on global polynomial approximations that take advantage of smooth behavior of the solution in the multidimensional parameter space. Hence, when there are steep gradients, sharp transitions, bifurcations, or finite discontinuities (e.g., piecewise processes) in stochastic space, these methods converge very slowly or even fail to converge. Such problems often arise in scientific and engineering problems due to the highly complex nature of most physical or biological phenomena. To be effective, refinement strategies must be guided by accurate estimations of errors while not expending significant computational effort approximating an output of interest within each random dimension. The papers [27, 45, 51, 52, 74] apply an adaptive sparse grid stochastic collocation strategy that follows the work of [14, 37]. This approach utilizes the hierarchical surplus as an error indicator to automatically detect regions of importance (e.g., discontinuities) in the stochastic parameter space and adaptively refine the collocation points in this region. To this end, grids are constructed in an adaptation process steered by the indicator in such a way that a prescribed global error tolerance is attained. This goal, however, might be achieved using more points than necessary due to the instability of this multi-scale basis.

The outline of this chapter is as follows. In Sect. 2, a generalized mathematical description of SPDEs is provided, together with the notations that are used

throughout. In Sect. 3, the general framework of stochastic finite element methods is discussed, followed by the notions of semi-discrete and fully discrete stochastic approximation, as well as several choices of multivariate polynomial subspaces. In Sect. 4, all discussions are based on the fact that the solution of the SPDE has very smooth dependence on the input random variables. Thus, SCMs approximate solutions using global approximations in parameter space. A generalized sparse grid interpolatory approximation is presented, followed by a detailed convergence analysis with respect to the total number of collocation points. In Sect. 5, it is assumed that the solution of the SPDE may have irregular dependence on the input random variables, as a result of which the global approximations are usually not appropriate. As an alternative, SCMs approximate the solutions using locally supported piecewise polynomial spaces for both spatial and stochastic discretization. We then extend this concept to include adaptive hierarchical stochastic collocation methods. Two comments about the content of this article are important to point out. First, the temporal dependence of solutions of SPDEs is ignored, i.e., it is assumed that coefficients, forcing functions, etc., and therefore solutions, only depend on spatial variables and random parameters. This is merely for economizing notation. Almost all discussions extend to problems that also involve temporal dependences. Second, only finite element methods are considered for effecting the spatial discretization of SPDEs, but most of the discussions also apply to finite difference, finite volume, and spectral methods for spatial discretization.

2 Problem Setting

This chapter considers a relevant model of boundary value problems, involving the simultaneous solution of a family of equations, parameterized by a vector $\mathbf{y} = (y_1, \dots, y_N) \in \Gamma = \prod_{i=1}^N \Gamma_i \subset \mathbb{R}^N$, on a bounded Lipschitz domain $D \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$. In particular, let \mathcal{L} denote a differential operator defined on D , and let $a(\mathbf{x}, \mathbf{y})$, with $\mathbf{x} \in D$ and $\mathbf{y} \in \Gamma$, represent the input coefficient associated with the operator \mathcal{L} . The forcing term $f = f(\mathbf{x}, \mathbf{y})$ is also assumed to be a parameterized field in $D \times \Gamma$. This chapter concentrates on the following parameterized boundary value problem: for all $\mathbf{y} \in \Gamma$, find $u(\cdot, \mathbf{y}) : \overline{D} \rightarrow \mathbb{R}$, such that the following equation holds

$$\mathcal{L}(a(\cdot, \mathbf{y})) [u(\cdot, \mathbf{y})] = f(\cdot, \mathbf{y}) \quad \text{in } D, \quad (21.1)$$

subject to suitable (possibly parameterized) boundary conditions. Such a problem arises in both contexts of deterministic and stochastic modeling. In the first case, the parameter vector \mathbf{y} is known or controlled by the user, and a typical goal is to study the dependence of u with respect to these parameters, e.g., optimizing an output of the equation with respect to \mathbf{y} (see [13, 56] for more details). In the second case, the parameters $\{y_n\}_{n=1}^N$ are random variables, and $\mathbf{y}(\omega) : \Omega \rightarrow \Gamma$ is a random vector. Here, $(\Omega, \mathcal{F}, \mathbb{P})$ is a complete probability space, Ω being the set of outcomes, $\mathcal{F} \subset 2^\Omega$ the σ -algebra of events, and $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ a probability measure. Moreover, the

components of $\mathbf{y}(\omega)$ have a joint probability density function (PDF) $\rho : \Gamma \rightarrow \mathbb{R}_+$, with $\rho \in L^\infty(\Gamma)$. In this setting, $(\Omega, \mathcal{F}, \mathbb{P})$ is mapped to $(\Gamma, \mathcal{B}(\Gamma), \rho(\mathbf{y})d\mathbf{y})$, where $\mathcal{B}(\Gamma)$ denotes the Borel σ -algebra on Γ and $\rho(\mathbf{y})d\mathbf{y}$ is the probability measure of \mathbf{y} .

Let $W(D)$ denote a Banach space of functions $v : D \rightarrow \mathbb{R}$ and define the following stochastic Banach spaces

$$L_\rho^q(\Gamma; W(D)) := \left\{ v : \Gamma \rightarrow W(D) \mid v \text{ is strongly measurable and} \right. \\ \left. \int_{\Gamma} \|v(\cdot, \mathbf{y})\|_{W(D)}^q \rho(\mathbf{y}) d\mathbf{y} < +\infty \right\}. \quad (21.2)$$

To guarantee the well posedness of the system (21.1) in a Banach space, the following assumptions are needed:

Assumption 1. (a) The solution to (21.1) has realizations in the Banach space $W(D)$, i.e., $u(\cdot, \mathbf{y}) \in W(D)$ ρ -almost surely satisfies that

$$\|u(\cdot, \mathbf{y})\|_{W(D)} \leq C \|f(\cdot, \mathbf{y})\|_{W^*(D)},$$

where $W^*(D)$ denotes the dual space of $W(D)$ and C denotes a constant having value independent of the realization $\mathbf{y} \in \Gamma$.

(b) The forcing term $f \in L_\rho^2(\Gamma; W^*(D))$ is such that the solution u is unique and bounded in $L_\rho^2(\Gamma; W(D))$.

Two examples of problems posed in this setting are as follows.

Example 1. The linear second-order elliptic problem

$$\begin{cases} -\nabla \cdot (a(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y})) = f(\mathbf{x}, \mathbf{y}) & \text{in } D \times \Gamma \\ u(\mathbf{x}, \mathbf{y}) = 0 & \text{on } \partial D \times \Gamma \end{cases} \quad (21.3)$$

with $a(\mathbf{x}, \mathbf{y})$ uniformly bounded from above and below, i.e.,

there exist $a_{\min}, a_{\max} \in (0, \infty)$ such that

$$\mathbb{P}\left(\mathbf{y} \in \Gamma : a(\mathbf{x}, \mathbf{y}) \in [a_{\min}, a_{\max}] \quad \forall \mathbf{x} \in \overline{D}\right) = 1$$

and $f(\mathbf{x}, \mathbf{y})$ square integrable with respect to $\rho(\mathbf{y})d\mathbf{y}$, i.e.,

$$\int_D \mathbb{E}[f^2] d\mathbf{x} = \int_D \int_{\Gamma} f^2(\mathbf{x}, \mathbf{y}) \rho(\mathbf{y}) d\mathbf{y} d\mathbf{x} < \infty,$$

such that Assumptions 1(a-b) are satisfied with $W(D) = H_0^1(D)$; see [1].

Example 2. Similarly, for $s \in \mathbb{N}_+$, the *nonlinear* second-order elliptic problem

$$\begin{cases} -\nabla \cdot (a(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y})) + u(\mathbf{x}, \mathbf{y})|u(\mathbf{x}, \mathbf{y})|^s = f(\mathbf{x}, \mathbf{y}) & \text{in } D \times \Gamma \\ u(\mathbf{x}, \mathbf{y}) = 0 & \text{on } \partial D \times \Gamma \end{cases} \quad (21.4)$$

with $a(\mathbf{x}, \mathbf{y})$ uniformly bounded from above and below and $f(\mathbf{x}, \mathbf{y})$ square integrable with respect to the probability measure such that Assumptions 1(a–b) are satisfied with $W(D) = H_0^1(D) \cap L^{s+2}(D)$; see [69].

In many applications, the stochastic input data may have a simple piecewise random representation, whereas, in other applications, the coefficients a and the right-hand side f in (21.1) may have spatial variation that can be modeled as a correlated random field, making them amenable to description by a Karhunen-Loëve (KL) expansion [37, 38]. In practice, one has to truncate such expansions according to the degree of correlation and the desired accuracy of the simulation. Examples of both types of random input data are given next.

Example 3 (Piecewise constant random fields). Assume that the spatial domain D is the union of non-overlapping subdomains D_n , $n = 1, \dots, N$. Then, consider a coefficient $a(\mathbf{x}, \mathbf{y})$ that is a random constant in each subdomain D_n , i.e., $a(\mathbf{x}, \mathbf{y})$ is the piecewise constant function

$$a(\mathbf{x}, \mathbf{y}) := a_0 + \sum_{n=1}^N a_n \mathbf{1}_{D_n}(\mathbf{x}),$$

where a_n , $n = 0, \dots, N$, denote constants, $\mathbf{1}_{D_n}(\mathbf{x})$ denotes the indicator function of the set $D_n \subset D$, and the random variables $y_n(\omega)$, $n = 1, \dots, N$ are bounded and independent.

Example 4 (Karhunen-Loëve expansions). According to Mercer's theorem [49], any second-order correlated random field with a continuous covariance function can be represented as an infinite sum of random variables. A commonly used example is the Karhunen-Loëve expansion. In this case, $a(\mathbf{x}, \mathbf{y})$ in (21.1) can be defined as a truncated Karhunen-Loëve expansion having the form

$$a(\mathbf{x}, \mathbf{y}) := \bar{a}(\mathbf{x}) + \sum_{n=1}^N \sqrt{\lambda_n} b_n(\mathbf{x}) y_n(\omega),$$

where λ_n and $b_n(\mathbf{x})$ for $n = 1, \dots, N$ are the dominant eigenvalues and corresponding eigenfunctions for the covariance function and $y_n(\omega)$ for $n = 1, \dots, N$ denote uncorrelated real-valued random variables. In addition, for the purpose of keeping

the property that a is bounded away from function zero, the logarithm of the random field is instead expanded so that $a(\mathbf{x}, \mathbf{y})$ has the form

$$a(\mathbf{x}, \mathbf{y}) := a_{\min} + \exp \left\{ \bar{a}(\mathbf{x}) + \sum_{n=1}^N \sqrt{\lambda_n} b_n(\mathbf{x}) y_n(\omega) \right\}, \quad (21.5)$$

where $a_{\min} > 0$ is the lower bound of a .

3 Stochastic Finite Element Method

Since the solution u of the SPDE in (21.1) can be expressed as $u(\mathbf{x}, y_1, \dots, y_N)$, it is natural to treat $u(\mathbf{x}, \mathbf{y})$, a function of d spatial variables and N random parameters, as a function of $d + N$ variables. This leads one to consider a *Galerkin weak formulation* of the SPDE, with respect to both physical and parameter space, of the form: seek $u \in W(D) \otimes L_\rho^q(\Gamma)$ such that

$$\begin{aligned} & \sum_{k=1}^K \int_{\Gamma} \int_D S_k(u; \mathbf{y}) T_k(v) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= \int_{\Gamma} \int_D v f(\mathbf{x}, \mathbf{y}) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \quad \forall v \in W(D) \otimes L_\rho^q(\Gamma), \end{aligned} \quad (21.6)$$

where $S_k(\cdot; \cdot)$, $k = 1, \dots, K$ are in general nonlinear operators and $T_k(\cdot, \cdot)$, $k = 1, \dots, K$ are linear operators.

Example 5. A weak formulation of the stochastic PDE in (21.4) is given by

$$\begin{aligned} & \int_{\Gamma} \int_D (a(\mathbf{y}) \nabla u) \cdot \nabla v \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} + \int_{\Gamma} \int_D (u(\mathbf{y}) |u(\mathbf{y})|^s) v \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= \int_{\Gamma} \int_D f(\mathbf{y}) v \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \quad \forall v \in H_0^1(D) \otimes L_\rho^q(\Gamma), \end{aligned}$$

where reference to the dependence of a , f , u , and v on the spatial variable \mathbf{x} is omitted for notational simplicity. For the first term on the left-hand side, the operators $S_1(u, \mathbf{y}) = a(\mathbf{y}) \nabla u$ and $T_1(v) = \nabla v$ are linear; for the second term, the operator $S_2(u, \mathbf{y}) = u(\mathbf{y}) |u(\mathbf{y})|^s$ is nonlinear and the operator $T_2(v) = v$ is linear.

Without loss of generality, it suffices to consider the single term form of (21.6), i.e.,

$$\int_{\Gamma} \int_D S(u; \mathbf{y}) T(v) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} = \int_{\Gamma} \int_D v f(\mathbf{y}) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \quad \forall v \in W(D) \otimes L_\rho^q(\Gamma), \quad (21.7)$$

where $T(\cdot)$ is a linear operator and, in general, $S(\cdot)$ is a nonlinear operator and where again the explicit reference to dependences on the spatial variable \mathbf{x} is suppressed.

3.1 Spatial Finite Element Semi-discretization

Let \mathcal{T}_h denote a conforming triangulation of D with maximum mesh size $h > 0$, and let $W_h(D) \subset W(D)$ denote a finite element space, parameterized by $h \rightarrow 0$, constructed using the triangulation \mathcal{T}_h . Let $\{\phi_j(\mathbf{x})\}_{j=1}^{J_h}$ denote a basis for $W_h(D)$ so that J_h denotes the dimension of $W_h(D)$. The *semi-discrete* approximation $u_{J_h}(\mathbf{x}, \mathbf{y}) \in W_h(D) \otimes L_\rho^q(\Gamma)$ has the form

$$u_{J_h}(\mathbf{x}, \mathbf{y}) := \sum_{j=1}^{J_h} c_j(\mathbf{y}) \phi_j(\mathbf{x}). \quad (21.8)$$

At each point in $\mathbf{y} \in \Gamma$, the coefficients $c_j(\mathbf{y})$ and thus u_{J_h} are determined by solving the problem

$$\int_D S\left(\sum_{j=1}^{J_h} c_j(\mathbf{y}) \phi_j(\mathbf{x}); \mathbf{y}\right) T(\phi_{j'}) d\mathbf{x} = \int_D \phi_{j'} f(\mathbf{y}) d\mathbf{x} \quad \text{for } j' = 1, \dots, J_h. \quad (21.9)$$

What this means is that *to obtain the semi-discrete approximation $u_{J_h}(\mathbf{x}, \mathbf{y})$ at any specific point $\mathbf{y}_0 \in \Gamma$, one only has to solve a deterministic finite element problem by fixing $\mathbf{y} = \mathbf{y}_0$ in (21.9)*. The subset of Γ in which (21.9) has no solution has zero measure with respect to $\rho d\mathbf{y}$. For convenience, it is assumed that the coefficient a and the forcing term f in (21.1) admit a smooth extension on $\rho d\mathbf{y}$ -zero measure sets. Then, (21.9) can be extended a.e. in Γ with respect to the Lebesgue measure, instead of the measure $\rho d\mathbf{y}$.

3.2 Stochastic Fully Discrete Approximation

Let $\mathcal{P}(\Gamma) \subset L_\rho^q(\Gamma)$ denote a global/local polynomial subspace, and let $\{\psi_m(\mathbf{y})\}_{m=1}^M$ denote a basis for $\mathcal{P}(\Gamma)$ with M being the dimension of $\mathcal{P}(\Gamma)$. A *fully discrete* approximation of the solution $u(\mathbf{x}, \mathbf{y})$ of (21.7) has the form

$$u_{J_h, M}(\mathbf{x}, \mathbf{y}) := \sum_{m=1}^M \sum_{j=1}^{J_h} c_{jm} \phi_j(\mathbf{x}) \psi_m(\mathbf{y}) \in W_h(D) \times \mathcal{P}(\Gamma), \quad (21.10)$$

where the coefficients c_{jm} and thus $u_{J_h, M}$ are determined by solving the problem

$$\int_\Gamma \int_D S(u_{J_h, M}; \mathbf{y}) T(v) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} = \int_\Gamma \int_D v f(\mathbf{y}) \rho(\mathbf{y}) d\mathbf{x} d\mathbf{y} \quad \forall v \in W_h(D) \times \mathcal{P}(\Gamma) \quad (21.11)$$

In general, the integrals in (21.11) cannot be evaluated exactly so that quadrature rules must be invoked to effect the approximate evaluation of both the integrals over Γ and D . However, by assuming that all methods discussed treat all aspects of the spatial discretization in the same manner, the quadrature rules for the integral over D will not be written out explicitly. As such, for some choice of quadrature points $\{\hat{\mathbf{y}}_r\}_{r=1}^R$ in Γ and quadrature weights $\{w_r\}_{r=1}^R$, (21.11) is further discretized, resulting in

$$\begin{aligned} & \sum_{r=1}^R w_r \rho(\hat{\mathbf{y}}_r) \psi_{m'}(\hat{\mathbf{y}}_r) \cdot \int_D S \left(\sum_{m=1}^M \sum_{j=1}^{J_h} c_{jm} \phi_j(\mathbf{x}) \psi_m(\hat{\mathbf{y}}_r), \hat{\mathbf{y}}_r \right) T(\phi_{j'}(\mathbf{x})) d\mathbf{x} \\ &= \sum_{r=1}^R w_r \rho(\hat{\mathbf{y}}_r) \psi_{m'}(\hat{\mathbf{y}}_r) \int_D \phi_{j'}(\mathbf{x}) f(\mathbf{x}, \hat{\mathbf{y}}_r) d\mathbf{x} \\ & \quad \text{for } j' \in \{1, \dots, J_h\} \text{ and } m' \in \{1, \dots, M\}. \end{aligned} \tag{21.12}$$

The discrete problem (21.12) is generally a fully coupled system of $J_h M$ equations in $J_h M$ degrees of freedom c_{jm} , $j = 1, \dots, J_h$ and $m = 1, \dots, M$. Both intrusive, e.g., stochastic Galerkin, and nonintrusive methods, e.g., MC sampling and stochastic collocation, can be viewed as being special cases of the problems in (21.12). They differ in the choices made for the stochastic polynomial subspace $\mathcal{P}(\Gamma)$, for the basis $\{\psi_m(\mathbf{y})\}_{m=1}^M$, and for the quadrature rule $\{\hat{\mathbf{y}}_r, \omega_r\}_{r=1}^R$. For example, in the context of stochastic Galerkin methods with $\{\psi_m(\mathbf{y})\}_{m=1}^M$ being an orthogonal polynomial basis, the fully discrete approximation $u_{J_h, M}(\mathbf{x}, \mathbf{y})$ of the solution $u(\mathbf{x}, \mathbf{y})$ of the SPDE can be obtained by solving the *single deterministic problem* (21.12). For stochastic collocation methods with the use of global Lagrange basis, set $R = M$ and choose $\{\hat{\mathbf{y}}_r, \omega_r\}_{r=1}^R$ to be the point set for interpolation. Then, the basis $\psi_m(\mathbf{y})$ satisfies the “delta property,” i.e., $\psi_m(\hat{\mathbf{y}}_r) = \delta_{mr}$ for $m = 1, \dots, M$, $r = 1, \dots, R$, so that the discrete problem of size $J_h M \times J_h M$ in (21.12) can be decoupled into M systems at each point $\hat{\mathbf{y}}_r$, each of size $J_h \times J_h$.

3.3 Stochastic Polynomial Subspaces

In this section, several choices of the stochastic polynomial subspace $\mathcal{P}(\Gamma)$ are discussed for constructing the fully discrete approximation in (21.10).

3.3.1 Standard Sparse Global Polynomial Subspaces

Let $\mathcal{S} := \{\mathbf{v} = (v_n)_{1 \leq n \leq N} : v_n \in \mathbb{N}\}$ denote an infinite multi-index set, where N is the dimension of \mathbf{y} . Let $p \in \mathbb{N}$ denote a single index denoting the polynomial order of the associated approximation and consider a sequence of increasing finite multi-index sets $\mathcal{J}(p) \subset \mathcal{S}$ such that $\mathcal{J}(0) = \{(0, \dots, 0)\}$ and $\mathcal{J}(p) \subseteq \mathcal{J}(p+1)$.

Let $\mathcal{P}(\Gamma) := \mathcal{P}_{\mathcal{J}(p)}(\Gamma) \subset L^2_\rho(\Gamma)$ denote the multivariate polynomial space over Γ corresponding to the index set $\mathcal{J}(p)$, defined by

$$\mathcal{P}_{\mathcal{J}(p)}(\Gamma) = \text{span} \left\{ \prod_{n=1}^N y_n^{v_n} \mid \mathbf{v} := (v_1, \dots, v_N) \in \mathcal{J}(p), y_n \in \Gamma_n \right\}, \quad (21.13)$$

where $M := \dim(\mathcal{P}_{\mathcal{J}(p)})$ is the dimension of the polynomial subspace.

Several choices for the index set and the corresponding polynomial spaces are available [5, 22, 30, 65]. The most obvious one is the tensor-product (TP) polynomial space, defined by choosing $\mathcal{J}(p) = \{\mathbf{v} \in \mathbb{N}^N \mid \max_n v_n \leq p\}$. In this case, $M = (p+1)^N$ which results in an explosion in computational effort for higher dimensions. For the same value of p , the same nominal rate of convergence is achieved at a substantially lower costs by the total degree (TD) polynomial spaces for which $\mathcal{J}(p) = \{\mathbf{v} \in \mathbb{N}^N \mid \sum_{n=1}^N v_n \leq p\}$ and $M = (N+p)!/(N!p!)$. Other subspaces having dimension smaller than the TP subspace include hyperbolic cross (HC) polynomial spaces for which $\mathcal{J}(p) = \{\mathbf{v} \in \mathbb{N}^N \mid \sum_{n=1}^N \log_2(v_n + 1) \leq \log_2(p+1)\}$ and sparse Smolyak (SS) polynomial spaces for which

$$\mathcal{J}(p) = \left\{ \mathbf{v} \in \mathbb{N}^N \mid \sum_{n=1}^N \gamma(v_n) \leq \gamma(p) \right\} \quad \text{with } \gamma(p) = \begin{cases} 0 & \text{for } p = 0 \\ 1 & \text{for } p = 1 \\ \lceil \log_2(p) \rceil & \text{for } p \geq 2. \end{cases} \quad (21.14)$$

As illustrated by Example 4, it is often the case that the stochastic input data exhibits anisotropic behavior with respect to the “directions” y_n , $n = 1, \dots, N$. To exploit this effect, it is necessary to approximate u_{J_h} in an anisotropic polynomial space. Following [57], the vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}_+^N$ of positive weights is introduced with $\alpha_{\min} = \min_{n=1, \dots, N} \alpha_n$. The anisotropic versions $\mathcal{J}_{\boldsymbol{\alpha}}(p)$ of the aforementioned polynomial spaces are described in [5]. Here, the anisotropic SS polynomial space is given by

$$\mathcal{J}_{\boldsymbol{\alpha}}(p) = \left\{ \mathbf{v} \in \mathbb{N}^N \mid \sum_{n=1}^N \alpha_n \gamma(v_n) \leq \alpha_{\min} \gamma(p) \right\}.$$

For $N = 2$ dimensions, in Fig. 21.1 examples are provided of both isotropic and anisotropic TP, TD, HC, and SS polynomial spaces, where $p = 8$ and $\boldsymbol{\alpha} = (2, 1)$.

3.3.2 Best M -Term Polynomial Subspaces

The construction of the fully discrete approximation in (21.10) combats the curse of dimensionality by imposing a sparse polynomial space $\mathcal{P}_{\mathcal{J}(p)}(\Gamma)$ a priori (e.g., total degree) and then adaptively selecting a set of polynomial indices (e.g., anisotropic refinement and/or coefficient thresholding). This results in methods with increased

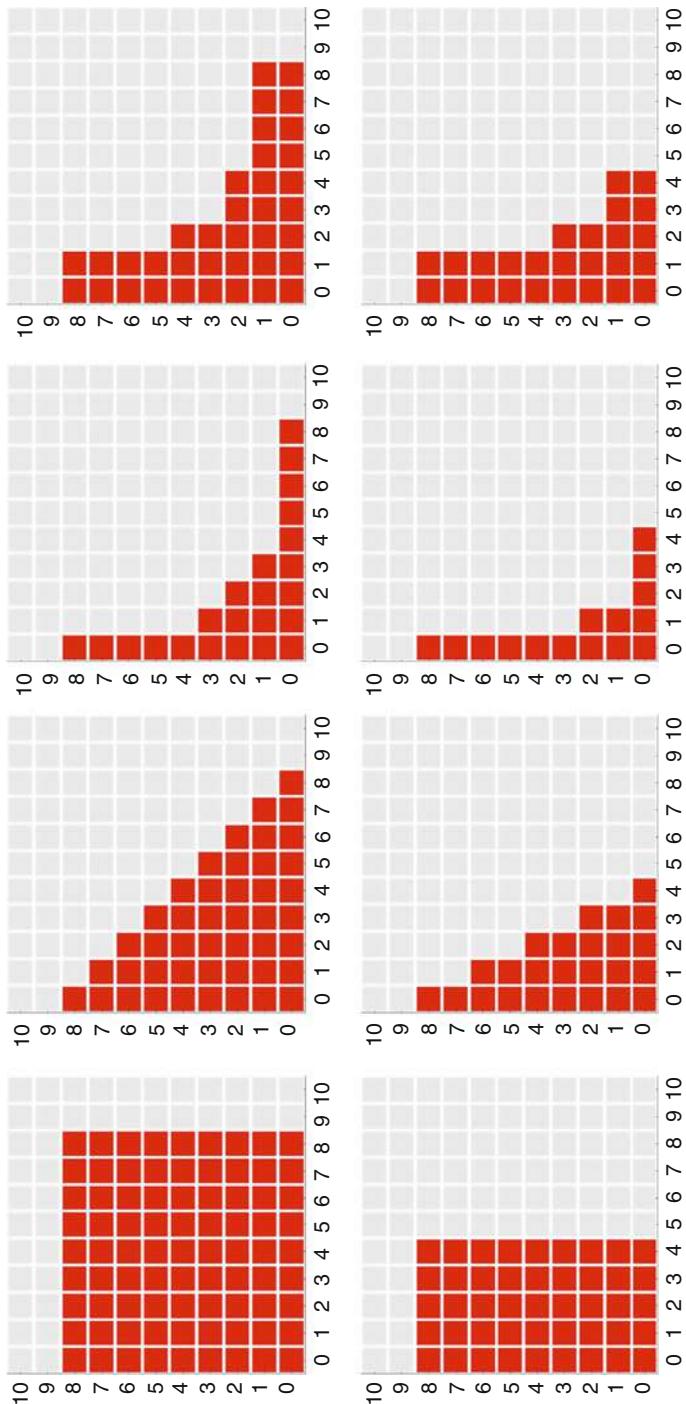


Fig. 21.1 For a finite dimensional Γ with $N = 2$ and a fixed polynomial index $p = 8$, *top row*: the indices $(p_1, p_2) \in \mathcal{J}(8)$ corresponding to the isotropic TP, TD, HC, and SS polynomial spaces. *Bottom row*: the indices $(p_1, p_2) \in \mathcal{J}_a(8)$ with $\alpha_1/\alpha_2 = 2$ corresponding to the anisotropic TP, TD, HC, and SS polynomial spaces

rates of convergence; however, the error estimates depend on the cardinality of the polynomial space, which can grow quickly with respect to the dimension N . Therefore, it is critical to construct a set of the most effective indices for approximation (21.10), which provides maximum accuracy for a given cardinality. In other words, one searches for an index set $\mathcal{J}_M^{\text{opt}}$ with cardinality $M = \#(\mathcal{J}_M^{\text{opt}})$ which minimizes the error $u_{J_h} - u_{J_h, M}$. This practice is known as best M -term approximations. In this case, the fully discrete approximation in (21.10) can be rewritten as

$$u_{J_h, M}(\mathbf{x}, \mathbf{y}) := \sum_{\nu \in \mathcal{J}_M^{\text{opt}}} t_\nu(\mathbf{x}) \psi_\nu(\mathbf{y}) := \sum_{\nu \in \mathcal{J}_M^{\text{opt}}} \left(\sum_{j=1}^{J_h} c_{j,\nu} \phi_j(\mathbf{x}) \right) \psi_\nu(\mathbf{y}), \quad (21.15)$$

where the basis functions of the best M -term subspace $\mathcal{P}_{\mathcal{J}_M^{\text{opt}}}(\Gamma)$ are indexed by $\nu \in \mathcal{J}_M^{\text{opt}}$ and the coefficient $t_\nu(\mathbf{x})$ is defined by $t_\nu(\mathbf{x}) := \sum_{j=1}^{J_h} c_{j,\nu} \phi_j(\mathbf{x})$.

The literature on the best M -term Taylor and Galerkin approximations has been growing fast recently; see [6, 7, 10, 15, 16, 21, 22, 42–44]. In the benchmark work [22], the analytical dependence of the solutions of stochastic elliptic PDEs on the parameters was proved under mild assumptions on the input coefficients, and convergence analysis of the best M -term Taylor and Legendre approximations was established subsequently. Consider, for example, the Taylor expansion of the semi-discrete solution $u_{J_h}(\mathbf{x}, \mathbf{y})$ with respect to \mathbf{y} on $\Gamma = [-1, 1]^N$. Application of the triangle inequality gives

$$\sup_{\mathbf{y} \in \Gamma} \left\| u_{J_h}(\mathbf{y}) - \sum_{\nu \in \mathcal{J}_M^{\text{opt}}} t_\nu \psi_\nu(\mathbf{y}) \right\|_{W(D)} \leq \sum_{\nu \notin \mathcal{J}_M^{\text{opt}}} \|t_\nu\|_{W(D)},$$

which suggests determining optimal index set $\mathcal{J}_M^{\text{opt}}$ by choosing for it the set of indices ν corresponding to M largest $\|t_\nu\|_{W(D)}$. In [22], the error of such approximation was estimated due to Stechkin inequality (see, e.g., [23]):

$$\sum_{\nu \notin \mathcal{J}_M^{\text{opt}}} \|t_\nu\|_{W(D)} \leq \|(\|t_\nu\|_{W(D)})\|_{\ell^q(\mathcal{S})} M^{1-\frac{1}{q}}, \quad (21.16)$$

where $q \in (0, 1)$ such that $(\|t_\nu\|_{W(D)})_{\nu \in \mathcal{S}}$ is ℓ^q -summable. It should be noted that the convergence rate (21.16) does not depend on the dimension of the parameter domain Γ (which is possibly countably infinite therein). This error estimate, however, has some limitations. First, sharp, explicit evaluation of coefficient $\|(\|t_\nu\|_{W(D)})\|_{\ell^q(\mathcal{S})}$ is inaccessible in general (thus so is the total estimate). Secondly, (21.16) often occurs with infinitely many values of q , and stronger rates, corresponding to smaller q , are also attached to bigger coefficients. For a specific range of M , the effective rate of convergence is unclear. In [66], the estimate (21.16) has been improved to a sharp sub-exponential convergence rates of the form

$M \exp(-(\kappa M)^{1/N})$, where κ is a constant depending on the shape and cardinality of multi-index sets. On the other hand, in implementation, finding the best index set and polynomial space is an infeasible task, since this requires computation of all of the t_ν . As a strategy to circumvent this challenge, adaptive algorithms which generate the index set in a near-optimal, greedy procedure were developed in [15]. This method gives the optimal rates; however, it comes with a high cost of exploring the polynomial space, which may be daunting in high-dimensional problems.

3.3.3 Quasi-optimal Polynomial Subspaces

Instead of building the index set based on exact values of polynomial coefficients t_ν , an attractive alternative approach (referred to as *quasi-optimal approximation*) is to establish sharp upper bounds of t_ν (by a priori or a posteriori methods) and then construct $\mathcal{J}_{M^{q\text{-opt}}}$ corresponding to M largest such bounds. For this strategy, the main computational work for the selection of the (near) best terms reduces to determining sharp coefficient estimates, which is expected to be significantly cheaper than exact calculations. Quasi-optimal polynomial approximation has been performed for some parametric elliptic models with optimistic results: while the upper bounds of $\|t_\nu\|_{V(D)}$, denoted by $B(\nu)$, were computed with a negligible cost, the method was comparably as accurate as best M -term approach; see [6, 7]. The first rigorous numerical analysis of quasi-optimal approximation was presented in [6] for $B(\nu) = \mathbf{r}^{-\nu}$ with \mathbf{r} being a vector $(r_n)_{1 \leq n \leq N}$ with $r_n > 1 \forall i$ (see Assumption 2). In that work, the asymptotic sub-exponential convergence rate was proved based on optimizing the Stechkin estimation. Briefly, the analysis applied Stechkin inequality to get

$$\sum_{\nu \notin \mathcal{J}_M^{q\text{-opt}}} B(\nu) \leq \|B(\nu)\|_{\ell^p(S)} M^{1-\frac{1}{q}} \quad (21.17)$$

and then took advantage of the formula of $B(\nu)$ to compute $q \in (0, 1)$ depending on M which minimizes $\|B(\nu)\|_{\ell^p(S)} M^{1-\frac{1}{q}}$.

Although known as an essential tool to study the convergence rate of best M -term approximations, the Stechkin inequality is probably less efficient for quasi-optimal methods. As a generic estimate, it does not fully exploit the available information of the decay of coefficient bounds. In this setting, a direct estimate of $\sum_{\nu \notin \mathcal{J}_M^{q\text{-opt}}} B(\nu)$ may be viable and advantageous to give a sharper result. On the other hand, the process of solving the minimization problem $q^* = \operatorname{argmin}_{q \in (0,1)} \|B(\nu)\|_{\ell^p(S)} M^{1-\frac{1}{q}}$ needs to be tailored to $B(\nu)$, making this approach not ideal for generalization. Currently, the minimization has been limited for some quite simple types of upper bounds. In many scenarios, the sharp estimates of the coefficients may involve complicated bounds which are not even explicitly computable, such as those proposed in [22]. The extension of this approach to such cases seems to be challenging.

In [66], a generalized methodology was proposed for convergence analysis of quasi-optimal polynomial approximations for parameterized elliptic PDEs, where the input coefficient depends affinely and non-affinely on the random parameters. However, since the error analysis only depends on the upper bounds of polynomial coefficients, it is expected that the presented results in [66] can be applied for other, more general model problems with finite parametric dimension, including nonlinear elliptic PDEs, initial value problems, and parabolic equations [16, 42–44]. The key idea of the effort in [66] is to seek for a direct estimate of $\sum_{\nu \notin \mathcal{J}_M^{\text{q-opt}}} B(\nu)$ without using Stechkin inequality. It involves a partition of $B(\mathcal{S} \setminus \mathcal{J}_M^{\text{q-opt}})$ into a family of small positive real intervals $(\mathcal{I}_k)_{k \in \mathcal{K}}$ and the corresponding splitting of $\mathcal{S} \setminus \mathcal{J}_M^{\text{q-opt}}$ into disjoint subsets \mathcal{Q}_k of indices ν such that $B(\nu) \in \mathcal{I}_k$. Under this process, the truncation error can be bounded as

$$\sum_{\nu \notin \mathcal{J}_M^{\text{q-opt}}} B(\nu) = \sum_{k \in \mathcal{K}} \sum_{\nu \in \mathcal{Q}_k} B(\nu) \leq \sum_{k \in \mathcal{K}} \#(\mathcal{Q}_k) \cdot \max(\mathcal{I}_k),$$

thus the quality of the error estimate mainly depends on the approximation of cardinality of \mathcal{Q}_k . To tackle this problem, the authors of [66] developed a strategy which extends \mathcal{Q}_k into continuous domain and, through relating the number of N -dimensional lattice points to continuous volume (Lebesgue measure), established a sharp estimate of the cardinality $\#(\mathcal{Q}_k)$ up to any prescribed accuracy. The development includes the utilization and extension of several results on lattice point enumeration; see [8, 38] for a survey. Under some weak assumptions on $B(\nu)$, an asymptotic sub-exponential convergence rate of truncation error of the form $M \exp(-(\kappa M)^{1/N})$ was achieved, where κ is a constant depending on the shape and size of quasi-optimal index sets. Through several examples, the authors explicitly derived κ and demonstrated the optimality of the estimate both theoretically (by proving a lower bound) and computationally (via comparison with exact calculation of truncation error). The advantage of the analysis framework is therefore twofold. First, it applies to a general class of quasi-optimal approximations; further, it gives sharp estimates for their asymptotic convergence rates.

3.3.4 Local Piecewise Polynomial Subspaces

The use of global polynomials discussed requires high regularity of the solution $u(\mathbf{x}, \mathbf{y})$ with respect to the random parameters $\{y_n\}_{n=1}^N$. They are therefore ineffective for the approximation of solutions that have irregular dependence with respect to those parameters. Motivated by finite element methods (FEMs) for spatial approximation, an alternate and potentially more effective approach for approximating irregular solutions is to use *locally supported piecewise polynomial* approaches for approximating the solution dependence on the random parameters. To achieve greater accuracy, global polynomial approaches increase the polynomial degree; piecewise polynomial approaches instead keep the polynomial degree fixed but refine the grid used to define the approximation space.

4 Global Stochastic Collocation Methods

This section focuses on the construction of the fully discrete approximation (21.10) in the subspace $W_h(D) \otimes \mathcal{P}_{\mathcal{J}(p)}(\Gamma)$. Rather than making use of a Galerkin projection in both the deterministic and stochastic domains, the semi-discrete approximation $u_{J_h}(\cdot, \mathbf{y})$ given in (21.8) is instead collocated on an appropriate set of points $\{\mathbf{y}_m\}_{m=1}^M \in \Gamma$ to determine $M := \dim(\mathcal{P}_{\mathcal{J}(p)})$ solutions $\{u_{J_h}(\cdot, \mathbf{y}_m)\}_{m=1}^M$. One can then use these solutions to construct a global, possibly interpolatory, polynomial to define the fully discrete approximation $u_{J_h, M}(\mathbf{x}, \mathbf{y}) := u_{J_h, M}^{\text{gSC}}(\mathbf{x}, \mathbf{y})$. This process is referred to as global stochastic collocation methods (gSCMs). Clearly, gSCMs are *nonintrusive* in the sense that the solution of (21.10) naturally decouples into a series of M deterministic solves, each of size $J_h \times J_h$.

In general, throughout this section, the following assumption is made about the regularity of the solution to (21.7).

Assumption 2. For $0 < \delta < a_{\min}$, there exists \mathbf{r} , with $(r_n)_{1 \leq n \leq N}$ and $r_n > 1 \ \forall i$, such that the complex extension of u to the (δ, \mathbf{r}) -polyellipse $\mathcal{E}_{\mathbf{r}}$, i.e., $u^* : \mathcal{E}_{\mathbf{r}} \rightarrow W(D)$ is well-defined and analytic in an open neighborhood of $\mathcal{E}_{\mathbf{r}}$, given by

$$\mathcal{E}_{\mathbf{r}} = \bigotimes_{1 \leq n \leq N} \left\{ z_n \in \mathbb{C} : \Re(z_n) = \frac{r_n + r_n^{-1}}{2} \cos \phi, \Im(z_n) = \frac{r_n - r_n^{-1}}{2} \sin \phi, \phi \in [0, 2\pi] \right\},$$

for all $x \in \overline{D}$ and all $\mathbf{z} = (z_n)_{1 \leq n \leq N}$ contained in the polyellipse.

4.1 Lagrange Global Polynomial Interpolation in Parameter Space

Interpolatory approximations in parameter space start with the selection of a set of distinct points $\{\mathbf{y}_m\}_{m=1}^M \in \Gamma$ and a set of basis functions $\{\psi_m(\mathbf{y})\}_{m=1}^M \subset \mathcal{P}_{\mathcal{J}(p)}(\Gamma)$. Then, an approximation $u_{J_h, M}^{\text{gSC}} \in W_h(D) \otimes \mathcal{P}_{\mathcal{J}(p)}(\Gamma)$ of the form

$$u_{J_h, M}^{\text{gSC}}(\mathbf{x}, \mathbf{y}) := \sum_{m=1}^M c_m(\mathbf{x}) \psi_m(\mathbf{y}). \quad (21.18)$$

The Lagrange interpolant is defined by first obtaining M realizations $u_{J_h}(\mathbf{x}, \mathbf{y}_m)$ of the finite element approximation of the solution $u(\mathbf{x}, \mathbf{y}_m)$ of the problem (21.1), i.e., one solves for the finite element approximation for each of the interpolation points in the set $\{\mathbf{y}_m\}_{m=1}^M$. Then, the coefficient functions $\{c_m(\mathbf{x})\}_{m=1}^M$ are determined by imposing the interpolation conditions

$$\sum_{m=1}^M c_m(\mathbf{x}) \psi_m(\mathbf{y}_{m'}) = u_{J_h}(\mathbf{x}, \mathbf{y}_{m'}) \quad \text{for } m' = 1, \dots, M. \quad (21.19)$$

Thus, each of the coefficient functions $\{c_m(\mathbf{x})\}_{m=1}^M$ is a linear combination of the finite element data $\{u_{J_h}(\mathbf{x}, \mathbf{y}_m)\}_{m=1}^M$; the specific linear combinations are determined in the usual manner from the entries of the inverse of the $M \times M$ interpolation matrix \mathbf{L} having entries $L_{m',m} = \psi_m(\mathbf{y}_{m'})$, $m, m' = 1, \dots, M$. The sparsity and conditioning of \mathbf{L} heavily depend on the choice of basis; that choice could result in matrices that range from fully dense to diagonal and from highly ill conditioned to perfectly well conditioned.

The main attraction of interpolatory approximations of parameter dependences is that it effects a complete decoupling of the spatial and probabilistic degrees of freedom. Clearly, once the interpolation points $\{\mathbf{y}_m\}_{m=1}^M$ are chosen, one can solve M deterministic finite element problems, one for each parameter point \mathbf{y}_m , with total disregard to what basis $\{\psi_m(\mathbf{y})\}_{m=1}^M$ one choose to use. Then, the coefficients $\{c_m(\mathbf{x})\}_{m=1}^M$ defining the approximation (21.18) are found from the interpolation conditions in (21.19); it is only in this last step that the choice of stochastic basis enters into the picture. Note that this decoupling property makes the implementation of Lagrange interpolatory approximations of parameter dependences almost as trivial as it is for Monte Carlo sampling. However, if that dependence is smooth, because of the higher accuracy of global polynomial approximations in the space $\mathcal{P}_{\mathcal{J}(p)}(\Gamma)$, interpolatory approximations require substantially fewer sampling points to achieve a desired error tolerance.

Given a set of interpolation points, to complete the setup of a Lagrange interpolation problem, one has to choose a basis. The simplest and most popular choices are the Lagrange fundamental polynomials, i.e., polynomials that have the *delta property* $\psi_{m'}(\mathbf{y}_m) = \delta_{m'm}$, where $\delta_{m'm}$ denotes the Kronecker delta. In this case, the interpolating conditions (21.19) reduce to $c_m(\mathbf{x}) = u_{J_h}(\mathbf{x}, \mathbf{y}_m)$ for $m = 1, \dots, M$, i.e., the interpolation matrix \mathbf{L} is simply the $M \times M$ identity matrix. In this sense, the use of Lagrange polynomial bases can be viewed as resulting in pure sampling methods, much the same as Monte Carlo methods, but instead of randomly sampling in the parameter space Γ , the sample points are deterministically structured. Mathematically, using the Lagrange fundamental polynomial basis $\{\psi_m\}_{m=1}^M$, this ensemble-based approach results in a fully discrete approximation of the solution $u(\mathbf{x}, \mathbf{y})$ of the form

$$u_{J_h M}^{gSC}(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^M u_{J_h}(\mathbf{x}, \mathbf{y}_m) \psi_m(\mathbf{y}). \quad (21.20)$$

The construction of multivariate Lagrange fundamental polynomials for a general set of interpolation points is not an easy matter. Fortunately, there exist means for so doing; see [59].

4.2 Generalized Sparse Grid Construction

By following [5, 57, 58, 61], a generalized version of the Smolyak sparse grid gSCM is described for interpolation and quadrature. For each $n = 1, \dots, N$, let $l_n \in \mathbb{N}_+$ denote the one-dimensional level of approximation, and let $\{y_{n,k}^{(l_n)}\}_{k=1}^{m(l_n)} \subset \Gamma_n$ denote

a sequence of one-dimensional interpolation points in Γ_n . Here, $m(l) : \mathbb{N}_+ \rightarrow \mathbb{N}_+$ is such that $m(0) = 0$, $m(1) = 1$, and $m(l) < m(l + 1)$ for $l = 2, 3, \dots$ so that $m(l)$ strictly increases with l ; $m(l_n)$ defines the total number of collocation points at level l_n . For a univariate function $v \in C^0(\Gamma_n)$, a sequence of one-dimensional Lagrange interpolation operators $\mathcal{U}_n^{m(l_n)} : C^0(\Gamma_n) \rightarrow \mathcal{P}_{m(l_n)-1}(\Gamma_n)$ is defined by

$$\mathcal{U}_n^{m(l_n)}[v](y_n) = \sum_{k=1}^{m(l_n)} v\left(y_{n,k}^{(l_n)}\right) \psi_{n,k}^{(l_n)}(y_n) \quad \text{for } l_n = 1, 2, \dots, \quad (21.21)$$

where $\psi_{n,k}^{(l_n)} \in \mathcal{P}_{m(l_n)-1}(\Gamma_n)$, $k = 1, \dots, m(l_n)$ are Lagrange fundamental polynomials of degree $p_{l_n} = m(l_n) - 1$ such that

$$\psi_{n,k}^{(l_n)}(y_n) = \prod_{\substack{k'=1 \\ k' \neq k}}^{\min(l_n)} \frac{\left(y_n - y_{n,k'}^{(l_n)}\right)}{\left(y_{n,k}^{(l_n)} - y_{n,k'}^{(l_n)}\right)}.$$

Using the convention that $\mathcal{U}_n^{m_0} = 0$, the difference operator can be defined by

$$\Delta_n^{m(l_n)} = \mathcal{U}_n^{m(l_n)} - \mathcal{U}_n^{m(l_n-1)}. \quad (21.22)$$

For the multivariate case, let $\mathbf{l} = (l_1, \dots, l_N) \in \mathbb{N}_+^N$ denote a multi-index and $L \in \mathbb{N}_+$ denote the total level of the sparse grid approximation. Then, for each $n = 1, \dots, N$, the N -dimensional hierarchical surplus operator can be defined by

$$\Delta^m = \bigotimes_{n=1}^N \Delta_n^{m(l_n)}, \quad (21.23)$$

and, from (21.22) and (21.23), the L -th level generalized sparse grid operator given by

$$\mathcal{I}_L^{m,g} = \sum_{g(\mathbf{l}) \leq L} \bigotimes_{n=1}^N \Delta_n^{m(l_n)}, \quad (21.24)$$

where $g : \mathbb{N}_+^N \rightarrow \mathbb{N}$ is another strictly increasing function that defines the mapping between the multi-index \mathbf{l} and the level L used to construct the sparse grid. Finally, given the functions m and g and a level L , the generalized sparse grid approximation of u_{J_h} can be constructed by

$$u_{J_h M_L}^{\text{gSC}} = \mathcal{I}_L^{m,g}[u_{J_h}] = \sum_{L-N+1 \leq g(\mathbf{l}) \leq L} \sum_{\substack{\mathbf{k} \in \{0,1\}^N \\ g(\mathbf{l+k}) \leq L}} (-1)^{|\mathbf{k}|} \bigotimes_{n=1}^N \mathcal{U}_n^{m(l_n)}[u_{J_h}]. \quad (21.25)$$

The fully discrete gSCM (21.25) requires the independent evaluation of the finite element approximation $u_{J_h}(\mathbf{x}, \mathbf{y})$ on a deterministic set of *distinct collocation points* given by

$$\mathcal{H}_L^{m,g} = \bigcup_{g(\mathbf{l}) \leq L} \bigotimes_{n=1}^N \left\{ y_{n,k}^{(l_n)} \right\}_{k=1}^{m(l_n)}$$

having cardinality M_L , i.e., $M = M_L$ in (21.10). Moreover, the construction of the sparse grid approximation naturally enables the evaluation of moments through simple sparse grid quadrature formulas, i.e.,

$$\mathbb{E}[u_{J_h M_L}^{\text{gSC}}](\mathbf{x}) = \sum_{m=1}^{M_L} u_{J_h}(\mathbf{x}, \mathbf{y}_m) \underbrace{\int_{\Gamma} \psi_m(\mathbf{y}) \rho(\mathbf{y}) d\mathbf{y}}_{\text{precomputed weights}} = \sum_{m=1}^{M_L} u_{J_h}(\mathbf{x}, \mathbf{y}_m) w_m$$

and

$$\mathbb{V}[u_{J_h M_L}^{\text{gSC}}](\mathbf{x}) = \sum_{m=1}^{M_L} \tilde{w}_m u_{J_h}^2(\mathbf{x}, \mathbf{y}_m) - \mathbb{E}[u_{J_h M_L}^{\text{gSC}}]^2(\mathbf{x}),$$

where $\tilde{w}_m = \mathbb{V}[\psi_m(\mathbf{y})]$, $m = 1, \dots, M_L$.

The work [5] constructs the underlying polynomial space associated with the approximation (21.25) for particular choices of m , g , and L . Let $m^{-1}(k) = \min\{l \in \mathbb{N}_+ \mid m(l) \geq k\}$ denote the left inverse of m such that $m^{-1}(m(l)) = l$ and $m(m^{-1}(k)) \geq k$. Then, let $\mathbf{s}(\mathbf{l}) = (m(l_1), \dots, m(l_N))$ and define the polynomial index set

$$\mathcal{J}^{m,g}(L) = \{ \mathbf{v} \in \mathbb{N}^N : g(\mathbf{m}^{-1}(\mathbf{v} + \mathbf{1})) \leq L \}.$$

With this definition in hand, the following proposition, whose proof can be found in [5, Proposition 1], characterizes the underlying polynomial space of the sparse grid approximation $\mathcal{I}_L^{m,g}[u_{J_h}]$.

Proposition 1. *Let $m : \mathbb{N}_+ \rightarrow \mathbb{N}_+$ and $g(\mathbf{l}) : \mathbb{N}_+^N \rightarrow \mathbb{N}$ denote strictly increasing functions, as described above, and $\{y_{n,k}^{(l)}\}_{k=1}^{m(l)} \subset \mathbb{C}_n$ denote arbitrary distinct points used in (21.21) to determine $\mathcal{U}_n^{m(l)}$, $l = 1, 2, \dots$. Then,*

- (1) *For any function $v \in C^0(\Gamma)$, the approximation $\mathcal{I}_L^{m,g}[v] \in \mathcal{P}_{\mathcal{J}^{m,g}(L)}(\Gamma)$.*
- (2) *For all $v \in \mathcal{P}_{\mathcal{J}^{m,g}(p)}(\Gamma)$, it holds that $\mathcal{I}_L^{m,g}[v] = v$.*

With Proposition 1 in hand, the sparse grid approximation $\mathcal{I}_L^{m,g}$ can be related with the corresponding polynomial subspaces, i.e., $\mathcal{P}_{\mathcal{J}^{m,g}(L)}(\Gamma)$ with $m(l) = l$ and $g(\mathbf{l}) = \max_{n=1,\dots,N} (l_n - 1) \leq L$, $g(\mathbf{l}) = \sum_{n=1}^N (l_n - 1) \leq L$, and

$g(\mathbf{l}) = \sum_{n=1}^N \log_2(l_n) \leq \log_2(L + 1)$ for the tensor-product, total degree, and hyperbolic cross polynomial subspaces, respectively. However, the most widely used polynomial subspace is the sparse Smolyak given by (21.14), which, in the context of the sparse grid approximation, is defined by

$$m(1) = 1, \quad m(l) = 2^{l-1} + 1, \quad \text{and} \quad g(\mathbf{l}) = \sum_{n=1}^N (l_n - 1). \quad (21.26)$$

Moreover, similar to the anisotropic polynomial spaces, the generalized gSCM enables anisotropic refinement with respect to the direction y_n by incorporating a weight vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}_+^N$ into the mapping $g : \mathbb{N}_+^N \rightarrow \mathbb{N}$, e.g., $g(\mathbf{l}) = \sum_{n=1}^N \alpha_n (l_n - 1) \leq \alpha_{\min} L$ in (21.26). Anisotropic refinement will be discussed further in the sections that follow, but first, two choices of points used for (21.25) are introduced, namely, the Clenshaw-Curtis and Gaussian points. See [68] for an insightful comparison of quadrature formulas based on these points.

4.2.1 Clenshaw-Curtis Points on Bounded Hypercubes

Without loss of generality, assume that $\Gamma_n = [-1, 1]$. The Clenshaw-Curtis points are the extrema of Chebyshev polynomials (CC) (see [19]) given by, for any choice of $m(l) > 1$,

$$y_k^{(l)} = -\cos\left(\frac{\pi(k-1)}{m(l)-1}\right) \quad \text{for } k = 1, \dots, M_l. \quad (21.27)$$

In addition, one sets $y_1^{(l)} = 0$ if $m(l) = 1$ and chooses the multi-index map g as well as the number of points $m(l)$, $l > 1$, at each level as in (21.26). Note that this particular choice corresponds to the most used sparse grid approximation because it leads to nested sequences of points, i.e., $\{y_k^{(l)}\}_{k=1}^{m(l)} \subset \{y_k^{(l+1)}\}_{k=1}^{m(l+1)}$ so that the sparse grids are also nested, i.e., $\mathcal{H}_L^{m,g} \subset \mathcal{H}_{L+1}^{m,g}$.

However, even though the CC choice of points results in a significantly reduced number of points used by $\mathcal{I}_L^{m,g}$, that number of points eventually increases exponentially fast with N . With this in mind, an alternative to the standard Clenshaw-Curtis (CC) family of rules is considered which attempts to retain the advantages of nestedness while reducing the excessive growth described above. To achieve this, it is necessary to exploit the fact that the CC interpolant is exact in the polynomial space $\mathcal{P}_{m(l)-1}$ to drop, in each direction, the requirement that the function m be strictly increasing. Instead, a new mapping $\tilde{m}(l) : \mathbb{N}_+ \rightarrow \mathbb{N}_+$ is defined such that $\tilde{m}(l) \leq \tilde{m}(l+1)$ and $\tilde{m}(l) = \tilde{m}(k)$, where $k = \operatorname{argmin}\{k' | 2^{k'-1} \geq L\}$. In other words, the current rule are reused for as many levels as possible, until the total degree subspace is properly included. Figure 21.2 shows the difference between the standard CC sparse grid and the “slow growth” CC (sCC) sparse grid for $l = 1, 2, 3, 4$. Figure 21.3 shows the corresponding polynomial accuracy of the CC and sCC sparse grids when used in a quadrature rule approximation (as opposed

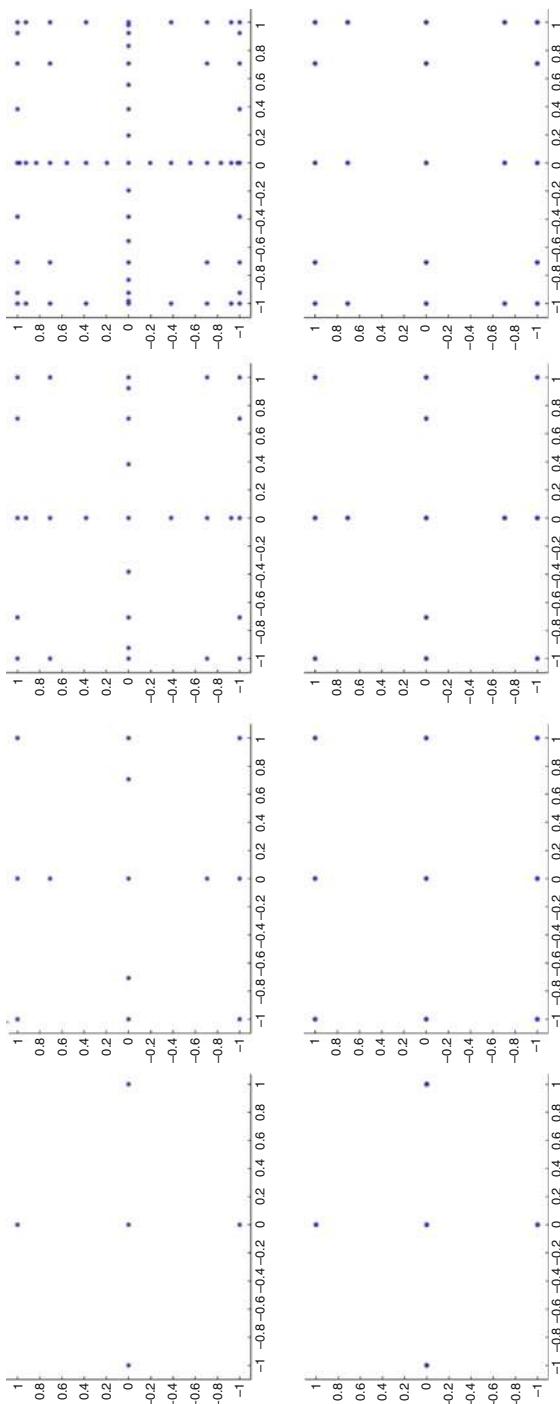


Fig. 21.2 For $\Gamma = [-1, 1]^2$, the sparse grids corresponding to levels $L = 1, 2, 3, 4$, using standard Clenshaw-Curtis points (*top*) and slow-growth Clenshaw-Curtis points (*bottom*)

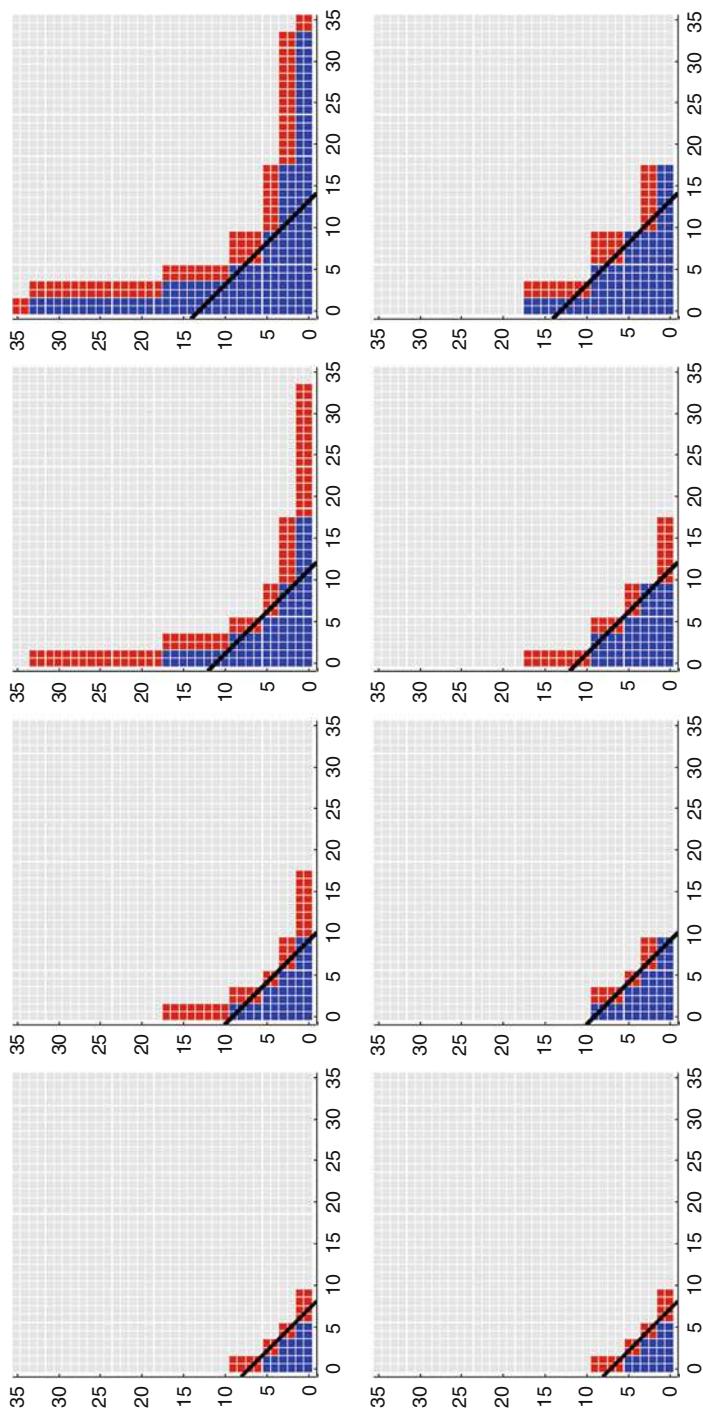


Fig. 21.3 For $\Gamma = [-1, 1]^2$, the polynomial subspaces associated with integrating a function $u \in C^0(\Gamma)$, using sparse grids corresponding to levels $L = 3, 4, 5, 6$ using standard Clenshaw-Curtis points (*top*) and slow-growth Clenshaw-Curtis points (*bottom*)

to an interpolant) of the integral a function in $C^0(\Gamma)$. Note that the concept of “slow growth” can also be applied to other nested one-dimensional rules, including, e.g., the Gauss-Patterson points [34].

4.2.2 Gaussian Points in Bounded or Unbounded Hypercubes

The Gaussian points $\{y_{n,k}^{(l_n)}\}_{k=1}^{m(l_n)} \subset \Gamma_n$ are the zeros of the orthogonal polynomials with respect to some positive weight function. In general, they are not nested. The natural choice for the weight function is the PDF $\rho(\mathbf{y})$ of the random variables \mathbf{y} . However, in the general multivariate case, if the random variables y_n are not independent, the PDF $\rho(\mathbf{y})$ does not factorize, i.e., $\rho(\mathbf{y}) \neq \prod_{n=1}^N \rho_n(y_n)$. As a result, an auxiliary probability density function $\widehat{\rho}(\mathbf{y}) : \Gamma \rightarrow \mathbb{R}^+$ is defined by

$$\widehat{\rho}(\mathbf{y}) = \prod_{n=1}^N \widehat{\rho}_n(y_n) \quad \forall \mathbf{y} \in \Gamma \quad \text{and such that} \quad \left\| \frac{\rho}{\widehat{\rho}} \right\|_{L^\infty(\Gamma)} < \infty.$$

Note that $\widehat{\rho}(\mathbf{y})$ factorizes so that it can be viewed as a joint PDF for N independent random variables.

For each parameter dimension $n = 1, \dots, N$, let the $m(l_n)$ Gaussian points be the roots of the $m(l_n)$ degree polynomial that is $\widehat{\rho}_n$ -orthogonal to all polynomials of degree $m(l_n - 1)$ on the interval Γ_n . The auxiliary density $\widehat{\rho}$ should be chosen as close as possible to the true density ρ so that the quotient $\rho/\widehat{\rho}$ is not too large.

4.2.3 Selection of the Anisotropic Weights for Example 3

In the special case of Example 3, the analytic dependence with respect to each of the random variables, i.e., Assumption 2, reduces to the following: for $n = 1, \dots, N$, let $\Gamma_n^* = \prod_{\substack{j=1 \\ j \neq n}}^N \Gamma_j$, and let \mathbf{y}_n^* denote an arbitrary element of Γ_n^* . Then there exist constants λ and $\tau_n \geq 0$ and regions $\Sigma_n \equiv \{z \in \mathbb{C}, \text{ dist}(z, \Gamma_n) \leq \tau_n\}$ in the complex plane for which

$$\max_{\mathbf{y}_n^* \in \Gamma_n^*} \max_{z \in \Sigma_n} \|u(\cdot, \mathbf{y}_n^*, z)\|_{W(D)} \leq \lambda,$$

such that, the solution $u(\mathbf{x}, \mathbf{y}_n^*, y_n)$ admits an analytic extension $u(\mathbf{x}, \mathbf{y}_n^*, z)$, $z \in \Sigma_n \subset \mathbb{C}$. The ability to treat the stochastic dimensions differently is a necessity because many practical problems exhibit highly anisotropic behavior, e.g., the size τ_n of the analyticity region associated to each random variable y_n increases with n .

In such a case, it is known that if the dependence on each random variable is approximated with polynomials, the best approximation error decays exponentially fast with respect to the polynomial degree. More precisely, for a bounded region Γ_n and a univariate analytic function, recall the following Lemma, whose proof can be found in [1, Lemma 7] and which is an immediate extension of the result given in [24, Chapter 7, Section 8].

Lemma 1. Given a function $v \in C^0(\Gamma_n; W(D))$, which admits an analytic extension in the region of the complex plane $\Sigma(\Gamma_n; \tau_n) = \{z \in \mathbb{C}, \text{dist}(z, \Gamma_n) \leq \tau_n\}$ for some $\tau_n > 0$, then

$$E_{m(l_n)} \equiv \min_{w \in \mathcal{P}_{m(l_n)}} \|v - w\|_{C_n^0} \leq \frac{2}{e^{2r_n} - 1} e^{-2M_{l_n}r_n} \max_{z \in \Sigma(\Gamma_n; \tau_n)} \|v(z)\|_{W(D)}$$

with

$$0 < r_n = \frac{1}{2} \log \left(\frac{2\tau_n}{|\Gamma_n|} + \sqrt{1 + \frac{4\tau_n^2}{|\Gamma_n|^2}} \right). \quad (21.28)$$

A related result with weighted norms holds for unbounded random variables whose probability density decays as the Gaussian density at infinity; see [1].

In the multivariate case, the size τ_n of the analyticity region depends, in general, on the direction n . As a consequence, the decay coefficient r_n will also depend on the direction. The key idea of the anisotropic sparse gSCM in [58] is to place more points in directions having slower convergence rate, i.e., with smaller value for r_n . In particular, the weights α_n can be linked with the rate of exponential convergence in the corresponding direction by

$$\alpha_n = r_n \quad \text{for all } n = 1, 2, \dots, N. \quad (21.29)$$

Let

$$\underline{\alpha} = \underline{r} = \min_{n=0,1,\dots,N} \{r_n\} \quad \text{and} \quad \mathcal{R}(N) = \sum_{n=1}^N r_n. \quad (21.30)$$

As observed in Remark 4, the choice $\alpha = \mathbf{r}$ is optimal with respect to the error bound derived in Theorem 1. Now, the problem of choosing α is transformed into the one of estimating the decay coefficients $\mathbf{r} = (r_1, \dots, r_N)$. In [58, Section 2.2], two rigorous estimation strategies are given. The first uses a priori knowledge about the error decay in each direction, whereas the second approach uses a posteriori information obtained from computations and fits the values of \mathbf{r} .

An illustration of the salubrious effect on the resulting sparse grid resulting from accounting for anisotropy is given in Fig. 21.4.

4.2.4 Sparse Grid gSCM Error Estimates for Example 3

Global sparse grid Lagrange interpolation gSCMs can be used to approximate the solution $u \in C^0(\Gamma; W(D))$ using finitely many function values. By Assumption 2, u admits an analytic extension. Furthermore, each function value is computed by means of a finite element technique. Recall that the fully discrete approximation is

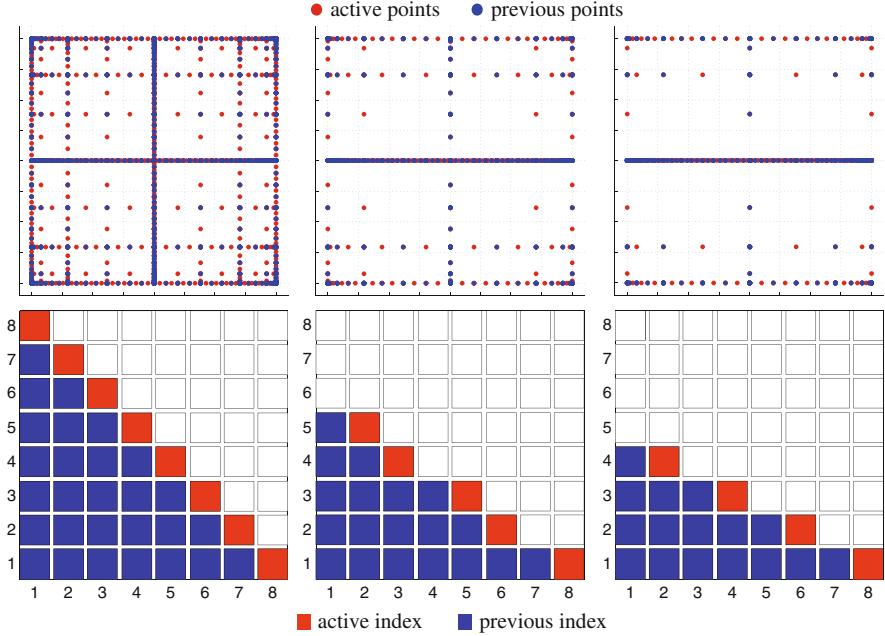


Fig. 21.4 For $\Gamma = [0, 1]^2$ and $L = 7$: the anisotropic sparse grids with $\alpha_2/\alpha_1 = 1$ (isotropic), $\alpha_2/\alpha_1 = 3/2$, and $\alpha_2/\alpha_1 = 2$ utilizing the Clenshaw-Curtis points (top row) and the corresponding indices (l_1, l_2) such that $\alpha_1(l_1 - 1) + \alpha_2(l_2 - 1) \leq \alpha_{\min} L$ (bottom row)

defined as $u_{J_h M_p}^{\text{gSC}} = \mathcal{I}_L^{m,g}[u_{J_h}]$, where the operator $\mathcal{I}_L^{m,g}$ is defined in (21.24). The aim is to provide an a priori estimates for the total error

$$\epsilon = u - \mathcal{I}_L^{m,g}[u_{J_h}].$$

The goal is to investigate the error

$$\|u - \mathcal{I}_L^{m,g}[u_{J_h}]\| \leq \underbrace{\|u - u_{J_h}\|}_{(I)} + \underbrace{\|u_{J_h} - \mathcal{I}_L^{m,g}[u_{J_h}]\|}_{(II)} \quad (21.31)$$

evaluated in the natural norm $L_\rho^2(\Gamma; W(D))$. Note that if the stochastic data, i.e., a and/or f , are not an exact representation but are instead an approximation in terms of N random variables, e.g., arising from a suitable truncation of infinite representations of random fields, then there would be an additional error $\|u - u_N\|$ to consider. This contribution to the total error was considered in [57, Section 4.2]. By controlling the error in this natural norm, the error can also be controlled in the expected value of the solution, e.g.,

$$\|\mathbb{E}[\epsilon]\|_{W(D)} \leq \mathbb{E}[\|\epsilon\|_{W(D)}] \leq \|\epsilon\|_{L_\rho^2(\Gamma; W(D))}.$$

The quantity (I) accounts for the error with respect to the spatial grid size h , i.e., the finite element error; it is estimated using standard approximability properties of the finite element space $W_h(D)$ and the spatial regularity of the solution u ; see, e.g., [11, 18]. Specifically,

$$\|u - u_{J_h}\|_{L^2_\rho(\Gamma; W(D))} \leq h^s \left(\int_\Gamma C_\pi(\mathbf{y}) C(s; u(\mathbf{y}))^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2}.$$

The primary concern will be to analyze the approximation error (II) , i.e.,

$$\|u_{J_h} - \mathcal{I}_L^{m,g}[u_{J_h}]\|_{L^2_\rho(\Gamma; W(D))}, \quad (21.32)$$

for the Clenshaw-Curtis points using the anisotropic sparse grid approximation with $m(l)$ and g defined as follows:

$$m(1) = 1, \quad m(l) = 2^{l-1} + 1, \quad \text{and} \quad g(l) = \sum_{n=1}^N \alpha_n (l_n - 1) \leq \alpha_{\min} L. \quad (21.33)$$

Error analysis of the sparse grid approximation with other isotropic or anisotropic choices of $m(l)$ and g can be found in [57, 58].

Under the very reasonable assumption that the semi-discrete finite element solution u_{J_h} admits an analytic extension as described in Assumption 2 with the same analyticity region as for u , the behavior of the error (21.32) will be analogous to $\|u - \mathcal{I}_L^{m,g}[u]\|_{L^2_\rho(\Gamma; W(D))}$. For this reason, the analysis presented next considers the latter.

Recall that even though in the global estimate (21.31) it is enough to bound the approximation error (II) in the $L^2_\rho(\Gamma; W(D))$ norm, we consider the more stringent $L^\infty(\Gamma; W(D))$ norm. In this chapter, the norm $\|\cdot\|_{\infty,n}$ is shorthand for $\|\cdot\|_{L^\infty(\Gamma_n; W(D))}$ and similarly, $\|\cdot\|_{\infty,N}$ is shorthand for $\|\cdot\|_{L^\infty(\Gamma; W(D))}$.

The multidimensional error estimate $\|u - \mathcal{I}_L^{m,g}[u]\|$ is constructed from a sequence of one-dimensional estimates and a tight bound on the number of distinct nodes on the sparse grid $\mathcal{H}_L^{m,g}$. To begin, let E_m denote the best approximation error, as in Lemma 1, to functions $u \in C^0(\Gamma_n; W(D))$ by polynomial functions $w \in \mathcal{P}_M$. Because the interpolation formula $\mathcal{U}_n^{M_{l_n}}$, $n = 1, \dots, N$ is exact for polynomials in $\mathcal{P}_{m(l_n)-1}$, the general formula can be applied

$$\|u - \mathcal{U}^{m(l_n)}[u]\|_{\infty,n} \leq (1 + \Lambda_{m(l_n)}) E_{m(l_n)-1}(u), \quad (21.34)$$

where Λ_m denotes the Lebesgue constant corresponding to the points (21.27). In this case, it is known that

$$\Lambda_m \leq \frac{2}{\pi} \log(m-1) + 1 \quad (21.35)$$

for $M_{l_n} \geq 2$; see [26]. On the other hand, using Lemma 1, the best approximation to functions $u \in C^0(\Gamma_n; W(D))$ that admit an analytic extension as described by Assumption 2 is bounded by

$$E_{m(l_n)}(u) \leq \frac{2}{e^{2r_n} - 1} e^{-2m(l_n)r_n} \theta(u), \quad (21.36)$$

where $\theta(u) = \max_{1 \leq n \leq N} \max_{y_n^* \in \Gamma_n^*} \max_{z \in \Sigma(\Gamma_n; \tau_n)} \|u(z)\|_{W(D)}$. For $n = 1, 2, \dots, N$, denote the one-dimensional identity operator by $I_n : C^0(\Gamma_n; W(D)) \rightarrow C^0(\Gamma_n; W(D))$ and use (21.34)–(21.36) to obtain the estimates

$$\|u - \mathcal{U}_n^{m(l_n)}[u]\|_{\infty,n} \leq \frac{4}{e^{2r_n} - 1} l_n e^{-r_n 2^{l_n}} \theta(u)$$

and

$$\|\Delta_n^{m(l_n)}[u]\|_{\infty,n} \leq \frac{8}{e^{2r_n} - 1} l_n e^{-r_n 2^{l_n} - 1} \theta(u). \quad (21.37)$$

Because the value $\theta(u)$ affects the error estimates as a multiplicative constant, from here on, it is assumed to be one without any loss of generality.

The next Theorem provides an error bound in terms of the total number M_L of Clenshaw-Curtis collocation points. The details of the proof can be found in [58, Section 3.1.1] and is therefore omitted. A similar result holds for the sparse grid $\mathcal{I}_L^{m,g}$ using Gaussian points and can be found in [58, Section 3.1.2].

Theorem 1. *Let $u \in L_p^2(\Gamma; W(D))$ and let the functions m and g satisfy (21.33) with weights $\alpha_n = r_n$. Then for the gSCM approximation based on the Clenshaw-Curtis points, the following estimates hold.*

- Algebraic convergence $(0 \leq L \leq \frac{\mathcal{R}(N)}{\underline{r} \log(2)})$.

$$\|u - \mathcal{I}_L^{m,g}[u]\|_{L^\infty(\Gamma^N; W(D))} \leq \widehat{C}(\mathbf{r}, N) M_L^{-\mu_1}$$

with $\mu_1 = \frac{\underline{r}(\log(2)e - 1/2)}{\log(2) + \sum_{n=1}^N \underline{r}/g(n)}.$ (21.38)

- Sub-exponential convergence $(L > \frac{\mathcal{R}(N)}{\underline{r} \log(2)})$

$$\|u - \mathcal{I}_L^{m,g}[u]\|_{L^\infty(\Gamma^N; W(D))} \leq \widehat{C}(\mathbf{r}, N) M_L^{\frac{\mu_2}{2}} \exp\left(-\mathcal{R}(N) M_L^{\frac{\log(2)}{\mathcal{R}(N)} \mu_2}\right)$$

with $\mu_2 = \frac{\underline{r}}{\left(\log(2) + \sum_{n=1}^N \underline{r}/g(n)\right)},$ (21.39)

where the constant $\widehat{C}(\mathbf{r}, N)$, defined in [58, (3.14)], is independent of M_L .

Remark 1. The estimate (21.39) may be improved when $L \rightarrow \infty$. Such an asymptotic estimate is obtained using the better counting result described in [58, Remark 3.7].

Remark 2. Observe that sub-exponential rate of convergence is always faster than the algebraic one when $L > \mathcal{R}(N)/(\underline{r} \log(2))$. Yet, this estimate is of little practical relevance since in practical computations, such a high level L is seldom reached.

Remark 3 (On the curse of dimensionality). Suppose that the stochastic input data are truncated expansions of random fields and that the values $\{r_n\}_{n=1}^\infty$ can be estimated. Whenever the sum $\sum_{n=1}^\infty r/r_n$ is finite, the algebraic exponent in (21.38) does not deteriorate as the truncation dimension N increases. This condition is satisfied. This is a clear advantage compared to the isotropic Smolyak method studied in [57] because $r_n \rightarrow +\infty$ is available and $\widehat{C}(\mathbf{r}, N)$ does not deteriorate with N , i.e., it is bounded, and therefore *the method does not suffer from the curse of dimensionality*. In fact, in such a case, the anisotropic Smolyak formula can be extended to infinite dimensions, i.e., $\sum_{n=1}^\infty (l_n - 1)r_n \leq L\underline{r}$.

The condition $\sum_{n=1}^\infty r/r_n < \infty$ is clearly sufficient to break the curse of dimensionality. In that case, even an anisotropic full tensor approximation also breaks the curse of dimensionality.

The algebraic exponent for the convergence of the anisotropic full tensor approximation again deteriorates with the value of $\sum_{n=1}^\infty r/r_n$, but the constant for such convergence is $\sum_{n=1}^N \frac{2}{e^{2r_n} - 1}$. This constant is worse than the one corresponding to the anisotropic Smolyak approximation $\widehat{C}(\mathbf{r}, N)$.

On the other hand, by considering the case where all r_n are equal and the results derived in [57], it can be seen that the algebraic convergence exponent has not be estimated sharply. It is expected that the anisotropic Smolyak method to break the curse of dimensionality for a wider set of problems, i.e., the condition $\sum_{n=1}^\infty r/r_n < \infty$, seems to be unnecessary to break the curse of dimensionality. This is in agreement with Remark 1.

Remark 4 (Optimal choice of weights α). Looking at the exponential term $e^{-h(\mathbf{l}, d)}$, where $h(\mathbf{l}, d) = \sum_{n=1}^N r_n 2^{l_n - 1}$ determining the rate of convergence, the weight α can be chosen as the solution to the optimization problem

$$\max_{\substack{\alpha \in \mathbb{R}_+^N \\ |\alpha|=1}} \min_{\tilde{g}(\mathbf{l}) \leq \alpha L} h(\mathbf{l}, N),$$

where $\tilde{g}(\mathbf{l}) = \sum_{n=1}^N \alpha_n (l_n - 1)$. This problem has the solution $\alpha = \mathbf{r}$, and hence, the choice of weights (21.29) is optimal.

4.3 Nonintrusive Sparse Interpolation in Quasi-optimal Subspaces

The anisotropic sparse grid gSCMs discussed above is an effective approach to alleviate the curse of dimensionality, by adaptively selecting the anisotropic weights based on the size of the analyticity region associated to each random parameter. However, to cover the best M -term polynomial subspace $\mathcal{P}_{\mathcal{J}_M^{\text{opt}}}(\Gamma)$, it is easy to see that the number of degrees of freedom of the sparse grid polynomial subspace $\mathcal{P}_{\mathcal{J}^{m,g}(L)}(\Gamma)$ is much larger than that of the quasi-optimal subspace $\mathcal{P}_{\mathcal{J}_M^{\text{q-opt}}}(\Gamma)$. Therefore, a sparse interpolation approach can be constructed for approximating the solution map $\mathbf{y} \mapsto u_{J_h}(\mathbf{y})$ in the quasi-optimal polynomial subspace $\mathcal{P}_{\mathcal{J}_M^{\text{q-opt}}}(\Gamma)$, by constructing a *nonintrusive* hierarchical interpolant $\mathcal{I}_{\mathcal{J}_M^{\text{q-opt}}}[u]$ on a set of *distinct collocation points* $\mathcal{H}_{\mathcal{J}_M^{\text{q-opt}}}$ given by

$$\mathcal{I}_{\mathcal{J}_M^{\text{q-opt}}}[u] = \sum_{v \in \mathcal{J}_M^{\text{q-opt}}} \bigotimes_{n=1}^N \Delta^{m(v_n)}[u] = \sum_{v \in \mathcal{J}_M^{\text{q-opt}}} \bigotimes_{n=1}^N (\mathcal{U}^{m(v_n)} - \mathcal{U}^{m(v_n-1)})[u] \quad (21.40)$$

and

$$\mathcal{H}_{\mathcal{J}_M^{\text{q-opt}}} = \bigcup_{v \in \mathcal{J}_M^{\text{q-opt}}} \bigotimes_{n=1}^N \{y_{n,k}\}_{k=1}^{m(v_n)}, \quad (21.41)$$

respectively. Here, for $n = 1, \dots, N$, $\mathcal{U}^{m(v_n)} : C^0(\Gamma_n) \rightarrow \mathcal{P}_{m(v_n)-1}(\Gamma_n)$ is a sequence of one-dimensional Lagrange interpolation operators using abscissas $\{y_{n,k}\}_{k=1}^{m(v_n)} \subset \Gamma_n$, with $m(v) : \mathbb{N}_+ \rightarrow \mathbb{N}_+$ a strictly increasing function such that $m(0) = 0$ and $m(v) < m(v+1)$. It is straightforward to construct a multidimensional interpolant that approximates u in the *quasi-optimal* subspace described above, i.e., $\mathcal{I}_{\mathcal{J}_M^{\text{q-opt}}}[u] \in \mathcal{P}_{\mathcal{J}_M^{\text{q-opt}}}(\Gamma)$; however, there are two difficult challenges that must be addressed in order to guarantee that the interpolation error recovers the convergence rate of the quasi-optimal approximation. First, the number of grid points must equal the dimension of the polynomial space, i.e., $\#(\mathcal{H}_{\mathcal{J}_M^{\text{q-opt}}}) = \dim(\mathcal{P}_{\mathcal{J}_M^{\text{q-opt}}}(\Gamma)) = M$, which implies the one-dimensional abscissas must be nested and increase linearly, i.e., $\{y_{n,k}\}_{k=1}^{m(v_n-1)} \subset \{y_{n,k}\}_{k=1}^{m(v_n)}$ for all $n = 1, \dots, N$ and $m(v) = v + 1$. Second, the Lebesgue constant $\mathbb{L}_{\mathcal{J}_M^{\text{q-opt}}}$ must grow slowly with respect to the total number of collocation points, so as to guarantee the accuracy of the interpolation operator $\mathcal{I}_{\mathcal{J}_M^{\text{q-opt}}}$, dictated by the inequality [12, 24, 60]:

$$\begin{aligned} \|u - \mathcal{I}_{\mathcal{J}_M^{\text{q-opt}}}[u]\|_{L^\infty(\Gamma)} &\leq \left(1 + \mathbb{L}_{\mathcal{J}_M^{\text{q-opt}}}\right) \inf_{v \in \mathcal{P}_{\mathcal{J}_M^{\text{q-opt}}}(\Gamma)} \|u - v\|_{L^\infty(\Gamma)} \\ &\leq \left(1 + \mathbb{L}_{\mathcal{J}_M^{\text{q-opt}}}\right) \|u - u_{\mathcal{J}_M^{\text{q-opt}}}\|_{L^\infty(\Gamma)}. \end{aligned} \quad (21.42)$$

Therefore, let $K = m(v_n)$ by suppressing n . Several choices of one-dimensional collocation points $\{y_k\}_{k=1}^K$ can be constructed that satisfy the criteria above. It is worth noting that the Lebesgue constant for the extrema points of the Chebyshev polynomials exhibits slow logarithmic growth, i.e., $\mathcal{O}(\log(K))$; however, evaluating Δ^K requires $2K + 1$ samples, resulting in $\#(\mathcal{H}_{\mathcal{J}_M^{q\text{-opt}}}) \gg M$. Even the nested version of the Chebyshev points, known as the Clenshaw-Curtis abscissas [19, 33, 68], requires $m(l) = 2^{l-1} + 1$, and hence capturing a general polynomial space requires excess interpolation points [41, 57, 58]. Instead, to ensure the sequence of one-dimensional abscissas is nested and grows linearly, greedy search approaches can be exploited, wherein, given the sequence $\{y_k\}_{k=1}^{K-1} = \mathcal{Z}_{K-1}$, the next abscissa is chosen as the extrema of some functional, e.g.,

$$\begin{aligned} \text{(i)} \quad & y_K = \underset{\xi \in \Gamma}{\operatorname{argmax}} \mathcal{R}_{\mathcal{Z}_{K-1}}(\xi), \\ \text{(ii)} \quad & y_K = \underset{\xi \in \Gamma}{\operatorname{argmax}} \mathcal{L}_{\mathcal{Z}_{K-1}}(\xi), \\ \text{(iii)} \quad & y_K = \underset{\xi \in \Gamma}{\operatorname{argmin}} \left[\max_{y \in \Gamma} \mathcal{L}_{\mathcal{Z}_{K-1}, \xi}(y) \right], \end{aligned} \quad (21.43)$$

where (i) is also known as the Leja sequence [9, 17], $\mathcal{R}_{\mathcal{Z}_{K-1}}(\xi) = \prod_{k=1}^{K-1} |\xi - y_k|$ is the residual function, and $\mathcal{L}_{\mathcal{Z}_{K-1}}(\xi)$ is the well-known Lebesgue function [12, 60]. The three separate one-dimensional optimization problems are referred to as (i) ‘‘Leja,’’ (ii) ‘‘Max-Lebesgue,’’ and (iii) ‘‘Min-Lebesgue.’’ Preliminary comparisons of the growth of the Lebesgue constants in $N = 1$ and $N = 2$ dimensions (using the sparse interpolant $\mathcal{I}_{\mathcal{J}_M^{q\text{-opt}}}$ in a total degree polynomial subspace) are given in Fig. 21.5. All three cases exhibit moderate growth of the Lebesgue constant, but theoretical estimates of growth of the Lebesgue constants are still open questions. Finally, it should be noted that the approach of constructing the

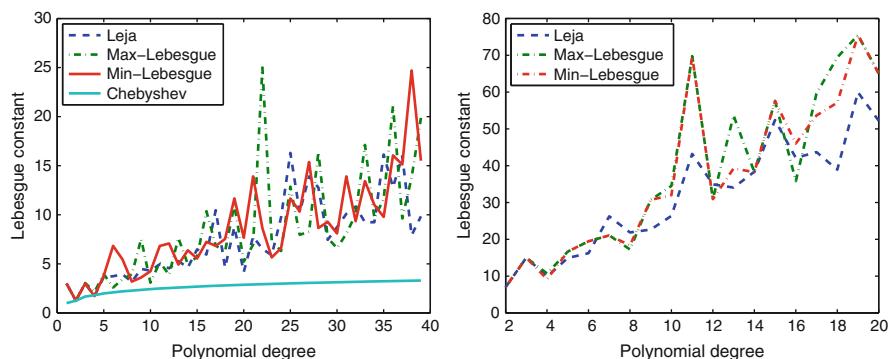


Fig. 21.5 (Left) Comparison between Lebesgue constants for one-dimensional interpolation rules; (right) comparison between Lebesgue constants for two-dimensional interpolation rules constructed by virtue of the quasi-optimal interpolant $\mathcal{I}_{\mathcal{J}_M^{q\text{-opt}}}[\mathbf{u}]$

interpolant $\mathcal{I}_{\mathcal{J}_M^{q,\text{opt}}}[u]$ using sparse tensor products of the one-dimensional abscissas is completely generalizable to N dimensions, whereas extensions of the greedy optimization procedures (i)–(iii) in multidimensions, e.g., the so-called *Magic points* [53], are typically complex, ill-conditioned, and computationally impractical for more than $N = 2$.

5 Local Stochastic Collocation Methods

To realize their high accuracy, the stochastic collocation methods based on the use of global polynomials discussed require high regularity of the solution $u(\mathbf{x}, \mathbf{y})$ with respect to the random parameters $\{y_n\}_{n=1}^N$. They are therefore ineffective for the approximation of solutions that have irregular dependence with respect to those parameters. Motivated by finite element methods (FEMs) for spatial approximation, an alternate and potentially more effective approach for approximating irregular solutions is to use *locally supported piecewise polynomial* approaches for approximating the solution dependence on the random parameters. To achieve greater accuracy, global polynomial approaches increase the polynomial degree; piecewise polynomial approaches instead keep the polynomial degree fixed but refine the grid used to define the approximation space.

5.1 Hierarchical Stochastic Collocation Methods

Several types of one-dimensional piecewise hierarchical polynomial bases [14] are first introduced, which are the foundation of hierarchical sparse grid stochastic collocation methods.

5.1.1 One-Dimensional Piecewise Linear Hierarchical Interpolation

The one-dimensional hat function having support $[-1, 1]$ is defined by

$$\psi(y) = \max\{0, 1 - |y|\}$$

from which an arbitrary hat function with support $(y_{l,i} - \tilde{h}_l, y_{l,i} + \tilde{h}_l)$ can be generated by dilation and translation, i.e.,

$$\psi_{l,i}(y) := \psi\left(\frac{y + 1 - i\tilde{h}_l}{\tilde{h}_l}\right),$$

where l denotes the resolution level, $\tilde{h}_l = 2^{-l+1}$ for $l = 0, 1, \dots$ denotes the grid size of the level l grid for the interval $[-1, 1]$, and $y_{l,i} = i\tilde{h}_l - 1$ for $i = 0, 1, \dots, 2^l$ denotes the grid points of that grid. The basis function $\psi_{l,i}(y)$ has local support and is centered at the grid point $y_{l,i}$; the number of grid points in the level l grid is $2^l + 1$.

With $Z = L^2_\rho(\Gamma)$, a sequence of subspaces $\{Z_l\}_{l=0}^\infty$ of Z of increasing dimension $2^l + 1$ can be defined as

$$Z_l = \text{span}\{\psi_{l,i}(y) \mid i = 0, 1, \dots, 2^l\} \quad \text{for } l = 0, 1, \dots$$

The sequence is dense in Z , i.e., $\cup_{l=0}^\infty Z_l = Z$, and nested, i.e.,

$$Z_0 \subset Z_1 \subset \dots \subset Z_l \subset Z_{l+1} \subset \dots \subset Z.$$

Each of the subspaces $\{Z_l\}_{l=0}^\infty$ is the standard finite element subspace of continuous piecewise linear polynomial functions on $[-1, 1]$ that is defined with respect to the grid having grid size \tilde{h}_l . The set $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ is the standard nodal basis for the space Z_l .

An alternative to the nodal basis $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ for Z_l is a *hierarchical* basis, the construction of which starts with the hierarchical index sets

$$B_l = \{i \in \mathbb{N} \mid i = 1, 3, 5, \dots, 2^l - 1\} \quad \text{for } l = 1, 2, \dots$$

and the sequence of hierarchical subspaces defined by

$$W_l = \text{span}\{\psi_{l,i}(y) \mid i \in B_l\} \quad \text{for } l = 1, 2, \dots$$

Due to the nesting property of $\{Z_l\}_{l=0}^\infty$, it is easy to see that $Z_l = Z_{l-1} \oplus W_l$ and $W_l = Z_l / \bigoplus_{l'=0}^{l-1} Z_{l'}$ for $l = 1, 2, \dots$. Then, the hierarchical subspace splitting of Z_l is given by

$$Z_l = Z_0 \oplus W_1 \oplus \dots \oplus W_l \quad \text{for } l = 1, 2, \dots$$

Then, the *hierarchical basis* for Z_l is given by

$$\{\psi_{0,0}(y), \psi_{0,1}(y)\} \cup \left(\cup_{l'=1}^l \{\psi_{l',i}(y)\}_{i \in B_{l'}} \right). \quad (21.44)$$

It is easy to verify that, for each l , the subspaces spanned by the hierarchical and the nodal basis bases are the same, i.e., they are both bases for Z_l .

The nodal basis $\{\psi_{L,i}(y)\}_{i=0}^{2^L}$ for Z_L possesses the delta property, i.e., $\psi_{L,i}(y_{L,i'}) = \delta_{i,i'}$ for $i, i' \in \{0, \dots, 2^L\}$. The hierarchical basis (21.44) for Z_L possesses only a partial delta property; specifically, the basis functions corresponding to a specific level possess the delta property with respect to its own level and coarser levels, but not with respect to finer levels, i.e., for $l = 0, 1, \dots, L$ and $i \in B_L$,

$$\begin{aligned} \text{for } 0 \leq l' < l, \quad & \psi_{l,i}(y_{l',i'}) = 0 && \text{for all } i' \in B_{l'} \\ \text{for } l' = l, \quad & \psi_{l,i}(y_{l,i'}) = \delta_{i,i'} && \text{for all } i' \in B_{l'} \\ \text{for } l < l' \leq L, \quad & \psi_{l,i}(y_{l',i'}) \neq 0 && \text{for all } i' \in B_{l'} \end{aligned} \quad (21.45)$$

A comparison between the linear hierarchical polynomial basis and the corresponding nodal basis for $L = 3$ is given in Fig. 21.6.

For each grid level l , the interpolant of a function $g(y)$ in the subspace Z_l in terms of its nodal basis $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ is given by

$$\mathcal{I}_l(g(y)) = \sum_{i=0}^{2^l} g(y_{l,i}) \psi_{l,i}(y). \quad (21.46)$$

Due to the nesting property $Z_l = Z_{l-1} \oplus W_l$, it is easy to see that $\mathcal{I}_{l-1}(g) = \mathcal{I}_l(\mathcal{I}_{l-1}(g))$, based on which the incremental interpolation operator is defined by

$$\begin{aligned} \Delta_l(g) &= \mathcal{I}_l(g) - \mathcal{I}_{l-1}(g) = \mathcal{I}_l(g - \mathcal{I}_{l-1}(g)) \\ &= \sum_{i=0}^{2^l} [g(y_{l,i}) - \mathcal{I}_{l-1}(g)(y_{l,i})] \psi_{l,i}(y) \\ &= \sum_{i \in B_l} [g(y_{l,i}) - \mathcal{I}_{l-1}(g)(y_{l,i})] \psi_{l,i}(y) = \sum_{i \in B_l} c_{l,i} \psi_{l,i}(y), \end{aligned} \quad (21.47)$$

where $c_{l,i} = g(y_{l,i}) - \mathcal{I}_{l-1}(g)(y_{l,i})$. Note that $\Delta_l(g)$ only involves the basis functions for W_l for $l \geq 1$. Because $\Delta_l(g)$ essentially approximates the difference between g and the interpolant $\mathcal{I}_{l-1}(g)$ on level $l-1$, the coefficients $\{c_{l,i}\}_{i \in B_l}$ are referred to as the *surpluses* on level l .

The interpolant $\mathcal{I}_l(g)$ for any level $l > 0$ can be decomposed in the form

$$\mathcal{I}_l(g) = \mathcal{I}_{l-1}(g) + \Delta_l(g) = \dots = \mathcal{I}_0(g) + \sum_{l'=1}^l \Delta_{l'}(g). \quad (21.48)$$

The delta property of the nodal basis implies that the interpolation matrix is diagonal. The interpolation matrix for the hierarchical basis is not diagonal, but the partial delta property (21.45) implies that it is triangular so that the coefficients in the interpolant can be solved for explicitly. This also can be seen from the definition (21.47) for $\Delta_l(\cdot)$ and the recursive form of $\mathcal{I}_l(\cdot)$ in (21.48) for which the surpluses can be computed explicitly.

5.1.2 Multidimensional Hierarchical Sparse Grid Interpolation

The interpolation of a multivariate function $g(\mathbf{y})$ is defined, again without loss of generality, over the unit hypercube $\Gamma = [-1, 1]^N \subset \mathbb{R}^N$. The one-dimensional hierarchical polynomial basis (21.44) can be extended to the N -dimensional parameter domain Γ using tensorization. Specifically, the N -variate basis function $\psi_{\mathbf{l},\mathbf{i}}(\mathbf{y})$ associated with the point $\mathbf{y}_{\mathbf{l},\mathbf{i}} = (y_{l_1,i_1}, \dots, y_{l_N,i_N})$ is defined using tensor products, i.e.,

$$\psi_{\mathbf{l},\mathbf{i}}(\mathbf{y}) := \prod_{n=1}^N \psi_{l_n,i_n}(y_n),$$

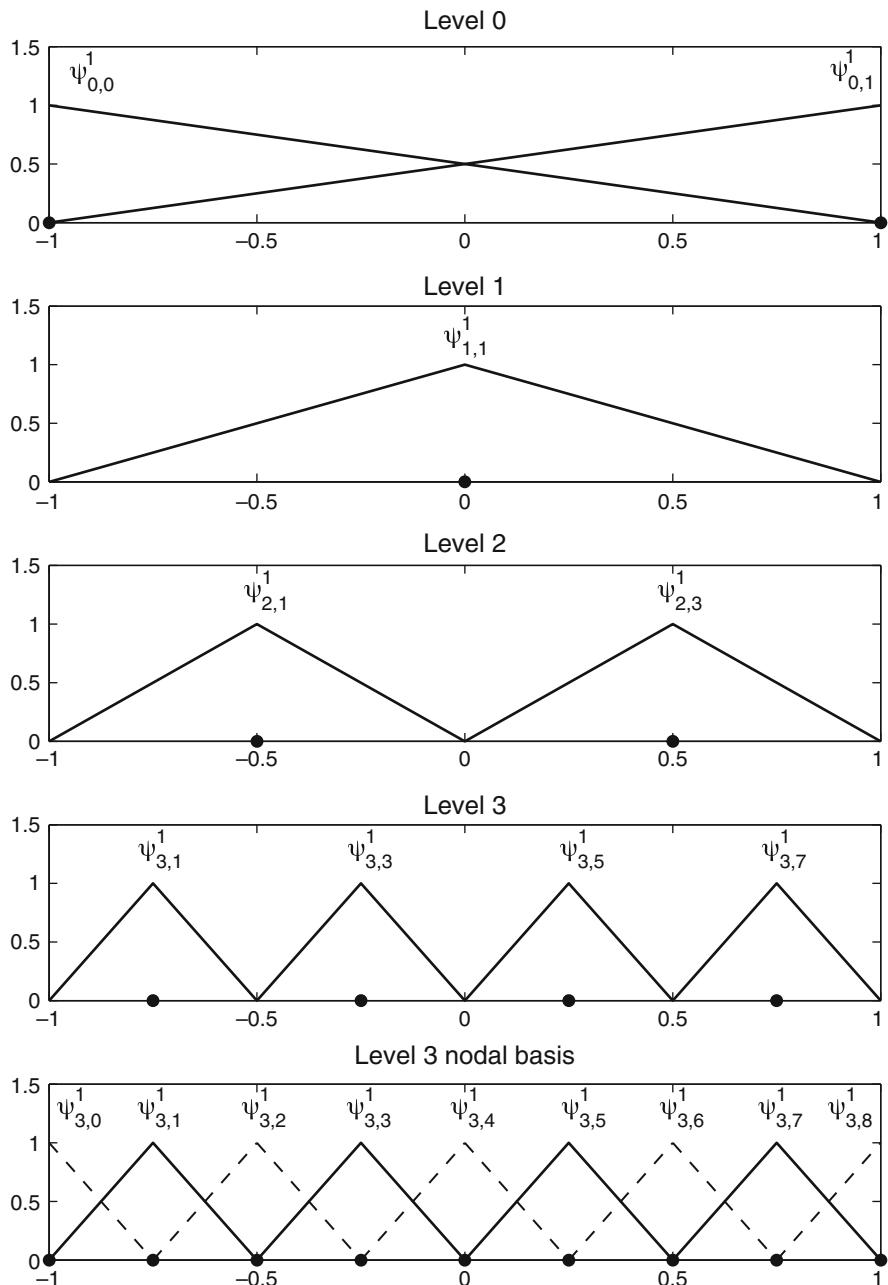


Fig. 21.6 Piecewise linear polynomial bases for $L = 3$. Top four rows: the basis functions for Z_0 , W_1 , W_2 , and W_3 , respectively; the hierarchical basis for Z_3 is the union of the functions in the top four rows. Bottom row: the nodal basis for Z_3

where $\{\psi_{l_n, i_n}(y_n)\}_{n=1}^N$ are the one-dimensional hierarchical polynomials associated with the point $y_{l_n, i_n} = i_n \tilde{h}_{l_n} - 1$ with $\tilde{h}_{l_n} = 2^{-l_n+1}$ and $\mathbf{l} = (l_1, \dots, l_N)$ is a multi-index indicating the resolution level of the basis function. The N -dimensional hierarchical incremental subspace $W_{\mathbf{l}}$ is defined by

$$W_{\mathbf{l}} = \bigotimes_{n=1}^N W_{l_n} = \text{span} \{ \psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}) \mid \mathbf{i} \in B_{\mathbf{l}} \},$$

where the multi-index set $B_{\mathbf{l}}$ is given by

$$B_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^N \mid \begin{array}{ll} i_n \in \{1, 3, 5, \dots, 2^{l_n} - 1\} & \text{for } n = 1, \dots, N \quad \text{if } l_n > 0 \\ i_n \in \{0, 1\} & \text{for } n = 1, \dots, N \quad \text{if } l_n = 0 \end{array} \right\}.$$

Similar to the one-dimensional case, a sequence of subspaces, again denoted by $\{Z_l\}_{l=0}^{\infty}$, of the space $Z := L^2(\Gamma)$, can be constructed as

$$Z_l = \bigoplus_{l'=0}^l W_{l'} = \bigoplus_{l'=0}^l \bigoplus_{\alpha(\mathbf{l}')=l'} W_{l'},$$

where the key is how the mapping $\alpha(\mathbf{l})$ is defined because it defines the incremental subspaces $W_{l'} = \bigoplus_{\alpha(\mathbf{l}')=l'} W_{l'}$. For example, $\alpha(\mathbf{l}) = \max_{n=1, \dots, N} l_n$ leads to a full tensor-product space, whereas $\alpha(\mathbf{l}) = |\mathbf{l}| = l_1 + \dots + l_N$ leads to a sparse polynomial space. As the full tensor-product space especially suffers from the curse of dimensionality as N increases, this choice is not feasible for even moderately high-dimensional problems. Thus, it is only considered the case that the sparse polynomial space obtained by setting $\alpha(\mathbf{l}) = |\mathbf{l}|$.

The level l hierarchical sparse grid interpolant of the multivariate function $g(\mathbf{y})$ is then given by

$$\begin{aligned} g_l(\mathbf{y}) &:= \sum_{l'=0}^l \sum_{|\mathbf{l}'|=l'} (\Delta_{l'_1} \otimes \dots \otimes \Delta_{l'_N}) g(\mathbf{y}) \\ &= g_{l-1}(\mathbf{y}) + \sum_{|\mathbf{l}'|=l} (\Delta_{l'_1} \otimes \dots \otimes \Delta_{l'_N}) g(\mathbf{y}) \\ &= g_{l-1}(\mathbf{y}) + \sum_{|\mathbf{l}'|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}'}} [g(\mathbf{y}_{\mathbf{l}', \mathbf{i}}) - g_{l'-1}(\mathbf{y}_{\mathbf{l}', \mathbf{i}})] \psi_{\mathbf{l}', \mathbf{i}}(\mathbf{y}) \\ &= g_{l-1}(\mathbf{y}) + \sum_{|\mathbf{l}'|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}'}} c_{\mathbf{l}', \mathbf{i}} \psi_{\mathbf{l}', \mathbf{i}}(\mathbf{y}), \end{aligned} \tag{21.49}$$

where $c_{\mathbf{l}', \mathbf{i}} = g(\mathbf{y}_{\mathbf{l}', \mathbf{i}}) - g_{l'-1}(\mathbf{y}_{\mathbf{l}', \mathbf{i}})$ is the multidimensional hierarchical surplus. This interpolant is a direct extension, via the Smolyak algorithm, of the one-dimensional

hierarchical interpolant. Analogous to (21.47), the definition of the surplus $c_{\mathbf{l}', \mathbf{i}}$ is based on the facts that $g_l(g_{l-1}(\mathbf{y})) = g_{l-1}(\mathbf{y})$ and $g_{l-1}(\mathbf{y}_{\mathbf{l}', \mathbf{i}}) - g(\mathbf{y}_{\mathbf{l}', \mathbf{i}}) = 0$ for $|\mathbf{l}'| = l$. In this case, $\mathcal{H}_{\mathbf{l}}(\Gamma) = \{\mathbf{y}_{\mathbf{l}, \mathbf{i}} \mid \mathbf{i} \in B_{\mathbf{l}}\}$ denotes the set of sparse grid points corresponding to subspace $W_{\mathbf{l}}$. Then, the sparse grid corresponding to the interpolant g_l is given by

$$\mathcal{H}_l^N(\Gamma) = \cup_{l'=0}^l \cup_{|\mathbf{l}'|=l'} \mathcal{H}_{\mathbf{l}'}(\Gamma),$$

and $\mathcal{H}_l^N(\Gamma)$ is also nested, i.e., $\mathcal{H}_{l-1}^N(\Gamma) \subset \mathcal{H}_l^N(\Gamma)$. In Fig. 21.7, the structure of a level $l = 2$ sparse grid is plotted in $N = 2$ dimensions, without consideration of boundary points. The left nine sub-grids $\mathcal{H}_{\mathbf{l}'}(\Gamma)$ correspond to the nine multi-index sets $B_{\mathbf{l}'}$, where

$$\mathbf{l}' \in \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)\}.$$

The level $l = 2$ sparse grid $\mathcal{H}_2^2(\Gamma)$ shown on the right-top includes only six of the nine sub-grids, with the three sub-grids depicted in gray not included because they fail the criterion $|\mathbf{l}'| \leq l' = 2$. Moreover, due to the nesting property of the

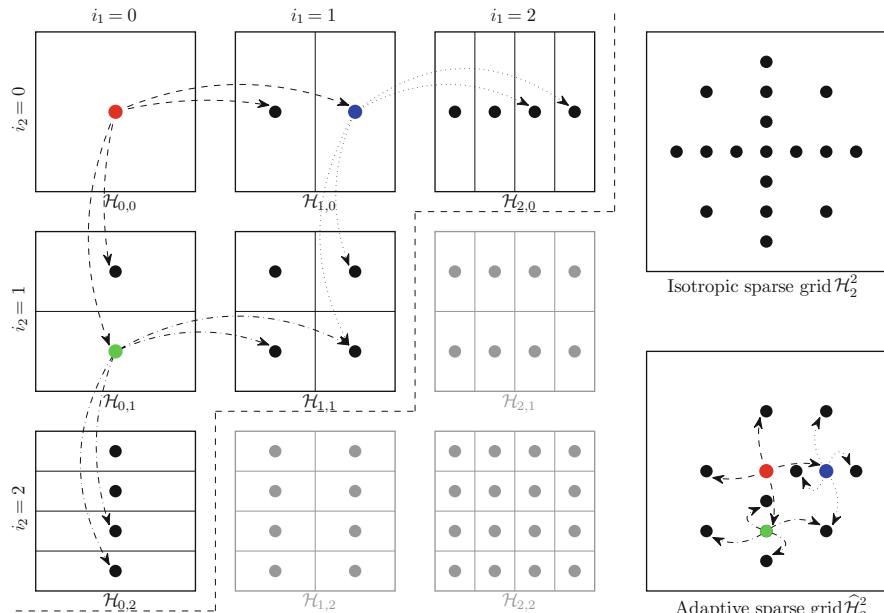


Fig. 21.7 Nine tensor-product sub-grids (left) for level $l = 0, 1, 2$ of which only the 6 sub-grids for which $l'_1 + l'_2 \leq l = 2$ are chosen to appear in the level $l = 2$ isotropic sparse grid $\mathcal{H}_2^2(\Gamma)$ (right-top) containing 17 points. With adaptivity, only points that correspond to a large surplus, i.e., the points in red, blue, and green, lead to 2 children points added in each direction resulting in the adaptive sparse grid $\tilde{\mathcal{H}}_2^2(\Gamma)$ (right-bottom) containing 12 points

hierarchical basis, $\mathcal{H}_2^2(\Gamma)$ has only 17 points, as opposed to the 49 points of the full tensor-product grid.

5.1.3 Hierarchical Sparse Grid Stochastic Collocation

Now the hierarchical sparse grid interpolation is used to approximate the parameter dependence of the solution $u(\mathbf{x}, \mathbf{y})$ of an SPDE. Specifically, the basis $\{\psi_m(\mathbf{y})\}_{m=1}^M$ entering into the fully discrete approximation (21.10) is chosen to be the hierarchical basis. In this case, the fully discrete approximate solution takes the form

$$u_{J_h M_L}(\mathbf{x}, \mathbf{y}) = \sum_{l=0}^L \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_l} c_{\mathbf{l}, \mathbf{i}}(\mathbf{x}) \psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}), \quad (21.50)$$

where now the coefficients are functions of \mathbf{x} to reflect that dependence of the function $u_{J_h M_L}(\mathbf{x}, \mathbf{y})$. In the usual manner, those coefficients are given in terms of the spatial finite element basis $\{\phi_j(\mathbf{x})\}_{j=1}^{J_h}$ by $c_{\mathbf{l}, \mathbf{i}}(\mathbf{x}) = \sum_{j=1}^{J_h} c_{j, \mathbf{l}, \mathbf{i}} \phi_j(\mathbf{x})$ so that, from (21.50), it can be obtained that

$$\begin{aligned} u_{J_h M_L}(\mathbf{x}, \mathbf{y}) &= \sum_{l=0}^L \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_l} \left(\sum_{j=1}^{J_h} c_{j, \mathbf{l}, \mathbf{i}} \phi_j(\mathbf{x}) \right) \psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}) \\ &= \sum_{j=1}^{J_h} \left(\sum_{l=0}^L \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_l} c_{j, \mathbf{l}, \mathbf{i}} \psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}) \right) \phi_j(\mathbf{x}). \end{aligned} \quad (21.51)$$

The number of parameter degrees of freedom M_L of $u_{J_h M_L}$ is equal to the number of the grid points of the sparse grid $\mathcal{H}_L^N(\Gamma)$.

The next step is to introduce how the coefficients $c_{j, \mathbf{l}, \mathbf{i}}$ in (21.51) are determined. In general, after running the deterministic FEM solver for all the sparse grid points, the dataset are obtained

$$u_h(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}, \mathbf{i}}) \quad \text{for } j = 1, \dots, J_h \text{ and } |\mathbf{l}| \leq L, \mathbf{i} \in B_l.$$

Then, it is easy to see from (21.51) that, for fixed j , $\{c_{j, \mathbf{l}, \mathbf{i}}\}_{|\mathbf{l}| \leq L, \mathbf{i} \in B_l}$ can be obtained by solving the linear system

$$\begin{aligned} u_{J_h M_L}(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}', \mathbf{i}'}) &= \sum_{l=0}^L \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_l} c_{j, \mathbf{l}, \mathbf{i}} \psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}_{\mathbf{l}', \mathbf{i}'}) \\ &= u_{N_h}(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}', \mathbf{i}'}) \quad \text{for } |\mathbf{l}'| \leq L, \mathbf{i}' \in B_l. \end{aligned} \quad (21.52)$$

Thus, the approximation $u_{J_h M_L}(\mathbf{x}, \mathbf{y})$ can be obtained by solving J_h linear systems. However, because the hierarchical bases $\psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y})$ satisfies $\psi_{\mathbf{l}, \mathbf{i}}(\mathbf{y}_{\mathbf{l}', \mathbf{i}'}) = 0$ if $l' \leq l$ (this is a consequence of the one-dimensional partialdelta property), the coefficient

$c_{j,\mathbf{l}',\mathbf{i}'}$ in the system (21.52) corresponding to the sparse grid point $\mathbf{y}_{\mathbf{l}',\mathbf{i}'}$ on level L , i.e., for $|\mathbf{l}'| = L$, reduces to

$$\begin{aligned} c_{j,\mathbf{l}',\mathbf{i}'} &= u_{N_h}(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}',\mathbf{i}'}) - \sum_{l=0}^{L-1} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_l} c_{j,\mathbf{l},\mathbf{i}} \psi_{\mathbf{l},\mathbf{i}}(\mathbf{y}_{\mathbf{l}',\mathbf{i}'}) \\ &= u_{N_h}(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}',\mathbf{i}'}) - u_{J_h M_{L-1}}(\mathbf{x}_j, \mathbf{y}_{\mathbf{l}',\mathbf{i}'}), \end{aligned} \quad (21.53)$$

so that linear system becomes a triangular system and all the coefficients can be computed explicitly by recursively using the (21.53). Note that (21.53) is consistent with the definition of the $c_{\mathbf{l},\mathbf{i}}(\mathbf{x})$ given in (21.49).

5.2 Adaptive Hierarchical Stochastic Collocation Methods

By virtue of the hierarchical surpluses $c_{j,\mathbf{l},\mathbf{i}}$, the approximation in (21.51) can be represented in a hierarchical manner, i.e.,

$$u_{J_h M_L}(\mathbf{x}, \mathbf{y}) = u_{J_h M_{L-1}}(\mathbf{x}, \mathbf{y}) + \Delta u_{J_h M_L}(\mathbf{x}, \mathbf{y}), \quad (21.54)$$

where $u_{J_h M_{L-1}}(\mathbf{x}, \mathbf{y})$ is the sparse grid approximation in Z_{L-1} and $\Delta u_{J_h M_L}(\mathbf{x}, \mathbf{y})$ is the hierarchical surplus interpolant in the subspace W_L . According to the analysis in [14], for smooth functions, the surpluses $c_{j,\mathbf{l},\mathbf{i}}$ of the sparse grid interpolant $u_{J_h M_L}$ in (21.51) tend to zero as the interpolation level l goes to infinity. For example, in the context of using piecewise linear hierarchical bases and assuming the spatial approximation $u_{N_h}(\mathbf{x}, \mathbf{y})$ of the solution has bounded second-order weak derivatives with respect to \mathbf{y} , i.e., $u_{N_h}(\mathbf{x}, \mathbf{y}) \in W_h(D) \otimes H_\rho^2(\Gamma)$, then the surplus $c_{j,\mathbf{l},\mathbf{i}}$ can be bounded as

$$|c_{j,\mathbf{l},\mathbf{i}}| \leq C 2^{-2|\mathbf{l}|} \quad \text{for } \mathbf{i} \in B_l \text{ and } j = 1, \dots, J_h, \quad (21.55)$$

where the constant C is independent of the level l . Furthermore, the smoother the target function is, the faster the surplus decays. This provides a good avenue for constructing adaptive sparse grid interpolants using the magnitude of the surplus as an error indicator, especially for irregular functions having, e.g., steep slopes or jump discontinuities.

The construction of one-dimensional adaptive grids is first introduced, and then such strategy for adaptivity will be extended to multidimensional sparse grids. As shown in Fig. 21.8, the one-dimensional hierarchical grid points have a treelike structure. In general, a grid point $y_{l,i}$ on level l has two children, namely, $y_{l+1,2i-1}$ and $y_{l+1,2i+1}$ on level $l+1$. Special treatment is required when moving from level 0 to level 1, where only one single child $y_{1,1}$ is added on level 1. On each successive interpolation level, the basic idea of adaptivity is to use the hierarchical surplus as an error indicator to detect the smoothness of the target function and refine

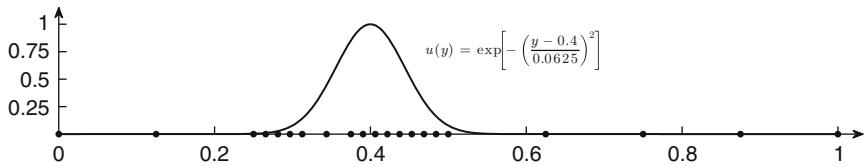
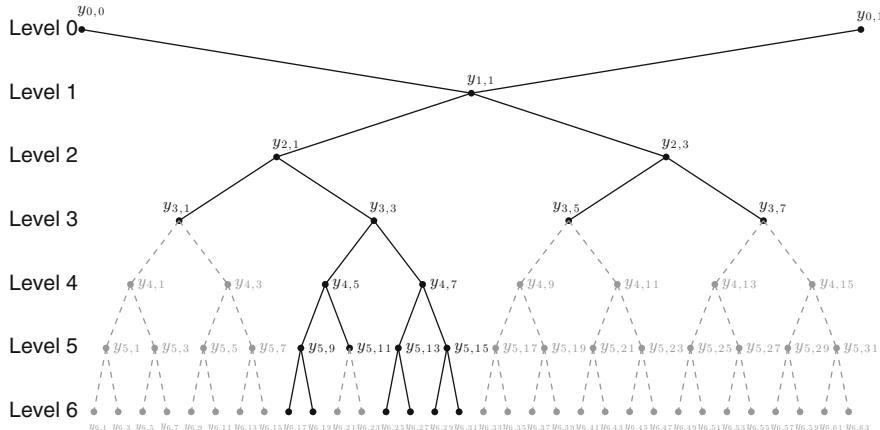


Fig. 21.8 A 6-level adaptive sparse grid for interpolating the one-dimensional function $g(y) = \exp[-(y - 0.4)^2 / 0.0625^2]$ on $[0, 1]$ with the error tolerance of 0.01. The resulting adaptive sparse grid has only 21 points (the black points), whereas the full grid has 65 points (the black and gray points)

the grid by adding two new points on the next level for each point for which the magnitude of the surplus is larger than the prescribed error tolerance. For example, in Fig. 21.8, the 6-level adaptive grid is illustrated for interpolating the function $g(y) = \exp[-(y - 0.4)^2 / 0.0625^2]$ on $[0, 1]$ with error tolerance 0.01. From level 0 to level 2, because the magnitude of every surplus is larger than 0.01, two points are added for each grid point on levels 0 and 2; as mentioned above, only one point is added for each grid point on level 1. However, on level 3, there is only 1 point, namely, $y_{3,3}$, whose surplus has a magnitude larger than 0.01, so only two new points are added on level 4. After adding levels 5 and 6, it ends up with the 6-level adaptive grid with only 21 points (points in black in Fig. 21.8), whereas the 6-level nonadaptive grid has a total of 65 points (points in black and gray in Fig. 21.8).

It is trivial to extend this adaptive approach from one-dimension to a multidimensional adaptive sparse grid. In general, as shown in Fig. 21.7, in N -dimensions a grid point has $2N$ children which are also its neighbor points. However, note that the children of a parent point correspond to hierarchical basis functions on the next interpolation level, so that the interpolant $u_{J_h M_L}$ in (21.51) can be built from level $L - 1$ to level L by only adding those points on level L whose parents have surpluses greater than the prescribed tolerance. At each sparse grid point $\mathbf{y}_{l,i}$,

the error indicator is set to the maximum magnitude of the j surpluses, i.e., to $\max_{j=1,\dots,J_h} |c_{j,l,i}|$. In this way, the sparse grid can be refined locally, resulting in an adaptive sparse grid which is a sub-grid of the corresponding isotropic sparse grid, as illustrated by the right-bottom plot in Fig. 21.7. The solution of the corresponding adaptive hSGSC approach is represented by

$$u_{J_h M_L}^\varepsilon(\mathbf{x}, \mathbf{y}) = \sum_{l=0}^L \sum_{|\mathbf{i}|=l} \sum_{\mathbf{i} \in B_1^\varepsilon} \left(\sum_{j=1}^{J_h} c_{j,l,i} \phi_j(\mathbf{x}) \right) \psi_{l,i}(\mathbf{y}), \quad (21.56)$$

where the multi-index set $B_1^\varepsilon \subset B_1$ is defined by

$$B_1^\varepsilon = \left\{ \mathbf{i} \in B_1 \mid \max_{j=1,\dots,J_h} |c_{j,l,i}| \geq \varepsilon \right\}.$$

Note that B_1^ε is an optimal multi-index set that contains only the indices of the basis functions with surplus magnitudes larger than the tolerance ε . However, in practice, the deterministic FEM solver needs to be executed at a certain number of grid points $y_{l,i}$ with $\max_{j=1,\dots,J_h} |c_{j,l,i}| < \varepsilon$ in order to detect when mesh refinement can stop. For example, in Fig. 21.8, the points $y_{3,1}$, $y_{3,5}$, $y_{3,7}$, and $y_{5,11}$ are this type of points. In this case, the numbers of degrees of freedom in (21.56) is usually smaller than the necessary number of executions of the deterministic FEM solver.

5.2.1 Other Choices of Hierarchical Basis

High-Order Hierarchical Polynomial Basis

One can generalize the piecewise linear hierarchical polynomials to high-order hierarchical polynomials [14]. The goal is to construct polynomial basis functions of order p , denoted by $\psi_{l,i}^p(y)$, without enlarging the support $[y_{l,i} - \tilde{h}_l, y_{l,i} + \tilde{h}_l]$ or increasing the degrees of freedom in the support. As shown in Fig. 21.6, for $l \geq 0$, a piecewise linear polynomial $\psi_{l,i}(y)$ is defined based on 3 supporting points, i.e., $\tilde{y}_{l,i}$ and its two ancestors that are also the endpoints of the support $[y_{l,i} - \tilde{h}_l, y_{l,i} + \tilde{h}_l]$. For $p \geq 2$, it is well known that $p+1$ supporting points are needed to define a Lagrange interpolating polynomial of order p . To achieve the goal, at each grid point $y_{l,i}$, additional ancestors outside of $[y_{l,i} - \tilde{h}_l, y_{l,i} + \tilde{h}_l]$ are borrowed to help build a higher-order Lagrange polynomial; then, the desired polynomial $\psi_{l,i}^p(y)$ is defined by restricting the resulting polynomial to the support $[y_{l,i} - \tilde{h}_l, y_{l,i} + \tilde{h}_l]$. The constructions of the cubic polynomial $\psi_{2,3}^3(y)$ and the quartic polynomial $\psi_{3,1}^4(y)$ are illustrated in Fig. 21.9 (right). For the cubic polynomial associated with $y_{2,3}$, the additional ancestor $y_{0,0}$ is needed to define a cubic Lagrange polynomial; for the quartic polynomial associated with $y_{3,1}$, two more ancestors $y_{1,1}$ and $y_{0,1}$ are added. After the construction of the cubic and quartic polynomials, the part within the support (solid curves) is retained, and the parts outside the support (dashed curves) are cut off. Using this strategy, high-order bases can be constructed, while

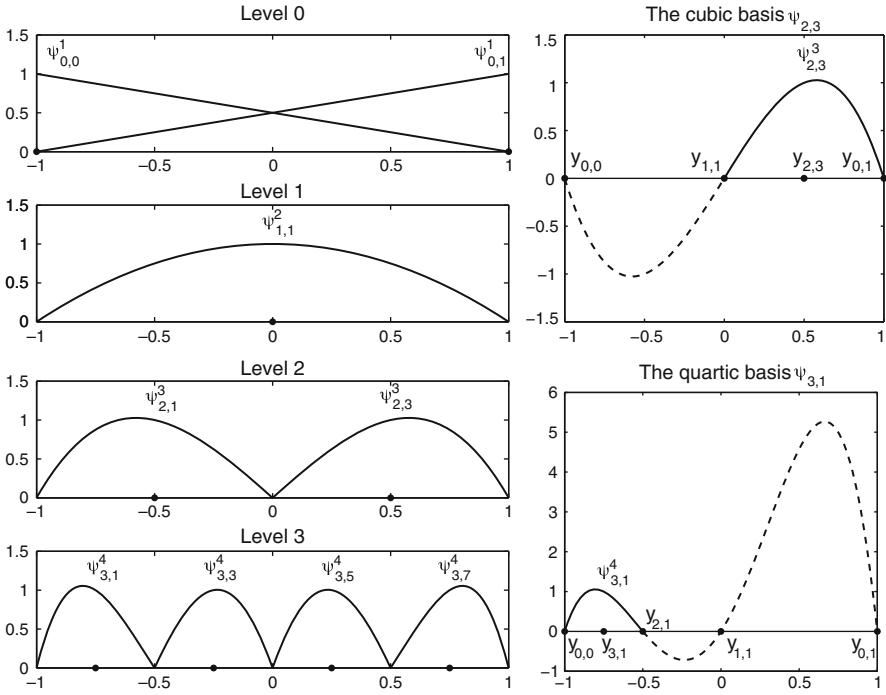


Fig. 21.9 Left: quartic hierarchical basis functions, where linear, quadratic, and cubic basis functions are used on levels 0, 1, and 2, respectively. Quartic basis functions appear beginning with level 3. Right: the construction of a cubic hierarchical basis function and a quartic hierarchical basis function, respectively

the hierarchical structure can be retained. It should be noted that because a total of p ancestors are needed, a polynomial of order p cannot be defined earlier than level $p - 1$. In other words, at level L , the maximum order of polynomials is $p = L + 1$. For example, a quartic polynomial basis of level 3 is plotted in Fig. 21.9 (left) where linear, quadratic, and cubic polynomials are used on levels 0, 1, and 2 due to the lack of ancestors. It is observed that there are multiple types of basis functions on each level when $p \geq 3$ because of the different distributions of supporting points for different grid points. In general, the hierarchical basis of order $p > 1$ contains 2^{p-2} types of p -th order polynomials. In Table 21.1, the supporting points used to define the hierarchical polynomial bases of order $p = 2, 3, 4$ are listed.

Wavelet Basis

Besides the hierarchical bases discussed above, wavelets form another important family of basis functions which can provide a stable subspace splitting because of their Riesz property. The second-generation wavelets, constructed using the lifting scheme discussed in [63, 64], will be briefly introduced in the following. Second-generation wavelets are a generalization of biorthogonal wavelets that are easier

Table 21.1 Supporting points for high-order hierarchical bases ($p = 2, 3, 4$)

Order	Grid point $y_{l,i}$	Supporting points of $\psi_{l,i}^p(y)$
$p = 2$	$l \geq 1, \text{ mod } (i, 2) = 1$	$y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l$
	$l \geq 2, \text{ mod } (i, 4) = 1$	$y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l, y_{l,i} + 3\tilde{h}_l$
$p = 3$	$l \geq 2, \text{ mod } (i, 4) = 3$	$y_{l,i} - 3\tilde{h}_l, y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l$
	$l \geq 3, \text{ mod } (i, 8) = 1$	$y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l, y_{l,i} + 3\tilde{h}_l, y_{l,i} + 7\tilde{h}_l$
	$l \geq 3, \text{ mod } (i, 8) = 3$	$y_{l,i} - 3\tilde{h}_l, y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l, y_{l,i} + 5\tilde{h}_l$
$p = 4$	$l \geq 3, \text{ mod } (i, 8) = 5$	$y_{l,i} - 5\tilde{h}_l, y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l, y_{l,i} + 3\tilde{h}_l$
	$l \geq 3, \text{ mod } (i, 8) = 7$	$y_{l,i} - 7\tilde{h}_l, y_{l,i} - 3\tilde{h}_l, y_{l,i} - \tilde{h}_l, y_{l,i}, y_{l,i} + \tilde{h}_l$

to apply for functions defined on bounded domains. The lifting scheme [63, 64] is a tool for constructing second-generation wavelets that are no longer dilates and translates of a single scaling function. The basic idea behind lifting is to start with simple multi-resolution analysis and gradually build a multi-resolution analysis with specific, a priori defined properties. The lifting scheme can be viewed as a process of taking an existing wavelet and modifying it by adding linear combinations of the scaling function at the coarse level. In the context of the piecewise linear basis, the second-generation wavelet on level $l \geq 1$, denoted by $\psi_{l,i}^w(y)$, is constructed by “lifting” the piecewise linear basis $\psi_{l,i}(y)$ as

$$\psi_{l,i}^w(y) := \psi_{l,i}(y) + \sum_{i'=0}^{2^l-1} \beta_{l,i}^{i'} \psi_{l-1,i'}(y),$$

where, for $i = 0, \dots, 2^l-1$, $\psi_{l-1,i}(y)$ are the nodal polynomials on level $l-1$ and the weights $\beta_{l,i}^{i'}$ in the linear combination are chosen in such a way that the wavelet $\psi_{l,i}^w(y)$ has more vanishing moments than $\psi_{l,i}(y)$ and thus provides a stabilization effect. Specifically, in the bounded domain $[-1, 1]$, there are three types of linear lifting wavelets:

$$\begin{aligned} \psi_{l,i}^w &:= \psi_{l,i} - \frac{1}{4}\psi_{l-1,\frac{i-1}{2}} - \frac{1}{4}\psi_{l-1,\frac{i+1}{2}} & \text{for } 1 < i < 2^l - 1, \text{ } i \text{ odd} \\ \psi_{l,i}^w &:= \psi_{l,i} - \frac{3}{4}\psi_{l-1,\frac{i-1}{2}} - \frac{1}{8}\psi_{l-1,\frac{i+1}{2}} & \text{for } i = 1 \\ \psi_{l,i}^w &:= \psi_{l,i} - \frac{1}{8}\psi_{l-1,\frac{i-1}{2}} - \frac{3}{4}\psi_{l-1,\frac{i+1}{2}} & \text{for } i = 2^l - 1, \end{aligned} \quad (21.57)$$

where the three equations define the central “mother” wavelet, the left-boundary wavelet, and the right-boundary wavelet, respectively. The three lifting wavelets are plotted in Fig. 21.10. For additional details, see [63].

Note that the property given in (21.45) is not valid for the lifting wavelets in (21.57) because neighboring wavelets at the same level have overlapping support. As a result, the coefficient matrix of the linear system (21.52) is no longer triangular.

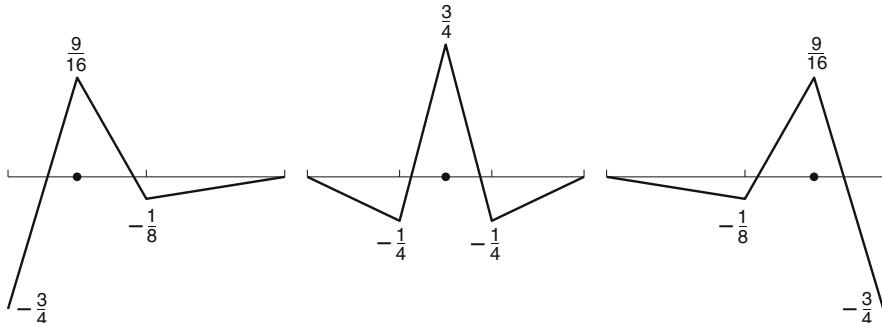


Fig. 21.10 Left-boundary wavelet (*left*), central wavelet (*middle*), right-boundary wavelet (*right*)

Thus, J_h linear systems, each of size $M_L \times M_L$, need be solved to obtain the surpluses in (21.51). However, note that for the second-generation wavelet defined in (21.57), the interpolation matrix is well conditioned. See [40] for details.

6 Conclusion

This chapter provides an overview of the stochastic collocation methods for PDEs with random input data. To alleviate the curse of dimensionality, sparse global polynomial subspaces and local hierarchical polynomial subspaces are incorporated into the framework of SCMs. By exploiting the inherent regularity of PDE solutions with respect to random parameters, the global SCMs can essentially match the fast convergence of the intrusive stochastic Galerkin methods and retain the nonintrusive nature that leads to massively parallel implementation. There are a variety of global sparse polynomial subspaces for the SCMs as alternatives to the standard isotropic full tensor-product space, in order to obtain better approximations to the best M -term polynomial subspace. For instance, the generalized sparse grid SCMs can efficiently approximate PDE solutions with anisotropic behavior by exploiting the size of the analyticity region associated to each random parameter and assigning an appropriate weight to each stochastic dimension. The nonintrusive sparse interpolation in quasi-optimal subspaces can provide more accurate approximations to the best M -term polynomial expansion by building the subspaces based on sharp upper bounds of the coefficients of the polynomial expansion of the PDE solutions.

In addition, there are many works on reducing complexity of implementing SCMs. For example, in [67], a multilevel version of the stochastic collocation method was proposed, which uses hierarchies of spatial approximations to reduce the overall computational complexity. In addition, this approach utilizes, for approximation in stochastic space, a sequence of multidimensional interpolants of increasing fidelity which can then be used for approximating statistics of the solution. With the use of interpolating polynomials, the multilevel SCM [39] can provide us with high-order surrogates featuring faster convergence rates, compared

to standard single-level SCMs as well as multilevel Monte Carlo methods. In [31], an acceleration technique was proposed to reduce the computational burden of hierarchical SCMs. Similar to the way multilevel methods take advantage of hierarchies of spatial approximation to reduce computational cost, our approach exploits the hierarchical structure of the sparse grid construction to seed the linear or nonlinear iterative solvers with improved initial guesses. Specifically, at each newly added sample point on the current level of a sparse grid, the solution of the SPDE is predicted using the sparse grid interpolant on the previous level, and then that prediction is used as the starting point of the chosen iterative solver. This approach can be applied to all the SCMs discussed in this effort, as well as to other hierarchically structured nonintrusive methods, e.g., nonintrusive generalized polynomial chaos, discrete least squares projection, multilevel methods, etc.

References

1. Babuska, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **45**, 1005–1034 (2007)
2. Babuška, I.M., Tempone, R., Zouraris, G.E.: Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.* **42**, 800–825 (2004) (electronic)
3. Babuška, I.M., Tempone, R., Zouraris, G.E.: Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Eng.* **194**, 1251–1294 (2005)
4. Barth, A., Lang, A., Schwab, C.: Multilevel Monte Carlo method for parabolic stochastic partial differential equations. *BIT Numer. Math.* **53**, 3–27 (2013)
5. Beck, J., Nobile, F., Tamellini, L., Tempone, R.: Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. *Lect. Notes Comput. Sci. Eng.* **76**, 43–62 (2011)
6. Beck, J., Nobile, F., Tamellini, L., Tempone, R.: Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients. *Comput. Math. Appl.* **67**, 732–751 (2014)
7. Beck, J., Tempone, R., Nobile, F., Tamellini, L.: On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. *Math. Models Methods Appl. Sci.* **22**, 1250023 (2012)
8. Beck, M., Robins, S.: Computing the Continuous Discretely: Integer-Point Enumeration in Polyhedra. Springer, New York (2007)
9. Białas-Cież, L., Calvi, J.-P.: Pseudo Leja sequences. *Annali di Matematica Pura ed Applicata* **191**, 53–75 (2012)
10. Bieri, M., Andreev, R., Schwab, C.: Sparse tensor discretization of elliptic sPDEs. *SIAM J. Sci. Comput.* **31**, 4281–4304 (2009)
11. Brenner, S.C., Scott, L.R.: The Mathematical Theory of Finite Element Methods. Springer, New York (1994)
12. Brutman, L.: On the Lebesgue function for polynomial interpolation. *SIAM J. Numer. Anal.* **15**, 694–704 (1978)
13. Buffa, A., Maday, Y., Patera, A., Prud'homme, C., Turinici, G.: A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Math. Model. Numer. Anal.* **46**, 595–603 (2012)
14. Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numer.* **13**, 1–123 (2004)

15. Chkifa, A., Cohen, A., DeVore, R., Schwab, C.: Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs. *Modél. Math. Anal. Numér.* **47**, 253–280 (2013)
16. Chkifa, A., Cohen, A., Schwab, C.: Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *J. Math. Pures Appl.* **103**, 400–428 (2015)
17. Chkifa, M.A.: On the Lebesgue constant of Leja sequences for the complex unit disk and of their real projection. *J. Approx. Theory* **166**, 176–200 (2013)
18. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. North-Holland, New York (1978)
19. Clenshaw, C.W., Curtis, A.R.: A method for numerical integration on an automatic computer. *Numer. Math.* **2**, 197–205 (1960)
20. Cliffe, K.A., Giles, M.B., Scheichl, R., Teckentrup, A.L.: Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.* **14**, 3–15 (2011)
21. Cohen, A., DeVore, R., Schwab, C.: Convergence rates of best n-term Galerkin approximations for a class of elliptic SPDEs. *Found. Comput. Math.* **10**, 615–646 (2010)
22. Cohen, A., DeVore, R., Schwab, C.: Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. *Anal. Appl.* **9**, 11–47 (2011)
23. DeVore, R.: Nonlinear approximation. *Acta Numer.* **7**, 51–150 (1998)
24. DeVore, R.A., Lorentz, G.G.: Constructive approximation. Volume 303 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer, Berlin (1993)
25. Dexter, N., Webster, C., Zhang, G.: Explicit cost bounds of stochastic Galerkin approximations for parameterized PDEs with random coefficients. ArXiv:1507.05545 (2015)
26. Dzjadyk, V.K., Ivanov, V.V.: On asymptotics and estimates for the uniform norms of the Lagrange interpolation polynomials corresponding to the Chebyshev nodal points. *Anal. Math.* **9**, 85–97 (1983)
27. Elman, H., Miller, C.: Stochastic collocation with kernel density estimation. Tech. Rep., Department of Computer Science, University of Maryland (2011)
28. Elman, H.C., Miller, C.W., Phipps, E.T., Tuminaro, R.S.: Assessment of collocation and Galerkin approaches to linear diffusion equations with random data. *Int. J. Uncertain. Quantif.* **1**, 19–33 (2011)
29. Fishman, G.: Monte Carlo. Springer Series in Operations Research. Springer, New York (1996)
30. Frauenfelder, P., Schwab, C., Todor, R.A.: Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Eng.* **194**, 205–228 (2005)
31. Galindo, D., Jantsch, P., Webster, C.G., Zhang, G.: Accelerating stochastic collocation methods for partial differential equations with random input data. Tech. Rep. ORNL/TM-2015/219, Oak Ridge National Laboratory (2015)
32. Ganapathysubramanian, B., Zabaras, N.: Sparse grid collocation schemes for stochastic natural convection problems. *J. Comput. Phys.* **225**, 652–685 (2007)
33. Gentleman, W.M.: Implementing Clenshaw-Curtis quadrature, II computing the cosine transformation. *Commun. ACM* **15**, 343–346 (1972)
34. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms* **18**, 209–232 (1998)
35. Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach. Springer, New York (1991)
36. Giles, M.B.: Multilevel Monte Carlo path simulation. *Oper. Res.* **56**, 607–617 (2008)
37. Griebel, M.: Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing* **61**, 151–179 (1998)
38. Gruber, P.: Convex and Discrete Geometry. Springer Grundlehren der mathematischen Wissenschaften (2007)
39. Gunzburger, M., Jantsch, P., Teckentrup, A., Webster, C.G.: A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA J. Uncertainty Quantification* **3**, 1046–1074 (2015)

40. Gunzburger, M., Webster, C.G., Zhang, G.: An adaptive wavelet stochastic collocation method for irregular solutions of partial differential equations with random input data. *Lect. Notes Comput. Sci. Eng.* **97**, 137–170. Springer (2014)
41. Gunzburger, M.D., Webster, C.G., Zhang, G.: Stochastic finite element methods for partial differential equations with random input data. *Acta Numer.* **23**, 521–650 (2014)
42. Hansen, M., Schwab, C.: Analytic regularity and nonlinear approximation of a class of parametric semilinear elliptic PDEs. *Math. Nachr.* **286**, 832–860 (2013)
43. Hansen, M., Schwab, C.: Sparse adaptive approximation of high dimensional parametric initial value problems. *Vietnam J. Math.* **41**, 181–215 (2013)
44. Hoang, V.H., Schwab, C.: Sparse tensor Galerkin discretizations for parametric and random parabolic PDEs – analytic regularity and generalized polynomial chaos approximation. *SIAM J. Math. Anal.* **45**, 3050–3083 (2013)
45. Jakeman, J.D., Archibald, R., Xiu, D.: Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. *J. Comput. Phys.* **230**, 3977–3997 (2011)
46. Kuo, F.Y., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo methods for high-dimensional integration: the standard (weighted Hilbert space) setting and beyond. *The ANZIAM J. Aust. N. Z. Ind. Appl. Math. J.* **53**, 1–37 (2011)
47. Kuo, F.Y., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.* **50**, 3351–3374 (2012)
48. Li, C.F., Feng, Y.T., Owen, D.R.J., Li, D.F., Davis, I.M.: A Fourier-Karhunen-Loève discretization scheme for stationary random material properties in SFEM. *Int. J. Numer. Methods Eng.* **73**, 1942–1965 (2007)
49. Loèvè, M.: Probability Theory. I. Graduate Texts in Mathematics, vol. 45, 4th edn. Springer, New York (1977)
50. Loèvè, M.: Probability Theory. II. Graduate Texts in Mathematics, vol. 46, 4th edn. Springer, New York (1978)
51. Ma, X., Zabaras, N.: An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* **228**, 3084–3113 (2009)
52. Ma, X., Zabaras, N.: An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *J. Comput. Phys.* **229**, 3884–3915 (2010)
53. Maday, Y., Nguyen, N., Patera, A., Pau, S.: A general multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.* **8**, 383–404 (2009)
54. Mathelin, L., Hussaini, M.Y., Zang, T.A.: Stochastic approaches to uncertainty quantification in CFD simulations. *Numer. Algorithms* **38**, 209–236 (2005)
55. Matthies, H.G., Keese, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**, 1295–1331 (2005)
56. Milani, R., Quarteroni, A., Rozza, G.: Reduced basis methods in linear elasticity with many parameters. *Comput. Methods Appl. Mech. Eng.* **197**, 4812–4829 (2008)
57. Nobile, F., Tempone, R., Webster, C.G.: An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.* **46**, 2411–2442 (2008)
58. Nobile, F., Tempone, R., Webster, C.G.: A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.* **46**, 2309–2345 (2008)
59. Sauer, T., Xu, Y.: On multivariate Lagrange interpolation. *Math. Comput.* **64**, 1147–1170 (1995)
60. Smith, S.J.: Lebesgue constants in polynomial interpolation. *Annales Mathematicae et Informaticae. Int. J. Math. Comput. Sci.* **33**, 109–123 (2006)
61. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR* **4**, 240–243 (1963) (English translation)
62. Stoyanov, M., Webster, C.G.: A gradient-based sampling approach for dimension reduction for partial differential equations with stochastic coefficients. *Int. J. Uncertain. Quantif.* **5**, 49–72 (2015)

63. Sweldens, W.: The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmonic Anal.* **3**, 186–200 (1996)
64. Sweldens, W.: The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.* **29**, 511–546 (1998)
65. Todor, R.A.: Sparse perturbation algorithms for elliptic PDE's with stochastic data. Diss. No. 16192, ETH Zurich (2005)
66. Tran, H., Webster, C.G., Zhang, G.: Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients. Tech. Rep. ORNL/TM-2015/341, Oak Ridge National Laboratory (2015)
67. Gunzburger, M., Jantsch, P., Teckentrup, A., Webster, C.G.: A multilevel stochastic collocation method for partial differential equations with random input data. Tech. Rep. ORNL/TM-2014/621, Oak Ridge National Laboratory (2014)
68. Trefethen, L.N.: Is gauss quadrature better than Clenshaw-Curtis? *SIAM Rev.* **50**, 67–87 (2008)
69. Webster, C.G.: Sparse grid stochastic collocation techniques for the numerical solution of partial differential equations with random input data. PhD thesis, Florida State University (2007)
70. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**, 897–936 (1938)
71. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**, 1118–1139 (2005)
72. Xiu, D., Karniadakis, G.E.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**, 619–644 (2002)
73. Zhang, G., Gunzburger, M.: Error analysis of a stochastic collocation method for parabolic partial differential equations with random input data. *SIAM J. Numer. Anal.* **50**, 1922–1940 (2012)
74. Zhang, G., Webster, C., Gunzburger, M., Burkardt, J.: A hyper-spherical adaptive sparse-grid method for high-dimensional discontinuity detection. *SIAM J. Numer. Anal.* **53**, 1508–1536 (2015)

Daniel M. Tartakovsky and Pierre A. Gremaud

Abstract

Parametric uncertainty, considered broadly to include uncertainty in system parameters and driving forces (source terms and initial and boundary conditions), is ubiquitous in mathematical modeling. The method of distributions, which comprises PDF and CDF methods, quantifies parametric uncertainty by deriving deterministic equations for either probability density function (PDF) or cumulative distribution function (CDF) of model outputs. Since it does not rely on finite-term approximations (e.g., a truncated Karhunen-Loëve transformation) of random parameter fields, the method of distributions does not suffer from the “curse of dimensionality.” On the contrary, it is exact for a class of nonlinear hyperbolic equations whose coefficients lack spatiotemporal correlation, i.e., exhibit an infinite number of random dimensions.

Keywords

Random • Stochastic • Probability density function (PDF) • Cumulative distribution function (CDF) • Langevin equation • White noise • Colored noise • Fokker-Planck equation

Contents

1	Introduction	764
1.1	Randomness in Mathematical Models	764
1.2	Uncertainty Quantification in Langevin sODEs	765
1.3	Uncertainty Quantification in PDEs with Random Coefficients	765

D.M. Tartakovsky (✉)

Department of Mechanical and Aerospace Engineering, University of California, San Diego,
La Jolla, CA, USA

e-mail: dmt@ucsd.edu

P.A. Gremaud

Department of Mathematics, North Carolina State University, Raleigh, NC, USA
e-mail: gremaud@ncsu.edu

2	Method of Distributions	766
2.1	PDF Methods	767
2.2	CDF Methods	770
3	Distribution Methods for PDEs	771
3.1	Weakly Nonlinear PDEs Subject to Random Initial Conditions	772
3.2	Weakly Nonlinear PDEs with Random Coefficients	773
3.3	Nonlinear PDEs with Shocks	774
3.4	Systems of PDEs	776
4	Conclusions	780
Appendix		781
References		782

1 Introduction

Probabilistic representations of uncertain parameters and forcings (e.g., initial and boundary conditions) are routinely used both to derive new effective mathematical models [4] and to quantify parametric uncertainty in existing ones (see this Handbook). Regardless of their *raison d'être*, such probabilistic approaches introduce randomness in quantitative predictions of system behavior. Randomness also stems from stochastic representations of subscale processes in mesoscopic models through either internally generated or externally imposed random excitations (Langevin forces) [22]. Despite their superficial similarity, these two sources of randomness pose different challenges. First, Langevin forces are time-dependent random processes, while uncertain parameters are spatially distributed but time-invariant random fields. Second, Langevin forces are (space-time uncorrelated) "white" noise or exhibit short-range correlations. Random (uncertain) coefficients, on the other hand, typically exhibit pronounced spatial correlations, reflecting the underlying structure of heterogeneous but fundamentally deterministic environments.

1.1 Randomness in Mathematical Models

The distinct challenges posed by the two types of randomness (Langevin forces and uncertain parameters) are illustrated by the following two examples. The first is provided by a Langevin (stochastic ordinary-differential) equation (sODE)

$$\frac{du}{dt} = h(u, t) + g(u, t)\xi(t, \omega). \quad (22.1a)$$

It describes the dynamics of the state variable $u(t, \omega)$, which consists of a slowly varying (deterministic) part h and a fast varying random part $g\xi$; the random fluctuations $\xi(t, \omega)$ have zero mean and a two-point covariance function

$$C(t, s) \equiv \langle \xi(t, \omega)\xi(s, \omega) \rangle = \sigma^2 \rho(t, s). \quad (22.1b)$$

Here ω is an element of a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, $\langle \cdot \rangle \equiv \mathbb{E}(\cdot)$ denotes the ensemble mean over this space, and σ^2 and ρ are the variance and correlation function of ξ , respectively. At time t , the system's state is characterized by the probability $\mathbb{P}[u(t) \leq U]$ or, equivalently, by its probability density function (PDF) $f_u(U; t)$.

The second example is an advection-reaction partial-differential equation (PDE)

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = b H(u) \quad (22.2)$$

where the spatially varying and uncertain coefficients a and b are correlated random fields $a(x, \omega)$ and $b(x, \omega)$. This equation is subject to deterministic or random initial and boundary conditions. Rather than having a unique solution, this problem admits an infinite set of solutions that is characterized by a PDF $f_u(U; x, t)$.

1.2 Uncertainty Quantification in Langevin sODEs

Derivation of deterministic equations governing the dynamics of $f_u(U; t)$ for Langevin equation (22.1) with white noise $\xi(t, \omega)$, i.e., with $\rho(t, s) = \tau \delta(t - s)$ where τ is a characteristic time, is relatively straightforward. For $\xi(t, \omega)$ with an arbitrary distribution, these equations are called the Kramers-Moyal expansion; the latter reduces to the Fokker-Planck equation (FPE) if $\xi(t, \omega)$ is Gaussian [22].

Non-Markovian Langevin equations, e.g., (22.1) with temporally correlated (colored) noise, pose an open challenge. Existing methods for their analysis fall into two categories. The *first approach* introduces an additional Markovian process describing the evolution of $\xi(t, \omega)$. The resulting enlarged system is Markovian and hence can be described by a FPE for the joint PDF of u and ξ [22, Sec. 3.5]. Then f_u is obtained by marginalizing a solution of this FPE with respect to ξ . Alternatively, under certain conditions, the enlarged (Markovian) system of Langevin equations can be solved with the unified colored noise approximation [13]. The *second approach* (see [20] for a review) is to derive a differential equation for f_u , which involves random variables and requires a closure. Such closures place restrictions on the noise properties: the decoupling theory [13] requires the nondimensionalized ξ to be second-order stationary and Gaussian, with variance $\sigma^2 \ll 1$; the small correlation-time expansion [9] is applicable to correlation times $\lambda \rightarrow 0$ and $\lambda/\sigma^2 \ll 1$; the functional integral [13] and path-integral [10] require ξ to be statistically homogeneous; and the large-eddy-diffusivity closure [23] requires ξ to have small variance σ^2 and short correlation length λ .

1.3 Uncertainty Quantification in PDEs with Random Coefficients

Monte Carlo simulations (MCS) provide the most robust and straightforward way to solve PDEs with random coefficients. In the case of (22.2), for instance, they consist of (i) generating multiple realizations of the input parameters a and b , (ii) solving deterministic PDEs for each realization, and (iii) evaluating ensemble statistics or PDFs of these solutions. MCS do not impose limitations on statistical properties of input parameters, entail no modifications of existing deterministic solvers, and are ideal for parallel computing. Yet MCS have a slow convergence rate which renders them computationally expensive (often, prohibitively so). Research in the field of

uncertainty quantification is driven by the goal of designing numerical techniques that are computationally more efficient than MCS.

Various types of stochastic finite element methods (FEMs) provide an alternative to MCS. They start by constructing (e.g., by means of truncated Karhunen-Loëve (K-L) expansions) a finite-dimensional probability space on which an SPDE solution is defined. The Galerkin FEM, often equipped with h -type and p -type adaptivity, approximates such solutions in the resulting composite probability-physical space. Stochastic Galerkin and collocation methods (both discussed elsewhere in this Handbook) employ orthogonal basis expansions of an SPDE solution in the chosen finite-dimensional probability space. These types of methods – sometimes referred to as generalized polynomial chaos (gPC) – outperform MCS when random parameter fields exhibit long correlations and, therefore, can be accurately represented by, e.g., a few terms of their K-L expansions. As the correlation length of an input parameter decreases, its K-L expansion requires more terms to maintain the same accuracy, thus increasing the dimensionality of the probability space on which solution is defined. Once the number of random variables exceeds a certain threshold, the stochastic FEMs become computationally less efficient than MCS, a phenomenon known as the curse of dimensionality.

This discussion illustrates the difficulty in developing efficient UQ tools capable of handling both short correlations (or lack thereof) typical of Langevin systems (22.1), and long correlations often present in PDEs with random coefficients such as (22.2). It also highlights the complementary nature of the method of distributions (of which Fokker-Planck equations are a classical example) and stochastic FEMs. While the former work best for random inputs with short correlation lengths (and are often exact in the absence of correlations), the latter outperform MCS when random inputs exhibit long correlations.

This chapter discusses the method of distributions, which aims to derive deterministic equations satisfied by a probabilistic distribution (e.g., PDF) of a system state, as a computationally efficient framework for dealing with both types of uncertainty (randomness). PDF methods originated in the statistical theory of turbulence [21] and have since been used to derive PDF (or Fokker-Planck) equations for systems of coupled Langevin (stochastic ordinary-differential) equations with colored noise [29], to homogenize ordinary-differential equations (ODEs) with random coefficients [19], and to quantify parametric uncertainty in advection-diffusion [23, 24, 28], shallow water [30], and multiphase flow [31] equations.

2 Method of Distributions

We use the term “the method of distributions” to designate a class of approaches based on the derivation of deterministic PDEs for either a probability density function (PDF) or a cumulative distribution function (CDF) of a state variable. The type of the distribution function used gives rise to PDF or CDF methods, respectively.

2.1 PDF Methods

We illustrate the nature of the PDF method by considering a deterministic version of (22.1) subject to an uncertain (random) initial condition,

$$\frac{du}{dt} = G(u, t), \quad u(0, \omega) = u_0(\omega). \quad (22.3)$$

Here $u_0 \in \mathbb{R}$ has a known PDF $f_0(U)$, and $G : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ is a known smooth function. In order to derive an equation for the PDF $f_u(U; t)$ of $u(t; \omega)$, we define a random “raw PDF function” Π in terms of the Dirac delta function $\delta(\cdot)$ such that

$$\Pi(U; t, \omega) = \delta[U - u(t, \omega)]. \quad (22.4)$$

Its ensemble mean $\mathbb{E}[\Pi]$ at any time t equals the PDF $f_u(U; t)$. Indeed,

$$\mathbb{E}[\Pi] \equiv \int_{-\infty}^{\infty} \delta(U - \tilde{u}) f_u(\tilde{u}; t) d\tilde{u} = f_u(U; t). \quad (22.5)$$

Multiplying (22.3) by $\partial_U \Pi$, using the fundamental properties of the Dirac delta function, and taking the expected value to a Cauchy problem:

$$\frac{\partial f_u}{\partial t} + \frac{\partial[G(U, t) f_u(U; t)]}{\partial U} = 0, \quad f_u(U; 0) = f_0(U), \quad (22.6)$$

whose solution yields the entire statistical profile of $u(t, \omega)$.

Challenges posed by the presence of multiplicative noise (parametric uncertainty) are often illustrated by the classic test problem [32]

$$\frac{du}{dt} = -ku, \quad u(0) = u_0, \quad (22.7)$$

in which both the coefficient $k(\omega)$ and initial condition $u_0(\omega)$ are random variables. While (22.7) looks simple, the determination of the statistical properties of its solution is a nonlinear problem. The need to deal with the mixed moment $\mathbb{E}[ku]$ suggests defining the raw joint PDF of $k(\omega)$ and $u(t, \omega)$,

$$\Pi(U, K; t, \omega) = \delta(U - u(t, \omega))\delta(K - k(\omega)). \quad (22.8)$$

Its ensemble mean is the joint PDF, $\mathbb{E}[\Pi] = f_{uk}(U, K; t)$. Manipulations similar to those used in the previous example lead to the PDF equation

$$\frac{\partial f_{uk}}{\partial t} = K \frac{\partial(U f_{uk})}{\partial U}. \quad (22.9)$$

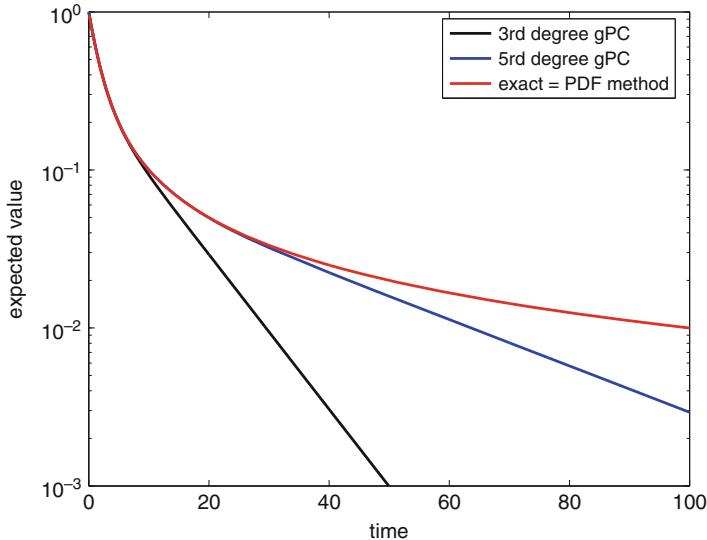


Fig. 22.1 Expected value of the solution to (22.7) with $k \sim \mathcal{U}(0, 1)$ and $u_0 \sim \mathcal{U}(-1, 1)$. The gPC approximations only capture the solution for small times. The PDF method is exact and yields $\mathbb{E}[u(t, \omega)] = (1 - e^{-t})/t$

This equation can be solved by the method of characteristics and the marginal $f_u(U; t)$ obtained by integration, see Fig. 22.1.

Numerical techniques such as gPC evolve approximations based on the statistical properties of the data (here k and u_0). These approximations become increasingly inappropriate under the stochastic drift generated by the nonlinear dynamics (see Fig. 22.1); in stark contrast to gPC, the PDF method is *exact*. Methods such as time-dependent and/or gPC (see the relevant chapters of this handbook) have been proposed to locally adapt the gPC polynomial bases; these improvements only partially solve the above problem at the price, however, of significant numerical costs and complexity.

The derivation of equations (22.6) and (22.9) for the PDFs of the solutions to (22.3) and (22.7), respectively, involves several delicate points that deserve further scrutiny. While PDF equations have been derived through the use of characteristic functions [14], we prefer a more explicit justification based on regularization arguments. We use the study of systems of nonlinear ODEs to provide a mathematical justification of the approach. Consider a set of state variables $\mathbf{u}(t) = (u_1, \dots, u_N)^\top$ whose dynamics are governed by a system of coupled ODEs subject to a random initial condition

$$\frac{du_i}{dt} = G_i(\mathbf{u}, t), \quad i = 1, \dots, N; \quad \mathbf{u}(0, \omega) = \mathbf{u}_0(\omega), \quad (22.10)$$

where \mathbf{u}_0 is an \mathbb{R}^N -valued random variable and $\mathbf{G} = (G_1, \dots, G_N)^\top : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a continuous function. For each $\omega \in \Omega$, (22.10) is an initial value problem which can be analyzed through deterministic calculus. An ω -wise modification of deterministic proofs [2] leads to the existence and uniqueness of solutions. It should be noticed however that the vector field $\mathbf{G}(\mathbf{u}, t)$ is generally only continuous in t but not differentiable even if \mathbf{G} is smooth.

We define a regularized PDF $f_{\mathbf{u}, \epsilon}$ of $\mathbf{u} = (u_1, \dots, u_N)^\top$, the solution to (22.10), as

$$f_{\mathbf{u}, \epsilon}(\mathbf{U}; t) = (\eta_\epsilon \star f_{\mathbf{u}})(\mathbf{U}, t) = \int_{-\infty}^{\infty} \eta_\epsilon(\mathbf{U} - \tilde{\mathbf{u}}) f_{\mathbf{u}}(\tilde{\mathbf{u}}; t) d\tilde{\mathbf{u}} = \mathbb{E}[\eta_\epsilon(\mathbf{U} - \mathbf{u})], \quad (22.11)$$

where $\mathbf{U} \in \mathbb{R}^N$, $f_{\mathbf{u}}$ is the PDF of \mathbf{u} , and $\eta_\epsilon \in \mathcal{C}_0^\infty(\mathbb{R}^N)$ is a standard mollifier, for instance,

$$\eta_\epsilon(\mathbf{x}) = \frac{\epsilon^{-N}}{\int \eta d\mathbf{x}} \eta\left(\frac{\mathbf{x}}{\epsilon}\right) \quad \text{where } \eta(\mathbf{x}) = \begin{cases} \exp\left(\frac{1}{|\mathbf{x}|^2 - 1}\right) & \text{if } |\mathbf{x}| < 1 \\ 0 & \text{if } |\mathbf{x}| \geq 1. \end{cases}$$

One can show (e.g., [8], Appendix C) that $f_{\mathbf{u}, \epsilon}$ is a smooth approximation of $f_{\mathbf{u}}$. Using this fact, we show in the Appendix that for any $\phi \in \mathcal{C}_c^1(\mathbb{R}^N \times [0, \infty))$, the PDF $f_{\mathbf{u}}$ satisfies

$$\int_0^\infty \int_{-\infty}^\infty f_{\mathbf{u}} \partial_t \phi d\mathbf{U} dt + \int_0^\infty \int_{-\infty}^\infty \mathbf{G} f_{\mathbf{u}} \partial_{\mathbf{U}} \phi d\mathbf{U} dt + \int_{-\infty}^\infty f_{\mathbf{u}}(\mathbf{U}; 0) \phi(\mathbf{U}, 0) d\mathbf{U} = 0. \quad (22.12)$$

In other words, $f_{\mathbf{u}}$ is a distributional solution to

$$\frac{\partial f_{\mathbf{u}}}{\partial t} + \nabla_{\mathbf{U}} \cdot [\mathbf{G}(\mathbf{U}, t) f_{\mathbf{u}}] = 0, \quad f_{\mathbf{u}}(\mathbf{U}, 0) = f_0(\mathbf{U}), \quad (22.13)$$

where $f_0(\mathbf{U})$ is the distribution corresponding to \mathbf{u}_0 . It is worthwhile emphasizing that the PDF equation (22.13) is *exact*. Figure 22.2 illustrates this approach in the case of the van der Pol equation

$$\frac{d}{dt} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_2 \\ \mu(1 - u_1^2)u_2 - u_1 \end{bmatrix} \quad (22.14a)$$

with $\mu \geq 0$ subject to Gaussian initial conditions

$$\begin{bmatrix} u_{1,0} \\ u_{2,0} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \sigma^2 \mathbb{I}\right). \quad (22.14b)$$

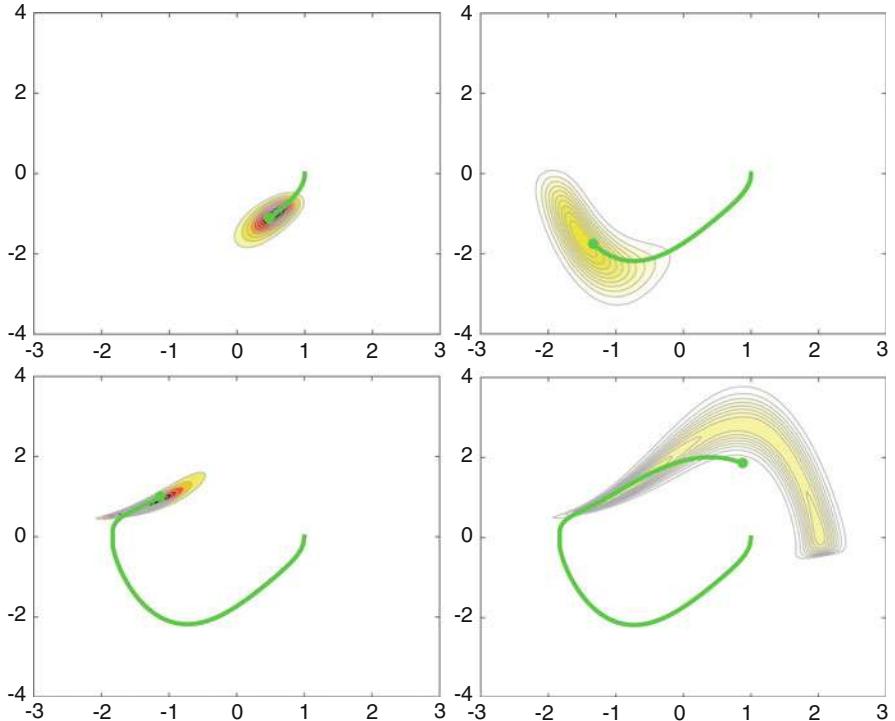


Fig. 22.2 Phase space trajectory of the PDF of the solution to the van der Pol equation (22.14a) with $\mu = 1.25$ subject to the Gaussian initial condition (22.14b) with $\sigma^2 = 0.01$ at times $t = 1.0, 2.0, 4.0$ and 5.2 ; the solid green line is the trajectory of the mean solution while the green “dot” corresponds to the expected value at the considered time. The PDF undergoes many “expansions” and “contractions”; this complex dynamics cannot be accurately described by a few moments of the solution

2.2 CDF Methods

CDF methods aim to derive deterministic PDEs governing the dynamics of a cumulative distribution function (CDF) of a state variable. For the ODE (22.3), instead of defining a raw PDF function (22.4), we introduce “raw CDF function”

$$\Pi(U; t, \omega) = \mathcal{H}[U - u(t, \omega)]. \quad (22.15)$$

where $\mathcal{H}(\cdot)$ is the Heaviside function. At time t , the ensemble mean $\mathbb{E}[\Pi]$ gives the CDF $F_u(U; t)$ of $u(t, \omega)$:

$$\mathbb{E}[\Pi] \equiv \int \mathcal{H}(U - \tilde{u}) f_u(\tilde{u}; t) d\tilde{u} = \int f_u(\tilde{u}; t) d\tilde{u} = F_u(U; t). \quad (22.16)$$

Multiplying (22.3) by $\partial_U \Pi$, using the fundamental properties of the Dirac delta function, and taking the expected value lead to a Cauchy problem for the CDF of the solution to (22.3):

$$\frac{\partial F_u}{\partial t} + G(U, t) \frac{\partial F_u}{\partial U} = 0, \quad F_u(U; 0) = F_0(U), \quad (22.17)$$

where $F_0(U)$ is the CDF of the random initial condition $u_0(\omega)$.

The PDF and CDF formulations of a specific system of ODEs, for instance, (22.6) and (22.17), are obviously related (through integration/derivation); they also display different properties which have implications for their numerical resolutions. First, PDF equations have a conservative form, while CDF equations do not. Second, solutions of CDF equations are monotonic in “space” (the U coordinate), while those of PDF equations are not. The situation is more involved in the case of PDEs and corresponding boundary conditions, as discussed in the next section.

3 Distribution Methods for PDEs

Consider a scalar balance law

$$\frac{\partial u}{\partial t} + \frac{\partial G(u)}{\partial x} = H(u, x, t), \quad u(x, 0, \omega) = u_0(x, \omega), \quad (22.18)$$

where $G : \mathbb{R} \rightarrow \mathbb{R}$ and $H : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ are known smooth functions of their arguments. Following the procedure outlined above, we define a raw PDF as $\Pi(U; x, t) = \delta(U - u(x, t; \omega))$ and elementary properties of δ to formally derive an equation governing its dynamics

$$\frac{\partial \Pi}{\partial t} - \frac{\partial}{\partial U} \left(\frac{dG}{dU} \Pi \frac{\partial u}{\partial x} \right) = -\frac{\partial}{\partial U} (H(U, x, t) \Pi). \quad (22.19)$$

The expected value of this equation takes the form

$$\frac{\partial f_u}{\partial t} - \frac{dG}{dU} \mathbb{E} \left[\frac{\partial \Pi}{\partial U} \frac{\partial u}{\partial x} \right] - \frac{d^2 G}{dU^2} \mathbb{E} \left[\Pi \frac{\partial u}{\partial x} \right] = -\frac{\partial H(U, x, t) f_u}{\partial U}, \quad (22.20)$$

which yields a closed exact integro-differential equation for the PDF $f_u(U; x, t)$

$$\frac{\partial f_u}{\partial t} + \frac{dG}{dU} \frac{\partial f_u}{\partial x} + \frac{d^2G}{dU^2} \frac{\partial}{\partial x} \int_{-\infty}^U f_u(\tilde{U}; x, t) d\tilde{U} + \frac{\partial H(U, x, t) f_u}{\partial U} = 0. \quad (22.21)$$

A CDF formulation can also be derived (through integration); it takes the simple form

$$\frac{\partial F_u}{\partial t} + \frac{dG}{dU} \frac{\partial F_u}{\partial x} + H(U, x, t) \frac{\partial F_u}{\partial U} = 0. \quad (22.22)$$

A few computational examples are presented below.

3.1 Weakly Nonlinear PDEs Subject to Random Initial Conditions

Consider a reaction-advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = H(u, x, t), \quad u(x, 0) = u_0(x, \omega) \quad (22.23)$$

Defining the raw PDF and CDF, $\Pi = \delta[U - u(x, t, \omega)]$ and $\Pi = \mathcal{H}[U - u(x, t, \omega)]$, and following the procedure described above yields Cauchy problems for the single-point PDF and CDF of u ,

$$\frac{\partial f_u}{\partial t} + \frac{\partial f_u}{\partial x} = -\frac{\partial[H(U) f_u]}{\partial U}, \quad f_u(U; x, 0) = f_{u_0}(U; x) \quad (22.24)$$

and

$$\frac{\partial F_u}{\partial t} + \frac{\partial F_u}{\partial x} = -H(U) \frac{\partial F_u}{\partial U}, \quad F_u(U; x, 0) = F_{u_0}(U; x). \quad (22.25)$$

The above equations can be numerically solved to high accuracy with standard methods. In fact, due to their linear structure, they can often be solved exactly through a characteristic analysis; the PDF solution for $H(u, x, t) \equiv u - u^3$ and a standard Gaussian initial condition $u_0(\omega)$ is shown in Fig. 22.3.

Additional probabilistic information can be gained by similar means. For instance, for an initial condition u_0 corresponding to a random field, the equation for the two-point CDF of u is given by

$$\frac{\partial F_{u,u'}}{\partial t} + \frac{\partial F_{u,u'}}{\partial x} + \frac{\partial F_{u,u'}}{\partial x'} = -H(U) \frac{\partial F_{u,u'}}{\partial U} - H(U') \frac{\partial F_{u,u'}}{\partial U'}, \quad (22.26)$$

where $F_{u,u'}(U, U'; x, x', t) = \mathbb{P}[u(x, t) \leq U, u(x', t) \leq U']$ and $F_{u,u'}(U, U'; x, x', 0)$ is a known initial condition.

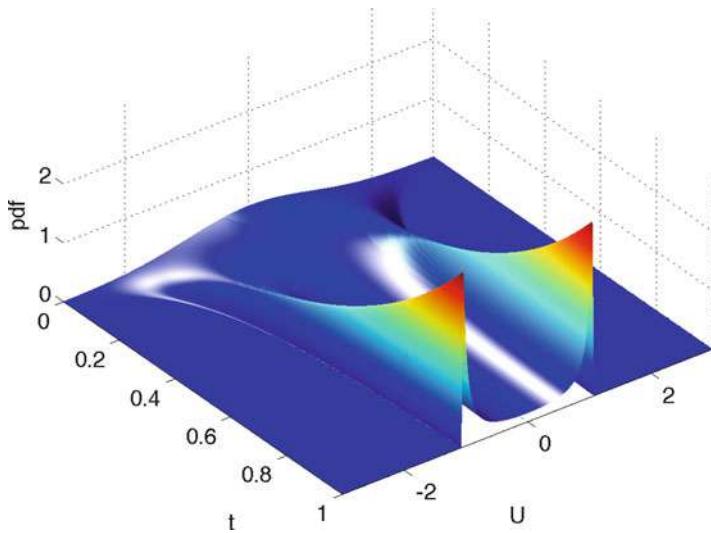


Fig. 22.3 PDF of the solution to (22.23) for a spatially constant initial condition $u_0 \sim \mathcal{N}(0, 1)$ and for $H(u, x, t) \equiv u - u^3$ (the solution u does not depend on x)

3.2 Weakly Nonlinear PDEs with Random Coefficients

Consider the advection-reaction equation (22.2) with random coefficients $a(x, \omega)$ and $b(x, \omega)$, which can be either correlated or uncorrelated in space. This equation is subject to an initial condition $u(x, 0) = u_0$. As in previous examples, the raw CDF $\Pi(U; x, t, \omega) = \mathcal{H}[U - u(x, t, \omega)]$ satisfies exactly an equation

$$\frac{\partial \Pi}{\partial t} + a \frac{\partial \Pi}{\partial x} = -bH(U, x, t) \frac{\partial \Pi}{\partial U}, \quad \Pi(U; x, 0, \omega) = \mathcal{H}(U - u_0). \quad (22.27)$$

It describes two-dimensional, $\mathbf{x} = (x, U)^\top \in \mathbb{R} \times \mathbb{R}$, advection of Π in the random velocity field $\mathbf{v} = (a, bH)^\top$. Stochastic averaging of this equation is a classical problem that requires a closure approximation (see, e.g., [5] and the references therein). Such closures can be constructed by representing the random coefficients $a(x, \omega)$ and $b(x, \omega)$ with their finite-term approximations obtained, e.g., via Karhunen-Loëve or Fourier transformations [25, 27, 28]. The number of terms in the resulting expansions of $a(x, \omega)$ and $b(x, \omega)$, which is necessary to achieve a required representational accuracy, increases as correlation lengths of $a(x, \omega)$ and $b(x, \omega)$ decrease. Beyond a certain range, solving (22.27) with Monte Carlo simulations becomes the most efficient option (curse of dimensionality).

Perturbation-based closures [5, 6, 24] provide a computational alternative which does not require approximations of random parameter fields such as $a(x, \omega)$ and $b(x, \omega)$. For example, the ensemble average of (22.27) yields a nonlocal equation [5]

$$\frac{\partial F}{\partial t} + \mathbb{E}[\mathbf{v}] \cdot \nabla F = \Phi(F), \quad \Phi(F; \mathbf{x}, t) \equiv \mathbb{E}[\mathbf{v}' \cdot \nabla \Pi'] \quad (22.28a)$$

where $\mathbf{v}' = \mathbf{v} - \mathbb{E}[\mathbf{v}]$ and $\Pi' = \Pi - F$; the nonlocal term $\Phi(F)$ is approximated by

$$\Phi(F; \mathbf{x}, t) \approx \int_0^t \int_D \mathbb{E}[v'_i(\mathbf{y}) v'_j(\mathbf{x})] \frac{\partial G}{\partial x_j}(\mathbf{x}, \mathbf{y}, t - \tau) \frac{\partial F}{\partial y_i}(\mathbf{y}, \tau) d\mathbf{y} d\tau, \quad (22.28b)$$

where $D \subset \mathbb{R}^2$. Here the Einstein notation is used to indicate summation over repeated indices, and $G(\mathbf{x}, \mathbf{y}, t - \tau)$ is the “mean-field” Green’s function for (22.27) defined as a solution of

$$\frac{\partial G}{\partial \tau} + \nabla_{\mathbf{y}} \cdot (\mathbb{E}[\mathbf{v}] G) = -\delta(\mathbf{x} - \mathbf{y}) \delta(t - \tau), \quad (22.29)$$

subject to the homogeneous initial and boundary conditions. Note that the CDF equation (22.28) accounts for arbitrary auto- and cross-correlations of the input parameter fields $a(x, \omega)$ and $b(x, \omega)$. Correlated random coefficients necessitate either space-time localization of (22.28b), as was done in [5, 6], or solving an integro-differential CDF equation (22.28). The accuracy of the localization approximation increases as the correlation lengths of $a(x, \omega)$ and $b(x, \omega)$, λ_a and λ_b become smaller ($\lambda_a, \lambda_b \rightarrow 0$). Thus, the methods of distribution work best in the regime in which the methods based on finite-term representations of parameter fields (e.g., gPC and stochastic collocation methods) fail.

3.3 Nonlinear PDEs with Shocks

The solutions to generic nonlinear hyperbolic balance laws such as (22.18) are in general non-smooth and can present shocks even for smooth initial conditions. The various types of probabilistic distributions describing such solutions are also expected to not be smooth. As a result, the resolution of the evolution equations describing such distributions (see, for instance, (22.21) and (22.22)) becomes problematic. A way to incorporate shock dynamics in the method of distributions (CDF equation) can be found in [31] and consists essentially in adapting the concept of front tracking to the present framework (see [30] for a different approach in the context of kinematic-wave equations).

Following [31], we illustrate the approach on a model of multiphase flow in porous media given by the Buckley-Leverett equation,

$$\frac{\partial u}{\partial t} + q \frac{\partial G(u)}{\partial x} = 0, \quad G = \frac{(u - s_{wi})^2}{(u - s_{wi})^2 + (1 - s_{oi} - u)^2 \epsilon_\mu}. \quad (22.30)$$

The above model describes the dynamics of water saturation $u(x, t) : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [s_{wi}, 1 - s_{oi}]$ due to displacement of oil by water with macroscopic velocity $q(t)$. The ratio of the water and oil viscosities is denoted by ϵ_μ . The porous medium is initially mostly saturated with oil and has a uniform (irreducible) water saturation s_{wi} , such that $u(x, t = 0) = s_{wi}$. Equation (22.30) is subject to the boundary condition $u(x = 0, t) = 1 - s_{oi}$, where s_{oi} is the irreducible oil saturation. Both s_{oi} and s_{wi} are treated as deterministic constants. The macroscopic flow velocity is uncertain and treated as a random function $q(t, \omega)$ with known PDF $f_q(Q; t)$.

Let $x_f(t)$ denote the position of a water-oil shock front. Ahead of the front, a rarefaction wave follows well-defined characteristic curves. Behind the front, the saturation remains at the initial value, $u^+ = s_{wi}$. The Rankine-Hugoniot condition defines the front location,

$$\frac{dx_f}{dt} = q \frac{G(u^-) - G(u^+)}{u^- - u^+}. \quad (22.31)$$

The saturation value ahead of the front, u^- , is constant along the characteristic curve defined by

$$\frac{dx}{dt} = v(u^-) = q \frac{dG}{du}(u^-),$$

which must match the shock speed:

$$\frac{G(u^-) - G(u^+)}{u^- - u^+} = \frac{dG}{du}(u^-). \quad (22.32)$$

Solving (22.32) gives u^- and hence the location of the shock front $x_f(t)$. The continuous rarefaction solution $u_r(x, t)$ ahead of the front is found by using the method of characteristics in the range $u^- \leq u_r \leq 1 - u_{oi}$. The complete solution is given by [31]:

$$u(x, t) = \begin{cases} u_r(x, t), & 0 \leq x < x_f(t) \\ s_{wi}, & x > x_f(t) \end{cases} \quad (22.33)$$

The raw CDF Π is subdivided into two parts, Π_a and Π_b , according to the saturation solution (22.33):

$$\Pi(U, x, t) = \begin{cases} \Pi_a = \mathcal{H}(U - s_{wi}), & U < u^-, \quad x > x_f(t) \\ \Pi_b = \mathcal{H}(U - s_i), & s^- < U, \quad x < x_f(t) \end{cases} \quad (22.34a)$$

where

$$\Pi_b = \mathcal{H}(U - 1 + s_{oi})\mathcal{H}(C - x) + \mathcal{H}(U - s_{wi})\mathcal{H}(x - C) \quad (22.34b)$$

with

$$C(U, t) = \frac{dG}{dU} \int_0^t q(\tau) d\tau. \quad (22.34c)$$

The ensemble average of (22.34) yields a formal expression for the CDF $F_u(U; x, t)$ [31],

$$F_u = \begin{cases} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pi_a \mathcal{H}(x - X_f) f_{u, x_f}^+(U, X_f; x, t) dU dX_f, & U < u^- \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [\Pi_a \mathcal{H}(x - X_f) f_{u, x_f}^+ + \Pi_b \mathcal{H}(X_f - x) f_{u, x_f}^-] dU dX_f, & U \geq u^-. \end{cases} \quad (22.35)$$

Here $f_{u, x_f}(U, X_f; x, t)$ is the (unknown) joint PDF of u and x_f , with the superscripts $-$ and $+$ indicating the parts of this PDF restricted to values of u before and after the front, respectively. Since $f_{u, x_f} dU dX_f = f_q dQ$, this yields a computable solution (see [31] for details)

$$F_u(U; x, t) = \begin{cases} \int_{-\infty}^{\infty} \Pi_a \mathcal{H}[x - X_f(Q)] p_q dQ, & U < u^- \\ \int_{-\infty}^{\infty} [\Pi_a \mathcal{H}(x - X_f(Q)) + \Pi_b \mathcal{H}[X_f(Q) - x]] p_q dQ, & U \geq u^-. \end{cases} \quad (22.36)$$

Evaluation of this expression requires one to compute the stochastic integral in (22.34c). The performance of three alternative approximations, whose performance depends on the ratio of the correlation length of the random input $q(t, \omega)$ and the integration time t , is shown in Fig. 22.4. Note that no approximation would be needed if $q(t, \omega)$ is white noise.

3.4 Systems of PDEs

The local nature in x and t of (22.21) and (22.22) is a direct result of the simple dependency structure associated to (22.18): solutions can be constructed using one family of characteristics. The situation is more involved for random hyperbolic systems of balance laws such as

$$\frac{\partial u_i}{\partial t} + \frac{\partial G_i(\mathbf{u})}{\partial x} = H_i(\mathbf{u}, x, t), \quad i = 1, \dots, N, \quad (22.37)$$

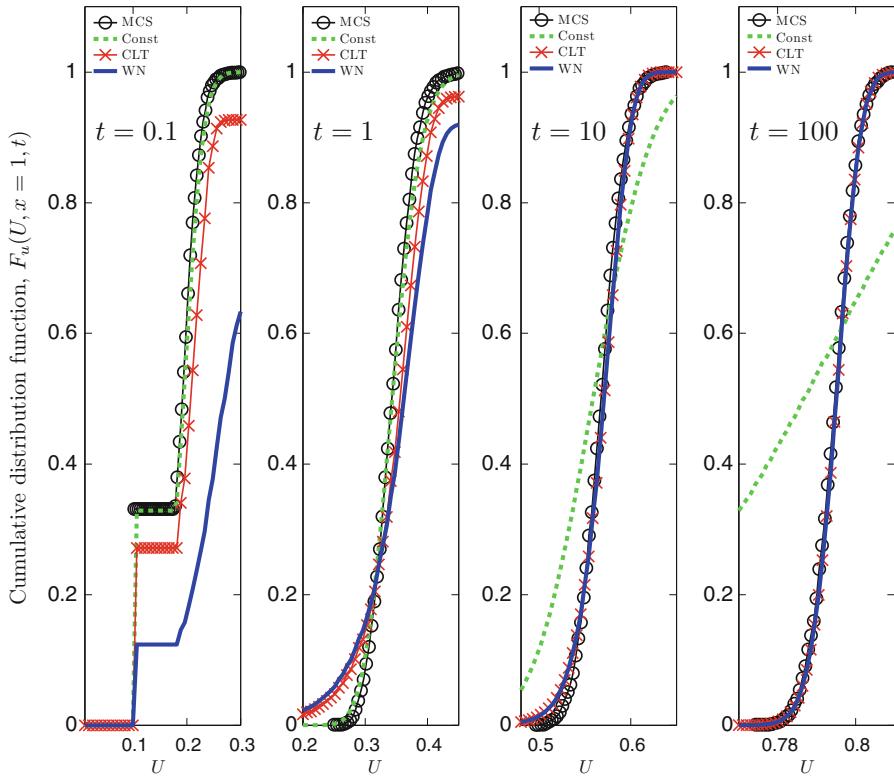


Fig. 22.4 CDF $F_u(U; x = 1, t)$ at $t = 0.1$, $t = 1$, $t = 10$, and $t = 100$ obtained with Monte Carlo simulations (MCS) and the CDF solution (22.36) under the three alternatively approximations: the random constant approximation (Const), an approximation based on the central limit theorem (CLT), and the white-noise approximation (WN). Only the portion of the domain in which the CDF varies is shown (This figure is reproduced from [31])

where $\mathbf{u} = (u_1, \dots, u_N)^\top$. Similar to the above, $G_i : \mathbb{R}^N \rightarrow \mathbb{R}$ and $H_i : \mathbb{R}^N \times \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ ($i = 1, \dots, N$) are known smooth functions with respect to all their arguments, and the initial condition is given by $\mathbf{u}(x, 0, \omega) = \mathbf{u}_0(x, \omega)$. The evolution equation for the raw PDF $\Pi(\mathbf{U}; x, t) \equiv \prod_{i=1}^N \delta(U_i - u_i(x, t; \omega))$ with $\mathbf{U} = (U_1, \dots, U_N)^\top$ is here

$$\frac{\partial \Pi}{\partial t} - \frac{\partial \Pi}{\partial U_i} \frac{\partial G_i(\mathbf{u})}{\partial x} = - \frac{\partial (H_i(\mathbf{U})\Pi)}{\partial U_i}, \quad (22.38)$$

where the Einstein notation is used to indicate the summation over the repeated indices. Relying again on properties of δ , we obtain the integro-differential equation

$$\frac{\partial \Pi}{\partial t} - \frac{\partial \Pi}{\partial U_i} \frac{\partial}{\partial x} \int G_i(\mathbf{U}^*) \Pi(\mathbf{U}^*; x, t) d\mathbf{U}^* = - \frac{\partial (H_i(\mathbf{U})\Pi)}{\partial U_i}. \quad (22.39)$$

An equation for the joint PDF $f_{\mathbf{u}}(\mathbf{U}; x, t)$ can in principle be obtained as the ensemble mean (over realizations of random u_i , $i = 1, \dots, N$ at point x and time t), i.e., $\langle \Pi \rangle = f_{\mathbf{u}}(\mathbf{U}; x, t)$; a closed exact equation – such as (22.21) or (22.22) in the scalar case – is however not available in general.

We illustrate this point on a simple example: the wave equation with constant speed of propagation $c > 0$

$$\frac{\partial}{\partial t} \begin{bmatrix} v \\ w \end{bmatrix} + \begin{bmatrix} 0 & -c^2 \\ -1 & 0 \end{bmatrix} \frac{\partial}{\partial x} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (22.40)$$

Equation (22.38) takes here the form

$$\frac{\partial \Pi}{\partial t} + \frac{\partial \Pi}{\partial W} \frac{\partial v}{\partial x} + c^2 \frac{\partial \Pi}{\partial V} \frac{\partial w}{\partial x} = 0, \quad (22.41)$$

the expectation of which yields

$$\frac{\partial f_{vw}}{\partial t} + \mathbb{E} \left[\frac{\partial \Pi}{\partial W} \frac{\partial v}{\partial x} \right] + c^2 \mathbb{E} \left[\frac{\partial \Pi}{\partial V} \frac{\partial w}{\partial x} \right] = 0. \quad (22.42)$$

Unlike the scalar case, there is here no simple way of expressing the arguments of the expectation(s) as exact differentials in order to close the equation.

A more direct approach illustrates this phenomenon in another way. Consider the wave equation (22.40) in the form

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0, \quad (22.43)$$

with

$$v = \frac{\partial u}{\partial x} \quad \text{and} \quad w = \frac{\partial u}{\partial t}.$$

Defining the raw PDF as $\Pi(U; x, t) = \delta(U - u(x, t; \omega))$ and differentiating twice with respect to x and t , we get

$$\frac{\partial^2 f_u}{\partial t^2} - c^2 \frac{\partial^2 f_u}{\partial x^2} = \frac{\partial}{\partial U^2} \mathbb{E} \left\{ \Pi \left[\left(\frac{\partial u}{\partial t} \right)^2 - c^2 \left(\frac{\partial u}{\partial x} \right)^2 \right] \right\}, \quad (22.44)$$

which, again, is an equation that does admit an elementary closure. This difficulty is not specific to the wave equation: the nonlocal nature and self-interacting character of most partial differential equations unfortunately precludes the existence of pointwise equations for the PDF of their solutions. The determination of PDF equations for the solutions of random PDEs is an active area of research; three approaches can be considered.

First, PDF equations can be obtained from the principles outlined above through the use of closure approximations. Such approximations are typically application dependent and attention must be to the resulting accuracy. For instance, we show

below that the simple approximation, when applied to the wave equation, leads to exact means but that the higher moments are not evolved exactly.

Second, using functional integral methods, sets of differential constraints satisfied by PDFs of random PDEs have been proposed [26]. We note however that this approach may still require closure approximations and that, in general, the existence of a set of differential constraints allowing the unique determination of a PDF is an open question.

Third, the need to invoke a closure approximation can, in principle, be avoided through discretization. Let's consider again the hyperbolic system of balance laws (22.37). Spatial semi-discretization of (22.37) by appropriate methods such as central schemes [15–18] leads to ODE systems akin to (22.10). A PDF approach can then be applied to the semi-discrete problems following the lines of Sect. 2. We note however that dimension reduction methods have to be considered for the numerical resolution of the resulting PDF equations. Analyzing the accuracy of the approach, i.e., the closeness of the PDF of solutions to a discretized problem to the PDF of solutions to the original problem, is an open question.

For most applications corresponding to systems of balance laws such as (22.37), the spatial domain is *finite*. This seemingly innocuous remark has profound consequences. The imposition of boundary conditions requires a characteristic analysis. Let $\mathbb{B} = (\partial G_i / \partial u_j)$ be the Jacobian of the system and let $\mathbb{B} = V \Lambda V^{-1}$ be its eigenvalue decomposition (which exists by hyperbolicity assumption). Assuming for now a linear system, i.e., a constant Jacobian and a problem defined on the half line $x > 0$, the number of boundary conditions to be imposed at $x = 0$ is q , the number of positive eigenvalues of \mathbb{B} . The boundary condition is then imposed on the characteristic variables with positive wave speed. If the problem is defined on an interval, the process is simply repeated at the other end by considering again the characteristics entering the computational domain there.

For nonlinear systems with smooth solutions, one can linearize the equations about an appropriate local state (frozen coefficient method). (The situation is much more involved for nonlinear problems admitting shock formation [3, 7, 11, 12].) The back and forth transformations between characteristic variables ($\mathbf{z} = V^{-1}\mathbf{u}$) and “conserved” variables (\mathbf{u}) are trivial in a deterministic setting. In the present framework, the unknowns are joint PDF functions or marginals of random variables that are *not* independent. Therefore, even linear transformations require additional, detailed analysis.

We now illustrate the effect of a closure approximation in the case of the wave equation (22.40). The application of the simple approximation to (22.42) yields

$$\mathbb{E} \left[\frac{\partial \Pi}{\partial W} \frac{\partial v}{\partial x} \right] \approx \mathbb{E} \left[\frac{\partial \Pi}{\partial W} \right] \mathbb{E} \left[\frac{\partial v}{\partial x} \right] = \frac{\partial \mathbb{E}[v]}{\partial x} \frac{\partial f_{vw}}{\partial W}, \quad (22.45)$$

where

$$\mathbb{E}[v] = \iint_{-\infty}^{\infty} V^* f_{vw}(V^*, W^*; x, t) dV^* dW^*,$$

and similarly for the other term in (22.42). This leads to the Fokker-Planck equation

$$\frac{\partial \tilde{f}_{vw}}{\partial t} + c^2 \frac{\partial \mathbb{E}[w]}{\partial x} \frac{\partial \tilde{f}_{vw}}{\partial V} + \frac{\partial \mathbb{E}[v]}{\partial x} \frac{\partial \tilde{f}_{vw}}{\partial W} = 0, \quad (22.46)$$

where \tilde{f}_{vw} is the approximation of f_{vw} under the above closure.

How accurate is this approximation? By multiplying (22.46) alternatively by V and W , and integrating, we obtain

$$\frac{\partial \tilde{\mathbb{E}}[v]}{\partial t} - c^2 \frac{\partial \mathbb{E}[w]}{\partial x} = 0 \quad \text{and} \quad \frac{\partial \tilde{\mathbb{E}}[w]}{\partial t} - \frac{\partial \mathbb{E}[v]}{\partial x} = 0, \quad (22.47)$$

where $\tilde{\mathbb{E}}$ denotes the expected value taken with respect to \tilde{f}_{vw} . On the other hand, by taking directly the expectation of (22.40), we get

$$\frac{\partial \mathbb{E}[v]}{\partial t} - c^2 \frac{\partial \mathbb{E}[w]}{\partial x} = 0 \quad \text{and} \quad \frac{\partial \mathbb{E}[w]}{\partial t} - \frac{\partial \mathbb{E}[v]}{\partial x} = 0. \quad (22.48)$$

Subtracting the two above systems equation by equation yields that the difference between $\tilde{\mathbb{E}}[v]$ and $\mathbb{E}[v]$ is constant in time and can be made zero through consistent initial conditions; a similar result holds for $\mathbb{E}[w]$. Therefore (22.46) evolves the means exactly. Similar arguments show that

$$\frac{\partial}{\partial t} \left(\mathbb{E}[v^2] - \tilde{\mathbb{E}}[v^2] \right) = 2 \operatorname{Cov} \left(v, c^2 \frac{\partial w}{\partial x} \right), \quad (22.49)$$

$$\frac{\partial}{\partial t} \left(\mathbb{E}[w^2] - \tilde{\mathbb{E}}[w^2] \right) = 2 \operatorname{Cov} \left(w, \frac{\partial v}{\partial x} \right), \quad (22.50)$$

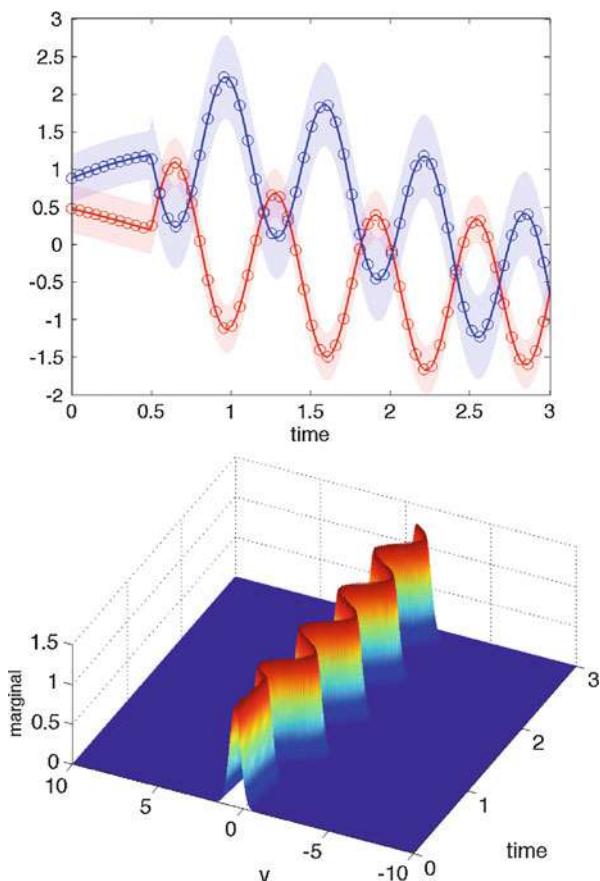
and thus, in general, (22.46) does not evolve the higher-order moments exactly.

Figure 22.5 illustrates the case of the wave equation on the half space $x > 0$ with random initial and boundary conditions. The retained closure is the simple mean field approximation.

4 Conclusions

The method of distributions, which comprises PDF and CDF methods, quantifies uncertainty in model parameters and driving forces (source terms and initial and boundary conditions) by deriving deterministic equations for either probability density function (PDF) or cumulative distribution function (CDF) of model outputs. Since it does not rely on finite-term approximations (e.g., a truncated Karhunen-Loëve transformation) of random parameter fields, the method of distributions does not suffer from the “curse of dimensionality.” On the contrary, it is exact for

Fig. 22.5 Solutions to the wave equation (22.40) in a half-space with random boundary condition. *Top:* the expected values of v and w (solid curves) computed from the PDF equation plus or minus one standard deviation are plotted at a fixed point in space. The boundary condition “hits” at about $t = 0.5$. The “circles” are the Monte Carlo results. *Bottom:* marginal for the v -component at the same spatial point



a class of nonlinear hyperbolic equations whose coefficients lack spatiotemporal correlation, i.e., exhibit an infinite number of random dimensions.

Appendix

It follows from (22.11) and the fact that \mathbf{u} solves (22.10) that

$$\begin{aligned} I &\equiv \int_0^\infty \int_{-\infty}^\infty f_{\mathbf{u},\epsilon}(\mathbf{U}; t) \frac{\partial \phi}{\partial t}(\mathbf{U}, t) d\mathbf{U} dt = \mathbb{E} \left[\int_0^\infty \int_{-\infty}^\infty \eta_\epsilon(\mathbf{U} - \mathbf{u}) \frac{\partial \phi}{\partial t}(\mathbf{U}, t) d\mathbf{U} dt \right] \\ &= \mathbb{E} \left[\int_0^\infty \int_{-\infty}^\infty \eta'_\epsilon(\mathbf{U} - \mathbf{u}) \mathbf{G}(\mathbf{u}) \phi(\mathbf{U}, t) d\mathbf{U} dt \right] - \int_{-\infty}^\infty \mathbb{E}[\eta_\epsilon(\mathbf{U} - \mathbf{u}_0)] \phi(\mathbf{U}, 0) d\mathbf{U}, \end{aligned}$$

for any $\phi \in \mathcal{C}_c^1(\mathbb{R}^N \times [0, \infty))$. Therefore

$$I = \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \eta'_\epsilon(\mathbf{U} - \tilde{\mathbf{u}}) \mathbf{G}(\tilde{\mathbf{u}}) \phi(\mathbf{U}, t) f_{\mathbf{u}}(\tilde{\mathbf{u}}; t) d\mathbf{U} d\tilde{\mathbf{u}} dt - \int_{-\infty}^\infty f_{\mathbf{u}, \epsilon}(\mathbf{U}; 0) \phi(\mathbf{U}, 0) d\mathbf{U}.$$

Integration by parts in \mathbf{U} yields

$$I = - \int_0^\infty \int_{-\infty}^\infty (\eta_\epsilon \star \mathbf{G} f_{\mathbf{u}})(\mathbf{U}, t) \nabla_{\mathbf{U}} \phi(\mathbf{U}, t) d\mathbf{U} dt - \int_{-\infty}^\infty f_{\mathbf{u}, \epsilon}(\mathbf{U}; 0) \phi(\mathbf{U}, 0) d\mathbf{U}.$$

By the above definition of I , we have established that, for any $\phi \in \mathcal{C}_c^1(\mathbb{R}^N \times [0, \infty))$,

$$\int_0^\infty \int_{-\infty}^\infty f_{\mathbf{u}, \epsilon} \frac{\partial \phi}{\partial t} d\mathbf{U} dt + \int_0^\infty \int_{-\infty}^\infty (\eta_\epsilon \star \mathbf{G} f_{\mathbf{u}}) \nabla_{\mathbf{U}} \phi d\mathbf{U} dt + \int_{-\infty}^\infty f_{\mathbf{u}, \epsilon}(\mathbf{U}; 0) \phi(\mathbf{U}, 0) d\mathbf{U} = 0.$$

Using standard arguments [1], taking the limit $\epsilon \rightarrow 0$ leads to (22.12).

References

1. Ambrosio, L., Fusco, N., Pallara, D.: Functions of Bounded Variation and Free Discontinuous Problems. The Clarendon Press/Oxford University Press, Oxford/New York (2000)
2. Arnold, L.: Random Dynamical Systems. Springer, Berlin/New York (1998)
3. Benabdallah, A., Serre, D.: Problèmes aux limites pour des systèmes hyperboliques non linéaires de deux équations à une dimension d'espace. C. R. Acad. Sci. Paris **305**, 677–680 (1986)
4. Bharucha-Reid, A.T. (ed.): Probabilistic Methods in Applied Mathematics. Academic, New York (1968)
5. Boso, F., Broyda, S.V., Tartakovsky, D.M.: Cumulative distribution function solutions of advection-reaction equations with uncertain parameters. Proc. R. Soc. A **470**(2166), 20140189 (2014)
6. Broyda, S., Dentz, M., Tartakovsky, D.M.: Probability density functions for advective-reactive transport in radial flow. Stoch. Environ. Res. Risk Assess. **24**(7), 985–992 (2010)
7. Dubois, F., LeFloch, P.: Boundary conditions for nonlinear hyperbolic systems. J. Differ. Equ. **71**, 93–122 (1988)
8. Evans, L.C.: Partial Differential Equations, 2nd edn. AMS, Providence (2010)
9. Fox, R.F.: Functional calculus approach to stochastic differential equations. Phys. Rev. A **33**, 467–476 (1986)
10. Fuentes, M.A., Wio, H.S., Toral, R.: Effective Markovian approximation for non-Gaussian noises: a path integral approach. Physica A **303**(1–2), 91–104 (2002)
11. Gisclon, M.: Etude des conditions aux limites pour un système strictement hyperbolique via l'approximation parabolique. PhD thesis, Université Claude Bernard, Lyon I (France) (1994)
12. Gisclon, M., Serre, D.: Etude des conditions aux limites pour un système strictement hyperbolique via l'approximation parabolique. C. R. Acad. Sci. Paris **319**, 377–382 (1994)
13. Hänggi, P., Jung, P.: Advances in Chemical Physics, chapter Colored Noise in Dynamical Systems, pp. 239–326. John Wiley & Sons, New York (1995)

14. Kozin, F.: On the probability densities of the output of some random systems. *Trans. ASME Ser. E J. Appl. Mech.* **28**, 161–164 (1961)
15. Kurganov, A., Lin, C.-T.: On the reduction of numerical dissipation in central-upwind schemes. *Commun. Comput. Phys.* **2**, 141–163 (2007)
16. Kurganov, A., Noelle, S., Petrova, G.: Semi-discrete central-upwind scheme for hyperbolic conservation laws and hamilton-jacobi equations. *SIAM J. Sci. Comput.* **23**, 707–740 (2001)
17. Kurganov, A., Petrova, G.: A third order semi-discrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems. *Numer. Math.* **88**, 683–729 (2001)
18. Kurganov, A., Tadmor, E.: New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *J. Comput. Phys.* **160**, 241–282 (2000)
19. Lichtner, P.C., Tartakovsky, D.M.: Stochastic analysis of effective rate constant for heterogeneous reactions. *Stoch. Environ. Res. Risk Assess.* **17**(6), 419–429 (2003)
20. Lindenberg, K., West, B.J.: *The Nonequilibrium Statistical Mechanics of Open and Closed Systems*. VCH Publishers, New York (1990)
21. Lundgren, T.S.: Distribution functions in the statistical theory of turbulence. *Phys. Fluids* **10**(5), 969–975 (1967)
22. Risken, H.: *The Fokker-Planck Equation: Methods of Solutions and Applications*, 2nd edn. Springer, Berlin/New York (1989)
23. Tartakovsky, D.M., Broyda, S.: PDF equations for advective-reactive transport in heterogeneous porous media with uncertain properties. *J. Contam. Hydrol.* **120–121**, 129–140 (2011)
24. Tartakovsky, D.M., Dentz, M., Lichtner, P.C.: Probability density functions for advective-reactive transport in porous media with uncertain reaction rates. *Water Resour. Res.* **45**, W07414 (2009)
25. Venturi, D., Karniadakis, G.E.: New evolution equations for the joint response-excitation probability density function of stochastic solutions to first-order nonlinear PDEs. *J. Comput. Phys.* **231**(21), 7450–7474 (2012)
26. Venturi, D., Karniadakis, G.E.: Differential constraints for the probability density function of stochastic solutions to wave equation. *Int. J. Uncertain. Quant.* **2**, 195–213 (2012)
27. Venturi, D., Sapsis, T.P., Cho, H., Karniadakis, G.E.: A computable evolution equation for the joint response-excitation probability density function of stochastic dynamical systems. *Proc. R. Soc. A* **468**(2139), 759–783 (2012)
28. Venturi, D., Tartakovsky, D.M., Tartakovsky, A.M., Karniadakis, G.E.: Exact PDF equations and closure approximations for advective-reactive transport. *J. Comput. Phys.* **243**, 323–343 (2013)
29. Wang, P., Tartakovsky, A.M., Tartakovsky, D.M.: Probability density function method for Langevin equations with colored noise. *Phys. Rev. Lett.* **110**(14), 140602 (2013)
30. Wang, P., Tartakovsky, D.M.: Uncertainty quantification in kinematic-wave models. *J. Comput. Phys.* **231**(23), 7868–7880 (2012)
31. Wang, P., Tartakovsky, D.M., Jarman K.D. Jr., Tartakovsky, A.M.: CDF solutions of Buckley-Leverett equation with uncertain parameters. *Multiscale Model. Simul.* **11**(1), 118–133 (2013)
32. Xiu, D., Karniadakis, G.E.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**, 619–644 (2002)

Sampling via Measure Transport: An Introduction

23

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini

Abstract

We present the fundamentals of a measure transport approach to sampling. The idea is to construct a deterministic coupling – i.e., a transport map – between a complex “target” probability measure of interest and a simpler reference measure. Given a transport map, one can generate arbitrarily many independent and unweighted samples from the target simply by pushing forward reference samples through the map. If the map is endowed with a triangular structure, one can also easily generate samples from conditionals of the target measure. We consider two different and complementary scenarios: first, when only evaluations of the unnormalized target density are available and, second, when the target distribution is known only through a finite collection of samples. We show that in both settings, the desired transports can be characterized as the solutions of variational problems. We then address practical issues associated with the optimization-based construction of transports: choosing finite-dimensional parameterizations of the map, enforcing monotonicity, quantifying the error of approximate transports, and refining approximate transports by enriching

Y. Marzouk (✉)

Massachusetts Institute of Technology, Cambridge, MA, USA

e-mail: ymarz@mit.edu

T. Moselhy

D. E. Shaw Group, New York, NY, USA

e-mail: tmoselhy@mit.edu

M. Parno

Massachusetts Institute of Technology, Cambridge, MA, USA

U. S. Army Cold Regions Research and Engineering Laboratory, Hanover, NH, USA

e-mail: matthew.d.parno@usace.army.mil; mparno@mit.edu

A. Spantini

Massachusetts Institute of Technology, Cambridge, MA, USA

e-mail: spantini@mit.edu

the corresponding approximation spaces. Approximate transports can also be used to “Gaussianize” complex distributions and thus precondition conventional asymptotically exact sampling schemes. We place the measure transport approach in broader context, describing connections with other optimization-based samplers, with inference and density estimation schemes using optimal transport, and with alternative transformation-based approaches to simulation. We also sketch current work aimed at the construction of transport maps in high dimensions, exploiting essential features of the target distribution (e.g., conditional independence, low-rank structure). The approaches and algorithms presented here have direct applications to Bayesian computation and to broader problems of stochastic simulation.

Keywords

Measure transport • Optimal transport • Knothe–Rosenblatt map • Monte Carlo methods • Bayesian inference • Approximate Bayesian computation • Density estimation • Convex optimization

Contents

1	Introduction	786
2	Transport Maps and Optimal Transport	789
3	Direct Transport: Constructing Maps from Unnormalized Densities	790
3.1	Preliminaries	790
3.2	Optimization Problems	791
3.3	Convergence, Bias, and Approximate Maps	794
4	Inverse Transport: Constructing Maps from Samples	797
4.1	Optimization Problem	797
4.2	Convexity and Separability of the Optimization Problem	799
4.3	Computing the Inverse Map	801
5	Parameterization of Transport Maps	803
5.1	Polynomial Representations	803
5.2	Radial Basis Functions	805
5.3	Monotonicity Constraints and Monotone Parameterizations	805
6	Related Work	806
7	Conditional Sampling	810
8	Example: Biochemical Oxygen Demand Model	813
8.1	Inverse Transport: Map from Samples	814
8.2	Direct Transport: Map from Densities	817
9	Conclusions and Outlook	819
	References	822

1 Introduction

Characterizing complex probability distributions is a fundamental and ubiquitous task in uncertainty quantification. In this context, the notion of “complexity” encompasses many possible challenges: non-Gaussian features, strong correlations

and nonlinear dependencies, high dimensionality, the high computational cost of evaluating the (unnormalized) probability density associated with the distribution, or even intractability of the probability density altogether. Typically one wishes to characterize a distribution by evaluating its moments, or by computing the probability of an event of interest. These goals can be cast as the computation of *expectations* under the distribution, e.g., the computation of $\mathbb{E}[g(\mathbf{Z})]$ where g is some measurable function and \mathbf{Z} is the random variable whose distribution we wish to characterize.

The workhorse algorithms in this setting are sampling or “simulation” methods, of which the most broadly useful are Markov chain Monte Carlo (MCMC) [16, 33, 69] or sequential Monte Carlo (SMC) [26, 49, 73] approaches. Direct sampling from the distribution of interest – i.e., generating independent and unweighted samples – is typically impossible. However, MCMC and SMC methods generate samples that can nonetheless be used to compute the desired expectations. In the case of MCMC, these samples are correlated, while in the case of importance sampling or SMC, the samples are endowed with weights. Nonzero correlations or nonuniform weights are in a sense the price to be paid for flexibility – for these approaches’ ability to characterize arbitrary probability distributions. But if the correlations between successive MCMC samples decay too slowly, or if importance sampling weights become too nonuniform and the sample population thus degenerates, all these approaches become extremely inefficient. Accordingly, enormous efforts have been devoted to the design of improved MCMC and SMC samplers – schemes that generate more nearly independent or unweighted samples. While these efforts are too diverse to summarize easily, they often rest on the design of improved (and structure-exploiting) proposal mechanisms within the algorithms [3, 22, 24, 26, 34, 37, 53, 62].

As an alternative to the sampling approaches described above, we will consider *transformations* of random variables, or perhaps more abstractly, *transport maps* between probability measures. Let $\mu_{\text{tar}} : \mathcal{B}(\mathbb{R}^n) \rightarrow \mathbb{R}_+$ be a probability measure that we wish to characterize, defined over the Borel σ -algebra on \mathbb{R}^n , and let $\mu_{\text{ref}} : \mathcal{B}(\mathbb{R}^n) \rightarrow \mathbb{R}_+$ be another probability measure from which we can easily generate independent and unweighted samples, e.g., a standard Gaussian. Then a transport map $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ pushes forward μ_{ref} to μ_{tar} if and only if $\mu_{\text{tar}}(A) = \mu_{\text{ref}}(T^{-1}(A))$ for any set $A \in \mathcal{B}(\mathbb{R}^n)$. We can write this compactly as

$$T_\sharp \mu_{\text{ref}} = \mu_{\text{tar}}. \quad (23.1)$$

In simpler terms, imagine generating samples $\mathbf{x}_i \in \mathbb{R}^n$ that are distributed according to μ_{ref} and then applying T to each of these samples. Then the transformed samples $T(\mathbf{x}_i)$ are distributed according to μ_{tar} .

Setting aside for a moment questions of how to find such a transformation T and what its properties might be, consider the significance of having T in hand. First of all, given T and the ability to sample directly from μ_{ref} , one can generate independent and unweighted samples from μ_{tar} and from any of its marginal distributions. Moreover one can generate these samples cheaply, regardless of the cost of evaluating the probability density associated with μ_{tar} ; with a map T in hand,

no further appeals to μ_{tar} are needed. A transport map can also be used to devise deterministic sampling approaches, i.e., quadratures for nonstandard measures μ_{tar} , based on quadratures for the reference measure μ_{ref} . Going further, if the transport map is endowed with an appropriate structure, it can enable direct simulation from particular conditionals of μ_{tar} . (We will describe this last point in more detail later.)

The potential to accomplish all of these tasks using measure transport is the launching point for this chapter. We will present a variational approach to the construction of transport maps, i.e., characterizing the desired maps as the solutions of particular optimization problems. We will also discuss the parameterization of transport maps – a challenging task since maps are, in general, high-dimensional multivariate functions, which ultimately must be approximated in finite-dimensional spaces. Because maps are sought via optimization, standard tools for assessing convergence can be used. In particular, it will be useful to quantify the error incurred when the map is not exact, i.e., when we have only $T_{\#}\mu_{\text{ref}} \approx \mu_{\text{tar}}$, and to develop strategies for refining the map parameterization in order to reduce this error. More broadly, it will be useful to understand how the structure of the transport map (e.g., sparsity and other low-dimensional features) depends on the properties of the target distribution and how this structure can be exploited to construct and represent maps more efficiently.

In discussing these issues, we will focus on two classes of map construction problems: (P1) constructing transport maps given the ability to evaluate only the unnormalized probability density of the target distribution and (P2) constructing transport maps given only *samples* from a distribution of interest, but no explicit density. These problems are frequently motivated by Bayesian inference, though their applicability is more general. In the Bayesian context, the first problem corresponds to the typical setup where one can evaluate the unnormalized density of the Bayesian posterior. The second problem can arise when one has samples from the joint distribution of parameters and observations and wishes to condition the former on a particular realization of the latter. This situation is related to approximate Bayesian computation (ABC) [9, 52], and here our ultimate goal is conditional simulation.

This chapter will present the basic formulations employed in the transport map framework and illustrate them with simple numerical examples. First, in Sect. 2, we will recall foundational notions in measure transport and *optimal transportation*. Then we will present variational formulations corresponding to Problems P1 and P2 above. In particular, Sect. 3 will explore the details of Problem P1: constructing transport maps from density evaluations. Section 4 will explore Problem P2: sample-based map construction. Section 5 then discusses useful finite-dimensional parameterizations of transport maps. After presenting these formulations, we will describe connections between the transport map framework and other work in Sect. 6. In Sect. 7 we describe how to simulate from certain conditionals of the target measure using a map, and in Sect. 8 we illustrate the framework on a Bayesian inference problem, including some comparisons with MCMC. Because this area is rapidly developing, this chapter will not attempt to capture all of the latest efforts and extensions. Instead, we will provide the reader with pointers to current work in Sect. 9, along with a summary of important open issues.

2 Transport Maps and Optimal Transport

A transport map T satisfying (23.1) can be understood as a deterministic coupling of two probability measures, and it is natural to ask under what conditions on μ_{ref} and μ_{tar} such a map exists. Consider, for instance, the case where μ_{ref} has as an atom but μ_{tar} does not; then there is no deterministic map that can push forward μ_{ref} to μ_{tar} , since the probability contained in the atom cannot be split. Fortunately, the conditions for the existence of a map are quite weak – for instance, that μ_{ref} be atomless [85]. Henceforth we will assume that both the reference measure μ_{ref} and the target measure μ_{tar} are absolutely continuous with respect to the Lebesgue measure on \mathbb{R}^n , thus assuring the existence of transport maps satisfying (23.1).

There may be infinitely many such transformations, however. One way of choosing a particular map is to introduce a transport cost $c : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ such that $c(\mathbf{x}, \mathbf{z})$ represents the “work” needed to move a unit of mass from \mathbf{x} to \mathbf{z} . The resulting cost of a particular map is then

$$C(T) = \int_{\mathbb{R}^n} c(\mathbf{x}, T(\mathbf{x})) \, d\mu_{\text{ref}}(\mathbf{x}). \quad (23.2)$$

Minimizing (23.2) while simultaneously satisfying (23.1) corresponds to a problem first posed by Monge [57] in 1781. The solution of this constrained minimization problem is the *optimal* transport map. Numerous properties of optimal transport have been studied in the centuries since. Of particular interest is the result of [15], later extended by [55], which shows that when $c(\mathbf{x}, T(\mathbf{x}))$ is quadratic and μ_{ref} is atomless, the optimal transport map exists and is unique; moreover this map is the gradient of a convex function and thus is monotone. Generalizations of this result accounting for different cost functions and spaces can be found in [2, 11, 19, 27]. For a thorough contemporary development of optimal transport, we refer to [84, 85]. The structure of the optimal transport map follows not only from the target and reference measures but also from the cost function in (23.2). For example, the quadratic cost of [15] and [55] leads to maps that are in general dense, i.e., with each output of the map depending on every input to the map. However, if the cost is taken to be

$$c(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^n t^{i-1} |x_i - z_i|^2, \quad t > 0, \quad (23.3)$$

then [18] and [13] show that the optimal map becomes lower triangular as $t \rightarrow 0$. Lower triangular maps take the form

$$T(\mathbf{x}) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \dots, x_n) \end{bmatrix} \quad \forall \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n, \quad (23.4)$$

where T^i represents output i of the map. Importantly, when μ_{tar} and μ_{ref} are absolutely continuous, a unique lower triangular map satisfying (23.1) exists; this map is exactly the Knothe–Rosenblatt rearrangement [13, 18, 70].

The numerical computation of the optimal transport map, for generic measures on \mathbb{R}^n , is a challenging task that is often restricted to very low dimensions [4, 10, 38, 50]. Fortunately, in the context of stochastic simulation and Bayesian inference, we are not particularly concerned with the optimality aspect of the transport; we just need to push forward the reference measure to the target measure. Thus we will focus on transports that are easy to compute, but that do not necessarily satisfy an optimality criterion based on transport cost. Triangular transports will thus be of particular interest to us. The triangular structure will make constructing transport maps feasible (see Sects. 3 and 4), conditional sampling straightforward (see Sect. 7), and map inversion efficient (see Sect. 4.3). Accordingly, we will require the transport map to be (lower) triangular and search for a transformation that satisfies (23.1). The optimization problems arising from this formulation are described in the next two sections.

3 Direct Transport: Constructing Maps from Unnormalized Densities

In this section we show how to construct a transport map that pushes forward a reference measure to the target measure when only evaluations of the *unnormalized target density* are available. This is a central task in Bayesian inference, where the target is the posterior measure.

3.1 Preliminaries

We assume that both target and reference measures are absolutely continuous with respect to the Lebesgue measure on \mathbb{R}^n . Let π and η be, respectively, the normalized target and reference densities with respect to the Lebesgue measure. In what follows and for the sake of simplicity, we assume that both π and η are smooth strictly positive functions on their support. We seek a diffeomorphism T (a smooth function with smooth inverse) that pushes forward the reference to the target measure,

$$\mu_{\text{tar}} = \mu_{\text{ref}} \circ T^{-1}, \quad (23.5)$$

where \circ denotes the composition of functions. In terms of densities, we will rewrite (23.5) as $T_{\sharp}\eta = \pi$ over the support of the reference density. $T_{\sharp}\eta$ is the pushforward of the reference density under the map T , and it is defined as:

$$T_{\sharp}\eta := \eta \circ T^{-1} |\det \nabla T^{-1}|, \quad (23.6)$$

where ∇T^{-1} denotes the Jacobian of the inverse of the map. (Recall that the Jacobian determinant $\det \nabla T^{-1}$ is equal to $1/(\det \nabla T \circ T^{-1})$.) As noted in the introduction, if $(\mathbf{x}_i)_i$ are independent samples from η , then $(T(\mathbf{x}_i))_i$ are independent samples from $T_{\sharp}\eta$. (Here and throughout the chapter, we use the notation $(\mathbf{x}_i)_i$ as a shorthand for $(\mathbf{x}_i)_{i=1}^M$ to denote a collection $(\mathbf{x}_1, \dots, \mathbf{x}_M)$ whenever the definition of the cardinality M is either unimportant or possibly infinite.) Hence, if we find a transport map T that satisfies $T_{\sharp}\eta = \pi$, then $(T(\mathbf{x}_i))_i$ will be independent samples from the target distribution. In particular, the change of variables formula:

$$\int g(\mathbf{x}) \pi(\mathbf{x}) d\mathbf{x} = \int [g \circ T](\mathbf{x}) \eta(\mathbf{x}) d\mathbf{x} \quad (23.7)$$

holds for any integrable real-valued function g on \mathbb{R}^n [1]. The map therefore allows for direct computation of posterior expectations.

3.2 Optimization Problems

Now we describe a series of optimization problems whose solution yields the desired transport map. Let $\mathcal{D}_{\text{KL}}(\pi_1 \parallel \pi_2)$ denote the Kullback–Leibler (K–L) divergence from a probability measure with density π_1 to a probability measure with density π_2 , i.e.,

$$\mathcal{D}_{\text{KL}}(\pi_1 \parallel \pi_2) = \mathbb{E}_{\pi_1} \left(\log \frac{\pi_1}{\pi_2} \right),$$

and let \mathcal{T} be an appropriate set of diffeomorphisms. Then, any global minimizer of the optimization problem:

$$\begin{aligned} & \min \mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi) \\ & \text{s.t. } \det \nabla T > 0, \quad (\eta - \text{a.e.}) \\ & \qquad T \in \mathcal{T} \end{aligned} \quad (23.8)$$

is a valid transport map that pushes forward the reference to the target measure¹. In fact, any global minimizer of (23.8) achieves the minimum cost $\mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi) = 0$ and implies that $T_{\sharp}\eta = \pi$. The constraint $\det \nabla T > 0$ ensures that the pushforward density $T_{\sharp}\eta$ is strictly positive on the support of the target. In particular, the constraint $\det \nabla T > 0$ ensures that the K–L divergence evaluates to finite values over \mathcal{T} and does not rule out any useful transport map since we assume that both target and reference densities are positive. The existence of global minimizers of (23.8) is a standard result in the theory of deterministic couplings between random variables [85].

¹See [61] for a discussion on the asymptotic equivalence of the K–L divergence and Hellinger distance in the context of transport maps.

Among these minimizers, a particularly useful map is given by the Knothe–Rosenblatt rearrangement [18]. In our hypothesis, the Knothe–Rosenblatt rearrangement is a *triangular* (in the sense that the k th component of the map depends only on the first k input variables) diffeomorphism T such that $\nabla T \succ 0$. That is, each eigenvalue of ∇T is real and positive. Thus it holds that $\det \nabla T > 0$. Notice that for a triangular map, the eigenvalues of ∇T are just the diagonal entries of this matrix. The Knothe–Rosenblatt rearrangement is also monotone increasing according to the lexicographic order on \mathbb{R}^{n^2} .

It turns out that we can further constrain (23.8) so that the Knothe–Rosenblatt rearrangement is the *unique* global minimizer of:

$$\begin{aligned} \min \mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi) \\ \text{s.t. } \nabla T \succ 0, \quad (\eta - \text{a.e.}) \\ T \in \mathcal{T}_{\Delta} \end{aligned} \tag{23.9}$$

where \mathcal{T}_{Δ} is now the vector space of smooth triangular maps. The constraint $\nabla T \succ 0$ suffices to enforce invertibility of a feasible triangular map. Equation (23.9) is a far better behaved optimization problem than the original formulation (23.8). Hence, for the rest of this section, we will focus on the computation of a Knothe–Rosenblatt rearrangement by solving (23.9). Recall that our goal is just to compute a transport map from μ_{ref} to μ_{tar} . If there are multiple transports, we can opt for the easiest one to compute. A possible drawback of a triangular transport is that the complexity of a parameterization of the map depends on the ordering of the input variables. This dependence motivates questions of what is the “best” ordering, or how to find at least a “good” ordering. We refer the reader to [75] for an in-depth discussion of this topic. For the computation of general non-triangular transports, see [61]; for a generalization of the framework to compositions of maps, see [61, 63]; and for the computation of optimal transports, see, for instance, [4, 10, 50, 85].

Now let $\bar{\pi}$ denote any *unnormalized* version of the target density. For any map T in the feasible set of (23.9), the objective function can be written as:

$$\begin{aligned} \mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi) &= \mathcal{D}_{\text{KL}}(\eta \parallel T_{\sharp}^{-1}\pi) \\ &= \mathbb{E}_{\eta}[-\log \bar{\pi} \circ T - \log \det \nabla T] + \mathfrak{C}, \end{aligned} \tag{23.10}$$

²The lexicographic order on \mathbb{R}^n is defined as follows. For $x, y \in \mathbb{R}^n$, we define $x \preceq y$ if and only if either $x = y$ or the first nonzero coordinate in $y - x$ is positive [32]. \preceq is a total order on \mathbb{R}^n . Thus, we define T to be a monotone increasing function if and only if $x \preceq y$ implies $T(x) \preceq T(y)$. Notice that monotonicity can be defined with respect to any order on \mathbb{R}^n (e.g., \preceq need not be the lexicographic order). There is no natural order on \mathbb{R}^n except when $n = 1$. It is easy to verify that for a triangular function T , monotonicity with respect to the lexicographic order is equivalent to the following: the k th component of T is a monotone function of the k th input variable.

where $\mathbb{E}_\eta[\cdot]$ denotes integration with respect to the reference measure, and \mathfrak{C} is a term independent of the transport map and thus a constant for the purposes of optimization. (In this case, $\mathfrak{C} = \log \beta + \log \eta$, where $\beta := \bar{\pi}/\pi$ is the normalizing constant of the target density.) The resulting optimization problem reads as:

$$\begin{aligned} & \min \mathbb{E}_\eta[-\log \bar{\pi} \circ T - \log \det \nabla T] \\ & \text{s.t. } \nabla T \succ 0, \quad (\eta \text{-a.e.}) \\ & \quad T \in \mathcal{T}_\Delta \end{aligned} \tag{23.11}$$

Notice that we can evaluate the objective of (23.11) given only the unnormalized density $\bar{\pi}$ and a way to compute the integral $\mathbb{E}_\eta[\cdot]$. There exist a host of techniques to approximate the integral with respect to the reference measure, including quadrature and cubature formulas, sparse quadratures, Monte Carlo methods, and quasi-Monte Carlo (QMC) methods. The choice between these methods is typically dictated by the dimension of the reference space. In any case, the reference measure is usually chosen so that the integral with respect to η can be approximated easily and accurately. For instance, if μ_{ref} is a standard Gaussian measure, then we can generate arbitrarily many independent samples to yield an approximation of $\mathbb{E}_\eta[\cdot]$ to any desired accuracy. This will be a crucial difference relative to the sample-based construction of the map described in Sect. 4, where samples from the target distribution are required to accurately solve the corresponding optimization problem.

Equation (23.11) is a linearly constrained nonlinear differentiable optimization problem. It is also non-convex unless, for instance, the target density is log concave [41]. That said, many statistical models have log-concave posterior distributions and hence yield convex map optimization problems; consider, for example, a log-Gaussian Cox process [34]. All n components of the map have to be computed simultaneously, and each evaluation of the objective function of (23.11) requires an evaluation of the unnormalized target density. The latter is also the minimum requirement for alternative sampling techniques such as MCMC. Of course, the use of derivatives in the context of the optimization problem is crucial for computational efficiency, especially for high-dimensional parameterizations of the map. The same can be said of the state-of-the-art MCMC algorithms that use gradient or Hessian information from the log-target density to yield better proposal distributions, such as Langevin or Hamiltonian MCMC [34]. In the present context, we advocate the use of quasi-Newton (e.g., BFGS) or Newton methods [90] to solve (23.11). These methods must be paired with a finite-dimensional parameterization of the map; in other words, we must solve (23.11) over a finite-dimensional space $\mathcal{T}_\Delta^h \subset \mathcal{T}_\Delta$ of triangular diffeomorphisms. In Sect. 5 we will discuss various choices for \mathcal{T}_Δ^h along with ways of enforcing the monotonicity constraint $\nabla T \succ 0$.

3.3 Convergence, Bias, and Approximate Maps

A transport map provides a deterministic solution to the problem of sampling from a given unnormalized density, avoiding classical stochastic tools such as MCMC. Once the transport map is computed, we can quickly generate independent and unweighted samples from the target distribution without further evaluating the target density [61]. This is a major difference with respect to MCMC. Of course, the density-based transport map framework essentially exchanges a challenging sampling task for a challenging optimization problem involving function approximation. Yet there are several advantages to dealing with an optimization problem rather than a sampling problem. Not only can we rely on a rich literature of robust algorithms for the solution of high-dimensional nonlinear optimization problems, but we also inherit the notion of convergence criteria. The latter point is crucial.

A major concern in MCMC sampling methods is the lack of clear and generally applicable convergence criteria. It is a nontrivial task to assess the stationarity of an ergodic Markov chain, let alone to measure the distance between the empirical measure given by the MCMC samples and the target distribution [36]. In the transport map framework, on the other hand, the convergence criterion is borrowed directly from standard optimization theory [51]. As shown in [61], the K–L divergence $\mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi)$ can be estimated as:

$$\mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi) \approx \frac{1}{2} \text{Var}_{\eta}[\log \eta - \log T_{\sharp}^{-1}\bar{\pi}] \quad (23.12)$$

up to second-order terms in the limit of $\text{Var}_{\eta}[\log \eta - \log T_{\sharp}^{-1}\bar{\pi}] \rightarrow 0$, even if the normalizing constant of the target density is unknown. (Notice that (23.12) contains only the unnormalized target density $\bar{\pi}$.) Thus one can monitor (23.12) to estimate the divergence between the pushforward of a given map and the desired target distribution. Moreover, the transport map algorithm also provides an estimate of the normalizing constant $\beta := \bar{\pi}/\pi$ of the target density as [61]:

$$\beta = \exp \mathbb{E}_{\eta}[\log \eta - \log T_{\sharp}^{-1}\bar{\pi}] . \quad (23.13)$$

The normalizing constant is a particularly useful quantity in the context of Bayesian model selection [30]. Reliably retrieving this normalizing constant from MCMC samples requires additional effort (e.g., [20, 31]). The numerical solution of (23.11) entails at least two different approximations. First, the infinite-dimensional function space \mathcal{T}_{Δ} must be replaced with a finite-dimensional subspace $\mathcal{T}_{\Delta}^h \subset \mathcal{T}_{\Delta}$. For example, each component of the map can be approximated in a total-degree polynomial space, as discussed in Sect. 5. Let h parameterize a sequence of possibly nested finite-dimensional approximation spaces $(\mathcal{T}_{\Delta}^h)_h$. Then, as the dimension of \mathcal{T}_{Δ}^h grows, we can represent increasingly complex maps. Second, the expectation with respect to the reference measure in the objective of (23.11) must also be approximated. As discussed earlier, one may take any of several approaches. As a concrete example, consider approximating $\mathbb{E}_{\eta}[\cdot]$ by a Monte Carlo sum with M

independent samples $(\mathbf{x}_i)_i$ from the reference measure. Clearly, as the cardinality of the sample set grows, the approximation of $\mathbb{E}_\eta[\cdot]$ becomes increasingly accurate. An instance of an approximation of (23.11) is then:

$$\begin{aligned} \min \frac{1}{M} \sum_{i=1}^M & \left(-\log \tilde{\pi}(T(\mathbf{x}_i)) - \sum_{k=1}^n \log \partial_k T^k(\mathbf{x}_i) \right) \\ \text{s.t. } & \partial_k T^k > 0, \quad k = 1, \dots, n, \quad (\eta - \text{a.e.}) \\ & T \in \mathcal{T}_\Delta^h \subset \mathcal{T}_\Delta \end{aligned} \quad (23.14)$$

where we have simplified the monotonicity constraint $\nabla T > 0$ by using the fact that ∇T is lower triangular for maps in \mathcal{T}_Δ . Above we require that the monotonicity constraint be satisfied over the whole support of the reference density. We will discuss ways of strictly guaranteeing such a property in Sect. 5. Depending on the parameterization of the map, however, the monotonicity constraint is sometimes relaxed (e.g., in the tails of η); doing so comprises a third source of approximation.

Given a fixed sample set, (23.14) is a sample-average approximation (SAA) [42] of (23.11), to which we can apply standard numerical optimization techniques [90]. Alternatively, one can regard (23.11) as a stochastic program and solve it using stochastic approximation techniques [43, 74]. In either case, the transport map framework allows efficient *global* exploration of the parameter space via optimization. The exploration is global since with a transport map T we are essentially trying to push forward the entire collection of reference samples $(\mathbf{x}_i)_i$ to samples $(T(\mathbf{x}_i))_i$ that fit the entire target distribution. Additionally, we can interpret the finite-dimensional parameterization of a candidate transport as a constraint on the relative motion of the pushforward particles $(T(\mathbf{x}_i))_i$.

The discretization of the integral $\mathbb{E}_\eta[\cdot]$ in (23.14) reveals another important distinction of the transport map framework from MCMC methods. At every optimization iteration, we need to evaluate the target density M times. If we want an accurate approximation of $\mathbb{E}_\eta[\cdot]$, then M can be large. But these M evaluations of the target density can be performed in an embarrassingly *parallel* manner. This is a fundamental difference from standard MCMC, where the evaluations of the target density are inherently sequential. (For exceptions to this paradigm, see, e.g., [17]).

A minimizer of (23.14) is an approximate transport map \tilde{T} ; this map can be written as $\tilde{T}(\cdot; M, h)$ to reflect dependence on the approximation parameters (M, h) defined above. Ideally, we would like the approximate map \tilde{T} to be as close as possible to the true minimizer $T \in \mathcal{T}_\Delta$ of (23.11). Yet it is also important to understand the potential of an approximate map alone. If \tilde{T} is not the exact transport map, then $\tilde{T}_\# \eta$ will not be the target density. The pushforward density $\tilde{T}_\# \eta$ instead defines an approximate target: $\tilde{\pi} := \tilde{T}_\# \eta$. We can easily sample from $\tilde{\pi}$ by pushing forward reference samples through \tilde{T} . If we are interested in estimating integrals of the form $\mathbb{E}_\pi[g]$ for some integrable function g , then we can try to use Monte Carlo estimators of $\mathbb{E}_{\tilde{\pi}}[g]$ to approximate $\mathbb{E}_\pi[g]$. This procedure will result in a biased

estimator for $\mathbb{E}_\pi[g]$ since $\tilde{\pi}$ is not the target density. It turns out that this bias can be bounded as:

$$\|\mathbb{E}_\pi[g] - \mathbb{E}_{\tilde{\pi}}[g]\| \leq \mathcal{C}(g, \pi, \tilde{\pi}) \sqrt{\mathcal{D}_{\text{KL}}(T_\sharp \eta || \pi)} \quad (23.15)$$

where $\mathcal{C}(g, \pi, \tilde{\pi}) := \sqrt{2} (\mathbb{E}_\pi[\|g\|^2] + \mathbb{E}_{\tilde{\pi}}[\|g\|^2])^{\frac{1}{2}}$. The proof of this result is in [79, Lemma 6.37] together with a similar result for the approximation of the second moments. Note that the K–L divergence on the right-hand side of (23.15) is exactly the quantity we minimize in (23.11) during the computation of a transport map, and it can easily be estimated using (23.12). Thus the transport map framework allows a systematic control of the bias resulting from estimation of $\mathbb{E}_\pi[g]$ by means of an approximate map \tilde{T} .

In practice, the mean-square error in approximating $\mathbb{E}_\pi[g]$ using $\mathbb{E}_{\tilde{\pi}}[g]$ will be entirely due to the bias described in (23.15). The reason is that a Monte Carlo estimator of $\mathbb{E}_{\tilde{\pi}}[g]$ can be constructed to have virtually no variance: one can cheaply generate an arbitrary number of independent and unweighted samples from $\tilde{\pi}$ using the approximate map. Hence the approximate transport map yields an essentially zero-variance biased estimator of the quantity of interest $\mathbb{E}_\pi[g]$. This property should be contrasted with MCMC methods which, while asymptotically unbiased, yield estimators of $\mathbb{E}_\pi[g]$ that have nonzero variance and bias for finite sample size.

If one is not content with the bias associated with an approximate map, then there are at least two ways to proceed. First, one can simply refine the approximation parameters (M, h) to improve the current approximation of the transport map. On the other hand, it is straightforward to apply any classical sampling technique (e.g., MCMC, importance sampling) to the *pullback* density $\tilde{T}^\# \pi = \tilde{T}_\sharp^{-1} \pi$. This density is defined as

$$\tilde{T}^\# \pi := \pi \circ \tilde{T} |\det \nabla \tilde{T}|, \quad (23.16)$$

and can be evaluated (up to a normalizing constant) at no significant additional cost compared to the original target density π . If $(\mathbf{x}_i)_i$ are samples from the pullback $\tilde{T}^\# \pi$, then $(\tilde{T}(\mathbf{x}_i))_i$ will be samples from the *exact* target distribution. But there are clear advantages to sampling $\tilde{T}^\# \pi$ instead of the original target distribution. Consider the following: if \tilde{T} were the exact transport map, then $\tilde{T}^\# \pi$ would simply be the reference density. With an approximate map, we still expect $\tilde{T}^\# \pi$ to be close to the reference density – more precisely, closer to the reference (in the sense of K–L divergence) than was the original target distribution. In particular, when the reference is a standard Gaussian, the pullback will be closer to a standard Gaussian than the original target. Pulling back through an approximate map thus “Gaussianizes” the target and can remove the correlations and nonlinear dependencies that make sampling a challenging task. In this sense, we can interpret an approximate map as a general *preconditioner* for any known sampling scheme. See [64] for a full development of this idea in the context of MCMC.

There is clearly a trade-off between the computational cost associated with constructing a more accurate transport map and the costs of “correcting” an approximate map by applying an exact sampling scheme to the pullback. Focusing on the former, it is natural to ask how to refine the finite-dimensional approximation space \mathcal{T}_Δ^h so that it can better capture the true map T . Depending on the problem at hand, a naïve finite-dimensional parameterization of the map might require a very large number of degrees of freedom before reaching a satisfactory approximation. This is particularly true when the parameter dimension n is large and is a challenge shared by any function approximation algorithm (e.g., high-dimensional regression). We will revisit this issue in Sect. 9, but the essential way forward is to realize that the structure of the target distribution is reflected in the structure of the transport map. For instance, conditional independence in the target distribution yields certain kinds of *sparsity* in the transport, which can be exploited when solving the optimization problems above. Many Bayesian inference problems also contain low-rank structure that causes the map to depart from the identity only on low-dimensional subspace of \mathbb{R}^n . From the optimization perspective, adaptivity can also be driven via a systematic analysis of the first variation of the K–L divergence $\mathcal{D}_{\text{KL}}(T_\sharp \eta \parallel \pi)$ as a function of $T \in \mathcal{T}$.

4 Inverse Transport: Constructing Maps from Samples

In the previous section, we focused on the computation of a transport map that pushes forward a reference measure to a target measure, in settings where the target density can be evaluated up to a normalizing constant and where it is simple to approximate integrals with respect to the reference measure (e.g., using quadrature, Monte Carlo, or QMC). In many problems of interest, including density estimation [81] and approximate Bayesian computation, however, it is not possible to evaluate the unnormalized target density $\bar{\pi}$.

In this section we assume that the target density is unknown and that we are only given a finite number of samples distributed according to the target measure. We show that under these hypotheses, it is possible to efficiently compute an *inverse transport* – a transport map that pushes forward the target to the reference measure – via convex optimization. The direct transport – a transport map that pushes forward the reference to the target measure – can then be easily recovered by inverting the inverse transport map pointwise, taking advantage of the map’s triangular structure.

4.1 Optimization Problem

We denote the inverse transport by $S : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and again assume that the reference and target measures are absolutely continuous with respect to the Lebesgue measure on \mathbb{R}^n , with smooth and positive densities. The inverse transport pushes forward the target to the reference measure:

$$\mu_{\text{ref}} = \mu_{\text{tar}} \circ S^{-1}. \quad (23.17)$$

We focus on the inverse triangular transport because it can be computed via convex optimization given samples from the target distribution. It is easy to see that the monotone increasing Knothe–Rosenblatt rearrangement that pushes forward μ_{tar} to μ_{ref} is the unique minimizer of

$$\begin{aligned} & \min \mathcal{D}_{\text{KL}}(S_{\sharp}\pi \parallel \eta) \\ \text{s.t. } & \nabla S \succ 0, \quad (\pi - \text{a.e.}) \\ & S \in \mathcal{T}_{\Delta} \end{aligned} \tag{23.18}$$

where \mathcal{T}_{Δ} is the space of smooth triangular maps. If S is a minimizer of (23.18), then $\mathcal{D}_{\text{KL}}(S_{\sharp}\pi \parallel \eta) = 0$ and thus $S_{\sharp}\pi = \eta$. For any map S in the feasible set of (23.18), the objective function can be written as:

$$\begin{aligned} \mathcal{D}_{\text{KL}}(S_{\sharp}\pi \parallel \eta) &= \mathcal{D}_{\text{KL}}(\pi \parallel S_{\sharp}^{-1}\eta) \\ &= \mathbb{E}_{\pi}[-\log \eta \circ S - \log \det \nabla S] + \mathfrak{C} \end{aligned} \tag{23.19}$$

where $\mathbb{E}_{\pi}[\cdot]$ denotes integration with respect to the target measure, and where \mathfrak{C} is once again a term independent of the transport map and thus a constant for the purposes of optimization. The resulting optimization problem is a stochastic program given by:

$$\begin{aligned} & \min \mathbb{E}_{\pi}[-\log \eta \circ S - \log \det \nabla S] \\ \text{s.t. } & \nabla S \succ 0, \quad (\pi - \text{a.e.}) \\ & S \in \mathcal{T}_{\Delta} \end{aligned} \tag{23.20}$$

Notice that (23.20) is equivalent to (23.11) if we interchange the roles of target and reference densities. Indeed, the K–L divergence is not a symmetric function. The direction of the K–L divergence, i.e., $\mathcal{D}_{\text{KL}}(S_{\sharp}\pi \parallel \eta)$ versus $\mathcal{D}_{\text{KL}}(T_{\sharp}\eta \parallel \pi)$, is one of the key distinctions between the sample-based map construction presented in this section and the density-based construction of Sect. 3. The choice of direction in (23.19) involves integration over the target distribution, as in the objective function of (23.20), which we approximate using the given samples. Let $(z_i)_{i=1}^M$ be M samples from the target distribution. Then, a sample-average approximation (SAA) [42] of (23.20) is given by:

$$\begin{aligned} & \min \frac{1}{M} \sum_{i=1}^M -\log \eta(S(z_i)) - \log \det \nabla S(z_i) \\ \text{s.t. } & \partial_k S^k > 0 \quad k = 1, \dots, n, \quad (\pi - \text{a.e.}) \\ & S \in \mathcal{T}_{\Delta} \end{aligned} \tag{23.21}$$

where we use the lower triangular structure of ∇S to rewrite the monotonicity constraint, $\nabla S \succ 0$, as a sequence of essentially one-dimensional monotonicity constraints: $\partial_k S^k > 0$ for $k = 1, \dots, n$. We note, in passing, that the monotonicity constraint can be satisfied automatically by using monotone parameterizations of the triangular transport (see Sect. 5.3). One can certainly use stochastic programming techniques to solve (23.20) depending on the availability of target samples (e.g., stochastic approximation [43, 74]). SAA, on the other hand, turns (23.20) into a deterministic optimization problem and does not require generating new samples from the target distribution, which could involve running additional expensive simulations or performing new experiments. Thus, SAA is generally the method of choice to solve (23.20). However, stochastic approximation may be better suited for applications involving streaming data or massive sample sets requiring single pass algorithms.

4.2 Convexity and Separability of the Optimization Problem

Note that (23.21) is a convex optimization problem as long as the reference density is log concave [41]. Since the reference density is a degree of freedom of the problem, it can always be chosen to be log concave. Thus, (23.21) can be a convex optimization problem *regardless* of the particular structure of the target. This is a major difference from the density-based construction of the map in Sect. 3, where the corresponding optimization problem (23.11) is convex only under certain conditions on the target (e.g., that the target density be log concave).

For smooth reference densities, the objective function of (23.21) is also smooth. Moreover, its gradients do not involve derivatives of the log-target density – which might require expensive adjoint calculations when the target density contains a PDE model. Indeed, the objective function of (23.21) does not contain the target density at all! This feature should be contrasted with the density-based construction of the map, where the objective function of (23.11) depends explicitly on the log-target density. Moreover, if the reference density can be written as the product of its marginals, then (23.21) is a separable optimization problem, i.e., each component of the inverse transport can be computed independently and in parallel.

As a concrete example, let the reference measure be standard Gaussian. In this case, (23.21) can be written as

$$\begin{aligned} \min \frac{1}{M} \sum_{i=1}^M \sum_{k=1}^n \left[\frac{1}{2} (S^k)^2(z_i) - \log \partial_k S^k(z_i) \right] \\ \text{s.t. } \partial_k S^k > 0 \quad k = 1, \dots, n, \quad (\pi - \text{a.e.}) \\ S \in \mathcal{T}_\Delta \end{aligned} \tag{23.22}$$

where we use the identity $\log \det \nabla S \equiv \sum_{k=1}^n \log \partial_k S^k$, which holds for triangular maps. Almost magically, the objective function and the constraining set of (23.22)

are *separable*: the k th component of the inverse transport can be computed as the solution of a single convex optimization problem,

$$\begin{aligned} \min \frac{1}{M} \sum_{i=1}^M \frac{1}{2} (S^k)^2(z_i) - \log \partial_k S^k(z_i) \\ \text{s.t. } \partial_k S^k > 0, \quad (\pi - \text{a.e.}) \\ S^k \in \mathcal{T}_k \end{aligned} \quad (23.23)$$

where \mathcal{T}_k denotes the space of smooth real-valued functions of k variables. Most importantly, (23.23) depends only on the k th component of the map. Thus, all the components of the inverse transport can be computed independently and in parallel by solving optimization problems of the form (23.23). This is another major difference from the density-based construction of the map, where all the components of the transport must be computed simultaneously as a solution of (23.11) (unless, for instance, the target density can be written as the product of its marginals).

As in the previous section, the numerical solution of (23.23) requires replacing the infinite-dimensional function space \mathcal{T}_k with a finite-dimensional subspace $\mathcal{T}_k^h \subset \mathcal{T}_k$. The monotonicity constraint $\partial_k S^k > 0$ can be discretized and enforced at only finitely many points in the parameter space, for instance at the samples $(z_i)_{i=1}^M$ (see Sect. 5). However, a better approach is to use monotone parameterizations of the triangular transport in order to turn (23.23) into an unconstrained optimization problem (see Sect. 5). In either case, the solution of (23.23) over a finite-dimensional approximation space \mathcal{T}_k^h yields a component of the approximate inverse transport. The quality of this approximation is a function of at least two parameters of the problem: the structure of the space \mathcal{T}_k^h and the number of target samples M . While enriching the space \mathcal{T}_k^h is often a straightforward task, increasing the number of target samples can be nontrivial, especially when exact sampling from the target density is impossible. This is an important difference from density-based construction of the map, wherein the objective function of (23.11) only requires integration with respect to the reference measure and thus can be approximated to any desired degree of accuracy during each stage of the computation. Of course, in many cases of interest, exact sampling from the target distribution is possible. Consider, for instance, the joint density of data and parameters in a typical Bayesian inference problem [63] or the forecast distribution in a filtering problem where the forward model is a stochastic difference equation.

Quantifying the quality of an approximate inverse transport is an important issue. If the target density can be evaluated up to a normalizing constant, then the K–L divergence between the pushforward of the target through the map and the reference density can be estimated as

$$\mathcal{D}_{\text{KL}}(S_\# \pi || \eta) \approx \frac{1}{2} \text{Var}_\pi [\log \bar{\pi} - \log S^\# \eta] \quad (23.24)$$

up to second-order terms in the limit of $\mathbb{V}\text{ar}_\pi[\log \bar{\pi} - \log S^\# \eta] \rightarrow 0$. This expression is analogous to (23.12) (see Sect. 3 for further details on this topic). When the target density cannot be evaluated, however, one can rely on statistical tests to monitor convergence to the exact inverse transport. For instance, if the reference density is a standard Gaussian, then we know that pushing forward the target samples $(z_i)_i$ through the inverse transport should yield jointly Gaussian samples, with independent and standard normal components. If the inverse transport is only approximate, then the pushforward samples will not have independent and standard normal components, and one can quantify their deviation from such normality using standard statistical tests [83].

4.3 Computing the Inverse Map

Up to now, we have shown how to compute the triangular inverse transport S via convex optimization given samples from the target density. In many problems of interest, however, the goal is to evaluate the direct transport T , i.e., a map that pushes forward the reference to the target measure. Clearly, the following relationship between the direct and inverse transports holds:

$$T(\mathbf{x}) = S^{-1}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (23.25)$$

Thus, if we want to evaluate the direct transport at a particular $\mathbf{x}^* \in \mathbb{R}^n$, i.e., $\mathbf{z}^* := T(\mathbf{x}^*)$, then by (23.25) we can simply invert S at \mathbf{x}^* to obtain \mathbf{z}^* . In particular, if $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$ and $\mathbf{z}^* = (z_1^*, \dots, z_n^*)$, then \mathbf{z}^* is a solution of the following lower triangular system of equations:

$$S(\mathbf{z}^*) = \begin{bmatrix} S^1(z_1^*) \\ S^2(z_1^*, z_2^*) \\ \vdots \\ S^n(z_1^*, z_2^*, \dots, z_n^*) \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix} = \mathbf{x}^* \quad (23.26)$$

where the k th component of S is just a function of the first k input variables. This system is in general nonlinear, but we can devise a simple recursion in k to compute each component of \mathbf{z}^* as

$$z_k^* := (S_{z_1^*, \dots, z_{k-1}^*}^k)^{-1}(x_k^*), \quad k = 1, \dots, n, \quad (23.27)$$

where $S_{z_1^*, \dots, z_{k-1}^*}^k : \mathbb{R} \rightarrow \mathbb{R}$ is a one-dimensional function defined as $w \mapsto S^k(z_1^*, \dots, z_{k-1}^*, w)$. That is, $S_{z_1^*, \dots, z_{k-1}^*}^k$ is the restriction of the k th component of the inverse transport obtained by fixing the first $k-1$ input variables z_1^*, \dots, z_{k-1}^* . Thus, \mathbf{z}^* can be computed recursively via a sequence of n one-dimensional root-finding problems. Monotonicity of the triangular maps guarantees that (23.27) has a

unique real solution for each k and any given \mathbf{x}^* . Here, one can use any off-the-shelf root-finding algorithm³. Whenever the transport is high-dimensional (e.g., hundreds or thousands of components), this recursive approach might become inaccurate, as it is sequential in nature. In this case, we recommend running a few Newton iterations of the form

$$\mathbf{z}_{j+1} = \mathbf{z}_j - \nabla S(\mathbf{z}_j)^{-1}(S(\mathbf{z}_j) - \mathbf{x}^*) \quad (23.28)$$

to clean up the approximation of the root \mathbf{z}^* obtained from the recursive algorithm (23.27).

An alternative way to evaluate the direct transport is to build a parametric representation of T itself via standard regression techniques. In particular, if $\{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ are samples from the target distribution, then $\{\mathbf{x}_1, \dots, \mathbf{x}_M\}$, with $\mathbf{x}_k := S(\mathbf{z}_k)$ for $k = 1, \dots, M$, are samples from the reference distribution. Note that there is a one-to-one correspondence between target and reference samples. Thus, we can use these pairs of samples to define a simple constrained least-squares problem to approximate the direct transport as:

$$\begin{aligned} & \min \sum_{k=1}^M \sum_{i=1}^n (T^i(\mathbf{x}_k) - z_k)^2 \\ & \text{s.t. } \partial_i T^i > 0 \quad i = 1, \dots, n, \quad (\eta - \text{a.e.}) \\ & \quad T \in \mathcal{T}_\Delta. \end{aligned} \quad (23.29)$$

In particular, each component of the direct transport can be approximated independently (and in parallel) as the minimizer of

$$\begin{aligned} & \min \sum_{k=1}^M (T^i(\mathbf{x}_k) - z_k)^2 \\ & \text{s.t. } \partial_i T^i > 0, \quad (\eta - \text{a.e.}) \\ & \quad T^i \in \mathcal{T}_i, \end{aligned} \quad (23.30)$$

where \mathcal{T}_i denotes the space of smooth real-valued functions of i variables. Of course, the numerical solution of (23.31) requires the suitable choice of a finite-dimensional approximation space $\mathcal{T}_i^h \subset \mathcal{T}_i$.

³Roots can be found using, for instance, Newton's method. When a component of the inverse transport is parameterized using polynomials, however, then a more robust root-finding approach is to use a bisection method based on Sturm sequences (e.g., [63]).

5 Parameterization of Transport Maps

As noted in the previous sections, the optimization problems that one solves to obtain either the direct or inverse transport must, at some point, introduce discretization. In particular, we must define finite-dimensional approximation spaces (e.g., \mathcal{T}_Δ^h) within which we search for a best map. In this section we describe several useful choices for \mathcal{T}_Δ^h and the associated map parameterizations. Closely related to the map parameterization is the question of how to enforce the monotonicity constraints $\partial_k T^k > 0$ or $\partial_k S^k > 0$ over the support of the reference and target densities, respectively. For some parameterizations, we will explicitly introduce discretizations of these monotonicity constraints. A different map parameterization, discussed in Sect. 5.3, will satisfy these monotonicity conditions automatically.

For simplicity, we will present the parameterizations below mostly in the context of the direct transport T . But these parameterizations can be used interchangeably for both the direct and inverse transports.

5.1 Polynomial Representations

A natural way to parameterize each component of the map T is by expanding it in a basis of multivariate polynomials. We define each multivariate polynomial ψ_j as a product of n univariate polynomials, specified via a multi-index $\mathbf{j} = (j_1, j_2, \dots, j_n) \in \mathbb{N}_0^n$, as:

$$\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^n \varphi_{j_i}(x_i), \quad (23.31)$$

where φ_{j_i} is a univariate polynomial of degree j_i . The univariate polynomials can be chosen from any standard orthogonal polynomial family (e.g., Hermite, Legendre, Laguerre) or they can even be monomials. That said, it is common practice in uncertainty quantification to choose univariate polynomials that are orthogonal with respect to the input measure, which in the case of the direct transport is μ_{ref} . If μ_{ref} is a standard Gaussian, the $(\varphi_i)_i$ above would be (suitably scaled and normalized) Hermite polynomials. The resulting map can then be viewed as a *polynomial chaos* expansion [46, 91] of a random variable distributed according to the target measure. From the coefficients of this polynomial expansion, moments of the target measure can be directly – that is, *analytically* – evaluated. In the case of inverse transports, however, μ_{tar} is typically not among the canonical distributions found in the Askey scheme for which standard orthogonal polynomials can be easily evaluated. While it is possible to construct orthogonal polynomials for more general measures [29], the relative benefits of doing so are limited, and hence with inverse transports we do not employ a basis orthogonal with respect to μ_{tar} .

Using multivariate polynomials given in (23.31), we can express each component of the transport map $T \in \mathcal{T}_\Delta^h$ as

$$T^k(\mathbf{x}) = \sum_{\mathbf{j} \in \mathcal{J}_k} \gamma_{k,\mathbf{j}} \psi_{\mathbf{j}}(\mathbf{x}), \quad k = 1, \dots, n, \quad (23.32)$$

where \mathcal{J}_k is a set of multi-indices defining the polynomial terms in the expansion for dimension k and $\gamma_{k,\mathbf{j}} \in \mathbb{R}$ is a scalar coefficient. Importantly, the proper choice of each multi-index set \mathcal{J}_k will force T to be lower triangular. For instance, a standard choice of \mathcal{J}_k involves restricting each map component to a total-degree polynomial space:

$$\mathcal{J}_k^{TO} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p \wedge j_i = 0, \forall i > k\}, \quad k = 1, \dots, n \quad (23.33)$$

The first constraint in this set, $\|\mathbf{j}\|_1 \leq p$, limits the total degree of each polynomial to p , while the second constraint, $j_i = 0, \forall i > k$, forces T to be lower triangular. Expansions built using \mathcal{J}_d^{TO} are quite “expressive” in the sense of being able to capture complex nonlinear dependencies in the target measure. However, the number of terms in \mathcal{J}_k^{TO} grows rapidly with k and p . A smaller multi-index set can be obtained by removing all the mixed terms in the basis:

$$\mathcal{J}_k^{NM} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p \wedge j_i j_\ell = 0, \forall i \neq \ell \wedge j_i = 0, \forall i > k\}.$$

An even more parsimonious option is to use diagonal maps, via the multi-index sets

$$\mathcal{J}_k^D = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p \wedge j_i = 0, \forall i \neq k\}.$$

Figure 23.1 illustrates the difference between these three sets for $p = 3$ and $k = 2$.

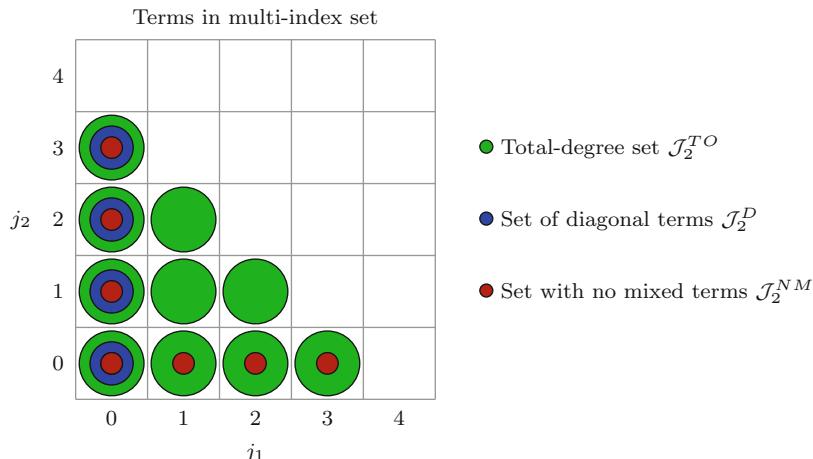


Fig. 23.1 Visualization of multi-index sets for the second component of a two-dimensional map, $T^2(x_1, x_2)$. In this case, j_1 is the degree of a basis polynomial in x_1 and j_2 is the degree in x_2 . A filled circle indicates that a term is present in the set of multi-indices

An alternative to using these standard and isotropic bases is to adapt the polynomial approximation space to the problem at hand. This becomes particularly important in high dimensions. For instance, beginning with linear maps (e.g., \mathcal{T}_k^{TO} with $p = 1$) [61] introduces an iterative scheme for enriching the polynomial basis, incrementing the degree of \mathcal{T}_{Δ}^h in a few input variables at a time. Doing so enables the construction of a transport map in $O(100)$ dimensions. In a different context, [65] uses the conditional independence structure of the posterior distribution in a multiscale inference problem to enable map construction in $O(1000)$ dimensions. Further comments on adapting \mathcal{T}_{Δ}^h are given in Sect. 9.

5.2 Radial Basis Functions

An alternative to a polynomial parameterization of the map is to employ a combination of linear terms and radial basis functions. This representation can be more efficient than a polynomial representation in certain cases – for example, when the target density is multimodal. The general form of the expansion in (23.32) remains the same, but we replace polynomials of degree greater than one with radial basis functions as follows:

$$T^k(\mathbf{x}) = a_{k,0} + \sum_{j=1}^k a_{k,j} x_j + \sum_{j=1}^{P_k} b_{k,j} \phi_j(x_1, x_2, \dots, x_k; \bar{\mathbf{x}}^{k,j}), \quad k = 1, \dots, n, \quad (23.34)$$

where P_k is the total number of radial basis functions used for the k th component of the map and $\phi_j(x_1, x_2, \dots, x_k; \bar{\mathbf{x}}^{k,j})$ is a radial basis function centered at $\bar{\mathbf{x}}^{k,j} \in \mathbb{R}^k$. Note that this representation ensures that the overall map T is lower triangular. The a and b coefficients can then be exposed to the optimization algorithm used to search for the map.

Choosing the centers and scales of the radial basis functions can be challenging in high dimensions, though some heuristics for doing so are given in [63]. To circumvent this difficulty, [63] also proposes using only univariate radial basis functions and embedding them within a composition of maps.

5.3 Monotonicity Constraints and Monotone Parameterizations

Neither the polynomial representation (23.32) nor the radial basis function representation (23.34) yields monotone maps for all values of the coefficients. With either of these choices for the approximation space \mathcal{T}_{Δ}^h , we need to enforce the monotonicity constraints explicitly. Recall that, for the triangular maps considered here, the monotonicity constraint reduces to requiring that $\partial_k T^k > 0$ over the entire support of the reference density, for $k = 1, \dots, n$. It is difficult to enforce this condition everywhere, so instead we choose a finite sequence of points $(\mathbf{x}_i)_i$ – a

stream of samples from the reference distribution, and very often the same samples used for the sample-average approximation of the objective (23.14) – and enforce local monotonicity at each point: $\partial_k T^k(\mathbf{x}_i) > 0$, for $k = 1, \dots, n$. The result is a finite set of linear constraints. Collectively these constraints are weaker than requiring monotonicity everywhere, but as the cardinality of the sequence $(\mathbf{x}_i)_i$ grows, we have stronger guarantees on the monotonicity of the transport map over the entire support of the reference density. When monotonicity is lost, it is typically only in the tails of μ_{ref} where samples are fewer. We should also point out that the $(-\log \det \nabla T)$ term in the objective of (23.11) acts as a barrier function for the constraint $\nabla T > 0$ [68].

A more elegant alternative to discretizing and explicitly enforcing the monotonicity constraints is to employ parameterizations of the map that are in fact *guaranteed* to be monotone [12]. Here we take advantage of the fact that monotonicity of a triangular function can be expressed in terms of one-dimensional monotonicity of its components. A smooth monotone increasing function of one variable, e.g., the first component of the lower triangular map, can be written as [66]:

$$T^1(x_1) = a_1 + \int_0^{x_1} \exp(b_1(w)) dw, \quad (23.35)$$

where $a_1 \in \mathbb{R}$ is a constant and $b_1 : \mathbb{R} \rightarrow \mathbb{R}$ is an arbitrary function. This can be generalized to the k th component of the map as:

$$T^k(x_1, \dots, x_k) = a_k(x_1, \dots, x_{k-1}) + \int_0^{x_k} \exp(b_k(x_1, \dots, x_{k-1}, w)) dw \quad (23.36)$$

for some functions $a_k : \mathbb{R}^{k-1} \rightarrow \mathbb{R}$ and $b_k : \mathbb{R}^k \rightarrow \mathbb{R}$. Note that a_k is not a function of the k th input variable. Of course, we now have to pick a finite-dimensional parameterization of the functions a_k, b_k , but the monotonicity constraint is automatically enforced since

$$\partial_k T^k(\mathbf{x}) = \exp(b_k(x_1, \dots, x_k)) > 0, \quad \forall \mathbf{x} \in \mathbb{R}^n, \quad (23.37)$$

and for all choices of a_k and b_k . Enforcing monotonicity through the map parameterization in this way, i.e., choosing \mathcal{T}_{Δ}^h so that it only contains monotone lower triangular functions, allows the resulting finite-dimensional optimization problem to be unconstrained.

6 Related Work

The idea of using nonlinear transformations to accelerate or simplify sampling has appeared in many different settings. Here we review several relevant instantiations.

Perhaps the closest analogue of the density-based map construction of Sect. 3 is the implicit sampling approach of [21, 22]. While implicit sampling was first

proposed in the context of Bayesian filtering [5, 21, 22, 58], it is in fact a more general scheme for importance simulation [60]. Consider the K–L divergence objective of the optimization problem (23.9). At optimality, the K–L divergence is zero. Rearranging this condition and explicitly writing out the arguments yields:

$$\mathbb{E}_\eta [\log \bar{\pi}(T(\mathbf{x})) + \log \det \nabla T(\mathbf{x}) - \log \beta - \log \eta(\mathbf{x})] = 0, \quad (23.38)$$

where β is the normalizing constant of the unnormalized target density $\bar{\pi}$ (23.13). Now let $\mathbf{z} = T(\mathbf{x})$. The central equation in implicit sampling methods is [22]:

$$\log \bar{\pi}(\mathbf{z}) - \mathcal{C} = \log \eta(\mathbf{x}), \quad (23.39)$$

where \mathcal{C} is an easily computed constant. Implicit sampling first draws a sample \mathbf{x}_i from the reference density η and then seeks a corresponding \mathbf{z}_i that satisfies (23.39). This problem is generally underdetermined, as terms in (23.39) are scalar valued while the samples $\mathbf{x}_i, \mathbf{z}_i$ are in \mathbb{R}^n . Accordingly, the random map implementation of implicit sampling [59] restricts the search for \mathbf{z}_i to a one-dimensional optimization problem along randomly oriented rays emanating from a point in \mathbb{R}^n , e.g., the mode of the target distribution. This scheme is efficient to implement, though it is restricted to target densities whose contours are star convex with respect to the chosen point [35]. Satisfying (23.39) in this way defines the *action* of a map from η to another distribution, and the intent of implicit sampling is that this pushforward distribution should be close to the target. There are several interesting contrasts between (23.38) and (23.39), however. First is the absence of the Jacobian determinant in (23.39). The samples \mathbf{z}_i produced by implicit sampling must then (outside of the Gaussian case) be endowed with weights, which result from the Jacobian determinant of the implicit map. The closeness of the implicit samples to the desired target is reflected in the variation of these weights. A second contrast is that (23.38) is a global statement about the action of a map T over the entire support of η , wherein the map T appears explicitly. On the other hand, (23.39) is a relationship between points in \mathbb{R}^n . The map does not appear explicitly in this relationship; rather, the way in which (23.39) is satisfied *implicitly* defines the map.

Another optimization-based sampling algorithm, similar in spirit to implicit sampling though different in construction, is the randomize-then-optimize (RTO) approach of [8]. This scheme is well defined for target distributions whose log densities can be written in a particular quadratic form following a transformation of the parameters, with some restrictions on the target’s degree of non-Gaussianity. The algorithm proceeds in three steps. First, one draws a sample \mathbf{x}_i from a Gaussian (reference) measure and uses this sample to fix the objective of an unconstrained optimization problem in n variables. Next, one solves this optimization problem to obtain a sample \mathbf{z}_i . And finally, this sample is “corrected” either via an importance weight or a Metropolis step. The goal, once again, is that the distribution of the samples \mathbf{z}_i should be close to the true target π – though as in implicit sampling, outside of the Gaussian case these two distributions will not be identical and the correction step is required.

Hence, another way of understanding the contrast between these optimization-based samplers and the transport map framework is that the latter defines an optimization problem *over maps*, where minimizing the left-hand side of (23.38) is the objective. Implicit sampling and RTO instead solve simpler optimization problems *over samples*, where each minimization yields the action of a particular transport. A crucial feature of these transports is that the pushforward densities they induce can be evaluated in closed form, thus allowing implicit samples and RTO samples to be reweighted or Metropolized in order to obtain asymptotically unbiased estimates of target expectations. Nonetheless, implicit sampling and RTO each implement a particular transport, and they are *bound* to these choices. In other words, these transports cannot be refined, and it is difficult to predict their quality for arbitrarily non-Gaussian targets. The transport map framework instead implements a search over a space of maps and therefore contains a tunable knob between computational effort and accuracy: by enriching the search space \mathcal{T}_Δ^h , one can get arbitrarily close to any target measure. Of course, the major disadvantage of the transport map framework is that one must then parameterize maps $T \in \mathcal{T}_\Delta^h$ rather than just computing the action of a particular map. But parameterization subsequently allows direct evaluation and sampling of the pushforward $T_\sharp \eta$ without appealing again to the target density.

Focusing for a moment on the specific problem of Bayesian inference, another class of approaches related to the transport map framework are the sampling-free Bayesian updates introduced in [47, 48, 54, 71, 72]. These methods treat Bayesian inference as a projection. In particular, they approximate the conditional expectation of any prescribed function of the parameters, where conditioning is with respect to the σ -field generated by the data. The approximation of the conditional expectation may be refined by enlarging the space of functions (typically polynomials) on which one projects; hence one can generalize linear Bayesian updates [71] to nonlinear Bayesian updates [48]. The precise goal of these approximations is different from that of the transport map framework, however. Both methods approximate random variables, but different ones: [48] focuses on the conditional expectation of a function of the parameters (e.g., mean, second moment) as a function of the data random variable, whereas the transport approach to inference [61] aims to fully characterize the posterior random variable for a particular realization of the data.

Ideas from *optimal* transportation have also proven useful in the context of Bayesian inference. In particular, [67] solves a *discrete* Kantorovich optimal transport problem to find an optimal transport plan from a set of unweighted samples representing the prior distribution to a weighted set of samples at the same locations, where the weights reflect the update from prior to posterior. This transport plan is then used to construct a linear transformation of the prior ensemble that yields consistent posterior estimates. The linear transformation can be understood as a resampling strategy, replacing the weighted samples with new samples that are convex combinations of the prior samples. The ability to “move” the samples within the convex hull of the prior ensemble leads to improved performance over other resampling strategies, though the prior samples should then have good coverage of the support of the posterior.

Turning to the sample-based map construction of Sect. 4, it is interesting to note that attempts to Gaussianize collections of samples using nonlinear transformations date back at least to 1964 [14]. In the geostatistics literature, the notion of Gaussian anamorphosis [86] uses the empirical cumulative distribution function (CDF), or Hermite polynomial approximations of the CDF, to Gaussianize the *marginals* of multivariate data. These transformations do not create joint Gaussianity, however.

To construct joint transformations of dependent multivariate data, [77] proposes a scheme employing discrete optimal transport. This approach generates an equivalent number of samples from a reference measure; solves a discrete assignment problem between the two sample sets, given a quadratic transport cost; and uses the resulting pairs to estimate a polynomial map using linear regression. This is a two-stage approach, in contrast with the single convex optimization problem proposed in Sect. 4. For reference and target distributions with compact support, it yields an approximation of the Monge optimal transport rather than the Knothe–Rosenblatt rearrangement.

Moving from polynomial approximations to nonparametric approaches, [45, 81, 82] introduce schemes for multivariate density estimation based on progressively transforming a given set of samples to a (joint) standard Gaussian by *composing* a sequence of monotone maps. The maps are typically chosen to be rather simple in form (e.g., sigmoid-type functions of one variable). In this context, we note that the empirical K–L divergence objective in (23.21) is the pullback density $S_{\sharp}^{-1}\eta$ evaluated at the samples $(z_i)_{i=1}^M$ and hence can be viewed as the log likelihood of the map S given the target samples. In [81], each new element of the composition of maps is guaranteed not to decrease this log-likelihood function. A related scheme is presented in [44]; here the sequence of maps alternates between rotations and diagonal maps that transform the marginals. Rotations are chosen via principle component analysis (PCA) or independent component analysis (ICA). The resulting composition of maps can Gaussianize remarkably complex distributions in hundreds of dimensions (e.g., samples from a face database). Both of these methods, however, reveal an interesting tension between the number of maps in the composition and the complexity of a single map. When each map in the composition is very simple (e.g., diagonal, or even constant in all but one variable), the maps are easy to construct, but their composition can converge very slowly to a Gaussianizing transformation. On the other hand, we know that there exist maps (e.g., the Knothe–Rosenblatt rearrangement or the Monge optimal transport map) that can Gaussianize the samples immediately, but approximating them directly requires much more effort. Some of these trade-offs are explored in [63].

Yet another use of transformations in stochastic simulation is the warp bridge sampling approach of [56]. The goal of bridge sampling is to estimate the ratio of normalizing constants of two probability densities (e.g., the ratio of Bayesian model evidences). Meng and Schilling [56] introduces several deterministic and/or stochastic transformations to increase the overlap of the two densities – by translating, scaling, and even symmetrizing them. These transformations can reduce the asymptotic variance of the bridge sampling estimator. More recent generalizations

use Gaussian mixture approximations of the densities to design transformations suitable for multimodal problems [88].

Finally, setting aside the notion of nonlinear transformations, it is useful to think of the minimization problem (23.9) in the broader context of variational Bayesian methods [6, 28, 40, 87]. As in typical variational Bayesian approaches, we seek to approximate some complex or intractable distribution (represented by π) with a simpler one. But the approximating distribution in the transport map framework is any pushforward of a reference density. In contrast with variational Bayesian approaches, this distribution can be found without imposing strong assumptions on its factorization (e.g., the mean field approximation) or on the family of distributions from which it is drawn (e.g., an exponential family). The transport map framework is also distinguished from variational Bayesian approaches due to the availability of the pullback density (23.16) – in an intuitive sense, the “leftover” after approximation with any given map. Using evaluations of the pullback density, one can compose sequences of maps, enrich the approximation space of any given map, or use the current transport to precondition an exact sampling scheme.

7 Conditional Sampling

In this section we will show how the triangular structure of the transport map allows efficient sampling from particular conditionals of the target density. This capability is important because, in general, the ability to sample from a distribution does not necessarily provide efficient techniques for also sampling its conditionals. As in the previous sections, assume that the reference and target measures are absolutely continuous with respect to the Lebesgue measure on \mathbb{R}^n with smooth and positive densities. Let $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a triangular and monotone increasing transport that pushes forward the reference to the target density, i.e., $T_{\sharp}\eta = \pi$, where T is the Knothe–Rosenblatt rearrangement.

We first need to introduce some additional notation. There is no loss of generality in thinking of the target as the joint distribution of some random vector \mathbf{Z} in \mathbb{R}^n . Consider a partition of this random vector as $\mathbf{Z} = (\mathbf{D}, \boldsymbol{\Theta})$ where $\mathbf{D} \in \mathbb{R}^{n_d}$, $\boldsymbol{\Theta} \in \mathbb{R}^{n_\theta}$, and $n = n_d + n_\theta$. In other words, \mathbf{D} simply comprises the first n_d components of \mathbf{Z} . We equivalently denote the joint density of $\mathbf{Z} = (\mathbf{D}, \boldsymbol{\Theta})$ by either π or $\pi_{\mathbf{D}, \boldsymbol{\Theta}}$. That is, $\pi \equiv \pi_{\mathbf{D}, \boldsymbol{\Theta}}$. We define the conditional density of $\boldsymbol{\Theta}$ given \mathbf{D} as

$$\pi_{\boldsymbol{\Theta}|\mathbf{D}} := \frac{\pi_{\mathbf{D}, \boldsymbol{\Theta}}}{\pi_{\mathbf{D}}}, \quad (23.40)$$

where $\pi_{\mathbf{D}} := \int \pi_{\mathbf{D}, \boldsymbol{\Theta}}(\cdot, \boldsymbol{\theta}) d\boldsymbol{\theta}$ is the marginal density of \mathbf{D} . In particular, $\pi_{\boldsymbol{\Theta}|\mathbf{D}}(\boldsymbol{\theta}|\mathbf{d})$ is the conditional density of $\boldsymbol{\Theta}$ at $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ given the event $\{\mathbf{D} = \mathbf{d}\}$. Finally, we define $\pi_{\boldsymbol{\Theta}|\mathbf{D}=\mathbf{d}}$ as a map from \mathbb{R}^{n_θ} to \mathbb{R} such that

$$\boldsymbol{\theta} \mapsto \pi_{\boldsymbol{\Theta}|\mathbf{D}}(\boldsymbol{\theta}|\mathbf{d}). \quad (23.41)$$

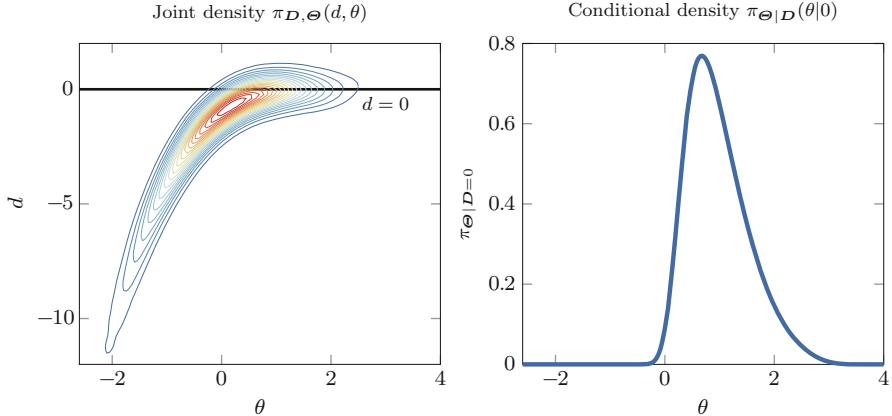


Fig. 23.2 (left) Illustration of a two-dimensional joint density $\pi_{D,\Theta}$ together with a particular slice at $d = 0$. (right) Conditional density $\pi_{\Theta|D}(\theta|0)$ obtained from a normalized slice of the joint density at $d = 0$

We can think of $\pi_{\Theta|D=d}$ as a particular normalized slice of the joint density $\pi_{D,\Theta}$ for $D = \mathbf{d}$, as shown in Fig. 23.2. Our goal is to show how the triangular transport map T can be used to efficiently sample the conditional density $\pi_{\Theta|D=d}$.

If T is a monotone increasing lower triangular transport on \mathbb{R}^n , we can denote its components by

$$T(\mathbf{x}) := T(\mathbf{x}_D, \mathbf{x}_\Theta) = \begin{bmatrix} T^D(\mathbf{x}_D) \\ T^\Theta(\mathbf{x}_D, \mathbf{x}_\Theta) \end{bmatrix}, \quad \forall \mathbf{x} \in \mathbb{R}^n, \quad (23.42)$$

where $T^D : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_d}$, $T^\Theta : \mathbb{R}^{n_d} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_\theta}$, and where $\mathbf{x} := (\mathbf{x}_D, \mathbf{x}_\Theta)$ is a partition of the dummy variable $\mathbf{x} \in \mathbb{R}^n$ as $\mathbf{x}_D \in \mathbb{R}^{n_d}$ and $\mathbf{x}_\Theta \in \mathbb{R}^{n_\theta}$, i.e., \mathbf{x}_D consists of the first n_d components of \mathbf{x} .

In the context of Bayesian inference, Θ could represent the inversion parameters or latent variables and D the observational data. In this interpretation, $\pi_{\Theta|D=d}$ is just the posterior distribution of the parameters for a particular realization of the data. Sampling this posterior distribution yields an explicit characterization of the Bayesian solution and is thus of crucial importance. This scenario is particularly relevant in the context of online Bayesian inference where one is concerned with fast posterior computations for multiple realizations of the data (e.g., [39, 63]). Of course, if one is only interested in $\pi_{\Theta|D=d}$ for a single realization of the data, then there is no need to first approximate the joint density $\pi_{D,\Theta}$ and subsequently perform conditioning to sample $\pi_{\Theta|D=d}$. Instead, one should simply pick $\pi_{\Theta|D=d}$ as the target density and compute the corresponding transport [61]. In the latter case, the dimension of the transport map would be independent of the size of the data.

The following lemma shows how to efficiently sample the conditional density $\pi_{\Theta|D=d}$ given a monotone increasing triangular transport T . In what follows we

assume that the reference density can be written as the product of its marginals; that is, $\eta(\mathbf{x}) = \eta_D(\mathbf{x}_D)\eta_\Theta(\mathbf{x}_\Theta)$ for all $\mathbf{x} = (\mathbf{x}_D, \mathbf{x}_\Theta)$ in \mathbb{R}^n and with marginal densities η_D and η_Θ . This hypothesis is not restrictive as the reference density is a degree of freedom of the problem (e.g., η is often a standard normal density).

Lemma 1. *For a fixed $\mathbf{d} \in \mathbb{R}^{n_d}$, define \mathbf{x}_d^* as the unique element of \mathbb{R}^{n_d} such that $T^D(\mathbf{x}_d^*) = \mathbf{d}$. Then, the map $T_d : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_\theta}$, defined as*

$$\mathbf{w} \mapsto T^\Theta(\mathbf{x}_d^*, \mathbf{w}), \quad (23.43)$$

pushes forward η_Θ to the desired conditional density $\pi_{\Theta|D=d}$.

Proof. First of all, notice that $\mathbf{x}_d^* := (T^D)^{-1}(\mathbf{d})$ is well defined since $T^D : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_d}$ is a monotone increasing and invertible function by definition of the Knothe–Rosenblatt rearrangement T . Then:

$$\begin{aligned} T_d^\# \pi_{\Theta|D=d}(\mathbf{w}) &= \pi_{\Theta|D}(T_d(\mathbf{w})|\mathbf{d}) |\det \nabla T_d(\mathbf{w})| \\ &= \frac{\pi_{D,\Theta}(\mathbf{d}, T^\Theta(\mathbf{x}_d^*, \mathbf{w}))}{\pi_D(\mathbf{d})} \det \nabla_w T^\Theta(\mathbf{x}_d^*, \mathbf{w}) \end{aligned} \quad (23.44)$$

Since, by definition, $T^D(\mathbf{x}_d^*) = \mathbf{d}$, we have for all $\mathbf{w} \in \mathbb{R}^{n_\theta}$:

$$\begin{aligned} T_d^\# \pi_{\Theta|D=d}(\mathbf{w}) &= \frac{\pi_{D,\Theta}(T^D(\mathbf{x}_d^*), T^\Theta(\mathbf{x}_d^*, \mathbf{w}))}{\pi_D(T^D(\mathbf{x}_d^*))} \det \nabla_w T^\Theta(\mathbf{x}_d^*, \mathbf{w}) \\ &= \frac{\pi_{D,\Theta}(T^D(\mathbf{x}_d^*), T^\Theta(\mathbf{x}_d^*, \mathbf{w}))}{(T^D)^\# \pi_D(\mathbf{x}_d^*)} \det \nabla T^D(\mathbf{x}_d^*) \det \nabla_w T^\Theta(\mathbf{x}_d^*, \mathbf{w}) \\ &= \frac{T^\# \pi_{D,\Theta}(\mathbf{x}_d^*, \mathbf{w})}{\eta_D(\mathbf{x}_d^*)} = \frac{\eta(\mathbf{x}_d^*, \mathbf{w})}{\eta_D(\mathbf{x}_d^*)} = \eta_\Theta(\mathbf{w}) \end{aligned}$$

where we used the identity $T_d^\# \eta_D = \pi_D$ which follows from the definition of Knothe–Rosenblatt rearrangement (e.g., [85]). \square

We can interpret the content of Lemma 1 in the context of Bayesian inference. If we observe a particular realization of the data, i.e., $\mathbf{D} = \mathbf{d}$, then we can easily sample the posterior distribution $\pi_{\Theta|D=d}$ as follows. First, solve the nonlinear triangular system $T^D(\mathbf{x}_d^*) = \mathbf{d}$ to get \mathbf{x}_d^* . Since T^D is a lower triangular and invertible map, one can solve this system using the techniques described in Sect. 4.3. Then, define a new map $T_d : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_\theta}$ as $T_d(\mathbf{w}) := T^\Theta(\mathbf{x}_d^*, \mathbf{w})$ for all $\mathbf{w} \in \mathbb{R}^{n_\theta}$, and notice that the pushforward through the map of the marginal distribution of the reference over the parameters, i.e., $(T_d)_\# \eta_\Theta$, is precisely the desired posterior distribution.

Notice that T_d is a single transformation parameterized by $\mathbf{d} \in \mathbb{R}^{nd}$. Thus it is straightforward to condition on a different value of \mathbf{D} , say $\mathbf{D} = \tilde{\mathbf{d}}$. We only need to solve a new nonlinear triangular system of the form $T^{\mathbf{D}}(\mathbf{x}_{\tilde{\mathbf{d}}}^*) = \tilde{\mathbf{d}}$ to define a transport $T_{\tilde{\mathbf{d}}}$ according to (23.43). Moreover, note that the particular triangular structure of the transport map T is essential to achieving efficient sampling from the conditional $\pi_{\Theta|\mathbf{D}=d}$ in the manner described by Lemma 1.

8 Example: Biochemical Oxygen Demand Model

Here we demonstrate some of the measure transport approaches described in previous sections with a simple Bayesian inference problem, involving a model of biochemical oxygen demand (BOD) commonly used in water quality monitoring [80]. This problem is a popular and interesting test case for sampling methods (e.g., MCMC [64], RTO [8]). The time-dependent forward model is defined by

$$\mathfrak{B}(t) = A(1 - \exp(-Bt)) + \mathcal{E}, \quad (23.45)$$

where A and B are unknown scalar parameters modeled as random variables, t represents time, and $\mathcal{E} \sim \mathcal{N}(0, 10^{-3})$ is an additive Gaussian observational noise that is statistically independent of A and B . In this example, the data \mathbf{D} consist of five observations of $\mathfrak{B}(t)$ at $t = \{1, 2, 3, 4, 5\}$ and is thus a vector-valued random variable defined by

$$\mathbf{D} := [\mathfrak{B}(1), \mathfrak{B}(2), \mathfrak{B}(3), \mathfrak{B}(4), \mathfrak{B}(5)].$$

Our goal is to characterize the joint distribution of A and B conditioned on the observed data. We assume that A and B are independent under the prior measure, with uniformly distributed marginals:

$$A \sim \mathcal{U}(0.4, 1.2), \quad B \sim \mathcal{U}(0.01, 0.31). \quad (23.46)$$

Instead of inferring A and B directly, we choose to invert for some new target parameters, Θ_1 and Θ_2 , that are related to the original parameters through the CDF of a standard normal distribution:

$$A := \left[0.4 + 0.4 \left(1 + \text{erf} \left(\frac{\Theta_1}{\sqrt{2}} \right) \right) \right] \quad (23.47)$$

$$B := \left[0.01 + 0.15 \left(1 + \text{erf} \left(\frac{\Theta_2}{\sqrt{2}} \right) \right) \right]. \quad (23.48)$$

Notice that these transformations are invertible. Moreover, the resulting prior marginal distributions over the target parameters Θ_1 and Θ_2 are given by

$$\Theta_1 \sim \mathcal{N}(0, 1), \quad \Theta_2 \sim \mathcal{N}(0, 1). \quad (23.49)$$

We denote the target random vector by $\Theta := (\Theta_1, \Theta_2)$. The main advantage of inferring Θ as opposed to the original parameters A and B directly is that the support of Θ is unbounded. Thus, there is no need to impose any geometric constraints on the range of the transport map.

8.1 Inverse Transport: Map from Samples

We start with a problem of efficient online Bayesian inference – where one is concerned with fast posterior computations for multiple realizations of the data – using the inverse transport of Sect. 4. Our first goal is to characterize the joint distribution of data and parameters, $\pi_{D,\Theta}$, by means of a lower triangular transport. As explained in Sect. 7, this transport will then enable efficient *conditioning* on different realizations of the data D . Posterior computations associated with the conditional $\pi_{\Theta|D=d}$, for arbitrary instances of d , will thus become computationally trivial tasks.

Note that having defined the prior and likelihood via (23.47)–(23.49) and (23.45), respectively, we can generate arbitrarily many independent “exact” samples from the joint target $\pi_{D,\Theta}$. For the purpose of this demonstration, we will pretend that we cannot evaluate the unnormalized target density and that we can access the target only through these samples. This is a common scenario in Bayesian inference problems with intractable likelihoods [23, 52, 89]. This sample-based setting is well suited to the computation of a triangular inverse transport – a transport map that pushes forward the target to a standard normal reference density – as the solution of a convex and separable optimization problem (23.22). The direct transport can then be evaluated implicitly using the techniques described in Sect. 4.3.

We will solve (23.22) for a matrix of different numerical configurations. The expectation with respect to the target measure is discretized using different Monte Carlo sample sizes (5×10^3 versus 5×10^4). The reference and target are measures on \mathbb{R}^7 , and thus the finite-dimensional approximation space for the inverse transport, $\mathcal{T}_\Delta^h \subset \mathcal{T}_\Delta$, is taken to be a total-degree polynomial space in $n = 7$ variables, parameterized with a Hermite basis and using range of different degrees (23.33). In particular, we will consider maps ranging from linear ($p = 1$) to seventh degree ($p = 7$). The result of (23.22) is an approximate inverse transport $\tilde{\mathcal{S}}$. An approximation to the direct transport, $\tilde{\mathcal{T}}$, is then obtained via standard regression techniques as explained in Sect. 4.3. In particular, the direct transport is sought in the same approximation space \mathcal{T}_Δ^h as the inverse transport.

In order to assess the accuracy of the computed transports, we will characterize the conditional density $\pi_{\Theta|D=d}$ (see Lemma 1 in Sect. 7) for a value of the data given by

$$d = [0.18, 0.32, 0.42, 0.49, 0.54].$$

Table 23.1 compares moments of the approximate conditional $\pi_{\Theta|D=d}$ computed via the inverse transports to the “true” moments of this distribution as estimated via a

Table 23.1 BOD problem of Section 8.1 via inverse transport and conditioning. First four moments of the conditional density $\pi_{\Theta|D=d}$, for $d = [0.18, 0.32, 0.42, 0.49, 0.54]$, estimated by a “reference” run of an adaptive Metropolis sampler with 6×10^6 steps, and by transport maps up to degree $p = 7$ with 3×10^4 samples

Map type	# training samples	Mean		Variance		Skewness		Kurtosis	
		Θ_1	Θ_2	Θ_1	Θ_2	Θ_1	Θ_2	Θ_1	Θ_2
MCMC “truth”	0.075	0.875	0.190	0.397	1.935	0.681	8.537	3.437	
$p = 1$	5000	0.199	0.717	0.692	0.365	-0.005	0.010	2.992	3.050
	50000	0.204	0.718	0.669	0.348	0.016	-0.006	3.019	3.001
$p = 3$	5000	0.066	0.865	0.304	0.537	0.909	0.718	4.042	3.282
	50000	0.040	0.870	0.293	0.471	0.830	0.574	3.813	3.069
$p = 5$	5000	0.027	0.888	0.200	0.447	1.428	0.840	5.662	3.584
	50000	0.018	0.907	0.213	0.478	1.461	0.843	6.390	3.606
$p = 7$	5000	0.090	0.908	0.180	0.490	2.968	0.707	29.589	16.303
	50000	0.034	0.902	0.206	0.457	1.628	0.872	7.568	3.876

“reference” adaptive Metropolis MCMC scheme [37]. While more efficient MCMC algorithms exist, the adaptive Metropolis algorithm is well known and enables a qualitative comparison of the computational cost of the transport approach to that of a widely used and standard method. The MCMC sampler was tuned to have an acceptance rate of 26%. The chain was run for 6×10^6 steps, 2×10^4 of which were discarded as burn-in. The moments of the approximate conditional density $\pi_{\Theta|D=d}$ given by each computed transport are estimated using 3×10^4 independent samples generated from the conditional map. The accuracy comparison in Table 23.1 shows that the cubic map captures the mean and variance of $\pi_{\Theta|D=d}$ but does not accurately capture the higher moments. Increasing the map degree, together with the number of target samples, yields better estimates of these moments. For instance, degree-seven maps constructed with 5×10^4 target samples can reproduce the skewness and kurtosis of the conditional density reasonably well. Kernel density estimates of the two-dimensional conditional density $\pi_{\Theta|D=d}$ using 50,000 samples are also shown in Fig. 23.3 for different orders of the computed transport. The degree-seven map gives results that are nearly identical to the MCMC reference computation.

In a typical application of Bayesian inference, we can regard the time required to compute an approximate inverse transport \tilde{S} and a corresponding approximate direct transport \tilde{T} as “offline” time. This is the expensive step of the computations, but it is independent of the observed data. The “online” time is that required to generate samples from the conditional distribution $\pi_{\Theta|D=d}$ when a new realization of the data $\{D = d\}$ becomes available. The online step is computationally inexpensive since it requires, essentially, only the solution of a single nonlinear triangular system of the dimension of the data (see Lemma 1).

In Table 23.2 we compare the computational time of the map-based approach to that of the reference adaptive Metropolis MCMC scheme. The online time column shows how long each method takes to generate 30,000 independent samples

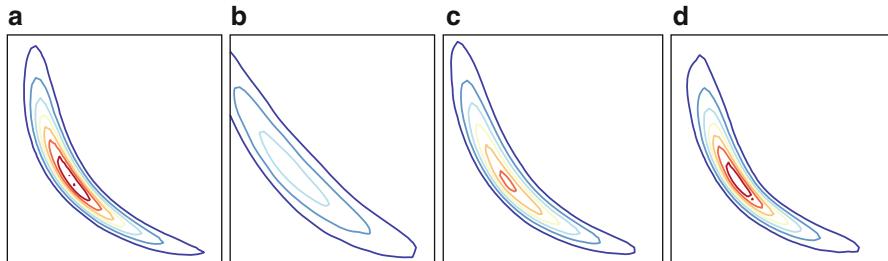


Fig. 23.3 BOD problem of Sect. 8.1 via inverse transport and conditioning. Kernel density estimates of the conditional density $\pi_{\theta|D=d}$, for $d = [0.18, 0.32, 0.42, 0.49, 0.54]$, using 50,000 samples from either a reference adaptive MCMC sampler (*left*) or conditioned transport maps of varying total degree. Contour levels and color scales are constant for all figures. **(a)** MCMC truth. **(b)** Degree $p = 3$. **(c)** Degree $p = 5$. **(d)** Degree $p = 7$

Table 23.2 BOD problem of Sect. 8.1 via inverse transport and conditioning. Efficiency of approximate Bayesian inference with inverse transport map from samples. The “offline” time is defined as the time it takes to compute an approximate inverse transport \tilde{S} and a corresponding approximate direct transport \tilde{T} via regression (see Sect. 4). The “online” time is the time required after observing $\{D = d\}$ to generate the equivalent of 30,000 independent samples from the conditional $\pi_{\theta|D=d}$. For MCMC, the “online” time is the average amount of time it takes to generate a chain with an effective sample size of 30,000

Map type	# training samples	Offline time (sec)		Online time (sec)
		\tilde{S} construction	\tilde{T} regression	
MCMC “truth”	NA			591.17
$p = 1$	5000	0.46	0.18	2.60
	50,000	4.55	1.65	2.32
$p = 3$	5000	4.13	1.36	3.54
	50,000	40.69	18.04	3.58
$p = 5$	5000	22.82	8.40	5.80
	50,000	334.25	103.47	6.15
$p = 7$	5000	145.00	40.46	8.60
	50,000	1070.67	432.95	8.83

from the conditional $\pi_{\theta|D=d}$. For MCMC, we use the average amount of time required to generate a chain with an effective sample size of 30,000. Measured in terms of online time, the polynomial transport maps are roughly two orders of magnitude more efficient than the adaptive MCMC sampler. More sophisticated MCMC samplers could be used, of course, but the conditional map approach will retain a significant advantage because, after solving for x_d^* in Lemma 1, it can generate *independent* samples at negligible cost. We must stress, however, that the samples produced in this way are from an *approximation* of the targeted conditional. In fact, the conditioning lemma holds true only if the computed joint transport is exact, and a solution of (23.22) is an approximate transport for the reasons discussed in Sect. 4.2. Put in another way, the conditioning lemma is exact for the pushforward

of the approximate map, $\tilde{\pi} = \widetilde{T}_\# \eta$. Nevertheless, if the conditional density $\pi_{\Theta|D=d}$ can be evaluated up to a normalizing constant, then one can quantify the error in the approximation of the conditional using (23.24). Under these conditions, if one is not satisfied with this error, then any of the approximate maps \widetilde{T}_d constructed here could be useful as a proposal mechanism for importance sampling or MCMC, to generate (asymptotically) exact samples from the conditional of interest. For example, the map constructed using the offline techniques discussed in this example could provide an excellent initial map for the MCMC scheme of [64]. Without this correction, however, the sample-based construction of lower triangular inverse maps, coupled with direct conditioning, can be seen as a flexible scheme for fast approximate Bayesian computation.

8.2 Direct Transport: Map from Densities

We now focus on the computation of a direct transport as described in Sect. 3, using evaluations of an unnormalized target density. Our goal here is to characterize a monotone increasing triangular transport map T that pushes forward a standard normal reference density η to the posterior distribution $\pi_{\Theta|D=d}$, i.e., the distribution of the BOD model parameters conditioned on a realization of the data d . We use the same realization of the data as in Sect. 8.1.

Figure 23.4 shows the reference and target densities. Notice that the target density exhibits a nonlinear dependence structure. This type of locally varying correlation can make sampling via standard MCMC methods (e.g., Metropolis–Hastings schemes with Gaussian proposals) quite challenging. In this example, the log-target density can be evaluated analytically up to a normalizing constant but direct sampling from the target distribution is impossible. Thus it is an ideal setting for the computation of the direct transport as the minimizer of the optimization problem (23.11). We just need to specify how to approximate integration with respect to the reference measure in the objective of (23.11) and to choose a

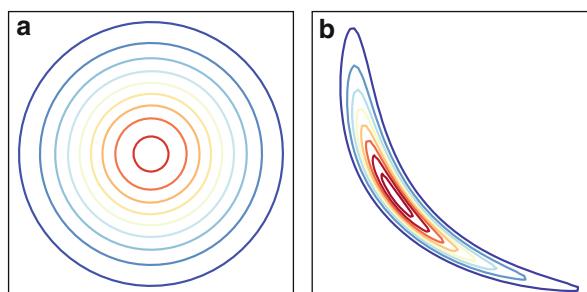


Fig. 23.4 BOD problem of Section 8.2 via direct transport. Observations are taken at times $t \in \{1, 2, 3, 4, 5\}$. The observed data vector is given by $d = [0.18; 0.32; 0.42; 0.49; 0.54]$. (a) Reference density. (b) Target density

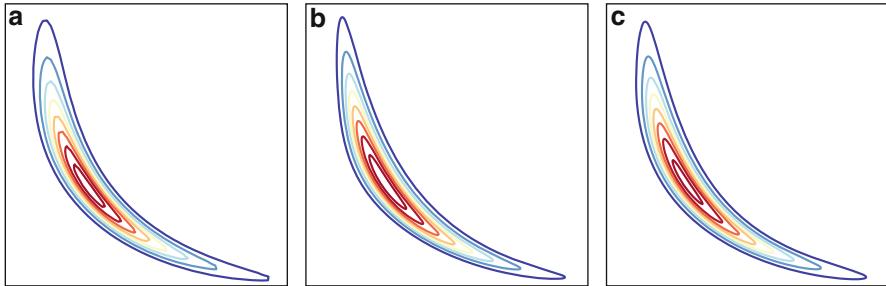


Fig. 23.5 BOD problem of Sect. 8.2 via direct transport: pushforwards of the reference density under a given total-degree triangular map. The basis of the map consists of multivariate Hermite polynomials. The expectation with respect to the reference measure is approximated with a full tensor product Gauss–Hermite quadrature rule. The approximation is already excellent with a map of degree $p = 3$. (a) Target density. (b) Pushforward $p = 3$. (c) Pushforward $p = 5$

finite-dimensional approximation space $\mathcal{T}_\Delta^h \subset \mathcal{T}_\Delta$ for the triangular map. Here we will approximate integration with respect to the reference measure using a tensor product of ten-point Gauss–Hermite quadrature rules. For this two-dimensional problem, this corresponds to 100 integration nodes (i.e., $M = 100$ in (23.14)). For \mathcal{T}_Δ^h we will employ a total-degree space of multivariate Hermite polynomials. In particular, we focus on third- and fifth-degree maps. The monotonicity constraint in (23.11) is discretized pointwise at the integration nodes as explained in Sect. 5.3.

Figure 23.5 shows the results of solving the discretized optimization problem (23.14) for the transport map. In particular, we show the pushforward of the reference density through the transport maps found by solving (23.14) for different truncations of \mathcal{T}_Δ^h . As we can see from Fig. 23.5, an excellent approximation of the target density is already achieved with a degree-three map (see Fig. 23.5b). This approximation improves and is almost indistinguishable from the true target density for a degree-five map (see Fig. 23.5c). Thus if we were to compute posterior expectations using the approximate map as explained in Sect. 3.3, we would expect virtually zero-variance estimators with extremely small bias. Moreover, we can estimate this bias using (23.15).

If one is not content with the bias of these estimators, then it is always possible to rely on asymptotically exact sampling of the pullback of the target distribution through the approximate map via, e.g., MCMC (see Sect. 3.3). Figure 23.6 illustrates such pullback densities for different total-degree truncations of the polynomial space \mathcal{T}_Δ^h . As we can see from Fig. 23.6b, c, the pullback density is progressively “Gaussianized” as the degree of the transport map increases. In particular, these pullbacks do not have the complex correlation structure of the original target density and are amenable to efficient sampling; for instance, even a Metropolis independence sampler [69] could be very effective. Thus, approximate transport maps can effectively precondition and improve the efficiency of existing sampling techniques.

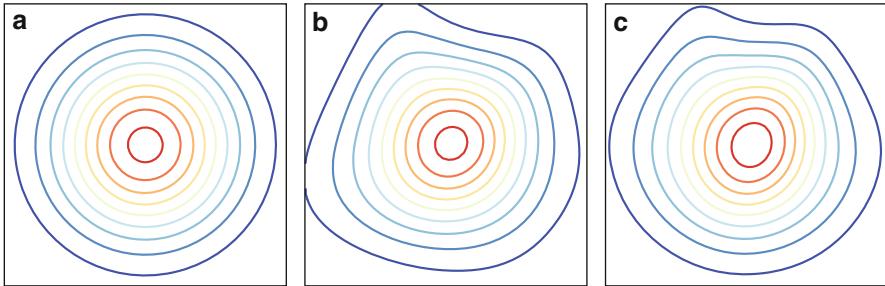


Fig. 23.6 BOD problem of Sect. 8.2 via direct transport: pullbacks under a given total order triangular map of the target density. Same setup of the optimization problem as in Fig. 23.5. The pullback density is progressively “Gaussianized” as the degree of the transport map increases. (a) Reference density. (b) Pullback $p = 3$. (c) Pullback $p = 5$

9 Conclusions and Outlook

In this chapter, we reviewed the fundamentals of the measure transport approach to sampling. The idea is simple but powerful. Assume that we wish to sample a given, possibly non-Gaussian, target measure. We solve this problem by constructing a deterministic transport map that pushes forward a reference measure to the target measure. The reference can be any measure from which we can easily draw samples or construct quadratures (e.g., a standard Gaussian). Under these assumptions, pushing forward independent samples from the reference through the transport map produces independent samples from the target. This construction turns sampling into a trivial task: we only need to evaluate a deterministic function. Of course, the challenge is now to determine a suitable transport. Though the existence of such transports is guaranteed under weak conditions [85], in this chapter we focused on target and reference measures that are absolutely continuous with respect to the Lebesgue measure, with smooth and positive densities. These hypotheses make the numerical computation of a continuous transport map particularly attractive. It turns out that a smooth triangular transport, the Knothe–Rosenblatt rearrangement [18, 70], can be computed via smooth and possibly unconstrained optimization.

To compute this transport, we considered two different scenarios. In Sect. 3 we addressed the computation of a monotone triangular *direct transport* – a transport map that pushes forward a reference measure to the target measure – given only the ability to evaluate the unnormalized target density [61]. This situation is very common in the context of Bayesian inference. The direct transport can be computed by solving a smooth optimization problem using standard gradient-based techniques. In Sect. 4, on the other hand, we focused on a setting where the target density is unavailable and we are instead given only finitely many samples from the target distribution. This scenario arises, for instance, in density estimation [81] or Bayesian inference with intractable likelihoods [23, 52, 89]. In this setting, we showed that a monotone triangular *inverse transport* – a transport

map that pushes forward the target measure to the reference measure – can be computed efficiently via separable convex optimization. The direct transport can then be evaluated by solving a nonlinear triangular system via a sequence of one-dimensional root findings (see Sect. 4.3). Moreover, we showed that characterizing the target distribution as the pushforward of a triangular transport enables efficient sampling from particular conditionals (and of course any marginal) of the target (see Sect. 7). This feature can be extremely useful in the context of online Bayesian inference, where one is concerned with fast posterior computations for multiple realizations of the data (see Sect. 8.1).

Ongoing efforts aim to expand the transport map framework by: (1) understanding the fundamental *structure* of transports and how this structure flows from certain properties of the target measure; (2) developing rigorous and automated methods for the adaptive refinement of maps; and (3) coupling these methods with more effective parameterizations and computational approaches.

An important preliminary issue, which we discussed briefly in Sect. 5.3, is how to enforce **monotonicity** and thus invertibility of the transport. In general, there is no easy way to parameterize a monotone map. However, as shown in Sect. 5.3 and detailed in [12], if we restrict our attention to triangular transports – that is, if we consider the computation of a Knothe–Rosenblatt rearrangement – then the monotonicity constraint can be enforced *strictly* in the parameterization of the map. This result is inspired by monotone regression techniques [66] and is useful in the transport map framework as it removes explicit monotonicity constraints altogether, enabling the use of unconstrained optimization techniques.

Another key challenge is the need to construct low-dimensional parameterizations of transport maps in high-dimensional settings. The critical observation in [75] is that Markov properties – i.e., the conditional independence structure – of the target distribution induce an intrinsic low dimensionality of the transport map in terms of **sparsity** and **decomposability**. A sparse transport is a multivariate map where each component is only a function of few input variables, whereas a decomposable transport is a map that can be written as the *exact* composition of a finite number of simple functions. The analysis in [75] reveals that these sparsity and decomposability properties can be predicted *before* computing the actual transport simply by examining the Markov structure of the target distribution. These properties can then be explicitly enforced in the parameterization of candidate transport maps, leading to optimization problems of considerably reduced dimension. Note that there is a constant effort in applications to formulate probabilistic models of phenomena of interest using sparse Markov structures; one prominent example is multiscale modeling [65]. A further source of low dimensionality in transports is **low-rank structure**, i.e., situations where a map departs from the identity only on a low-dimensional subspace of the input space [75]. This situation is fairly common in large-scale Bayesian inverse problems where the data are informative, relative to the prior, only about a handful of directions in the parameter space [25, 76].

Building on these varieties of low-dimensional structure, we still need to construct *explicit* representations of the transport. In this chapter, we have opted for a

parametric paradigm, seeking the transport map within a finite-dimensional approximation class. Parameterizing high-dimensional functions is broadly challenging (and can rapidly become intractable), but exploiting the sparsity, decomposability, and low-rank structure of transports can dramatically reduce the burden associated with explicit representations. Within any structure of this kind, however, we would still like to introduce the fewest degrees of freedom possible: for instance, we may know that a component of the map should depend only on a small subset of the input variables, but what are the best basis functions to capture this dependence? A possible approach is the **adaptive enrichment** of the approximation space of the map during the optimization routine. The main question is how to drive the enrichment. A standard approach is to compute the gradient of the objective of the optimization problem over a slightly richer approximation space and to detect the new degrees of freedom that should be incorporated in the parameterization of the transport. This is in the same spirit as adjoint-based techniques in adaptive finite element methods for differential equations [7]. In the context of transport maps, however, it turns out that one can *exactly* evaluate the first variation of the objective over an infinite-dimensional function space containing the transport. A rigorous and systematic analysis of this first variation can guide targeted enrichment of the approximation space for the map [12]. Alternatively, one could try to construct rather complex transports by **composing** simple maps and rotations of the space. This idea has proven successful in high-dimensional applications (see [63] for the details of an algorithm and [44, 81] for related approaches).

Even after finding efficient parameterizations that exploit available low-dimensional structure, we must still search for the best transport map within a finite-dimensional approximation space. As a result, our transports will in general be only approximate. This fact should not be surprising or alarming. It is the same issue that one faces, for instance, when solving a differential equation using the finite element method [78]. The important feature of the transport map framework, however, is that we can estimate the quality of an approximate transport and decide whether to enrich the approximation space to improve the accuracy of the map or to accept the bias resulting from use of an approximate map to sample the target distribution. In Sect. 3.3 we reviewed many properties and possible applications of approximate transports. Perhaps the most notable is the use of approximate maps to precondition existing sampling techniques such as MCMC. In particular, we refer to [64] for a use of approximate transport maps in the context of adaptive MCMC, where a low-order map is learned from MCMC samples and used to construct efficient non-Gaussian proposals that allow long-range global moves even for highly correlated targets.

So far, the transport map framework has been deployed successfully in a number of challenging applications: high-dimensional non-Gaussian Bayesian inference involving expensive forward models [61], multiscale methods for Bayesian inverse problems [65], non-Gaussian proposals for MCMC algorithms [64], and Bayesian optimal experimental design [39]. Ongoing and future applications of the framework include sequential data assimilation (Bayesian filtering and smoothing), statistical modeling via non-Gaussian Markov random fields, density estimation

and inference in likelihood-free settings (e.g., with radar and image data), and rare event simulation.

References

1. Adams, M.R., Guillemin, V.: *Measure Theory and Probability*. Birkhäuser Basel (1996)
2. Ambrosio, L., Gigli, N.: A user's guide to optimal transport. In: Benedetto, P., Michel, R. (eds) *Modelling and Optimisation of Flows on Networks*, pp. 1–155. Springer, Berlin/Heidelberg (2013)
3. Andrieu, C., Moulines, E.: On the ergodicity properties of some adaptive MCMC algorithms. *Ann. Appl. Probab.* **16**(3), 1462–1505 (2006)
4. Agangen, S., Haker, S., Tannenbaum, A.: Minimizing flows for the Monge–Kantorovich problem. *SIAM J. Math. Anal.* **35**(1), 61–97 (2003)
5. Atkins, E., Morzfeld, M., Chorin, A.J.: Implicit particle methods and their connection with variational data assimilation. *Mon. Weather Rev.* **141**(6), 1786–1803 (2013)
6. Attias, H.: Inferring parameters and structure of latent variable models by variational Bayes. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, pp. 21–30. Morgan Kaufmann Publishers Inc. (1999)
7. Bangerth, W., Rannacher, R.: *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Basel (2013)
8. Bardsley, J.M., Solonen, A., Haario, H., Laine, M.: Randomize-then-optimize: a method for sampling from posterior distributions in nonlinear inverse problems. *SIAM J. Sci. Comput.* **36**(4), A1895–A1910 (2014)
9. Beaumont, M.A., Zhang, W., Balding, D.J.: Approximate Bayesian computation in population genetics. *Genetics* **162**(4), 2025–2035 (2002)
10. Benamou, J.D., Brenier, Y.: A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numer. Math.* **84**(3), 375–393 (2000)
11. Bernard, P., Buffoni, B.: Optimal mass transportation and Mather theory. *J. Eur. Math. Soc.* **9**, 85–121 (2007)
12. Bigoni, D., Spantini, A., Marzouk, Y.: On the computation of monotone transports (2016, preprint)
13. Bonnotte, N.: From Knothe's rearrangement to Brenier's optimal transport map. *SIAM J. Math. Anal.* **45**(1), 64–87 (2013)
14. Box, G., Cox, D.: An analysis of transformations. *J. R. Stat. Soc. Ser. B* **26**(2), 211–252 (1964)
15. Brenier, Y.: Polar factorization and monotone rearrangement of vector-valued functions. *Commun. Pure Appl. Math.* **44**(4), 375–417 (1991)
16. Brooks, S., Gelman, A., Jones, G., Meng, X.L. (eds.): *Handbook of Markov Chain Monte Carlo*. Boca Raton (2011)
17. Calderhead, B.: A general construction for parallelizing Metropolis-Hastings algorithms. *Proc. Natl. Acad. Sci.* **111**(49), 17408–17413 (2014)
18. Carlier, G., Galichon, A., Santambrogio, F.: From Knothe's transport to Brenier's map and a continuation method for optimal transport. *SIAM J. Math. Anal.* **41**(6), 2554–2576 (2010)
19. Champion, T., De Pascale, L.: The Monge problem in \mathbb{R}^d . *Duke Math. J.* **157**(3), 551–572 (2011)
20. Chib, S., Jeliazkov, I.: Marginal likelihood from the Metropolis-Hastings output. *J. Am. Stat. Assoc.* **96**(453), 270–281 (2001)
21. Chorin, A., Morzfeld, M., Tu, X.: Implicit particle filters for data assimilation. *Commun. Appl. Math. Comput. Sci.* **5**(2), 221–240 (2010)
22. Chorin, A.J., Tu, X.: Implicit sampling for particle filters. *Proc. Natl. Acad. Sci.* **106**(41), 17,249–17,254 (2009)
23. Csilléry, K., Blum, M.G.B., Gaggiotti, O.E., François, O.: Approximate Bayesian computation (ABC) in practice. *Trends Ecol. Evol.* **25**(7), 410–8 (2010)

24. Cui, T., Law, K.J.H., Marzouk, Y.M.: Dimension-independent likelihood-informed MCMC. *J. Comput. Phys.* **304**(1), 109–137 (2016)
25. Cui, T., Martin, J., Marzouk, Y.M., Solonen, A., Spantini, A.: Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Probl.* **30**(11), 114,015 (2014)
26. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. R. Stat. Soc. B* **68**(3), 411–436 (2006)
27. Feyel, D., Üstünel, A.S.: Monge-Kantorovitch measure transportation and Monge-Ampere equation on Wiener space. *Probab. Theory Relat. Fields* **128**(3), 347–385 (2004)
28. Fox, C.W., Roberts, S.J.: A tutorial on variational Bayesian inference. *Artif. Intell. Rev.* **38**(2), 85–95 (2012)
29. Gautschi, W.: Orthogonal polynomials: applications and computation. *Acta Numer.* **5**, 45–119 (1996)
30. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: *Bayesian Data Analysis*, 2nd edn. Chapman and Hall, Boca Raton (2003)
31. Gelman, A., Meng, X.L.: Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Stat. Sci.* **13**, 163–185 (1998)
32. Ghorpade, S., Limaye, B.V.: *A Course in Multivariable Calculus and Analysis*. Springer, New York (2010)
33. Gilks, W., Richardson, S., Spiegelhalter, D. (eds.): *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London (1996)
34. Girolami, M., Calderhead, B.: Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B* **73**, 1–37 (2011)
35. Goodman, J., Lin, K.K., Morzfeld, M.: Small-noise analysis and symmetrization of implicit Monte Carlo samplers. *Commun. Pure Appl. Math.* **2–4**, n/a (2015)
36. Gorham, J., Mackey, L.: Measuring sample quality with Stein’s method. In: *Advances in Neural Information Processing Systems*, Montréal, Canada, pp. 226–234 (2015)
37. Haario, H., Saksman, E., Tamminen, J.: An adaptive metropolis algorithm. *Bernoulli* **7**(2), 223–242 (2001)
38. Haber, E., Rehman, T., Tannenbaum, A.: An efficient numerical method for the solution of the L_2 optimal mass transfer problem. *SIAM J. Sci. Comput.* **32**(1), 197–211 (2010)
39. Huan, X., Parno, M., Marzouk, Y.: Adaptive transport maps for sequential Bayesian optimal experimental design (2016, preprint)
40. Jaakkola, T.S., Jordan, M.I.: Bayesian parameter estimation via variational methods. *Stat. Comput.* **10**(1), 25–37 (2000)
41. Kim, S., Ma, R., Mesa, D., Coleman, T.P.: Efficient Bayesian inference methods via convex optimization and optimal transport. *IEEE Symp. Inf. Theory* **6**, 2259–2263 (2013)
42. Kleywegt, A., Shapiro, A., Homem-de-Mello, T.: The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* **12**(2), 479–502 (2002)
43. Kushner, H., Yin, G.: *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, New York (2003)
44. Laparra, V., Camps-Valls, G., Malo, J.: Iterative gaussianization: from ICA to random rotations. *IEEE Trans. Neural Netw.* **22**(4), 1–13 (2011)
45. Laurence, P., Pignol, R.J., Tabak, E.G.: Constrained density estimation. In: *Quantitative Energy Finance*, pp. 259–284. Springer, New York (2014)
46. Le Maître, O., Knio, O.M.: *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Springer, Dordrecht/New York (2010)
47. Litvinenko, A., Matthies, H.G.: Inverse Problems and Uncertainty Quantification. arXiv:1312.5048 (2013)
48. Litvinenko, A., Matthies, H.G.: Uncertainty quantification and non-linear Bayesian update of PCE coefficients. *PAMM* **13**(1), 379–380 (2013)
49. Liu, J.S.: *Monte Carlo Strategies in Scientific Computing*. Springer, New York (2004)
50. Loeper, G., Rapetti, F.: Numerical solution of the Monge–Ampère equation by a Newton’s algorithm. *Comptes Rendus Math.* **340**(4), 319–324 (2005)
51. Luenberger, D.G.: *Optimization by Vector Space Methods*. Wiley, New York (1968)

52. Marin, J.M., Pudlo, P., Robert, C.P., Ryder, R.J.: Approximate Bayesian computational methods. *Stat. Comput.* **22**(6), 1167–1180 (2012)
53. Martin, J., Wilcox, L., Burstedde, C., Ghattas, O.: A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM J. Sci. Comput.* **34**(3), 1460–1487 (2012)
54. Matthies, H.G., Zander, E., Rosić, B.V., Litvinenko, A., Pajonk, O.: Inverse problems in a Bayesian setting. arXiv:1511.00524 (2015)
55. McCann, R.: Existence and uniqueness of monotone measure-preserving maps. *Duke Math. J.* **80**(2), 309–323 (1995)
56. Meng, X.L., Schilling, S.: Warp bridge sampling. *J. Comput. Graph. Stat.* **11**(3), 552–586 (2002)
57. Monge, G.: Mémoire sur la théorie des déblais et de remblais. In: *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, pp. 666–704 (1781)
58. Morzfeld, M., Chorin, A.J.: Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation. arXiv:1109.3664 (2011)
59. Morzfeld, M., Tu, X., Atkins, E., Chorin, A.J.: A random map implementation of implicit filters. *J. Comput. Phys.* **231**(4), 2049–2066 (2012)
60. Morzfeld, M., Tu, X., Wilkening, J., Chorin, A.: Parameter estimation by implicit sampling. *Commun. Appl. Math. Comput. Sci.* **10**(2), 205–225 (2015)
61. Moselhy, T., Marzouk, Y.: Bayesian inference with optimal maps. *J. Comput. Phys.* **231**(23), 7815–7850 (2012)
62. Neal, R.M.: MCMC using Hamiltonian dynamics. In: Brooks, S., Gelman, A., Jones, G.L., Meng, X.L. (eds.) *Handbook of Markov Chain Monte Carlo*, chap. 5, pp. 113–162. Taylor and Francis, Boca Raton (2011)
63. Parno, M.: Transport maps for accelerated Bayesian computation. Ph.D. thesis, Massachusetts Institute of Technology (2014)
64. Parno, M., Marzouk, Y.: Transport Map Accelerated Markov Chain Monte Carlo. arXiv:1412.5492 (2014)
65. Parno, M., Moselhy, T., Marzouk, Y.: A Multiscale Strategy for Bayesian Inference Using Transport Maps. arXiv:1507.07024 (2015)
66. Ramsay, J.: Estimating smooth monotone functions. *J. R. Stat. Soc. Ser. B* **60**(2), 365–375 (1998)
67. Reich, S.: A nonparametric ensemble transform method for Bayesian inference. *SIAM J. Sci. Comput.* **35**(4), A2013–A2024 (2013)
68. Renegar, J.: *A Mathematical View of Interior-Point Methods in Convex Optimization*, vol. 3. SIAM, Philadelphia (2001)
69. Robert, C.P., Casella, G.: *Monte Carlo Statistical Methods*, 2nd edn. Springer, New York (2004)
70. Rosenblatt, M.: Remarks on a multivariate transformation. *Ann. Math. Stat.* **23**(3), 470–472 (1952)
71. Rosić, B.V., Litvinenko, A., Pajonk, O., Matthies, H.G.: Sampling-free linear Bayesian update of polynomial chaos representations. *J. Comput. Phys.* **231**(17), 5761–5787 (2012)
72. Saad, G., Ghanem, R.: Characterization of reservoir simulation models using a polynomial chaos-based ensemble Kalman filter. *Water Resour. Res.* **45**(4), n/a (2009)
73. Smith, A., Doucet, A., de Freitas, N., Gordon, N. (eds.): *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
74. Spall, J.C.: *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, vol. 65. Wiley, Hoboken (2005)
75. Spantini, A., Marzouk, Y.: On the low-dimensional structure of measure transports (2016, preprint)
76. Spantini, A., Solonen, A., Cui, T., Martin, J., Tenorio, L., Marzouk, Y.: Optimal low-rank approximations of Bayesian linear inverse problems. *SIAM J. Sci. Comput.* **37**(6), A2451–A2487 (2015)

-
77. Stavropoulou, F., Müller, J.: Parameterization of random vectors in polynomial chaos expansions via optimal transportation. *SIAM J. Sci. Comput.* **37**(6), A2535–A2557 (2015)
 78. Strang, G., Fix, G.J.: *An Analysis of the Finite Element Method*, vol. 212. Prentice-Hall, Englewood Cliffs (1973)
 79. Stuart, A.M.: Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559 (2010)
 80. Sullivan, A.B., Snyder, D.M., Rounds, S.A.: Controls on biochemical oxygen demand in the upper Klamath River, Oregon. *Chem. Geol.* **269**(1-2), 12–21 (2010)
 81. Tabak, E., Turner, C.V.: A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics* **66**(2), 145–164 (2013)
 82. Tabak, E.G., Trigila, G.: Data-driven optimal transport. *Commun. Pure Appl. Math.* **10**, 1002 (2014)
 83. Thode, H.C.: *Testing for Normality*, vol. 164. Marcel Dekker, New York (2002)
 84. Villani, C.: *Topics in Optimal Transportation*, vol. 58. American Mathematical Society, Providence (2003)
 85. Villani, C.: *Optimal Transport: Old and New*, vol. 338. Springer, Berlin/Heidelberg (2008)
 86. Wackernagel, H.: *Multivariate Geostatistics: An Introduction with Applications*. Springer-Verlag Berlin Heidelberg (2013)
 87. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* **1**(1–2), 1–305 (2008)
 88. Wang, L.: Methods in Monte Carlo computation, astrophysical data analysis and hypothesis testing with multiply-imputed data. Ph.D. thesis, Harvard University (2015)
 89. Wilkinson, D.J.: *Stochastic Modelling for Systems Biology*. CRC Press, Boca Raton (2011)
 90. Wright, S.J., Nocedal, J.: *Numerical Optimization*, vol. 2. Springer, New York (1999)
 91. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)

Jerrad Hampton and Alireza Doostan

Abstract

A salient task in uncertainty quantification (UQ) is to study the dependence of a quantity of interest (QoI) on input variables representing system uncertainties. Relying on linear expansions of the QoI in orthogonal polynomial bases of inputs, polynomial chaos expansions (PCEs) are now among the widely used methods in UQ. When there exists a smoothness in the solution being approximated, the PCE exhibits sparsity in that a small fraction of expansion coefficients are significant. By exploiting this sparsity, *compressive sampling*, also known as *compressed sensing*, provides a natural framework for accurate PCE using relatively few evaluations of the QoI and in a manner that does not require intrusion into legacy solvers. The PCE possesses a rich structure between the QoI being approximated, the polynomials, and input variables used to perform the approximation and where the QoI is evaluated. In this chapter insights are provided into this structure, summarizing a portion of the current literature on PCE via compressive sampling within the context of UQ.

Keywords

Legendre Polynomials • Hermite Polynomials • Orthogonal Polynomials • Compressed Sensing • Polynomial Chaos Expansions • Markov Chain Monte Carlo • ℓ_1 -minimization • Basis Pursuit • Sparse Approximation

Contents

1	Introduction	828
1.1	Problem Formulation	828
1.2	The Askey Scheme	830
1.3	Polynomial Order	831
1.4	Sparsity in PCE	832

J. Hampton (✉) • A. Doostan

Aerospace Engineering Sciences, University of Colorado, Boulder, CO, USA
e-mail: alireza.doostan@colorado.edu

2	Solution Computation	833
2.1	Basis Pursuit Denoising (BPDN)	834
2.2	Orthogonal Matching Pursuit (OMP)	835
3	Measures for Recovery	835
3.1	Phase Transition Diagrams	835
3.2	Restricted Isometry Constant (RIC)	836
3.3	Coherence of Bounded Orthonormal Systems	837
3.4	Mutual Coherence	838
4	Improving Recovery	839
4.1	Sampling Distribution	839
4.2	Column Weighting	842
4.3	Incorporating Derivative Evaluations	844
5	Thermally Driven Cavity Flow Example	846
6	Conclusion	851
	References	851

1 Introduction

One task of uncertainty quantification (UQ) is to understand how various quantities of interest (QoI) behave as functions of uncertain system inputs. An ineffective understanding may give unfounded confidence in the QoI or lead to unnecessary restrictions in the system inputs due to lack of knowledge of the QoI. In the probabilistic framework, input uncertainties are modeled by random variables. Polynomial chaos expansions (PCEs) have been significantly studied as a method to identify an approximation to the map between the inputs and the QoI that, for a variety of practical purposes, are both tractably computable and sufficiently accurate [36, 56, 84, 86]. The main idea behind PCEs is to expand a finite variance QoI in a basis of (multivariate) polynomials that are orthogonal with respect to the joint probability density function of the inputs. Estimating the expansion coefficients has classically relied on methods based on Galerkin projections [36, 86], Monte Carlo and quadrature integrations, [56, 85], as well as standard least-squares regression [8, 39, 41]. When the QoI depends *smoothly* on the inputs, the PCE coefficients associated with higher-order basis converge to zero rapidly. This results in a natural *sparsity* in the coefficient vector or, equivalently, a smaller set of unknown coefficients to be estimated. In such cases, compressive sampling techniques [12, 15, 16, 22, 25, 26, 33] have been recently employed and proven as a tool for generating PCEs with a reduced computational cost [1, 9, 10, 31, 32, 38, 40, 43, 46, 49, 50, 57, 59, 64, 65, 70, 72–74, 78, 83, 88–91].

This chapter will investigate how UQ problems may be formulated for use with PCE, how PCE may be computed via compressive sampling, theoretical measures that can gauge this recovery, and methods which attempt to accelerate the recovery.

1.1 Problem Formulation

Probability is a natural framework for modeling uncertain inputs by assuming that the input depends on a d -dimensional random vector $\boldsymbol{\xi} := (\xi_1, \dots, \xi_d)$ with some joint probability density function $f(\boldsymbol{\xi})$, where $\boldsymbol{\xi}$ represents a realization of

the random vector, $\boldsymbol{\Xi}$. In this manner a scalar QoI, denoted by $u(\boldsymbol{\Xi})$, is modeled as an unknown function of the random input, and the goal is to approximate this function. This approximation then forms a surrogate model for $u(\boldsymbol{\Xi})$, which can be accessed with negligible cost.

Often $u(\boldsymbol{\Xi})$ depends on a function $v(\mathbf{x}, t, \boldsymbol{\Xi})$, the solution to a set of partial or ordinary differential equations,

$$\mathcal{R}(\mathbf{x}, t, \boldsymbol{\Xi}; v) = \mathbf{0}, \quad (\mathbf{x}, t) \in \mathcal{D} \times [0, T], \quad (24.1)$$

with appropriate boundary and initial conditions. Here, \mathcal{R} denotes the set of governing equations, $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^D$, $D = 1, 2, 3$, the spatial variable, and $t \in [0, T]$ the time.

As the computation of realizations of $v(\mathbf{x}, t, \boldsymbol{\Xi})$, and hence of $u(\boldsymbol{\Xi})$, requires potentially the use of legacy codes, it is often preferred not to alter the solver for v to handle $\boldsymbol{\Xi}$ in a sophisticated manner, e.g., by considering the Galerkin projection of \mathcal{R} into a subspace spanned by a basis in $\boldsymbol{\Xi}$. Additionally, the evaluation of a realized $u(\boldsymbol{\Xi})$ can become a significant computational bottleneck, and in these cases it is prudent to identify an approximation using as few realizations of $u(\boldsymbol{\Xi})$ as is practical.

Approximating $u(\boldsymbol{\Xi})$, assumed to have finite variance, using a basis expansion in multivariate orthogonal polynomials, each of which is denoted by $\psi_k(\boldsymbol{\Xi})$, yields the PCE

$$u(\boldsymbol{\Xi}) = \sum_{k=1}^{\infty} c_k \psi_k(\boldsymbol{\Xi}). \quad (24.2)$$

The unknown coefficients $c_k, k = 1, 2, \dots$ can be computed via the projection equation

$$c_k = \int u(\boldsymbol{\xi}) \psi_k(\boldsymbol{\xi}) f(\boldsymbol{\xi}) d\boldsymbol{\xi} = \mathbb{E}[u(\boldsymbol{\Xi}) \psi_k(\boldsymbol{\Xi})], \quad (24.3)$$

leading to the mean-squares convergence of (24.2), when appropriately truncated. In (24.3), \mathbb{E} denotes the mathematical expectation operator and $\psi_k(\boldsymbol{\Xi})$ are assumed to be normalized such that $\mathbb{E}[\psi_k^2(\boldsymbol{\Xi})] = 1$. Although it is possible to construct polynomials that are orthogonal with respect to a larger class of $f(\boldsymbol{\xi})$, here it is assumed that the coordinates of $\boldsymbol{\Xi}$ are independent, which implies that $f(\boldsymbol{\xi})$ is a product of its d marginal distributions. Orthogonal polynomials are well understood for a variety of distributions, significantly easing the identification and evaluation of the orthogonal polynomials, as well as the analysis of approximation in those polynomials.

Within the PCE framework, it is possible to identify useful sets of basis polynomials, methods to truncate the series to be computationally tractable, and methods that cheaply and effectively approximate the coefficients c_k . These form the basis of the subsequent discussions.

Table 24.1 The orthogonal polynomials corresponding to several commonly used continuous distributions from the Askey scheme

Distribution of random variable, $f(\xi)$	Normal	Uniform	Gamma	Beta
Associated orthogonal polynomial, $\psi_k(\Xi)$	Hermite	Legendre	Laguerre	Jacobi

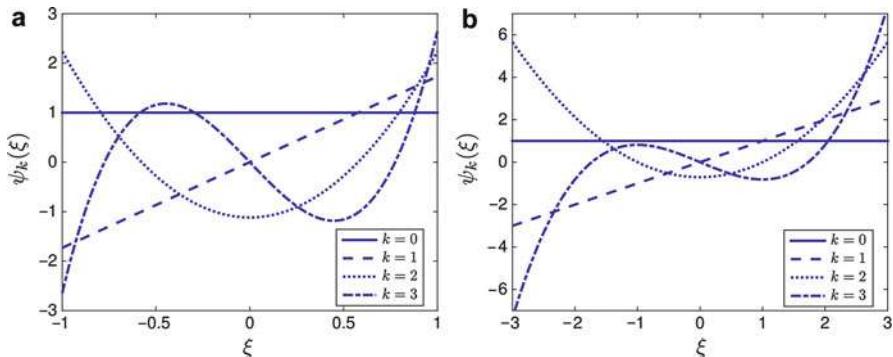


Fig. 24.1 Plots of orthonormal Legendre (a) and Hermite (b) polynomials of degree up to three

1.2 The Askey Scheme

In the one-dimensional case, where $\Xi = \Xi$ is distributed according to $f(\xi)$, $\psi_k(\Xi)$ are polynomials of order k that are orthogonal with respect to $f(\xi)$. For several commonly used distributions, such an association between $\psi_k(\Xi)$ and $f(\xi)$ is given by the Askey scheme of polynomials [4], as seen in Table 24.1.

The order k of the one-dimensional polynomial refers to the highest degree of that polynomial, allowing an indexing of these polynomials from zero to infinity. Figure 24.1 shows low-order orthonormal Legendre and Hermite examples, which are the focus in this chapter. Notice how the S-shape of the third-order Legendre polynomial is contained within $[-1, 1]$, while the third-order Hermite polynomial has an S-shape oscillation that is stretched over a larger domain. This behavior continues at higher orders, with high-order Hermite polynomials being oscillatory over ever-expanding domains, while Legendre polynomials are oscillatory within $[-1, 1]$, achieving their extreme values in that interval at the endpoints. Additionally, the Legendre polynomials tend to reach higher peaks in their oscillations than their Hermite counterparts. For results concerning properties of these and other orthogonal polynomial systems, the interested reader is referred to [3, 4, 75, 77].

In the d -dimensional case, the orthogonal polynomials are a tensor product of the d orthogonal polynomials in each respective dimension, thus allowing different orthogonal polynomials in the various coordinates of Ξ , when following different distributions. As an example, if Ξ_1 is normally distributed while Ξ_2 is uniformly distributed, then each basis polynomial is a product of a Hermite polynomial in Ξ_1 and a Legendre polynomial in Ξ_2 . For notational simplicity, considerations are needed to index the order of any polynomial with respect to each dimension. For this

purpose, a multi-index by the degree of polynomial along each dimension is utilized here. Specifically, if \mathbf{k} is a d -index, then k_i refers to the order of the polynomial in the i th dimension, and

$$\psi_{\mathbf{k}}(\boldsymbol{\Xi}) = \prod_{i=1}^d \psi_{k_i}(\Xi_i),$$

is the appropriate d -dimensional orthogonal polynomial, where the polynomial is of the appropriate type in each dimension. There are several potential notions of order for multidimensional polynomials, as described in the next section.

1.3 Polynomial Order

Definitions of multivariate order determine which multi-indices \mathbf{k} correspond to included basis functions, thus defining which basis functions are used in a given approximation. For a given approximation of a certain order, denote the set of basis functions in the set of order p by \mathcal{K}_p and the number of elements in that set by P or $|\mathcal{K}_p|$. The definition of order also determines which functions, u , may be accurately recovered for a given \mathcal{K}_p , in that the accuracy of the approximation relates to the basis functions available for the corresponding approximation,

$$u(\boldsymbol{\Xi}) \approx \hat{u}(\boldsymbol{\Xi}) := \sum_{\mathbf{k} \in \mathcal{K}_p} c_{\mathbf{k}} \psi_{\mathbf{k}}(\boldsymbol{\Xi}), \quad (24.4)$$

which by (24.3) is the approximation minimizing $\|u - \hat{u}\|_{\mathcal{L}_2(\boldsymbol{\Xi})}$. Perhaps, the simplest formulation of polynomial order is that of tensor order, defined to be

$$\mathcal{K}_p^{(\infty)} := \{\mathbf{k} : \|\mathbf{k}\|_{\infty} \leq p\},$$

where the superscript (∞) is chosen due to the norm on \mathbf{k} when viewed as a vector. This set has $|\mathcal{K}_p^{(\infty)}| = (p+1)^d$ polynomials in it, which is exponentially dependent on d , but has the benefit of being able to handle a high level of interaction among the different dimensions. However, it is rare that u has such a high dependence between dimensions so as to simultaneously require high-order polynomials in all dimensions to accurately reconstruct u . For this reason, a more popular definition of order for PCE, total order, is defined by

$$\mathcal{K}_p^{(1)} := \{\mathbf{k} : \|\mathbf{k}\|_1 \leq p\}. \quad (24.5)$$

A combinatorial consideration reveals that this definition yields $|\mathcal{K}_p^{(1)}| = \binom{p+d}{d} \leq \min\{(p+1)^d, (d+1)^p\}$ basis functions, which often gives a significant reduction in the number of basis functions though it grows rapidly if both p and d are large.

Most functions of interest have a weak enough interaction among dimensions to be effectively represented by this type of truncation for modest p .

A more general notion of order which has been used in building bases and generalizes the above definitions is given by

$$\mathcal{K}_p^{(\alpha)} := \left\{ \mathbf{k} : \left(\sum_{i=1}^d k_i^\alpha \right)^{1/\alpha} \leq p \right\}, \quad (24.6)$$

where $\alpha > 0$ is a parameter controlling the shape of this set of basis functions. Smaller α reduces the number of basis functions more dramatically, particularly by removing polynomials of higher orders in multiple dimensions. In contrast, larger α more rapidly increases the number of basis functions with the advantage of being more robust to recovering solutions possessing highly interactive dimensions.

In this chapter, the focus is on the total-order bases as defined in (24.5), corresponding to $\alpha = 1$ in (24.6), and the superscript is dropped. As seen in the next section, approximations to u using such a basis often require a relatively small proportion of these basis functions, leading to a sparsity in computed solutions.

1.4 Sparsity in PCE

The sparsity in the PCE is closely intertwined with recovery via compressive sampling. Intuitively, a PCE is sparse if a small but unknown subset of the basis set may still accurately approximate the function u . Refer to the set of indices of these basis functions as \mathcal{S}_p , noting that it depends on p , in the sense that with a higher-order approximation, previously unavailable basis functions may improve the approximation. Intuitively, smooth u are well approximated by a truncated Taylor series expansion, which can itself be constructed from a small number of basis polynomials. Specifically, a function u is s -sparse if for some $\mathcal{S}_p \subset \mathcal{K}_p$ with $|\mathcal{S}_p| = s$,

$$u(\boldsymbol{\xi}) = \sum_{\mathbf{k} \in \mathcal{S}_p} c_{\mathbf{k}} \psi_{\mathbf{k}}(\boldsymbol{\xi}).$$

Practically, this will not hold with equality, but rather only as an accurate approximation, often referred to as approximate sparsity. An example of such sparsity, adapted from [64], is a solution computed from a thermally driven cavity flow problem shown in Fig. 24.2 and discussed later, exhibiting a significant decay in coefficient magnitude as order increases and for polynomials which vary with respect to dimensions that are less useful for constructing an accurate approximation. This solution is for a problem summarized in Fig. 24.8. Though a priori information as to which basis functions are useful in approximating u is often unavailable, there is often reasonable certainty that the basis could be compressed to a smaller number

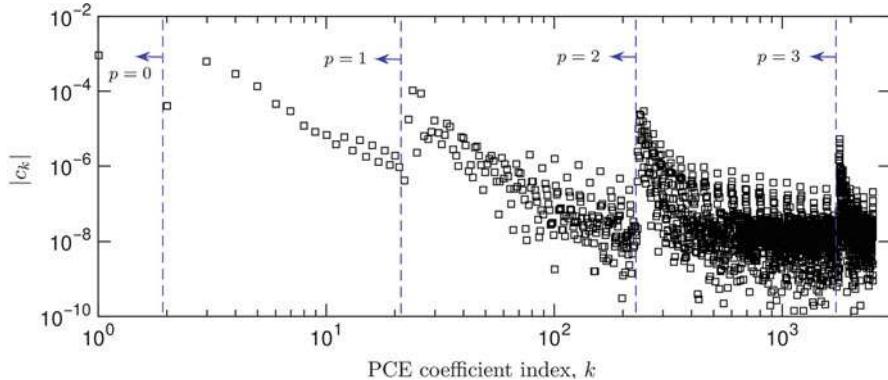


Fig. 24.2 Solution coefficient magnitude for basis functions of varied order, demonstrating coefficient decay motivating a sparse reconstruction. Coefficient indices to the left of the *vertical dashed lines* are included in the set of PC basis functions of order not larger than the specified p

of basis functions, in a set \mathcal{S}_p of size s . If it is known which basis functions were in \mathcal{S}_p , then one could accurately approximate the s corresponding coefficients using a relatively small, possibly linear in s , number of realizations. However, there is typically a large number of basis functions in the set \mathcal{K}_p , and no knowledge of what basis functions may or may not be in \mathcal{S}_p . Compressive sampling discerns which basis functions are appropriate at the time that the coefficients are accurately computed, revealing the sparsity in the solution while computing an approximation in this sparse basis.

2 Solution Computation

While there is motivation to identify a solution that has small support \mathcal{S}_p , finding the smallest such support is a difficult combinatorial problem [26], and instead a similar, tractable problem is solved. Starting from (24.4) a matrix equation can be constructed from realized data that may be solved to identify the coefficients c_k .

Specifically, utilizing N realizations of Ξ , each denoted by $\xi^{(i)}$, one can define N equations in a linear system, where methods to select these realizations are discussed in a later section. Here, let \mathbf{u} be a vector with $u_i = u(\xi^{(i)})$. For some indexing of polynomials that maps $\mathbf{k} \in \mathcal{K}_p$ to $j \in \{1, \dots, P\}$, let \mathbf{c} be a vector with entries c_j , and $\Psi_{i,j} = \psi_j(\xi^{(i)})$. This gives an $N \times 1$ vector \mathbf{u} , a $P \times 1$ vector \mathbf{c} , and an $N \times P$ matrix Ψ , referred to as the *measurement matrix*. Denote the approximation to \mathbf{c} by $\hat{\mathbf{c}}$. It is reasonable to approximate \mathbf{c} by $\hat{\mathbf{c}}$ computed using Ψ and \mathbf{u} . Moreover, compressive sampling allows this to be done while undersampling the $N \times P$ matrix Ψ , that is when $N < P$. One problem to investigate is

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \Psi \mathbf{c} = \mathbf{u}, \quad (24.7)$$

where $\|\cdot\|_0$ is the number of nonzero elements of the argument. However, this is an NP-hard combinatorial problem [26], and so it is practical to instead consider approximate solutions. Two methods are considered here, though several other methods to approximately solve (24.7) exist; see, e.g., [6, 24, 51, 61, 62, 87].

2.1 Basis Pursuit Denoising (BPDN)

One useful method is the convex relaxation of the $\|\cdot\|_0$ objective function in (24.7) to the ℓ_1 norm $\|\cdot\|_1$,

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \Psi \mathbf{c} = \mathbf{u}. \quad (24.8)$$

This problem, referred to as basis pursuit [23], is a convex optimization and may be solved efficiently via, for instance, primal-dual interior-point methods as implemented in [7, 30]. The ℓ_1 -minimization problem (24.8) is the main focus here, along with a greedy method to approximate the solution to (24.7). The relation between (24.7) and (24.8) is an important point of analysis, which is partially addressed by Theorems 2 and 4, whose presentation is deferred until later sections.

For many practical problems, the condition $\Psi \mathbf{c} = \mathbf{u}$ is often too restrictive, and so a nonzero residual is admitted, leading to a problem such as

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \|\mathbf{u} - \Psi \mathbf{c}\|_2 < \epsilon, \quad (24.9)$$

which is known as basis pursuit denoising (BPDN). It is possible to select ϵ via the statistical technique of cross-validation [2, 11, 31, 82]. In many practical contexts, the truncation error defined via (24.4) as $u - \hat{u}$ implies a need for positive ϵ to account for the QoI not being fully explained by the chosen basis.

Closely connected with (24.9) is the regularized formulation (for some positive parameter τ),

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{u} - \Psi \mathbf{c}\|_2 \quad \text{subject to} \quad \|\mathbf{c}\|_1 \leq \tau, \quad (24.10)$$

which is known as LASSO [79]. Related to this is the Bayesian formulation [5, 45, 49, 63, 72], based on a model for the entries of the mismatch $\mathbf{u} - \Psi \mathbf{c}$, e.g., a centered Gaussian with prescribed or parametric standard deviation and a Laplace prior distribution on \mathbf{c} ,

$$\pi(\mathbf{c} | \lambda) := \prod_{k \in \mathcal{K}_p} \frac{\lambda}{2} e^{-\lambda|c_k|}, \quad (24.11)$$

for some parameter λ . For computational purposes, often an alternative, but similar in nature, prior distribution on \mathbf{c} is employed [5].

2.2 Orthogonal Matching Pursuit (OMP)

Orthogonal matching pursuit (OMP) is a popular greedy method which approximates a solution to (24.7) via a greedy search [80, 81]. Matching pursuits are greedy methods which update the approximation by using a matching criterion with the residual to iteratively include a new basis function to an active basis set, denoted here by \mathcal{A} , where coefficients are nonzero. Orthogonal matching pursuit, presented in Algorithm 1, recomputes the coefficients by a least-squares approximation after each iterative addition to the active set and updates the residual accordingly. Note that each iterative least-squares solve involves only $|\mathcal{A}|$ columns of Ψ , and these least squares can be computed efficiently and accurately [39, 60]. In practice this algorithm is faster than BPDN, while providing a good approximation to the solution of (24.9).

Algorithm 1 Orthogonal Matching Pursuit (OMP)

```

 $\mathcal{A} = \emptyset, \mathbf{r} = \mathbf{u}.$ 
while  $\|\mathbf{r}\|_2 > \delta$ 
     $k = \arg \max_{j \notin \mathcal{A}} \frac{\mathbf{r}^T \Psi(:, j)}{\|\Psi(:, j)\|_2^2}.$ 
     $\mathcal{A} = \mathcal{A} \cup k.$ 
     $\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{u} - \Psi \mathbf{c}\|_2 \quad \text{subject to} \quad c_k = 0 \quad \forall k \notin \mathcal{A}.$ 
     $\mathbf{r} = \mathbf{u} - \Psi \hat{\mathbf{c}}.$ 
end while

```

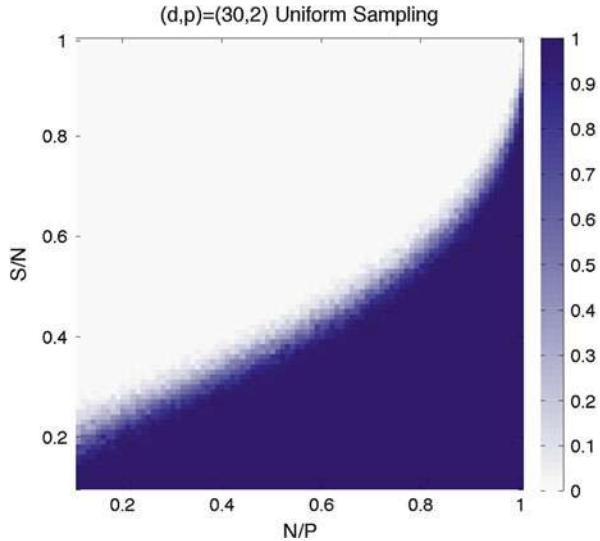
3 Measures for Recovery

Several measures for recovery are focused on identifying when the solution of the ℓ_1 -minimization problem (24.9) or OMP without noise is equivalent to the solution of (24.7). This recovery is also shown to be robust to noise, which includes the stability of the problem with respect to the truncation error as well as potential errors in identifying $u(\xi)$. Though a variety of measures are considered here, other properties relating to the spark and null space of a matrix are additionally useful [25, 47, 48].

3.1 Phase Transition Diagrams

A direct demonstration of recovery is how well randomly generated, sparse signals are recovered. An effective normalization to consider is the level of sparsity relative to the sampling s/N , as well as the degree of sampling relative to the number of basis functions N/P . There is a limiting transition with recovery in many cases, which describes the number of samples needed for a particular sparsity to recover a truly sparse solution [28]. For a large class of random matrices and solvers, there is a limiting behavior [28] as seen in Fig. 24.3. This phase transition is seen in many PCEs [39].

Fig. 24.3 Phase transition diagram associated with ℓ_1 -minimization's recovery of random sparse solutions with Legendre polynomials from a $d = 30, p = 2$ total-order basis with samples drawn uniformly from the d -dimensional hypercube, displaying archetypal transition behavior. The colors represent the probability of successful solution recovery (Figure adapted from [40])



3.2 Restricted Isometry Constant (RIC)

In this context, the Restricted isometry constant (RIC) [18, 34, 58], closely related to the uniform uncertainty principle [14], is defined to be the smallest constant δ_s satisfying

$$(1 - \delta_s) \|\mathbf{c}_s\|_2^2 \leq \frac{1}{N} \|\Psi \mathbf{c}_s\|_2^2 \leq (1 + \delta_s) \|\mathbf{c}_s\|_2^2, \quad (24.12)$$

for all \mathbf{c}_s having s or fewer nonzero elements. Note that smaller δ_s implies a near isometry of any s columns of Ψ , and it is for this reason that the RIC is analytically useful. Further, when δ_s is less than some threshold for an appropriate s , then the matrix Ψ is said to satisfy a restricted isometry property (RIP), guaranteeing several results. An example of such an RIP result is a theorem restated from [69].

Theorem 1 ([69]). Let $\mathbf{c} \in \mathbb{R}^P$ represent a solution to be approximated and $\hat{\mathbf{c}}$ be the solution to (24.9). Let

$$\boldsymbol{\eta} := \Psi \mathbf{c} - \mathbf{u}$$

denote the contribution from sources of error, and let the ϵ from (24.9) be chosen such that $\|\boldsymbol{\eta}\|_2 < \epsilon$. If

$$\delta_{2s} < \delta_\star := 3/(4 + \sqrt{6}) \approx 0.4652,$$

then the following error estimates hold, where c_1, c_2, c_3 , and c_4 depend only on δ_{2s} .

$$\|\mathbf{c} - \hat{\mathbf{c}}\|_2 \leq \frac{c_1}{\sqrt{s}} \inf_{\|\mathbf{c}_s\|_0 \leq s} \|\mathbf{c}_s - \mathbf{c}\|_1 + c_2 \epsilon;$$

$$\|\mathbf{c} - \hat{\mathbf{c}}\|_1 \leq c_3 \inf_{\|\mathbf{c}_s\|_0 \leq s} \|\mathbf{c}_s - \mathbf{c}\|_1 + c_4 \epsilon \sqrt{s},$$

and $\|\cdot\|_0$ again refers to the number of nonzero elements of the vector.

Unfortunately, for practically sized s , the RIC is difficult to compute for a given matrix, and as a result any RIP can be difficult to verify. Still, the RIC can be guaranteed in a probabilistic sense and lead to a minimum sampling size N for a successful solution recovery via (24.8). We defer the statement of any particular theorem until the coherence parameter of the next section is defined.

3.3 Coherence of Bounded Orthonormal Systems

Let $\Psi(i, :)$ denote the i th row of Ψ and $\Psi(:, j)$ denote the j th column of Ψ . The rows of Ψ are assumed to be independent and identically distributed. Recall that as basis polynomials are taken according to the Askey scheme, an appropriate normalization for the basis polynomials gives that

$$\frac{1}{N} \mathbb{E}(\Psi^T \Psi) = \mathbb{E}((\Psi(1, :))^T \Psi(1, :)) = I, \quad (24.13)$$

which is equivalent to using the orthonormalized basis polynomials. A useful measure is the parameter [19] given by

$$\mu := \sup_{k \in \mathcal{K}_p, \xi \in \Omega} |\psi_k(\xi)|^2, \quad (24.14)$$

where Ω is the support of Ξ . μ is often referred to as an incoherence or coherence parameter; here it is referred to as a coherence parameter. From (24.14), μ corresponds to the largest absolute value taken by any of the basis functions evaluated in the domain of the random input. This definition of coherence is related to the recovery of solutions by (24.7), (24.8), (24.9) and the RIC [13]. Often, such as when considering Hermite polynomials, (24.14) is unbounded over all of Ω , and it is necessary to restrict the domain of Ξ to a smaller set as in [19, 40], giving a boundedness with high probability. With μ defined in this way, one may bound the RIC as in Theorem 8.1 of [68], reproduced here.

Theorem 2 ([68]). Suppose $\eta \in (0, 1)$, and $\delta \in (0, 0.5]$. If $C \leq 243150$ and

$$\frac{N}{\log(10N)} \geq C \mu \delta^{-2} s \log^2(100s) \log(4P) \log(7\eta^{-1}), \quad (24.15)$$

then with probability at least $1 - \eta$, it follows that the RIC for Ψ satisfies $\delta_s \leq \delta$.

This result may be combined with Theorem 1 to bound recovery. Although not needed here, it is worthwhile noting that the result in Theorem 2 can be improved slightly to that of Theorem 8.4 in [68]. The RIC implies uniform recovery over all potential c , but this guarantee of uniform recovery is often unneeded in applications.

If one instead wishes to assure recovery for an arbitrary but fixed \mathbf{c} , then one may invoke Theorem 1.1 of [19], reproduced here.

Theorem 3 ([19]). *Let \mathbf{c} be a fixed but otherwise arbitrary vector satisfying $\|\mathbf{c}\|_0 = s$. Then with probability $1 - 5/P - \exp(-\beta)$, \mathbf{c} solves (24.8) provided that*

$$N \geq C_0(1 + \beta)\mu s \log(P). \quad (24.16)$$

We note that this result is stated for the solution to (24.8), though a similar result holds for the solution to (24.9) [19].

3.4 Mutual Coherence

Mutual coherence [27, 29], also referred to as coherence, is used to analyze recovery via BPDN, but is particularly useful for identifying when OMP reproduces the solution of (24.7). The mutual coherence is defined by

$$M := \sup_{1 \leq i \neq j \leq P} \frac{|\Psi^T(:, i)\Psi(:, j)|}{\|\Psi(:, i)\|_2 \|\Psi(:, j)\|_2}. \quad (24.17)$$

To eliminate potential confusion with the coherence parameter, μ , of the previous section, (24.17) is here referred to as mutual coherence. Note that this definition coincides with $M = \delta_2$ from (24.12), and Section 2.5 of [68] considers other relations in more depth. This definition relates closely to recovery via OMP, which relies heavily on inner products, as Theorem 5.1 of [29], restated here.

Theorem 4 ([29]). *Let ϵ be the error parameter in OMP, and let \mathbf{c} satisfy*

$$\|\mathbf{c}\|_0 \leq \frac{1+M}{2M} - \frac{\epsilon}{M} \cdot \frac{1}{\min_{c_k \neq 0} |c_k|}.$$

If $\hat{\mathbf{c}}$ is the recovered approximation to \mathbf{c} as calculated via OMP, then the supports are identical, that is

$$\text{supp}(\mathbf{c}) = \text{supp}(\hat{\mathbf{c}}).$$

Additionally, with regard to the ℓ_2 norm of error,

$$\|\hat{\mathbf{c}} - \mathbf{c}\|_2^2 \leq \frac{\epsilon^2}{1 - M(\|\mathbf{c}\|_0 - 1)}.$$

While this coherence involves the inner products between vectors, a different definition is useful for random matrices commonly utilized in PCE. This related

definition of mutual coherence is often given in terms of inner products of columns from two separate matrices [12, 53], although this two-matrix approach is not considered here. In this approach one matrix is the current matrix Ψ associated with realizations of $u(\Xi)$, and the second matrix is used in place of Ψ for computation of \hat{c} . This allows analysis of methods where the matrix used in the computation of c differs from the matrix used to recover u .

4 Improving Recovery

Knowing how the recovery of compressive sampling solutions relates to measurable quantities can provide insights into how to adjust the method to elicit change. Presented here are simple changes to standard approaches that may give considerable improvements. Other methods not discussed here which can significantly improve solution recovery include Bayesian adaptations [45, 49, 55] and iteratively adjusting the basis to generate quality approximations using fewer basis functions [43, 72].

4.1 Sampling Distribution

Several methods exist for identifying the $\xi^{(i)}$ used to form the linear system in (24.7). A straightforward approach is to draw $\xi^{(i)}$ independently from the distribution of Ξ , i.e., $f(\xi)$. This distribution is, in general, not an ideal sampling method for solution recovery [40], and it may be desired to sample from a different distribution, $g(\xi)$. Unfortunately, this would destroy the orthogonality of the basis functions in that (24.13) would no longer hold. This, however, can be corrected within the context of importance sampling by noting if $g(\xi) > 0$ for $\xi \in \Omega$, then

$$\delta_{i,j} = \int_{\Omega} \psi_i(\xi) \psi_j(\xi) f(\xi) d\xi = \int_{\Omega} \frac{f(\xi)}{g(\xi)} \psi_i(\xi) \psi_j(\xi) g(\xi) d\xi.$$

Thus, if Ξ is sampled according to $g(\xi)$, and each basis sample is weighted proportionally to $(f(\xi)/g(\xi))^{1/2}$, then orthogonality is restored. Motivated by this, define $w(\xi) := (f(\xi)/g(\xi))^{1/2}$ and W to be a diagonal matrix such that $W_{i,i} = w(\xi^{(i)})$. In practice, it is sufficient to know $w(\xi)$ up to a normalizing constant. Consider the solution to the preconditioned ℓ_1 -minimization problem

$$\hat{c} = \arg \min_c \|c\|_1 \quad \text{subject to} \quad \|Wu - W\Psi c\|_2 < \epsilon, \quad (24.18)$$

where the weight must also be applied to the corresponding realization of QoI, $u(\xi)$, as the matrix relation $W\Psi c = Wu$ insures that $\Psi c = u$. Here, the choice of $g(\xi)$ depends in particular on (24.14), as described in the next section. Note that the inclusion of $w(\xi)$ leads to a new definition for (24.14) given by

$$\mu_w := \sup_{k \in \mathcal{K}_p, \xi \in \Omega} |w(\xi) \psi_k(\xi)|^2, \quad (24.19)$$

for an appropriately normalized $w(\xi)$. This admits a modification to this parameter based solely on the sampling distribution and the corresponding weight.

4.1.1 Sampling for Minimizing (24.19)

To minimize (24.19), take $w(\xi)$ such that $\sup_{k \in \mathcal{K}_p} |w(\xi)\psi_k(\xi)|^2$ is constant in ξ [40].

This gives

$$w(\xi) := \left(\sup_{k \in \mathcal{K}_p} |\psi_k(\xi)| \right)^{-1}, \quad (24.20)$$

which implies the sampling distribution

$$g(\xi) := c \left(\sup_{k \in \mathcal{K}_p} |\psi_k(\xi)| \right)^2 f(\xi), \quad (24.21)$$

where c is the necessary normalizing constant to insure that $g(\xi)$ is a probability distribution and is typically unneeded in practice. Some of these coherence-minimizing distributions, $g(\xi)$, for one-dimensional Hermite and Legendre polynomials, are provided for reference in Fig. 24.4. Here, the limit for the Legendre sampling distribution when $p \rightarrow \infty$ is the Chebyshev distribution, defined to be

$$g(\xi) := \frac{1}{\pi \sqrt{1 - \xi^2}}. \quad (24.22)$$

In contrast, the limiting behavior for the Hermite sampling can be derived from theoretical results in [54] and is more nuanced, as depicted in Fig. 24.5. Here, the input on the horizontal axis scales as $\tilde{\xi} = \xi / (\sqrt{2} \sqrt{2p+1})$, and for $\tilde{\xi}$ defined this way the distribution has theoretical form $g(\tilde{\xi}) = \frac{6}{5} |\tilde{\xi}|^{-1/6}$ for $\tilde{\xi} \in [-1, 1]$.

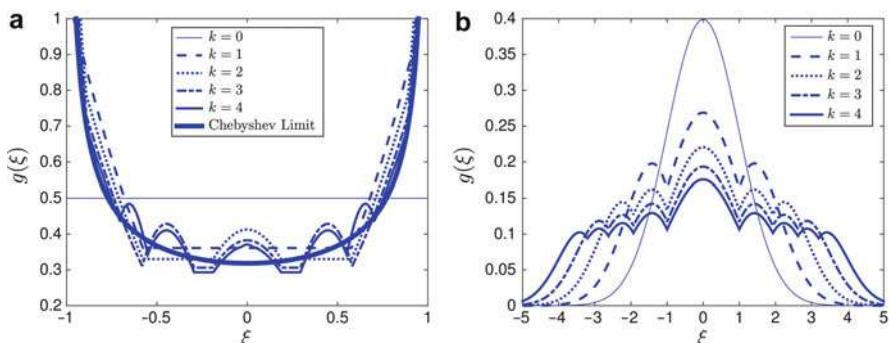
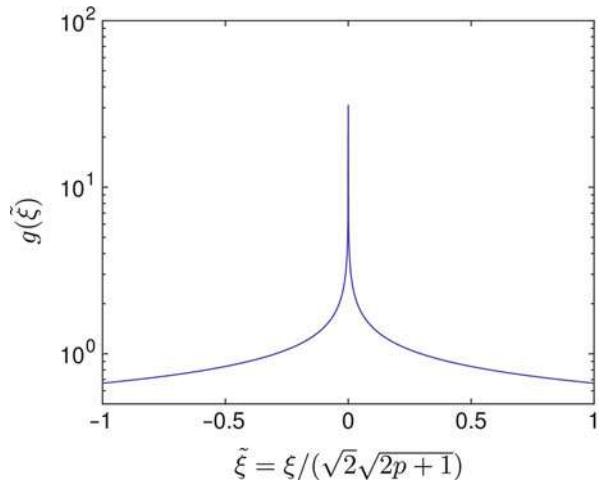


Fig. 24.4 Plots of a few coherence-minimizing sampling distributions for Legendre (a) and Hermite (b) polynomials

Fig. 24.5 Limiting behavior of $g(\xi)$ for Hermite polynomials in one dimension. Note that the scaled input maintains a dependence on the order p



As the support grows with p , a normalization dividing by $\sqrt{2}\sqrt{2p+1}$ is required to scale the support of the distribution, and so the limit of the coherence-minimizing distributions for a basis of total-order p as $p \rightarrow \infty$ is not itself a distribution. In a practical sense, sampling uniformly from a d -dimensional hypersphere of radius $\sqrt{2p+1}$ yields reasonable coherence values [40] and is used here as a distribution for high-order expansions.

There is a practical method to sample from $g(\xi)$ so as to minimize (24.19) based on Markov chain Monte Carlo (MCMC) sampling [37], which does not require the computation of the normalizing constant c in (24.21). A Metropolis-Hastings [55] type version of MCMC sampling is presented in Algorithm 2 and uses random samples which are rejected and accepted in such a way that the distribution of samples converges to the desired distribution, which is the stationary distribution of a Markov chain. This is a particularly useful tool when the sampling distribution is accessible up to an unknown normalizing constant, as in the case of $g(\xi)$. There are several considerations [71] to take into account when using Algorithm 2 to sample from $g(\xi)$. One is that as MCMC samples are dependent, there may be significant serial correlation between samples, and so a number of intermediate samples, denoted by T in Algorithm 2, may need to be discarded to reduce this effect. Another consideration is that a number of initial discarded samples are required to insure that the samples are being drawn approximately from the desired distribution. Both of these considerations depend on the choice of proposal distribution $g_p(\xi)$ in Algorithm 2. Choices of proposal distributions depend on whether the PCE is of a higher dimension or order [40] and in the latter case can be derived from asymptotic considerations of the polynomials, leading to the asymptotic distributions shown in Figs. 24.4 and 24.5. For example, if a high-order Legendre approximation is used, $g_p(\xi)$ can be taken to be the Chebyshev distribution, while a low-order Legendre approximation would be better served by a uniformly distributed $g_p(\xi)$. Similarly, for a high-order Hermite approximation,

one may use a uniform distribution on a d -dimensional hypersphere of radius $\sqrt{2}\sqrt{2p+1}$, as in [40].

Algorithm 2 Metropolis-Hastings Sampler with Static Proposal Distribution

```

Let  $g(\xi)$  be the desired distribution from which to draw.
Draw  $x$  according to proposal distribution  $g_p(\xi)$ .
for The number of desired samples.
  Set  $k = 0$ .
  while  $k \leq T$ 
    Draw  $y$  according to  $g_p(\xi)$ .
    Let  $r = \frac{g(y)g_p(x)}{g(x)g_p(y)}$ . Note that this does not require a normalizing constant for  $g(\xi)$ .
    With probability  $\min\{1, r\}$ , set  $x$  to be  $y$ .
    Increment  $k$ .
  end while
  Save sample  $x$ , approximately distributed according to  $g(\xi)$ .
end for

```

Sampling in this way has been seen to lead to phase-transition diagrams as in Fig. 24.3 over a range of dimensions and orders for both Hermite and Legendre polynomials [40]. The effects of the various sampling methods on the phase-transition diagrams of randomly generated Legendre expansions are presented in Fig. 24.6.

4.1.2 Quadrature-Based Sampling

Random samples may not be ideal in all applications as the solution may depend, possibly largely in highly under-sampled cases, on the actual samples used. This has motivated the utilization of more *structured* samples. Quadratures for the computation of one-dimensional integrals have a rich history and structure [77], allowing a surprising level of accuracy with relatively few evaluations. A tensor product of one-dimensional quadrature rules can provide a d -dimensional quadrature, but this leads to a dense grid, growing exponentially in d , that is most appropriate for tensor order bases and undesirable for most applications. This tensor grid is often subsampled using the Smolyak sparse grid construction [35, 76], allowing for a significantly smaller number of evaluations that may still accurately integrate/interpolate the solution. In the context of compressive sampling, a random subsampling of the grid of quadrature nodes may be utilized to solve the ℓ_1 -minimization problem (24.8) [78].

4.2 Column Weighting

If a given, positive, diagonal, weight matrix W is added, (24.8) can be changed to

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{W}\mathbf{c}\|_1 \quad \text{subject to} \quad \Psi\mathbf{c} = \mathbf{u}. \quad (24.23)$$

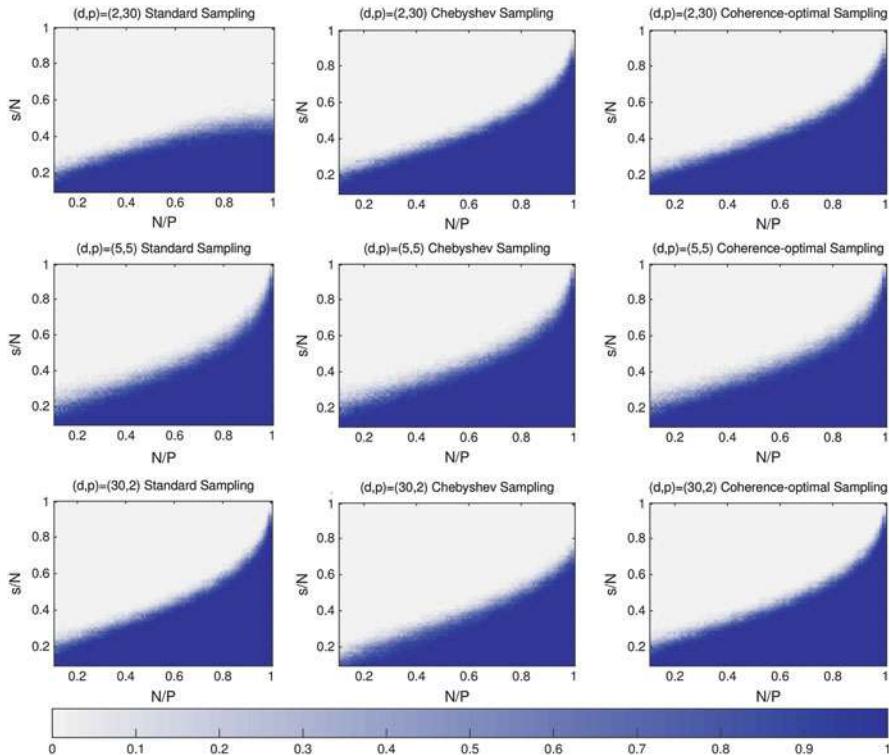


Fig. 24.6 Phase transition diagrams for Legendre PCE under various sampling strategies. Different rows correspond to different problems ranging from high dimensional to high order. Different columns correspond to different sampling distributions (Figure adapted from [40] with stricter ℓ_1 -minimization solver tolerance and larger number of iterations)

Notice that values $\mathbf{W}_{k,k}$ which are relatively large impose a higher penalty for larger \hat{c}_k , and so the corresponding values of \hat{c}_k will tend to be smaller than the solution to (24.8). Similarly, relatively small values of $\mathbf{W}_{k,k}$ will yield a smaller penalty for larger \hat{c}_k , and so the values of \hat{c}_k will tend to be larger than the solution to (24.8). In this way, \mathbf{W} distorts the ℓ_1 -ball and via the effect displayed in Fig. 24.7 may improve solution recovery [1, 17, 21, 42, 51, 61, 64, 70, 87, 90]. Weights may be adjusted iteratively and (24.23) recomputed, in a process called iteratively reweighted ℓ_1 -minimization [17, 90]. This adjustment is also related to a variant of (24.7) via the concept of weighted sparsity, as in [1, 70]. The choice of weights influences the computed solutions based on which coordinates are more heavily or lightly weighted, and so the weights must be identified carefully. Sometimes these weights can be derived by a careful analysis of decay rates [64], other times by a simplified or lower-fidelity model. In the context of function interpolation via solution of (24.8), column weighting is used to reduce aliasing, insuring that high-order basis functions are not overrepresented [1, 70].

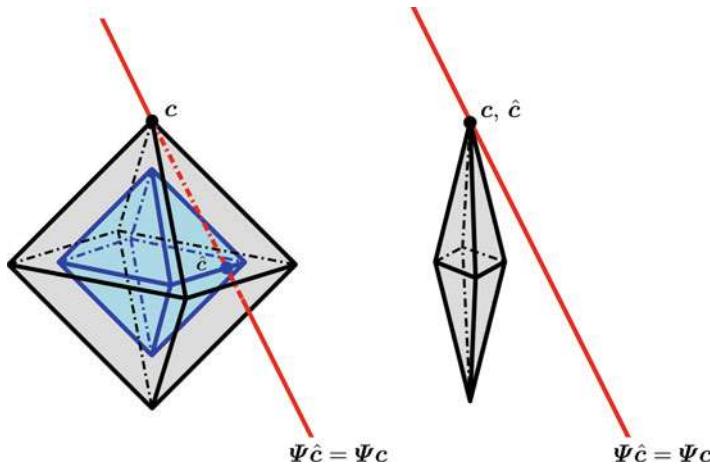


Fig. 24.7 Schematic of approximation of a sparse $c_0 \in \mathbb{R}^3$ via standard and weighted ℓ_1 -minimization. (a) Standard ℓ_1 -minimization for a particular solution, leading to a suboptimally sparse solution \hat{c} with more nonzero entries than c . (b) Weighted ℓ_1 -minimization for the same solution, leading to an optimally sparse computed solution, $\hat{c} = c$ [20]

4.3 Incorporating Derivative Evaluations

In practical applications the computation of a realized QoI may be accompanied with the computation of derivatives. These derivatives may be computed from automatic differentiation [67] or from direct or adjoint sensitivity equations [52]. These derivatives are often used to identify sensitivity, for design decisions, and for the solutions of inverse problems. Derivatives are also used to improve the quality of regression by insuring that the approximation is accurate not only for function values but also for its derivatives [44, 65], and this is how they are utilized here in the context of PCE. To change (24.8) and (24.9) appropriately, define

$$\mathbf{u}_\partial := \left(\frac{\partial u}{\partial \xi_1}(\xi^{(1)}), \dots, \frac{\partial u}{\partial \xi_d}(\xi^{(1)}), \dots, \frac{\partial u}{\partial \xi_1}(\xi^{(N)}), \dots, \frac{\partial u}{\partial \xi_d}(\xi^{(N)}) \right)^T \quad (24.24)$$

and

$$\Psi_\partial((i-1) \cdot d + k, j) := \frac{\partial \psi_j}{\partial \xi_k}(\xi^{(i)}), \quad k = 1, \dots, d. \quad (24.25)$$

In this way, \mathbf{u}_∂ contains all the partial derivative information for $u(\xi)$ with respect to each coordinate of ξ . Similarly, Ψ_∂ contains the partial derivatives of each basis function with respect to each coordinate of ξ . Whereas before Ψ had independent rows, here instead it has independent submatrices of size $(d+1) \times P$ for the

measurement matrix. Let the submatrix of independent information related with the k th sample be given by \mathbf{X}_k , with a generic realization given by \mathbf{X} . Specifically,

$$\begin{aligned}\mathbf{X}(i, j) &= \frac{\partial \psi_j}{\partial \xi_i}(\xi), \quad i = 1, \dots, d; \\ \mathbf{X}(d + 1, j) &= \psi_j(\xi).\end{aligned}$$

For simplicity of presentation, let

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\Psi} \\ \boldsymbol{\Psi}_\partial \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} \mathbf{u} \\ \mathbf{u}_\partial \end{pmatrix}. \quad (24.26)$$

With these definitions, \mathbf{c} can be approximated by the solution to the ℓ_1 -minimization problem

$$\mathbf{c} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \|\mathbf{v} - \boldsymbol{\Phi} \mathbf{W}_\partial \mathbf{c}\|_2 \leq \delta. \quad (24.27)$$

Here, \mathbf{W}_∂ is a fixed matrix depending on the basis functions and their derivatives chosen to insure that

$$\mathbf{W}_\partial^T \mathbb{E}(\mathbf{N}^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Phi}) \mathbf{W}_\partial = \mathbf{I}. \quad (24.28)$$

It should be noted that $\mathbb{E}(\mathbf{N}^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Phi}) = \mathbb{E}(\mathbf{X}^T \mathbf{X})$ is a symmetric positive definite matrix, admitting the Cholesky decomposition of $\mathbb{E}(\mathbf{X}^T \mathbf{X}) = \mathbf{L} \mathbf{L}^T$. With this decomposition $\mathbf{W}_\partial^T = \mathbf{L}^{-1}$. As a concrete example for identifying \mathbf{W}_∂ , consider the case of a d -dimensional Hermite polynomial basis. In this case, \mathbf{W}_∂ can be derived from Lemma 3.1 of [65], restated here.

Theorem 5 ([65]). *For d -dimensional orthonormal probabilists' Hermite polynomials ψ_i and ψ_j with order i_k, j_k in dimension k ,*

$$\mathbb{E} \left(\psi_i(\xi) \psi_j(\xi) + \sum_{k=1}^d \frac{\partial \psi_i}{\partial \xi_k}(\xi) \frac{\partial \psi_j}{\partial \xi_k}(\xi) \right) = \delta_{i,j} \left(1 + \sum_{k=1}^d i_k \right), \quad (24.29)$$

where $\delta_{i,j}$ is the Kronecker delta.

This theorem implies that \mathbf{W}_∂ is diagonal and having entries determined by the degree of the polynomial in each of the various dimensions. In general, \mathbf{W}_∂ will not be diagonal, for example, when considering d -dimensional Legendre polynomials.

It is possible to define coherence in the presence of derivative information in a manner that extends (24.14). Specifically, let

$$\mu_\nabla := \sup_{k \in \mathcal{K}_p, \xi \in \Omega} \|X(:, k)\|_2^2, \quad (24.30)$$

where the supremum is taken over all columns of X and potential realizations of X . From (24.30), μ_V represents the largest ℓ_2 -norm of a column of X evaluated at the domain of the random input. Stated another way, μ_V represents the largest ℓ_2 -contribution to Φ from any specific basis function over any potential realization. This definition of coherence is related to the recovery of solutions by (24.27), both directly and via the RIC in (24.12). As for (24.14), for Hermite polynomials, (24.30) is unbounded, and it is necessary to restrict the domain of Ξ to a smaller set as in [19, 40], giving a boundedness with high probability.

The same guarantees of Theorem 2 hold in this case [65]. The coherence in (24.30) is generally less than that of (24.14) [65] and is compatible with Theorem 3, implying that there is no need to increase sample sizes when including derivative information. In terms of the conditioning of the measurement matrix Ψ , relative to a comparable Φ , including derivative information as described does not reduce the ability of the matrix to uniformly recover solutions, although contributions from evaluation of the QoI and the appropriate derivatives may effect error in the solution computation [65]. In practice, the computational saving associated with including derivative information on the ℓ_1 -minimization solution depends on both the cost of computing the derivatives for the QoI and the benefit of including derivative information.

Note that the importance sampling to minimize (24.30) involves the weight function

$$w(\xi) := \left(\sup_{k \in \mathcal{K}_p} \|X(:, k)\|_2 \right)^{-1}. \quad (24.31)$$

Here, the realized weight function would multiply the entire submatrix for the corresponding sample. The associated sampling distribution is

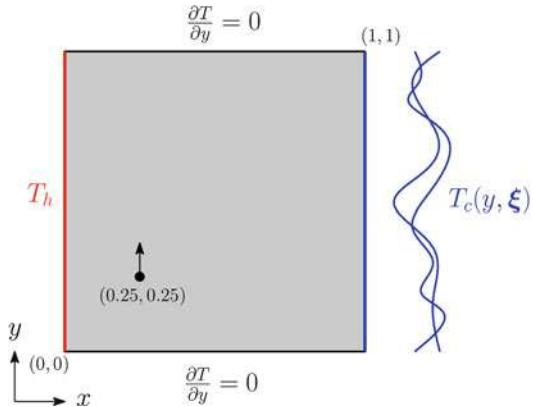
$$g(\xi) := c \left(\sup_{k \in \mathcal{K}_p} \|X(:, k)\|_2 \right)^2 f(\xi), \quad (24.32)$$

where c is a normalizing constant that need not be identified when samples from $g(\xi)$ are drawn via MCMC.

5 Thermally Driven Cavity Flow Example

Following [56, 64, 66], a 2-D heat-driven square cavity flow problem with uncertain wall temperature, as shown in Fig. 24.8, is considered. The left vertical wall has a deterministic, constant temperature \tilde{T}_h , while the right vertical wall has a stochastic temperature $\tilde{T}_c < \tilde{T}_h$ with constant mean $\bar{\tilde{T}}_c$. The superscript tilde (\sim) denotes the nondimensional quantities. Both the top and bottom walls are assumed to be adiabatic. The reference temperature and the reference temperature difference are

Fig. 24.8 Schematic for the cavity flow problem (Figure adapted from [64])



defined as $\tilde{T}_{ref} = (\tilde{T}_h + \tilde{T}_c)/2$ and $\Delta\tilde{T}_{ref} = \tilde{T}_h - \tilde{T}_c$, respectively. Under the assumption of small temperature differences, i.e., the Boussinesq approximation, the governing equations in dimensionless variables are given by

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{\text{Pr}}{\sqrt{\text{Ra}}} \nabla^2 \mathbf{u} + \text{Pr} T \hat{\mathbf{y}}, \\ \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) &= \frac{1}{\sqrt{\text{Ra}}} \nabla^2 T, \end{aligned} \quad (24.33)$$

where $\hat{\mathbf{y}}$ is the unit vector $(0, 1)$, $\mathbf{u} = (u, v)$ is the velocity vector field, $T = (\tilde{T} - \tilde{T}_{ref})/\Delta\tilde{T}_{ref}$ is normalized temperature, p is pressure, and t is time. Nondimensional Prandtl and Rayleigh numbers are defined, respectively, as $\text{Pr} = \tilde{\mu}\tilde{c}_p/\tilde{\kappa}$ and $\text{Ra} = \tilde{\rho}g\beta\Delta\tilde{T}_{ref}\tilde{L}^3/(\tilde{\mu}\tilde{\kappa})$. Specifically, $\tilde{\mu}$ is molecular viscosity, $\tilde{\kappa}$ is thermal diffusivity, $\tilde{\rho}$ is density, g is gravitational acceleration, the coefficient of thermal expansion is given by β , and \tilde{L} is the reference length. In this example, the Prandtl and Rayleigh numbers are set to $\text{Pr} = 0.71$ and $\text{Ra} = 10^6$, respectively. For more details on the nondimensional variables in (24.33), the interested reader is referred to [56, Chapter 6.2]. On the right vertical wall, a (normalized) temperature distribution with stochastic fluctuations of the form

$$\begin{aligned} T_c(x = 1, y, \boldsymbol{\Xi}) &= \bar{T}_c + T'_c, \\ T'_c &= \sigma_T \sum_{i=1}^d \sqrt{\lambda_i} \varphi_i(y) \Xi_i \end{aligned} \quad (24.34)$$

is used, where \bar{T}_c is a constant mean temperature. In (24.34), Ξ_i , $i = 1, \dots, d$ are independent random variables uniformly distributed on $[-1, 1]$. $\{\lambda_i\}_{i=1}^d$ and $\{\varphi_i(y)\}_{i=1}^d$ are the d largest eigenvalues and the corresponding eigenfunctions of

the exponential covariance kernel $C_{T_c T_c}(y_1, y_2) = \exp\left(-\frac{|y_1 - y_2|}{l_c}\right)$, where l_c is the correlation length. In this setting, a (semi-)analytic representation of the eigenpairs $(\lambda_i, \varphi_i(y))$ in (24.34) is available; see, e.g., [36].

Here, $(T_h, \bar{T}_c) = (0.5, -0.5)$, $d = 20$, $l_c = 1/21$, and $\sigma_T = 11/100$. Our QoI is the vertical velocity component at $(x, y) = (0.25, 0.25)$ denoted by $u(\mathcal{E})$.

Figures 24.9a and 24.10a contrast the approximate PCE coefficients obtained via ℓ_1 -minimization, with $N = 200$ and $N = 1000$, respectively, against the reference solution, confirming an improved solution recovery when N is increased.

As an example of the improvement that can be achieved via weighted ℓ_1 -minimization, Figs. 24.9b and 24.10b display approximations to the solution, with and without weights as in Fig. 24.11, generated from a simplified model which does not require the solution to (24.33), [64]. Note that in Fig. 24.9, considerable improvement occurs even though the weights are not a very accurate representation of the solution decay. Figure 24.10 verifies that the weighted and unweighted

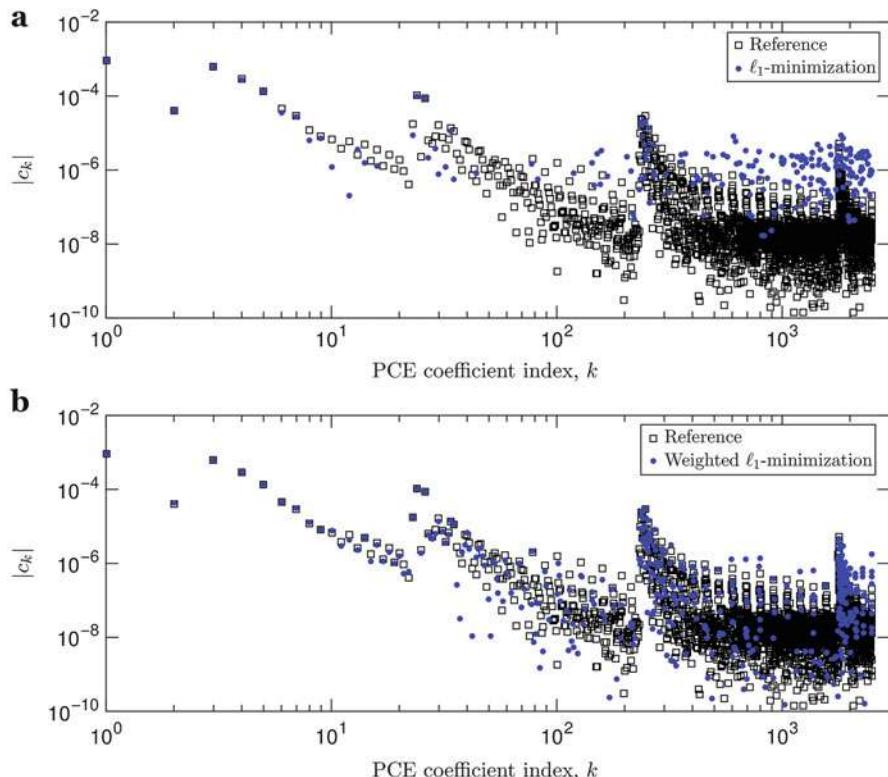


Fig. 24.9 Reference solution overlaid with a computed solution via standard ℓ_1 -minimization (a) and weighted ℓ_1 -minimization (b) using 200 realizations of the input random vector (Figure adapted from [64])

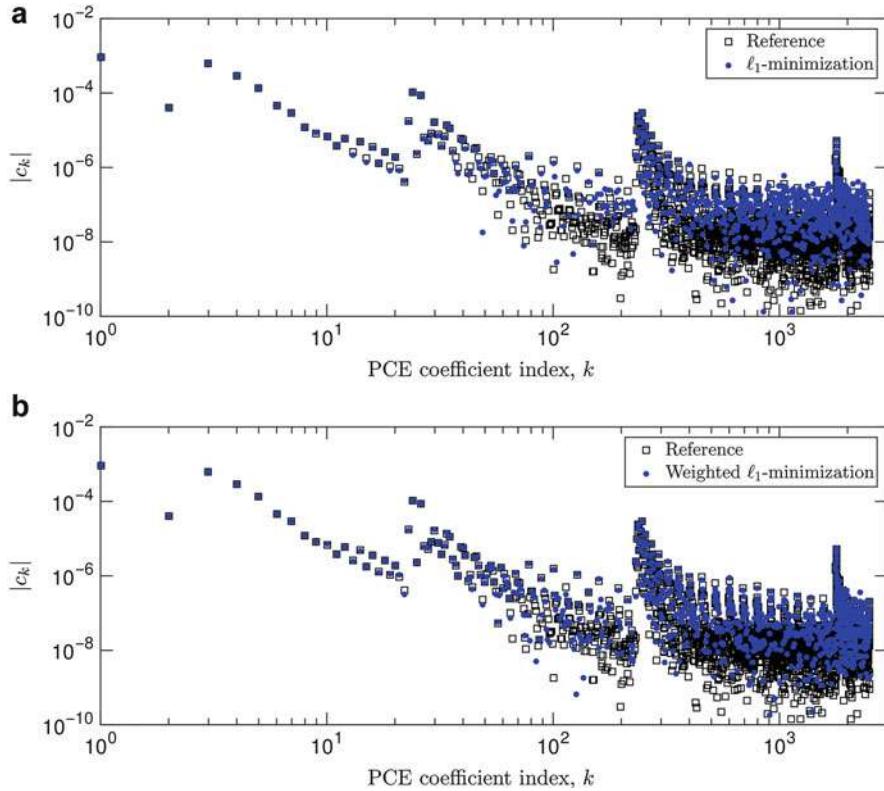


Fig. 24.10 Reference solution overlaid with a computed solution via standard ℓ_1 -minimization (a) and weighted ℓ_1 -minimization (b) using 1000 realizations of the input random vector (Figure adapted from [64])

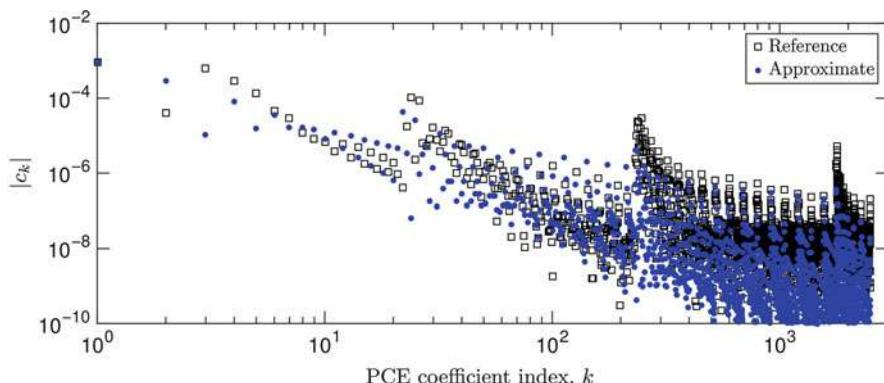


Fig. 24.11 Approximate PCE coefficient magnitudes of the QoI $v(0.25, 0.25)$ for the cavity problem of Fig. 24.8. The approximation is derived from a simplified model, [64], and is used to construct weights for solving (24.23) (Figure adapted from [64])

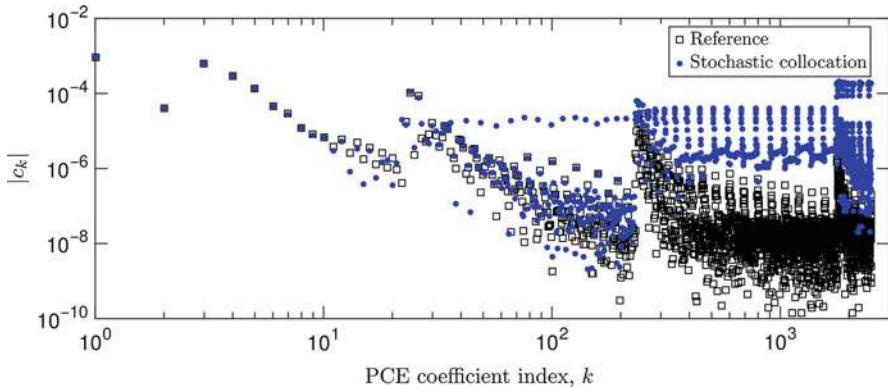
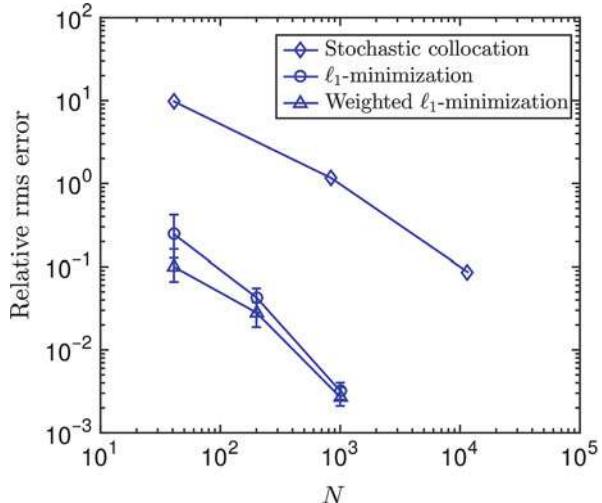


Fig. 24.12 Reference solution overlaid with a computed solution via stochastic collocation with Smolyak sparse grid based on Clenshaw-Curtis abscissas with $N = 841$ samples

Fig. 24.13 Comparison of relative error in statistics of $v(0.25, 0.25)$ computed via ℓ_1 -minimization, weighted ℓ_1 -minimization, and stochastic collocation. The error bars are generated using 100 independent replications with fixed samples size N (Figure adapted from [64])



solutions closely coincide when a sufficient number of samples is available, as also seen in Fig. 24.13. To compare the performance of standard and weighted ℓ_1 -minimization with that of the stochastic collocation, the relative root-mean-square error of the PCE solution as a function of the sample size N is displayed in Fig. 24.12. The stochastic collocation approximates c_k in (24.3) via numerical integration using Smolyak sparse grids [76] generated from Clenshaw-Curtis abscissas with multiple sizes. Two remarks are important to highlight. Firstly, unlike in the stochastic collocation, both standard and weighted ℓ_1 -minimization solutions are computed using random samples. To illustrate the associated variability, solutions with 100 independent replications of the same sample size N are used to establish the uncertainty bars in Fig. 24.13. As can be observed, the solution variability

reduces when larger number of samples are used. Secondly, improving the accuracy of the stochastic collocation (by using a larger sample size N) requires the simulation of an additional number of samples that is dictated by the Smolyak grid construction, as opposed to the available computational budget. This limitation, however, does not apply to the compressive sampling methods described above, as they rely on random samples.

6 Conclusion

The use of polynomial basis functions can impart a sparsity in solutions for many problems, allowing a tractable computation of solutions to a large number of problems. This chapter has looked at how a QoI is built from a computation of realized random variables, how orthogonal basis polynomials are identified in those random variables, and how coefficients for those basis functions are then computed. A variety of theoretical measures were investigated which are useful in gauging how well a solution may be recovered with a given number of samples. Adjustments to the standard approach were considered that may lead to an improvement of recovery.

References

1. Adcock, B.: Infinite-dimensional ℓ_1 minimization and function approximation from pointwise data. arXiv preprint arXiv:150302352 (2015)
2. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Stat. Surv. **4**, 40–79 (2010)
3. Askey, R., Wainger, S.: Mean convergence of expansions in Laguerre and hermite series. Am. J. Math. **87**(3), 695–708 (1965)
4. Askey, R.A., Arthur, W.J.: Some Basic Hypergeometric Orthogonal Polynomials That Generalize Jacobi Polynomials, vol. 319. AMS, Providence (1985)
5. Babacan, S., Molina, R., Katsaggelos, A.: Bayesian compressive sensing using laplace priors. IEEE Trans. Image Process. **19**(1), 53–63 (2010)
6. Becker, S., Bobin, J., Candès, E.J.: NESTA: A fast and accurate first-order method for sparse recovery. ArXiv e-prints (2009). Available from <http://arxiv.org/abs/0904.3367>
7. Berg, E.v., Friedlander, M.P.: SPGL1: a solver for large-scale sparse reconstruction (2007). Available from <http://www.cs.ubc.ca/labs/scl/spgl1>
8. Berveiller, M., Sudret, B., Lemaire, M.: Stochastic finite element: a non intrusive approach by regression. Eur. J. Comput. Mech. Revue (Européenne de Mécanique Numérique) **15**(1–3), 81–92 (2006)
9. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. J. Comput. Phys. **230**, 2345–2367 (2011)
10. Bouchot, J.L., Bykowski, B., Rauhut, H., Schwab, C.: Compressed sensing Petrov-Galerkin approximations for parametric PDEs. In: International Conference on Sampling Theory and Applications (SampTA 2015), pp. 528–532. IEEE (2015)
11. Boufounos, P., Duarte, M., Baraniuk, R.: Sparse signal reconstruction from noisy compressive measurements using cross validation. In: Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing (SSP'07), Madison, pp. 299–303. IEEE Computer Society (2007)

12. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.* **51**(1), 34–81 (2009)
13. Candès, E., Tao, T.: Decoding by linear programming. *IEEE Trans. Inf. Theory* **51**(12), 4203–4215 (2005)
14. Candès, E., Tao, T.: Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory* **52**(12), 5406–5425 (2006)
15. Candès, E., Wakin, M.: An introduction to compressive sampling. *IEEE Signal Process. Mag.* **25**(2), 21–30 (2008)
16. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **52**(2), 489–509 (2006)
17. Candès, E., Wakin, M., Boyd, S.: Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.* **14**(5), 877–905 (2008)
18. Candès, E.J.: The restricted isometry property and its implications for compressed sensing. *C. R. Math.* **346**(9), 589–592 (2008)
19. Candès, E.J., Plan, Y.: A probabilistic and ripless theory of compressed sensing. *IEEE Trans. Inf. Theory* **57**(11), 7235–7254 (2010)
20. Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.* **59**(8), 1207–1223 (2006)
21. Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas (2008)
22. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.* **20**, 33–61 (1998)
23. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. *SIAM Rev.* **43**(1), 129–159 (2001)
24. Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory* **55**(5), 2230–2249 (2009)
25. Davenport, M.A., Duarte, M.F., Eldar, Y.C., Kutyniok, G.: Introduction to Compressed Sensing. Cambridge University Press, Cambridge (2012)
26. Donoho, D.: Compressed sensing. *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006)
27. Donoho, D., Huo, X.: Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inf. Theory* **47**(7), 2845–2862 (2001). doi:10.1109/18.959265
28. Donoho, D., Tanner, J.: Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philos. Trans. R Soc. A Math. Phys. Eng. Sci.* **367**(1906), 4273–4293 (2009)
29. Donoho, D., Elad, M., Temlyakov, V.: Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory* **52**(1), 6–18 (2006)
30. Donoho, D., Stodden, V., Tsai, Y.: About SparseLab (2007)
31. Doostan, A., Owhadi, H.: A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.* **230**, 3015–3034 (2011)
32. Doostan, A., Owhadi, H., Lashgari, A., Iaccarino, G.: Non-adapted sparse approximation of PDEs with stochastic inputs. Tech. Rep. Annual Research Brief, Center for Turbulence Research, Stanford University (2009)
33. Eldar, Y., Kutyniok, G.: Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge (2012)
34. Foucart, S.: A note on guaranteed sparse recovery via, ℓ_1 -minimization. *Appl. Comput. Harmonic Anal.* **29**(1), 97–103 (2010). Elsevier
35. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms* **18**(3–4):209–232 (1998)
36. Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach. Dover, Minneola (2002)
37. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: Markov Chain Monte Carlo in Practice, vol 2. CRC Press, Boca Raton (1996)

38. Hadigol, M., Maute, K., Doostan, A.: On uncertainty quantification of lithium-ion batteries. arXiv preprint arXiv:150507776 (2015)
39. Hampton, J., Doostan, A.: Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. Comput. Methods Appl. Mech. Eng. **290**, 73–97 (2015)
40. Hampton, J., Doostan, A.: Compressive sampling of polynomial chaos expansions: convergence analysis and sampling strategies. J. Comput. Phys. **280**, 363–386 (2015)
41. Hosder, S., Walters, R., Perez, R.: A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-891, Reno (NV) (2006)
42. Huang, A.: A re-weighted algorithm for designing data dependent sensing dictionary. Int. J. Phys. Sci. **6**(3), 386–390 (2011)
43. Jakeman, J., Eldred, M., Sargsyan, K.: Enhancing ℓ_1 -minimization estimates of polynomial chaos expansions using basis selection. J. Comput. Phys. **289**, 18–34 (2015)
44. Jakeman, J.D., Eldred, M.S., Sargsyan, K.: Enhancing ℓ_1 -minimization estimates of polynomial chaos expansions using basis selection. ArXiv e-prints 1407.8093 (2014)
45. Ji, S., Xue, Y., Carin, L.: Bayesian compressive sensing. IEEE Trans. Signal Process. **56**(6), 2346–2356 (2008)
46. Jones, B., Parrish, N., Doostan, A.: Postmaneuver collision probability estimation using sparse polynomial chaos expansions. J. Guidance Control Dyn. **38**(8), 1–13 (2015)
47. Juditsky, A., Nemirovski, A.: Accuracy guarantees for ℓ_1 -recovery. IEEE Trans. Inf. Theory **57**, 7818–7839 (2011)
48. Juditsky, A., Nemirovski, A.: On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization. Math. Program. **127**(1), 57–88 (2011)
49. Karagiannis, G., Lin, G.: Selection of polynomial chaos bases via Bayesian model uncertainty methods with applications to sparse approximation of PDEs with stochastic inputs. J. Comput. Phys. **259**, 114–134 (2014)
50. Karagiannis, G., Konomi, B., Lin, G.: A Bayesian mixed shrinkage prior procedure for spatial–stochastic basis selection and evaluation of gPC expansions: Applications to elliptic SPDEs. J. Comput. Phys. **284**, 528–546 (2015)
51. Khajehnejad, M.A., Xu, W., Avestimehr, A.S., Hassibi, B.: Improved sparse recovery thresholds with two-step reweighted ℓ_1 minimization. In: 2010 IEEE International Symposium on Information Theory Proceedings (ISIT), Austin, pp. 1603–1607. IEEE (2010)
52. Komkov, V., Choi, K., Haug, E.: Design Sensitivity Analysis of Structural Systems, vol. 177. Academic, Orlando (1986)
53. Krahmer, F., Ward, R.: Beyond incoherence: stable and robust sampling strategies for compressive imaging. arXiv preprint arXiv:12102380 (2012)
54. Krasicov, I.: New bounds on the Hermite polynomials. ArXiv Mathematics e-prints [math/0401310](#) (2004)
55. Ma, X., Zabaras, N.: An efficient Bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method. Inverse Probl. **25**, 035,013+ (2009)
56. Maitre, O.L., Knio, O.: Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics. Springer, Dordrecht/New York (2010)
57. Mathelin, L., Gallivan, K.: A compressed sensing approach for partial differential equations with random input data. Commun. Comput. Phys. **12**, 919–954 (2012)
58. Mo, Q., Li, S.: New bounds on the restricted isometry constant δ_{2k} . Appl. Comput. Harmonic Anal. **31**(3), 460–468 (2011)
59. Narayan, A., Zhou, T.: Stochastic collocation on unstructured multivariate meshes. Commun. Comput. Phys. **18**, 1–36 (2015)
60. Narayan, A., Jakeman, J.D., Zhou, T.: A Christoffel function weighted least squares algorithm for collocation approximations. arXiv preprint arXiv:14124305 (2014)
61. Needell, D.: Noisy signal recovery via iterative reweighted ℓ_1 -minimization. In: Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove (2009)
62. Needell, D., Tropp, J.: CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmonic Anal. **26**(3), 301–321 (2008)

63. Park, T., Casella, G.: The Bayesian lasso. *J. Am. Stat. Assoc.* **103**(482), 681–686 (2008)
64. Peng, J., Hampton, J., Doostan, A.: A weighted ℓ_1 -minimization approach for sparse polynomial chaos expansions. *J. Comput. Phys.* **267**, 92–111 (2014)
65. Peng, J., Hampton, J., Doostan, A.: On polynomial chaos expansion via gradient-enhanced ℓ_1 -minimization. arXiv preprint arXiv:150600343 (2015)
66. Quéré, P.L.: Accurate solutions to the square thermally driven cavity at high rayleigh number. *Comput. Fluids* **20**(1), 29–41 (1991)
67. Rall, L.B.: Automatic Differentiation: Techniques and Applications, vol. 120. Springer, Berlin (1981)
68. Rauhut, H.: Compressive sensing and structured random matrices. *Theor. Found. Numer. Methods Sparse Recover.* **9**, 1–92 (2010)
69. Rauhut, H., Ward, R.: Sparse Legendre expansions via ℓ_1 -minimization. *J. Approx. Theory* **164**(5), 517–533 (2012)
70. Rauhut, H., Ward, R.: Interpolation via weighted minimization. *Appl. Comput. Harmonic Anal.* **40**, 321–351 (2015)
71. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics. Springer, New York (2004)
72. Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., Thornton, P.: Dimensionality reduction for complex models via Bayesian compressive sensing. *Int. J. Uncertain. Quantif.* **4**, 63–93 (2013)
73. Savin, E., Resmini, A., Peter, J.: Sparse polynomial surrogates for aerodynamic computations with random inputs. arXiv preprint arXiv:150602318 (2015)
74. Schiavazzi, D., Doostan, A., Iaccarino, G.: Sparse multiresolution regression for uncertainty propagation. *Int. J. Uncertain. Quantif.* (2014). doi:10.1615/Int.J.UncertaintyQuantification.2014010147
75. Schoutens, W.: Stochastic Processes and Orthogonal Polynomials. Springer, New York (2000)
76. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady* **4**, 240–243 (1963)
77. Szegő G (1939) Orthogonal Polynomials. American Mathematical Society, American Mathematical Society
78. Tang, G., Iaccarino, G.: Subsampled gauss quadrature nodes for estimating polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.* **2**(1), 423–443 (2014)
79. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. (Ser. B)* **58**, 267–288 (1996)
80. Tropp, J.: Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory* **50**(10), 2231–2242 (2004). doi:10.1109/TIT.2004.834793
81. Tropp, J.A., Anna, G.C.: Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **53**, 4655–4666 (2007)
82. Ward, R.: Compressed sensing with cross validation. *IEEE Trans. Inf. Theory* **55**(12), 5773–5782 (2009)
83. West, T., Brune, A., Hosder, S., Johnston, C.: Uncertainty analysis of radiative heating predictions for titan entry. *J. Thermophys. Heat Transf.* 1–14 (2015)
84. Xiu, D.: Numerical Methods for Stochastic Computations: A Spectral Method Approach. Princeton University Press, Princeton (2010)
85. Xiu, D., Hesthaven, J.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
86. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
87. Xu, W., Khajehnejad, M., Avestimehr, A., Hassibi, B.: Breaking through the thresholds: an analysis for iterative reweighted ℓ_1 minimization via the Grassmann Angle Framework (2009). ArXiv e-prints Available from <http://arxiv.org/abs/0904.0994>
88. Xu, Z., Zhou, T.: On sparse interpolation and the design of deterministic interpolation points. *SIAM J. Sci. Comput.* **36**(4), A1752–A1769 (2014)

89. Yan, L., Guo, L., Xiu, D.: Stochastic collocation algorithms using ℓ_1 -minimization. *Int. J. Uncertain. Quantif.* **2**(3), 279–293 (2012)
90. Yang, X., Karniadakis, G.E.: Reweighted ℓ_1 minimization method for stochastic elliptic differential equations. *J. Comput. Phys.* **248**, 87–108 (2013)
91. Yang, X., Lei, H., Baker, N., Lin, G.: Enhancing sparsity of hermite polynomial expansions by iterative rotations. arXiv preprint arXiv:150604344 (2015)

Anthony Nouy

Abstract

Parameter-dependent models arise in many contexts such as uncertainty quantification, sensitivity analysis, inverse problems, or optimization. Parametric or uncertainty analyses usually require the evaluation of an output of a model for many instances of the input parameters, which may be intractable for complex numerical models. A possible remedy consists in replacing the model by an approximate model with reduced complexity (a so-called reduced order model) allowing a fast evaluation of output variables of interest. This chapter provides an overview of low-rank methods for the approximation of functions that are identified either with order-two tensors (for vector-valued functions) or higher-order tensors (for multivariate functions). Different approaches are presented for the computation of low-rank approximations, either based on samples of the function or on the equations that are satisfied by the function, the latter approaches including projection-based model order reduction methods. For multivariate functions, different notions of ranks and the corresponding low-rank approximation formats are introduced.

Keywords

High-dimensional problems • Low-rank approximation • Model order reduction • Parameter-dependent equations • Stochastic equations • Uncertainty quantification

Contents

1	Introduction	858
2	Low-Rank Approximation of Order-Two Tensors	860
2.1	Best Rank- m Approximation and Optimal Subspaces	860

A. Nouy (✉)

Department of Computer Science and Mathematics, GeM, Ecole Centrale Nantes, Nantes, France
e-mail: anthony.nouy@ec-nantes.fr

2.2	Characterization of Optimal Subspaces	861
2.3	Singular Value Decomposition	861
3	Projection-Based Model Order Reduction Methods	862
3.1	Projections on a Given Subspace	862
3.2	Construction of Subspaces	865
4	Low-Rank Approximation of Multivariate Functions	869
4.1	Tensor Ranks and Corresponding Low-Rank Formats	870
4.2	Relation with Other Structured Approximations	871
4.3	Properties of Low-Rank Formats	872
5	Low-Rank Approximation from Samples of the Function	873
5.1	Least Squares	873
5.2	Interpolation/Projection	874
6	Low-Rank Tensor Methods for Parameter-Dependent Equations	875
6.1	Tensor-Structured Equations	875
6.2	Iterative Solvers and Low-Rank Truncations	876
6.3	Optimization on Low-Rank Manifolds	877
7	Concluding Remarks	879
	References	879

1 Introduction

Parameter-dependent models arise in many contexts such as uncertainty quantification, sensitivity analysis, inverse problems, and optimization. These models are typically given under the form of a parameter-dependent equation:

$$R(u(\xi); \xi) = 0, \quad (25.1)$$

where ξ are parameters taking values in some set Ξ and where the solution $u(\xi)$ is in some vector space V , say \mathbb{R}^M . Parametric or uncertainty analyses usually require the evaluation of the solution for many instances of the parameters, which may be intractable for complex numerical models (with large M) for which one single solution requires hours or days of computation time. Therefore, one usually relies on approximations of the solution map $u : \Xi \rightarrow V$ allowing for a rapid evaluation of output quantities of interest. These approximations take on different names such as meta-model, surrogate model, or reduced order model. They are usually of the form

$$u_m(\xi) = \sum_{i=1}^m v_i s_i(\xi), \quad (25.2)$$

where the v_i are elements in V and the s_i are elements of some space S of functions defined on Ξ . Standard linear approximation methods rely on the introduction of generic bases (e.g., polynomials, wavelets, etc.) allowing an accurate approximation of a large class of models to be constructed but at the price of requiring expansions with a high number of terms m . These generic approaches usually result in a very high computational complexity.

Model order reduction methods aim at finding an approximation u_m with a small number of terms ($m \ll M$) that are adapted to the particular function u . One can distinguish approaches relying (i) on the construction of a reduced basis $\{v_1, \dots, v_m\}$ in V , (ii) on the construction of a reduced basis $\{s_1, \dots, s_m\}$ in S , or (iii) directly on the construction of an approximation under the form (25.2). These approaches are closely related. They all result in a *low-rank approximation* u_m , which can be interpreted as a rank- m element of the tensor space $V \otimes S$. Approaches of type (i) are usually named *projection-based model order reduction methods* since they define $u_m(\xi)$ as a certain projection of $u(\xi)$ onto a low-dimensional subspace of V . They include *reduced basis*, *proper orthogonal decomposition*, *Krylov subspace*, *balanced truncation methods*, and also subspace-based variants of *proper generalized decomposition* methods. Corresponding reduced order models usually take the form of a small system of equations which defines the projection $u_m(\xi)$ for each instance of ξ . Approaches (i) and (ii) are in some sense dual to each other. Approaches (ii) include *sparse approximation methods* which consist in selecting $\{s_1, \dots, s_m\}$ in a certain dictionary of functions (e.g., a polynomial basis), based on prior information on u or based on a posteriori error estimates (adaptive selection). They also include methods for the construction of reduced bases in S that exploit some prior information on u . *Low-rank tensor methods* enter the family of approaches (iii), where approximations of the form (25.2) directly result from an optimization on low-rank manifolds.

When one is interested not only in evaluating the solution $u(\xi)$ at a finite number of samples of ξ but in obtaining an explicit representation of the solution map $u : \Xi \rightarrow V$, approximations of functions of multiple parameters $\xi = (\xi_1, \dots, \xi_d)$ are required. This constitutes a challenging issue for high-dimensional problems. Naive approximation methods which consist in using tensorized bases yield an exponential increase in storage or computational complexity when the dimension d increases, which is the so-called curse of dimensionality. Specific structures of the functions have to be exploited in order to reduce the complexity. Standard structured approximations include additive approximations $u_1(\xi_1) + \dots + u_d(\xi_d)$, separated approximations $u_1(\xi_1) \dots u_d(\xi_d)$, sparse approximations, or rank-structured approximations. Rank-structured approximation methods include several types of approximation depending on the notion of rank.

The present chapter provides an overview of model order reduction methods based on low-rank tensor approximation methods. In a first section, we recall some basic notions about low-rank approximations of an order-two tensor $u \in V \otimes S$. In the second section, which is devoted to projection-based model reduction methods, we present different definitions of projections onto subspaces and different possible constructions of these subspaces. In the third section, we introduce the basic concepts of low-rank tensor methods for the approximation of a multivariate function, which is identified with a high-order tensor. The last two sections present different methods for the computation of low-rank approximations, either based on samples of the function (fourth section) or on the equations satisfied by the function (fifth section).

2 Low-Rank Approximation of Order-Two Tensors

Let us assume that $u : \mathcal{E} \rightarrow V$ with V a Hilbert space equipped with a norm $\|\cdot\|_V$ and inner product $(\cdot, \cdot)_V$. For the sake of simplicity, let us consider that $V = \mathbb{R}^M$. Let us further assume that u is in the Bochner space $L_\mu^p(\mathcal{E}; V)$ for some $p \geq 1$, where μ is a probability measure supported on \mathcal{E} . The space $L_\mu^p(\mathcal{E}; V)$ can be identified with the algebraic tensor space $V \otimes L_\mu^p(\mathcal{E})$ (or the completion of this space when V is infinite dimensional; see, e.g., [17]) which is the space of functions w that can be written under the form $w(\xi) = \sum_{i=1}^m v_i s_i(\xi)$ for some $v_i \in V$ and $s_i \in L_\mu^p(\mathcal{E})$ and some $m \in \mathbb{N}$. The rank of an element w , denoted $\text{rank}(w)$, is the minimal integer m such that w admits such an m -term representation. A rank- m approximation of u then takes the form

$$u_m(\xi) = \sum_{i=1}^m v_i s_i(\xi), \quad (25.3)$$

and can be interpreted as a certain projection of $u(\xi)$ onto an m -dimensional subspace V_m in V , where $\{v_1, \dots, v_m\}$ constitutes a basis of V_m .

2.1 Best Rank- m Approximation and Optimal Subspaces

The set of elements in $V \otimes L_\mu^p(\mathcal{E})$ with a rank bounded by m is denoted $\mathcal{R}_m = \{w \in V \otimes L_\mu^p(\mathcal{E}) : \text{rank}(w) \leq m\}$. The definition of a best approximation of u from \mathcal{R}_m requires the introduction of a measure of error. The best rank- m approximation with respect to the Bochner norm $\|\cdot\|_p$ in $L_\mu^p(\mathcal{E}; V)$ is the solution of

$$\min_{v \in \mathcal{R}_m} \|u - v\|_p = \min_{v \in \mathcal{R}_m} \|\|u(\xi) - v(\xi)\|_V\|_{L_\mu^p(\mathcal{E})} := d_m^{(p)}(\mathcal{E}). \quad (25.4)$$

The set \mathcal{R}_m admits the subspace-based parametrization $\mathcal{R}_m = \{w \in V_m \otimes L_\mu^p(\mathcal{E}) : V_m \subset V, \dim(V_m) = m\}$. Then the best rank- m approximation problem can be reformulated as an optimization problem over the set of m -dimensional spaces:

$$d_m^{(p)}(u) = \min_{\dim(V_m)=m} \min_{v \in V_m \otimes L_\mu^p} \|u - v\|_p = \min_{\dim(V_m)=m} \|u - P_{V_m} u\|_p, \quad (25.5)$$

where P_{V_m} is the orthogonal projection from V to V_m which provides the best approximation $P_{V_m} u(\xi)$ of $u(\xi)$ from V_m , defined by

$$\|u(\xi) - P_{V_m} u(\xi)\|_V = \min_{v \in V_m} \|u(\xi) - v\|_V. \quad (25.6)$$

That means that the best rank- m approximation problem is equivalent to the problem of finding an optimal subspace of dimension m for the projection of the solution.

2.2 Characterization of Optimal Subspaces

The numbers $d_m^{(p)}(u)$, which are the best rank- m approximation errors with respect to norms $\|\cdot\|_p$, are so-called linear widths of the *solution manifold*

$$\mathcal{K} := u(\Xi) = \{u(\xi) : \xi \in \Xi\},$$

and measure how well the set of solutions \mathcal{K} can be approximated by m -dimensional subspaces. They provide a quantification of the ideal performance of model order reduction methods.

For $p = \infty$, assuming that \mathcal{K} is compact, the number

$$d_m^{(\infty)}(u) = \min_{\dim(V_m)=m} \sup_{\xi \in \Xi} \|u(\xi) - P_{V_m} u(\xi)\|_V = \min_{\dim(V_m)=m} \sup_{v \in \mathcal{K}} \|v - P_{V_m} v\|_V \quad (25.7)$$

corresponds to the *Kolmogorov m -width* $d_m(\mathcal{K})_V$ of \mathcal{K} . This measure of error is particularly pertinent if one is interested in computing approximations that are uniformly accurate over the whole parameter set Ξ . For $p < \infty$, the numbers

$$d_m^{(p)}(u) = \min_{\dim(V_m)=m} \left(\int_{\Xi} \|u(\xi) - P_{V_m} u(\xi)\|_V^p \mu(d\xi) \right)^{1/p} \quad (25.8)$$

provide measures of the error that take into account the measure μ on the parameter set. This is particularly relevant for uncertainty quantification, where these numbers $d_m^{(p)}(u)$ directly control the error on the moments of the solution map u (such as mean, variance, or higher-order moments).

Some general results on the convergence of linear widths are available in approximation theory (see, e.g., [51]). More interesting results which are specific to some classes of parameter-dependent equations have been recently obtained [16, 38]. These results usually exploit smoothness and anisotropy of the solution map $u : \Xi \rightarrow V$ and are typically upper bounds deduced from results on polynomial approximation. Even if a priori estimates for $d_m^{(p)}(u)$ are usually not available, a challenging problem is to propose numerical methods that provide approximations u_m of the form (25.3) with an error $\|u - u_m\|_p$ of the order of the best achievable accuracy $d_m^{(p)}(u)$.

2.3 Singular Value Decomposition

Of particular importance is the case $p = 2$ where u is in the Hilbert space $L_\mu^2(\Xi; V) = V \otimes L_\mu^2(\Xi)$ and can be identified with a compact operator $U : v \in V \mapsto (u(\cdot), v)_V \in L_\mu^2(\Xi)$ which admits a *singular value decomposition*

$$U = \sum_{i \geq 1} \sigma_i v_i \otimes s_i,$$

where the σ_i are the singular values and where v_i and s_i are the corresponding normalized right and left singular vectors, respectively. Denoting by U^* the adjoint operator of U , defined by $U^* : s \in L^2_\mu(\Xi) \mapsto \int_\Xi u(\xi)s(\xi)d\mu(\xi) \in V$, the operator

$$U^*U := C(u) : v \in V \mapsto \int_\Xi u(\xi)(u(\xi), v)_V d\mu(\xi) \in V \quad (25.9)$$

is the correlation operator of u , with eigenvectors v_i and corresponding eigenvalues σ_i^2 . Assuming that singular values are sorted in decreasing order, $V_m = \text{span}\{v_1, \dots, v_m\}$ is a solution of (25.4) (which means an optimal subspace) and a best approximation of the form (25.2) is given by a rank- m truncated singular value decomposition $u_m(\xi) = \sum_{i=1}^m \sigma_i v_i s_i(\xi)$ which satisfies $\|u - u_m\|_2 = d_m^{(2)}(u) = (\sum_{i>m} \sigma_i^2)^{1/2}$.

3 Projection-Based Model Order Reduction Methods

Here we adopt a subspace point of view for the low-rank approximation of the solution map $u : \Xi \rightarrow V$. We first describe how to define projections onto a given subspace. Then we present methods for the practical construction of subspaces.

3.1 Projections on a Given Subspace

Here, we consider that a finite-dimensional subspace V_m is given to us. The best approximation of $u(\xi)$ from V_m is given by the projection $P_{V_m}u(\xi)$ defined by (25.6). However, in practice, projections of the solution $u(\xi)$ onto a given subspace V_m must be defined using computable information.

3.1.1 Interpolation

When the subspace V_m is the linear span of evaluations (samples) of the solution u at m given points $\{\xi^1, \dots, \xi^m\}$ in Ξ , i.e.,

$$V_m = \text{span}\{u(\xi^1), \dots, u(\xi^m)\}, \quad (25.10)$$

projections $u_m(\xi)$ of $u(\xi)$ onto V_m can be obtained by interpolation. An interpolation of u can be written in the form

$$u_m(\xi) = \sum_{i=1}^m u(\xi^i) s_i(\xi),$$

where functions s_i satisfy the interpolation conditions $s_i(\xi^j) = \delta_{ij}$, $1 \leq i, j \leq m$. The best approximation $P_{V_m} u(\xi)$ is a particular case of interpolation. However, its computation is not of practical interest since it requires the knowledge of $u(\xi)$. Standard polynomial interpolations can be used when Ξ is an interval in \mathbb{R} . In higher dimensions, polynomial interpolation can still be constructed for structured interpolation grids. For arbitrary sets of points, other interpolation formulae can be used, such as Kriging, nearest neighbor, Shepard, or radial basis interpolations. These standard interpolation formulae provide approximations $u_m(\xi)$ that only depend on the value of u at points $\{\xi^1, \dots, \xi^m\}$.

Generalized interpolation formulae that take into account the function over the whole parameter set can be defined by

$$u_m(\xi) = \sum_{i=1}^m u(\xi^i) \varphi_i(u(\xi)), \quad (25.11)$$

with $\{\varphi_i\}$ a dual system to $\{u(\xi^i)\}$ such that $\varphi_i(u(\xi^j)) = \delta_{ij}$ for $1 \leq i, j \leq m$. This yields an interpolation $u_m(\xi)$ depending not only on the value of u at the points $\{\xi^1, \dots, \xi^m\}$ but also on $u(\xi)$. This is of practical interest if $\varphi_i(u(\xi))$ can be efficiently computed without a complete knowledge of $u(\xi)$. For example, if the coefficients of $u(\xi)$ on some basis of V can be estimated without computing $u(\xi)$, then a possible choice consists in taking for $\{\varphi_1, \dots, \varphi_m\}$ a set of functions that associate to an element of V a set of m of its coefficients. This is the idea behind the *empirical interpolation method* [41] and its generalization [40].

3.1.2 Galerkin Projections

For models described by an equation of type (25.1), the most prominent methods are *Galerkin projections* which define the approximation $u_m(\xi)$ from the residual $R(u_m(\xi); \xi)$, e.g., by imposing orthogonality of the residual with respect to an m -dimensional space or by minimizing some residual norm. Galerkin projections do not provide the best approximation, but under some usual assumptions, they can provide quasi-best approximations $u_m(\xi)$ satisfying

$$\|u(\xi) - u_m(\xi)\|_V \leq c(\xi) \min_{v \in V_m} \|u(\xi) - v\|_V, \quad (25.12)$$

where $c(\xi) \geq 1$. As an example, let us consider the case of a linear problem where

$$R(v; \xi) = A(\xi)v - b(\xi), \quad (25.13)$$

with $A(\xi)$ a linear operator from V into some Hilbert space W , and let us consider a Galerkin projection defined by minimizing some residual norm, that means

$$\|A(\xi)u_m(\xi) - b(\xi)\|_W = \min_{v \in V_m} \|A(\xi)v - b(\xi)\|_W. \quad (25.14)$$

Assuming

$$\alpha(\xi)\|v\|_V \leq \|A(\xi)v\|_W \leq \beta(\xi)\|v\|_V, \quad (25.15)$$

the resulting projection $u_m(\xi)$ satisfies (25.12) with a constant $c(\xi) = \frac{\beta(\xi)}{\alpha(\xi)}$ which can be interpreted as the condition number of the operator $A(\xi)$. This reveals the interest of introducing efficient preconditioners in order to better exploit the approximation power of the subspace V_m (see, e.g., [13, 58] for the construction of parameter-dependent preconditioners). The resulting approximation can be written

$$u_m(\xi) = P_{V_m}^G(\xi)u(\xi),$$

where $P_{V_m}^G(\xi)$ is a parameter-dependent projection from V onto V_m . Note that if V_m is generated by samples of the solution, as in (25.10), then the Galerkin projection is also an interpolation which can be written under the form (25.11) with parameter-dependent functions φ_i that depend on the operator.

Some technical assumptions on the residual are required for a practical computation of Galerkin projections. More precisely, for $u_m(\xi) = \sum_{i=1}^m v_i s_i(\xi)$, the residual should admit a low-rank representation

$$R(u_m; \xi) = \sum_j R_j \gamma_j(s(\xi); \xi),$$

where the R_j are independent of $s(\xi) = (s_1(\xi), \dots, s_m(\xi))$ and where the γ_j can be computed with a complexity depending on m (the dimension of the reduced order model) but not on the dimension of V . This allows Galerkin projections to be computed by solving a reduced system of m equations on the unknown coefficients $s(\xi)$, with a computational complexity independent of the dimension of the full order model. For linear problems, such a property is obtained when the operator $A(\xi)$ and the right-hand side $b(\xi)$ admit low-rank representations (so-called affine representations)

$$A(\xi) = \sum_{i=1}^L A_i \alpha_i(\xi), \quad b(\xi) = \sum_{i=1}^R b_i \beta_i(\xi). \quad (25.16)$$

If $A(\xi)$ and $b(\xi)$ are not explicitly given under the form (25.16), then a preliminary approximation step is needed. Such expressions can be obtained by the empirical interpolation method [5, 11] or other low-rank truncation methods. Note that preconditioners for parameter-dependent operators should also have such representations in order to preserve a reduced order model with a complexity independent of the dimension of the full order model.

3.2 Construction of Subspaces

The computation of an optimal m -dimensional subspace V_m with respect to the natural norm in $L_\mu^p(\mathcal{E}; V)$, defined by (25.5), is not feasible in practice since it requires the knowledge of u . Practical constructions of subspaces must rely on computable information on u , which can be samples of the solution or the model equations (when available).

3.2.1 From Samples of the Function

Here, we present constructions of subspaces V_m which are based on evaluations of the function u at a set of points $\mathcal{E}_K = \{\xi^1, \dots, \xi^K\}$ in \mathcal{E} . Of course, V_m can be chosen as the span of evaluations of u at m points chosen independently of u , e.g., through random sampling. Here, we present methods for obtaining subspaces V_m that are closer to the optimal m -dimensional spaces, with $K \geq m$.

L^2 Optimality

When one is interested in optimal subspaces with respect to the norm $\|\cdot\|_2$, the definition (25.5) of optimal subspaces can be replaced by

$$\min_{\dim(V_m)=m} \frac{1}{K} \sum_{k=1}^K \|u(\xi^k) - P_{V_m} u(\xi^k)\|_V^2 = \min_{v \in \mathcal{R}_m} \frac{1}{K} \sum_{k=1}^K \|u(\xi^k) - v(\xi^k)\|_V^2, \quad (25.17)$$

where ξ^1, \dots, ξ^K are K independent random samples drawn according the probability measure μ . The resulting subspace V_m is the dominant eigenspace of the empirical correlation operator

$$C_K(u) = \frac{1}{K} \sum_{k=1}^K u(\xi^k)(u(\xi^k), \cdot)_V, \quad (25.18)$$

which is a statistical estimate of the correlation operator $C(u)$ defined by (25.9). This approach, which requires the computation of K evaluations of the solution u , corresponds to the classical *principal component analysis*. It is at the basis of proper orthogonal decomposition methods for parameter-dependent equations [33]. A straightforward generalization consists in defining the subspace V_m by

$$\min_{\dim(V_m)=m} \sum_{k=1}^K \omega_k \|u(\xi^k) - P_{V_m} u(\xi^k)\|_V^2 = \min_{v \in \mathcal{R}_m} \sum_{k=1}^K \omega_k \|u(\xi^k) - v(\xi^k)\|_V^2, \quad (25.19)$$

using a suitable quadrature rule for the integration over \mathcal{E} , e.g., exploiting the smoothness of the solution map $u : \mathcal{E} \rightarrow V$ in order to improvethe convergence

with K and therefore decrease the number of evaluations of u for a given accuracy. The resulting subspace V_m is obtained as the dominant eigenspace of the operator

$$C_K(u) = \sum_{k=1}^K \omega^k u(\xi^k)(u(\xi^k), \cdot)_V. \quad (25.20)$$

L^∞ Optimality

If one is interested in optimality with respect to the norm $\|\cdot\|_\infty$, the definition (25.5) of optimal spaces can be replaced by

$$\min_{\dim(V_m)=m} \sup_{\xi \in \Xi_K} \|u(\xi) - P_{V_m} u(\xi)\|_V = \min_{v \in \mathcal{R}_m} \sup_{\xi \in \Xi_K} \|u(\xi) - v(\xi)\|_V, \quad (25.21)$$

with $\Xi_K = \{\xi^1, \dots, \xi^K\}$ a set of K points in Ξ . A computationally tractable definition of subspaces can be obtained by adding the constraint that subspaces V_m are generated from m samples of the solution, that means $V_m = \text{span}\{u(\xi_*^1), \dots, u(\xi_*^m)\}$. Therefore, problem (25.21) becomes

$$\min_{\xi_*^1, \dots, \xi_*^m \in \Xi_K} \max_{\xi \in \Xi_K} \|u(\xi) - P_{V_m} u(\xi)\|_V,$$

where the m points are selected in the finite set of points Ξ_K . In practice, this combinatorial problem can be replaced by a *greedy algorithm*, which consists in selecting the points adaptively: given the first m points and the corresponding subspace V_m , a new interpolation point ξ_*^{m+1} is defined such that

$$\|u(\xi_*^{m+1}) - P_{V_m} u(\xi_*^{m+1})\|_V = \max_{\xi \in \Xi_K} \|u(\xi) - P_{V_m} u(\xi)\|_V. \quad (25.22)$$

This algorithm corresponds to the empirical interpolation method [5, 6, 41].

3.2.2 From Approximations of the Correlation Operator

An optimal m -dimensional space V_m for the approximation of u in $L_\mu^2(\Xi; V)$ is given by a dominant eigenspace of the correlation operator $C(u)$ of u . Approximations of optimal subspaces can then be obtained by computing dominant eigenspaces of an approximate correlation operator. These approximations can be obtained by using numerical integration in the definition of the correlation operator, yielding approximations (25.18) or (25.20) which require evaluations of u . Another approach consists in using the correlation operator $C(u_K)$ (or the singular value decomposition) of an approximation u_K of u which can be obtained by a projection of u onto a low-dimensional subspace $V \otimes S_K$ in $V \otimes L_\mu^2(\Xi)$. For example, an approximation can be sought after in the form $u_K(\xi) = \sum_{k=1}^K v_k \psi_k(\xi)$ where $\{\psi_k(\xi)\}_{k=1}^K$ is a

polynomial basis. Let us note that the statistical estimate $C_K(u)$ in (25.18) can be interpreted as the correlation operator $C(u_K)$ of a piecewise constant interpolation $u_K(\xi) = \sum_{k=1}^K u(\xi^k) 1_{\xi \in O_k}$ of $u(\xi)$, where the sets $\{O_1, \dots, O_K\}$ form a partition of Ξ such that $\xi^k \in O_k$ and $\mu(O_k) = 1/K$ for all k .

Remark 1. Let us mention that the optimal rank- m singular value decomposition of u can be equivalently obtained by computing the dominant eigenspace of the operator $\hat{C}(u) = UU^* : L_\mu^2(\Xi) \rightarrow L_\mu^2(\Xi)$. Then, a dual approach for model order reduction consists in defining a subspace S_m in $L_\mu^2(\Xi)$ as the dominant eigenspace of $\hat{C}(u_K)$, where u_K is an approximation of u . An approximation of u can then be obtained by a Galerkin projection onto the subspace $V \otimes S_m$ (see, e.g., [20] where an approximation u_K of the solution of a parameter-dependent partial differential equation is first computed using a coarse finite element approximation).

3.2.3 From the Model Equations

In the definition (25.5) of optimal spaces, $\|u(\xi) - v(\xi)\|$ can be replaced by a function $\Delta(v(\xi); \xi)$ which is computable without having $u(\xi)$ and such that $v \mapsto \Delta(v; \xi)$ has $u(\xi)$ as a minimizer over V . The choice for Δ is natural for problems where $u(\xi)$ is the minimizer of a functional $\Delta(\cdot; \xi) : V \rightarrow \mathbb{R}$. When $u(\xi)$ is solution of an equation of the form (25.1), a typical choice consists in taking for $\Delta(v(\xi); \xi)$ a certain norm of the residual $R(v(\xi); \xi)$.

L^2 Optimality (Proper Generalized Decomposition Methods)

When one is interested in optimality in the norm $\|\cdot\|_2$, an m -dimensional subspace V_m can be defined as the solution of the following optimization problem over the Grassmann manifold of subspaces of dimension m :

$$\min_{\dim(V_m)=m} \min_{v \in V_m \otimes L_\mu^2(\Xi)} \int_{\Xi} \Delta(v(\xi); \xi)^2 d\mu(\xi), \quad (25.23)$$

which can be equivalently written as an optimization problem over the set of m -dimensional bases in V :

$$\min_{v_1, \dots, v_m \in V} \min_{s_1, \dots, s_m \in L_\mu^2(\Xi)} \int_{\Xi} \Delta\left(\sum_{i=1}^m v_i s_i(\xi); \xi\right)^2 d\mu(\xi). \quad (25.24)$$

This problem is an optimization problem over the set \mathcal{R}_m of rank- m tensors which can be solved using optimization algorithms on low-rank manifolds such as an alternating minimization algorithm which consists in successively minimizing over the v_i and over the s_i . Assuming that $\Delta(\cdot; \xi)$ defines a distance to the solution $u(\xi)$ which is uniformly equivalent to the one induced by the norm $\|\cdot\|_V$, i.e.,

$$\alpha_\Delta \|u(\xi) - v\|_V \leq \Delta(v; \xi) \leq \beta_\Delta \|u(\xi) - v\|_V, \quad (25.25)$$

the resulting subspace V_m is quasi-optimal in the sense that

$$\|u - P_{V_m} u\|_2 \leq c_\Delta \min_{\dim(V_m)=m} \|u - P_{V_m} u\|_2 = c_\Delta d_m^{(2)}(u),$$

with $c_\Delta = \frac{\beta_\Delta}{\alpha_\Delta}$. For linear problems where $R(v; \xi) = A(\xi)v - b(\xi) \in W$ and $\Delta(v(\xi); \xi) = \|R(v(\xi); \xi)\|_W$, (25.25) results from the property (25.15) of the operator $A(\xi)$. A suboptimal but constructive variant of algorithm (25.24) is defined by

$$\min_{v_m \in V} \min_{s_1, \dots, s_m \in L^2_\mu(\Xi)} \int_{\Xi} \Delta\left(\sum_{i=1}^m v_i s_i(\xi; \xi)^2 d\mu(\xi)\right), \quad (25.26)$$

which is a greedy construction of the reduced basis $\{v_1, \dots, v_m\}$. It yields a nested sequence of subspaces V_m . This is one of the variants of proper generalized decomposition methods (see [44–46]).

Note that in practice, for solving (25.24) or (25.26) when Ξ is not a finite set, one has either to approximate functions s_i in a finite-dimensional subspace of $L^2_\mu(\Xi)$ [44, 45] or to approximate the integral over Ξ by using a suitable quadrature [26].

L^∞ Optimality (Reduced Basis Methods)

When one is interested in optimality in the norm $\|\cdot\|_\infty$, a subspace V_m could be defined by

$$\min_{\dim(V_m)=m} \max_{\xi \in \Xi} \Delta(u_m(\xi); \xi),$$

where $u_m(\xi)$ is some projection of $u(\xi)$ onto V_m (typically a Galerkin projection). A modification of the above definition consists in searching for spaces V_m that are generated from evaluations of the solution at m points selected in a subset Ξ_K of K points in Ξ (a training set). In practice, this combinatorial optimization problem can be replaced by a greedy algorithm for the selection of points: given a set of interpolation points $\{\xi_*^1, \dots, \xi_*^m\}$ and an approximation $u_m(\xi)$ in $V_m = \text{span}\{u(\xi_*^1), \dots, u(\xi_*^m)\}$, a new point ξ_*^{m+1} is selected such that

$$\Delta(u_m(\xi_*^{m+1}); \xi_*^{m+1}) = \max_{\xi \in \Xi_K} \Delta(u_m(\xi); \xi). \quad (25.27)$$

This results in an adaptive interpolation algorithm which was first introduced in [52]. It is the basic idea behind the so-called reduced basis methods (see, e.g., [50, 53]).

Assuming that Δ satisfies (25.25) and that the projection $u_m(\xi)$ verifies the quasi-optimality condition (25.12), the selection of interpolation points defined by (25.27) is quasi-optimal in the sense that

$$\|u(\xi_*^{m+1}) - P_{V_m} u(\xi_*^{m+1})\|_V \geq \gamma \max_{\xi \in \Xi_K} \|u(\xi) - P_{V_m} u(\xi)\|_V,$$

with $\gamma = c_{\Delta}^{-1} \inf_{\xi \in \Xi} c(\xi)^{-1}$. That makes (25.27) a suboptimal version of the greedy algorithm (25.22) (so-called weak greedy algorithm). Convergence results for this algorithm can be found in [8, 9, 18], where the authors provide explicit comparisons between the resulting error $\|u(\xi) - u_m(\xi)\|_V$ and the Kolmogorov m -width of $u(\Xi_K)$ which is the best achievable error by projections onto m -dimensional spaces.

The above definitions of interpolation points, and therefore of the resulting subspaces V_m , do not take into account explicitly the probability measure μ . However, this measure is taken into account implicitly when working with a sample set Ξ_K drawn according the probability measure μ . A construction that takes into account the measure explicitly has been proposed in [12], where $\Delta(v; \xi)$ is replaced by $\omega(\xi)\Delta(v; \xi)$, with $\omega(\xi)$ a weight function depending on the probability measure μ .

4 Low-Rank Approximation of Multivariate Functions

For the approximation of high-dimensional functions $u(\xi_1, \dots, \xi_d)$, a standard approximation tool consists in searching for an expansion on a multidimensional basis obtained by tensorizing univariate bases:

$$u(\xi_1, \dots, \xi_d) \approx \sum_{v_1=1}^{n_1} \dots \sum_{v_d=1}^{n_d} a_{v_1, \dots, v_d} \phi_{v_1}^1(\xi_1) \dots \phi_{v_d}^d(\xi_d). \quad (25.28)$$

This results in an exponential growth of storage and computational complexities with the dimension d . Low-rank tensor methods aim at reducing the complexity by exploiting high-order low-rank structures of multivariate functions, considered as elements of tensor product spaces. This section presents basic notions about low-rank approximations of high-order tensors. The reader is referred to the textbook [29] and the surveys [28, 34, 36] for further details on the subject. For simplicity, we consider the case of a real-valued function $u : \Xi \rightarrow \mathbb{R}$. The presentation naturally extends to the case $u : \Xi \rightarrow V$ by the addition of a new dimension.

Here, we assume that $\Xi = \Xi_1 \times \dots \times \Xi_d$ and that μ is a product measure $\mu_1 \otimes \dots \otimes \mu_d$. Let S_v denote a space of univariate functions defined on Ξ_v , $1 \leq v \leq d$. The elementary tensor product of functions $v^v \in S_v$ is defined by $(v^1 \otimes \dots \otimes v^d)(\xi_1, \dots, \xi_d) = v^1(\xi_1) \dots v^d(\xi_d)$. The algebraic tensor space $S = S_1 \otimes \dots \otimes S_d$ is defined as the set of elements that can be written as a finite sum of elementary tensors, which means

$$v(\xi_1, \dots, \xi_d) = \sum_{i=1}^r a_i v_i^1(\xi_1) \dots v_i^d(\xi_d), \quad (25.29)$$

for some $v_i^v \in S_v$, $a_i \in \mathbb{R}$, and $r \in \mathbb{N}$. For the sake of simplicity, we consider that S_v is a finite-dimensional approximation space in $L_{\mu_v}^2(\Xi_v)$ (e.g., a space of polynomials, wavelets, splines, etc.), with $\dim(S_v) = n_v \leq n$, so that S is a

subspace of the algebraic tensor space $L^2_{\mu_1}(\mathcal{E}_1) \otimes \dots \otimes L^2_{\mu_d}(\mathcal{E}_d)$ (whose completion is $L^2_\mu(\mathcal{E})$).

4.1 Tensor Ranks and Corresponding Low-Rank Formats

The *canonical rank* of a tensor v in S is the minimal integer m such that v can be written under the form (25.29). An approximation of the form (25.29) is called an approximation in *canonical tensor format*. It has a storage complexity in $O(rnd)$. For order-two tensors, this is the standard and unique notion of rank. For higher-order tensors, other notions of rank can be introduced, therefore yielding different types of rank-structured approximations. First, for a certain subset of dimensions $\alpha \subset D := \{1, \dots, d\}$ and its complementary subset $\alpha^c = D \setminus \alpha$, S can be identified with the space $S_\alpha \otimes S_{\alpha^c}$ of order-two tensors, where $S_\alpha = \bigotimes_{v \in \alpha} S_v$. The α -rank of a tensor v , denoted $\text{rank}_\alpha(v)$, is then defined as the minimal integer r_α such that

$$v(\xi_1, \dots, \xi_d) = \sum_{i=1}^{r_\alpha} v_i^\alpha(\xi_\alpha) v_i^{\alpha^c}(\xi_{\alpha^c}),$$

where $v_i^\alpha \in S_\alpha$ and $v_i^{\alpha^c} \in S_{\alpha^c}$ (here ξ_α denotes the collection of variables $\{\xi_v : v \in \alpha\}$), which is the standard and unique notion of rank for order-two tensors. Low-rank Tucker formats are then defined by imposing the α -rank for a collection of subsets $\alpha \subset D$. The *Tucker rank* (or *multilinear rank*) of a tensor is the tuple $(\text{rank}_{\{1\}}(v), \dots, \text{rank}_{\{d\}}(v))$ in \mathbb{N}^d . A tensor with Tucker rank bounded by (r_1, \dots, r_d) can be written as

$$v(\xi_1, \dots, \xi_d) = \sum_{i_1=1}^{r_1} \dots \sum_{i_d=1}^{r_d} a_{i_1 \dots i_d} v_{i_1}^1(\xi_1) \dots v_{i_d}^d(\xi_d), \quad (25.30)$$

where $v_{i_v}^v \in S_v$ and where $a \in \mathbb{R}^{r_1 \times \dots \times r_d}$ is a tensor of order d , called the *core tensor*. An approximation of the form (25.30) is called an approximation in *Tucker format*. It can be seen as an approximation in a tensor space $U_1 \otimes \dots \otimes U_d$, where $U_v = \text{span}\{v_{i_v}^v\}_{i_v=1}^{r_v}$ is a r_v -dimensional subspace of S_v . The storage complexity of this format is in $O(rnd + r^d)$ (with $r = \max_v r_v$) and grows exponentially with the dimension d . Additional constraints on the ranks of v (or of the core tensor a) have to be imposed in order to reduce this complexity. *Tree-based (or hierarchical) Tucker formats* [25, 30] are based on a notion of rank associated with a dimension partition tree T which is a tree-structured collection of subsets α in D , with D as the root of the tree and the singletons $\{1\}, \dots, \{d\}$ as the leaves of the tree (see Fig. 25.1). The *tree-based (or hierarchical) Tucker rank* associated with T is then defined as the tuple $(\text{rank}_\alpha(v))_{\alpha \in T}$. A particular case of interest is the *tensor train (TT) format* [47] which is associated with a simple rooted tree T with interior nodes $I = \{\{v, \dots, d\}, 1 \leq v \leq d\}$ (represented on Fig. 25.1b). The *TT-rank* of a

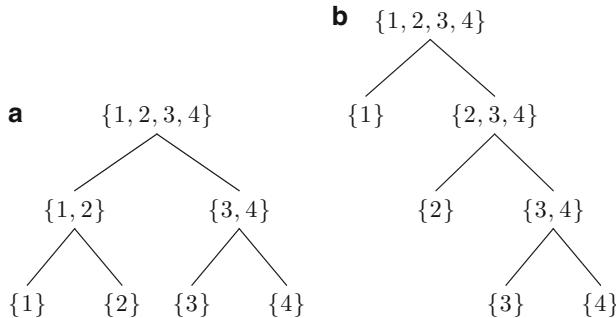


Fig. 25.1 Examples of dimension partition trees over $D = \{1, \dots, 4\}$. (a) Balanced tree. (b) Unbalanced tree

tensor v is defined as the tuple of ranks $(\text{rank}_{\{v+1, \dots, d\}}(v))_{1 \leq v \leq d-1}$. A tensor v with TT-rank bounded by (r_1, \dots, r_{d-1}) can be written under the form

$$v(\xi_1, \dots, \xi_d) = \sum_{i_1=1}^{r_1} \dots \sum_{i_{d-1}=1}^{r_{d-1}} v_{1,i_1}^1(\xi_1) v_{1,i_2}^2(\xi_2) \dots v_{i_{d-1},1}^d(\xi_d), \quad (25.31)$$

with $v_{i_{v-1}, i_v}^v \in S_v$, which is very convenient in practice (for storage, evaluations, and algebraic manipulations). The storage complexity for the TT format is in $O(dr^2n)$ (with $r = \max_v r_v$).

4.2 Relation with Other Structured Approximations

Sparse tensor methods consist in searching for approximations of the form (25.28) with only a few nonzero terms that means approximations $v(\xi) = \sum_{i=1}^m a_i s_i(\xi)$ where the m functions s_i are selected (with either adaptive or nonadaptive methods) in the collection (*dictionary*) of functions $\mathcal{D} = \{\phi_{k_1}^1(\xi_1) \dots \phi_{k_d}^d(\xi_d) : 1 \leq k_v \leq n_v, 1 \leq v \leq d\}$. A typical choice consists in taking for \mathcal{D} a basis of multivariate polynomials. Recently, theoretical results have been obtained on the convergence of sparse polynomial approximations of the solution of parameter-dependent equations (see [15]). Also, algorithms have been proposed that can achieve convergence rates comparable to the best m -term approximations. For such a dictionary \mathcal{D} containing rank-one functions, a sparse m -term approximation is a tensor with canonical rank bounded by m (usually lower than m , see [29, Section 7.6.5]). Therefore, an approximation in canonical tensor format (25.29) can be seen as a sparse m -term approximation where the m functions are selected in the dictionary of all rank-one (separated) functions $\mathcal{R}_1 = \{s(\xi) = s^1(\xi_1) \dots s^d(\xi_d) : s^v \in S^v\}$. Convergence results for best rank- m approximations can therefore be deduced from convergence results for sparse tensor approximation methods.

Let us mention some other standard structured approximations that are particular cases of low-rank tensor approximations. First, a function $v(\xi) = v(\xi_\nu)$ depending on a single variable ξ_ν is a rank-one (elementary) tensor. It has an α -rank equal to 1 for any subset α in D . A low-dimensional function $v(\xi) = v(\xi_\alpha)$ depending on a subset of variables ξ_α , $\alpha \subset D$, has a β -rank equal to 1 for any subset of dimensions β containing α or such that $\beta \cap \alpha = \emptyset$. An additive function $v(\xi) = v_1(\xi_1) + \dots + v_d(\xi_d)$, which is a sum of d elementary tensors, is a tensor with canonical rank d . Also, such an additive function has $\text{rank}_\alpha(v) \leq 2$ for any subset $\alpha \in D$, which means that it admits an exact representation in any hierarchical Tucker format (including TT format) with a rank bounded by $(2, \dots, 2)$.

Remark 2. Let us note that low-rank structures (as well as other types of structures) can be revealed only after a suitable change of variables. For example, let $\eta = (\eta_1, \dots, \eta_m)$ be the variables obtained by an affine transformation of variables ξ , with $\eta_i = \sum_{j=1}^d a_{ij} \xi_j + b_j$, $1 \leq j \leq m$. Then the function $v(\xi) = \sum_{i=1}^m v_i(\eta_i) := \hat{v}(\eta)$, as a function m variables, can be seen as an order- m tensor with canonical rank less than m . This type of approximation corresponds to the projection pursuit regression model.

4.3 Properties of Low-Rank Formats

Low-rank tensor approximation methods consist in searching for approximations in a subset of tensors

$$\mathcal{M}_{\leq r} = \{v : \text{rank}(v) \leq r\},$$

where different notions of rank yield different approximation formats. A first important question is to characterize the approximation power of low-rank formats, which means to quantify the best approximation error:

$$\inf_{v \in \mathcal{M}_{\leq r}} \|u - v\| := \sigma_r$$

for a given class of functions u and a given low-rank format. A few convergence results have been obtained for functions with standard Sobolev regularity (see, e.g., [55]). An open and challenging problem is to characterize approximation classes of the different low-rank formats, which means the class of functions u such that σ_r has a certain (e.g., algebraic or exponential) decay with r .

The characterization of topological and geometrical properties of subsets $\mathcal{M}_{\leq r}$ is important for different purposes such as proving the existence of best approximations in $\mathcal{M}_{\leq r}$ or deriving algorithms. For $d \geq 3$, the subsets $\mathcal{M}_{\leq r}$ associated with the notion of canonical rank are not closed and best approximation problems in $\mathcal{M}_{\leq r}$ are ill-posed. Subsets of tensors associated with the notions of tree-based (hierarchical) Tucker ranks have better properties. Indeed, they are closed sets,

which ensures the existence of best approximations. Also, they are differentiable manifolds [24, 32, 57]. This has useful consequences for optimization [57] or for the projection of dynamical systems on these manifolds [39].

For the different notions of rank introduced above, an interesting property of the corresponding low-rank subsets is that they admit simple parametrizations:

$$\mathcal{M}_{\leq r} = \{v = F(p_1, \dots, p_\ell) : p_k \in P_k\}, \quad (25.32)$$

where F is a multilinear map and P_k are vector spaces and where the number of parameters ℓ is in $O(d)$. An optimization problem on $\mathcal{M}_{\leq r}$ can then be reformulated as an optimization problem on $P_1 \times \dots \times P_\ell$, for which simple alternating minimization algorithms (block coordinate descent) can be used [23].

5 Low-Rank Approximation from Samples of the Function

Here we present some strategies for the construction of low-rank approximations of a multivariate function $u(\xi)$ from point evaluations of the function.

5.1 Least Squares

Let us assume that $u \in L^2_\mu(\mathcal{E})$. Given a set of K samples ξ^1, \dots, ξ^K of ξ drawn according the probability measure μ , a least-squares approximation of u in a low-rank subset $\mathcal{M}_{\leq r}$ is defined by

$$\min_{v \in \mathcal{M}_{\leq r}} \frac{1}{K} \sum_{k=1}^K (u(\xi^k) - v(\xi^k))^2. \quad (25.33)$$

Using a multilinear parametrization of low-rank tensor subsets (see (25.32)), the optimization problem (25.33) can be solved using alternating minimization algorithms, each iteration corresponding to a standard least-squares minimization for linear approximation [7, 14, 21]. An open question concerns the analysis of the number of samples which is required for a stable approximation in a given low-rank format. Also, standard regularizations can be introduced, such as sparsity-inducing regularizations [14]. In this statistical framework, cross-validation methods can be used for the selection of tensor formats, ranks, and approximation spaces S_k (see [14]).

Remark 3. Note that for other objectives in statistical learning (e.g., classification), (25.33) can be replaced by

$$\min_{v \in \mathcal{M}_{\leq r}} \frac{1}{K} \sum_{k=1}^K \ell(u(\xi^k), v(\xi^k)),$$

where ℓ is a so-called *loss function* measuring a certain distance between $u(\xi^k)$ and the approximation $v(\xi^k)$.

5.2 Interpolation/Projection

Here we present interpolation and projection methods for the approximation of u in $S = S_1 \otimes \dots \otimes S_d$. Let $\{\phi_{k_v}^v\}_{k_v \in \Lambda_{n_v}}$ be a basis of S_v , with $\Lambda_{n_v} = \{1, \dots, n_v\}$. If $\{\phi_{k_v}^v\}_{k_v \in \Lambda_{n_v}}$ is a set of interpolation functions associated with a set of points $\{\xi_v^{k_v}\}_{k_v \in \Lambda_{n_v}}$ in Ξ_v , then $\{\phi_k(\xi) = \phi_{k_1}^1(\xi_1) \dots \phi_{k_d}^d(\xi_d)\}_{k \in \Lambda}$, $\Lambda = \Lambda_{n_1} \times \dots \times \Lambda_{n_d}$, is a set of interpolation functions associated with the tensorized grid $\{\xi^k = (\xi_1^{k_1}, \dots, \xi_d^{k_d})\}_{k \in \Lambda}$ composed of $N = \prod_{v=1}^d n_v$ points. An interpolation u_N of u is then given by

$$u_N(\xi) = \sum_{k \in \Lambda} u(\xi^k) \phi_k(\xi),$$

so that u_N is completely characterized by the order- d tensor $U \in \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$ whose components $U_{k_1, \dots, k_d} = u(\xi_1^{k_1}, \dots, \xi_d^{k_d})$ are the evaluations of u on the interpolation grid. Now, if $\{\phi_{k_v}^v\}_{k_v \in \Lambda_{n_v}}$ is an orthonormal basis of S_v (e.g., orthonormal polynomials) and if $\{(\xi_v^{k_v}, \omega_{k_v}^v)\}_{k_v \in \Lambda_{n_v}}$ is a quadrature rule on Ξ_v (associated with the measure μ_v), an approximate L^2 -projection of u can also be defined by

$$u_N(\xi) = \sum_{k \in \Lambda} u_k \phi_k(\xi), \quad u_k = \sum_{k \in \Lambda} \omega_k u(\xi^k) \phi_k(\xi^k),$$

with $\omega_k = \omega_{k_1}^1 \dots \omega_{k_d}^d$. Here again, u_N is completely characterized by the order- d tensor U whose components are the evaluations of u on the d -dimensional quadrature grid.

Then, low-rank approximation methods can be used in order to obtain an approximation of U using only a few entries of the tensor (i.e., a few evaluations of the function u). This is related to the problem of tensor completion. A possible approach consists in evaluating some entries of the tensor taken at random and then in reconstructing the tensor by the minimization of a least-squares functional (this is the algebraic version of the least-squares approach described in the previous section) or dual approaches using regularizations of rank minimization problems (see [54]). In this statistical framework, a challenging question is to determine the number of samples required for a stable reconstruction of low-rank approximations in different tensor formats (see [54] for first results). An algorithm has been introduced in [22] for the approximation in canonical format, using least-squares minimization with a structured set of entries selected adaptively. Algorithms have also been proposed for an adaptive construction of low-rank approximations of U in tensor train format [49] or hierarchical Tucker format [4]. These algorithms are extensions of adaptive

cross approximation (ACA) to high-order tensors and provide approximations that interpolate the tensor U at some adaptively chosen entries.

6 Low-Rank Tensor Methods for Parameter-Dependent Equations

Here, we present numerical methods for the direct computation of low-rank approximations of the solution u of a parameter-dependent equation $R(u(\xi); \xi) = 0$, where u is seen as a two-order tensor in $V \otimes L^2_\mu(\Xi)$ or as a higher-order tensor by exploiting an additional tensor space structure of $L^2_\mu(\Xi)$ (for a product measure μ).

Remark 4. When exploiting only the order-two tensor structure, the methods presented here are closely related to projection-based model reduction methods. Although they provide a directly exploitable low-rank approximation of u , they can also be used for the construction of a low-dimensional subspace in V (a candidate for projection-based model reduction) which is extracted from the obtained low-rank approximation.

6.1 Tensor-Structured Equations

Here, we describe how the initial equation can be reformulated as a tensor-structured equation. In practice, a preliminary discretization of functions defined on Ξ is required. A possible discretization consists in introducing an N -dimensional approximation space S in $L^2_\mu(\Xi)$ (e.g., a polynomial space) and a standard Galerkin projection of the solution onto $V \otimes S$ (see, e.g., [42, 46]). The resulting approximation can then be identified with a two-order tensor \mathbf{u} in $V \otimes \mathbb{R}^N$. When S is the tensor product of n_v -dimensional spaces S_v , $1 \leq v \leq d$, the resulting approximation can be identified with a higher-order tensor \mathbf{u} in $V \otimes \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$. Another simple discretization consists in considering only a finite (possibly large) set of N points in Ξ (e.g., an interpolation grid) and the corresponding finite set of equations $R(u(\xi^k); \xi^k) = 0$, $1 \leq k \leq N$, on the set of samples of the solution $(u(\xi^k))_{k=1}^N \in V^N$, which can be identified with a tensor $\mathbf{u} \in V \otimes \mathbb{R}^N$. If the set of points is obtained by the tensorization of unidimensional grids with n_v points, $1 \leq v \leq d$, then $(u(\xi^k))_{k=1}^N$ can be identified with a higher-order tensor \mathbf{u} in $V \otimes \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$. Both types of discretization yield an equation:

$$\mathbf{R}(\mathbf{u}) = 0, \tag{25.34}$$

with

$$\mathbf{u} \text{ in } V \otimes \mathbb{R}^N \text{ or } V \otimes \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}.$$

In order to clarify the structure of Eq. (25.34), let us consider the case of a linear problem where $R(v; \xi) = A(\xi)v - b(\xi)$ and assume that $A(\xi)$ and $b(\xi)$ have low-rank (or affine) representations of the form

$$A(\xi) = \sum_{i=1}^L A_i \alpha_i(\xi) \quad \text{and} \quad b(\xi) = \sum_{i=1}^R b_i \beta_i(\xi). \quad (25.35)$$

Then (25.34) takes the form of a tensor-structured equation $\mathbf{Au} - \mathbf{b} = 0$, with

$$\mathbf{A} = \sum_{i=1}^L A_i \otimes \tilde{A}_i \quad \text{and} \quad \mathbf{b} = \sum_{i=1}^R b_i \otimes \tilde{b}_i, \quad (25.36)$$

where, for the second type of discretization, $\tilde{A}_i \in \mathbb{R}^{N \times N}$ is a diagonal matrix whose diagonal is the vector of evaluations of $\alpha_i(\xi)$ at the sample points and $\tilde{b}_i \in \mathbb{R}^N$ is the vector of evaluations of $\beta_i(\xi)$ at the sample points. If $A(\xi)$ and $b(\xi)$ have higher-order low-rank representations of the form

$$A(\xi) = \sum_{i=1}^L A_i \alpha_i^1(\xi_1) \dots \alpha_i^d(\xi_d) \quad \text{and} \quad b(\xi) = \sum_{i=1}^R b_i \beta_i^1(\xi_1) \dots \beta_i^d(\xi_d), \quad (25.37)$$

then (25.34) takes the form of a tensor-structured equation $\mathbf{Au} - \mathbf{b} = 0$ on \mathbf{u} in $V \otimes \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$, with

$$\mathbf{A} = \sum_{i=1}^L A_i \otimes \tilde{A}_i^1 \otimes \dots \otimes \tilde{A}_i^d \quad \text{and} \quad \mathbf{b} = \sum_{i=1}^R b_i \otimes \tilde{b}_i^1 \otimes \dots \otimes \tilde{b}_i^d, \quad (25.38)$$

where, for the second type of discretization (with a tensorized grid in $\mathcal{E}_1 \times \dots \times \mathcal{E}_d$), $\tilde{A}_i^\nu \in \mathbb{R}^{n_\nu \times n_\nu}$ is a diagonal matrix whose diagonal is the vector of evaluations of $\alpha_i^\nu(\xi_\nu)$ on the unidimensional grid in \mathcal{E}_ν and $\tilde{b}_i^\nu \in \mathbb{R}^{n_\nu}$ is the vector of evaluations of $\beta_i^\nu(\xi_\nu)$ on this grid (for the first type of discretization, see [46] for the definition of tensors \mathbf{A} and \mathbf{b}). Note that when $A(\xi)$ and $b(\xi)$ do not have low-rank representations (25.35) or (25.37) (or any other higher-order low-rank representation), then a preliminary approximation step is required in order to obtain such approximate representations (see, e.g., [5, 11, 19]). This is crucial for reducing the computational and storage complexities.

6.2 Iterative Solvers and Low-Rank Truncations

A first solution strategy consists in using standard iterative solvers (e.g., Richardson, conjugate gradient, Newton, etc.) with efficient low-rank truncation methods

of the iterates [1–3, 35, 37, 43]. A simple iterative algorithm takes the form $\mathbf{u}^{k+1} = M(\mathbf{u}^k)$, where M is an iteration map involving simple algebraic operations between tensors (additions, multiplications) which requires the implementation of a tensor algebra. Low-rank truncation methods can be systematically used for limiting the storage complexity and the computational complexity of algebraic operations. This results in approximate iterations $\mathbf{u}^{k+1} \approx M(\mathbf{u}^k)$, and the resulting algorithm can be analyzed as an inexact version (or perturbation) of the initial algorithm (see, e.g., [31]).

As an example, let us consider a linear tensor-structured problem $\mathbf{A}\mathbf{u} - \mathbf{b} = 0$. An approximate Richardson algorithm takes the form

$$\mathbf{u}^{k+1} = \Pi_\epsilon(\mathbf{u}^k + \alpha(\mathbf{b} - \mathbf{A}\mathbf{u}^k)),$$

where Π_ϵ is a map which associates to a tensor \mathbf{w} a low-rank approximation $\Pi_\epsilon(\mathbf{w})$ such that $\|\mathbf{w} - \Pi_\epsilon(\mathbf{w})\|_p \leq \epsilon \|\mathbf{w}\|_p$, with $p = 2$ or $p = \infty$ depending on the desired control of the error (mean-square or uniform error control over the parameter set). Provided some standard assumptions on the operator \mathbf{A} and the parameter α , the generated sequence \mathbf{u}^k is such that $\limsup_{k \rightarrow \infty} \|\mathbf{u} - \mathbf{u}^k\|_p \leq C(\epsilon)$ with $C(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0$. For $p = 2$, efficient low-rank truncations of a tensor can be obtained using SVD for an order-two tensor or generalizations of SVD for higher-order tensor formats [27, 48]. A selection of the ranks based on the singular values of (matricizations of) the tensor allows a control of the error. In [2], the authors propose an alternative truncation strategy based on soft thresholding. For $p = \infty$, truncations can be obtained with an *adaptive cross approximation* algorithm (or empirical interpolation method) [6] for an order-two tensor or with extensions of this algorithm for higher-order tensors [49]. Note that low-rank representations of the form (25.36) or (25.38) for \mathbf{A} and \mathbf{b} are crucial since they ensure that algebraic operations between tensors can be done with a reduced complexity. Also, iterative methods usually require good preconditioners. In order to maintain a low computational complexity, these preconditioners must also admit low-rank representations.

6.3 Optimization on Low-Rank Manifolds

Another solution strategy consists in directly computing a low-rank approximation by minimizing some functional \mathcal{J} whose minimizer on $V \otimes \mathbb{R}^N$ (or $V \otimes \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$) is the solution of Eq. (25.34), i.e., by solving

$$\min_{\mathbf{v} \in \mathcal{M}_{\leq r}} \mathcal{J}(\mathbf{v}), \quad (25.39)$$

where $\mathcal{M}_{\leq r}$ is a low-rank manifold. There is a natural choice of functional for problems where (25.34) corresponds to the stationary condition of a functional

\mathcal{J} [26]. Also, $\mathcal{J}(\mathbf{v})$ can be taken as a certain norm of the residual $\mathbf{R}(\mathbf{v})$. For linear problems, choosing

$$\mathcal{J}(\mathbf{v}) = \|\mathbf{Av} - \mathbf{b}\|^2$$

yields a quadratic optimization problem over a low-rank manifold. Optimization problems on low-rank manifolds can be solved either by using algorithms which exploit the manifold structure (e.g., Riemannian optimization) or by using simple alternating minimization algorithms given a parametrization (25.32) of the low-rank manifold. Under the assumption that \mathcal{J} satisfies $\alpha\|\mathbf{u} - \mathbf{v}\|_2 \leq \mathcal{J}(\mathbf{v}) \leq \beta\|\mathbf{u} - \mathbf{v}\|_2$, the solution \mathbf{u}_r of (25.39) is quasi-optimal in the sense that

$$\|\mathbf{u} - \mathbf{u}_r\|_2 \leq \frac{\beta}{\alpha} \min_{\mathbf{v} \in \mathcal{M}_{\leq r}} \|\mathbf{u} - \mathbf{v}\|_2,$$

where β/α is the condition number of the operator A . Here again, the use of preconditioners allows us to better exploit the approximation power of a given low-rank manifold $\mathcal{M}_{\leq r}$.

Constructive algorithms are also possible, the most prominent algorithm being a greedy algorithm which consists in computing a sequence of low-rank approximations u_m obtained by successive rank-one corrections, i.e.,

$$\mathcal{J}(\mathbf{u}_m) = \min_{w \in \mathcal{R}_1} \mathcal{J}(\mathbf{u}_{m-1} + w).$$

This algorithm is a standard greedy algorithm [56] with a dictionary of rank-one (elementary) tensors \mathcal{R}_1 , which was first introduced in [44] for the solution of parameter-dependent (stochastic) equations. Its convergence has been established under standard assumptions for convex optimization problems [10, 24]. The utility of this algorithm is that it is adaptive, and it only requires the solution of optimization problems on the low-dimensional manifold \mathcal{R}_1 . However, for many practical problems, greedy constructions of low-rank approximations in canonical low-rank format are observed to converge slowly. Improved constructive algorithms which better exploit the tensor structure of the problem have been proposed [46] and convergence results are also available for some general convex optimization problems [24].

Remark 5. The above algorithms can also be used within iterative algorithms for which an iteration takes the form

$$\mathbf{C}_k \mathbf{u}^{k+1} = \mathbf{F}_k(\mathbf{u}^k),$$

where $\mathbf{F}_k(\mathbf{u}^k)$ can be computed with a low complexity using low-rank tensor algebra (with potential low-rank truncations), but where the inverse of the operator \mathbf{C}_k is not known explicitly, so that $\mathbf{C}_k^{-1} \mathbf{F}_k(\mathbf{u}^k)$ cannot be obtained from simple algebraic

operations. Here, a low-rank approximation of \mathbf{u}^{k+1} can be computed using the above algorithms with the residual-based functional $\mathcal{J}(\mathbf{v}) = \|\mathbf{C}_k \mathbf{v} - \mathbf{F}_k(\mathbf{u}^k)\|^2$.

7 Concluding Remarks

Low-rank tensor methods have emerged as a very powerful tool for the solution of high-dimensional problems arising in many contexts and in particular in uncertainty quantification. However, there remain many challenging issues to address for a better understanding of this type of approximation and for a diffusion of these methods in a wide class of applications. From a theoretical point of view, open questions include the characterization of the approximation classes of a given low-rank tensor format, which means the class of functions for which a certain type of convergence (e.g., algebraic or exponential) can be expected, and also the characterization of the problems yielding solutions in these approximation classes. Also, quantitative results on the approximation of a low-rank function (or tensor) from samples of this function (or entries of this tensor) could allow us to answer some practical issues such as the determination of the number of samples required for a stable approximation in a given low-rank format or the design of sampling strategies which are adapted to particular low-rank formats. From a numerical point of view, challenging issues include the development of efficient algorithms for global optimization on low-rank manifolds, with guaranteed convergence properties, and the development of adaptive algorithms for the construction of controlled low-rank approximations, with an adaptive selection of ranks and potentially of the tensor formats (e.g., based on tree optimization for tree-based formats). From a practical point of view, low-rank tensor methods exploiting model equations (Galerkin-type methods) are often seen as “intrusive methods” in the sense that they require (a priori) the development of specific softwares. An important issue is then to develop weakly intrusive implementations of these methods which may allow the use of existing computational frameworks and which would therefore contribute to a large diffusion of these methods.

References

1. Bachmayr, M., Dahmen, W.: Adaptive near-optimal rank tensor approximation for high-dimensional operator equations. *Found. Comput. Math.* **15**(4), 839–898 (2015)
2. Bachmayr, M., Schneider, R.: Iterative Methods Based on Soft Thresholding of Hierarchical Tensors (Jan 2015). ArXiv e-prints 1501.07714
3. Ballani, J., Grasedyck, L.: A projection method to solve linear systems in tensor format. *Numer. Linear Algebra Appl.* **20**(1), 27–43 (2013)
4. Ballani, J., Grasedyck, L., Kluge, M.: Black box approximation of tensors in hierarchical tucker format. *Linear Algebra Appl.* **438**(2), 639–657 (2013). Tensors and Multilinear Algebra
5. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math.* **339**(9), 667–672 (2002)

6. Bebendorf, M., Maday, Y., Stamm, B.: Comparison of some reduced representation approximations. In: Quarteroni, A., Rozza, G. (eds.) Reduced Order Methods for Modeling and Computational Reduction. Volume 9 of MS&A – Modeling, Simulation and Applications, pp. 67–100. Springer International Publishing, Cham (2014)
7. Beylkin, G., Garcke, B., Mohlenkamp, M.J.: Multivariate regression and machine learning with sums of separable functions. *J. Comput. Phys.* **230**, 2345–2367 (2011)
8. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P.: Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.* **43**(3), 1457–1472 (2011)
9. Buffa, A., Maday, Y., Patera, A.T., Prud’Homme, C., Turinici, G.: A priori convergence of the Greedy algorithm for the parametrized reduced basis method. *ESAIM: Math. Model. Numer. Anal.* **46**(3), 595–603 (2012). Special volume in honor of Professor David Gottlieb
10. Cancès, E., Ehrlacher, V., Lelièvre, T.: Convergence of a greedy algorithm for high-dimensional convex nonlinear problems. *Math. Models Methods Appl. Sci.* **21**(12), 2433–2467 (2011)
11. Casenave, F., Ern, A., Lelièvre, T.: A nonintrusive reduced basis method applied to aeroacoustic simulations. *Adv. Comput. Math.* **41**(5), 961–986 (2015)
12. Chen, P., Quarteroni, A., Rozza, G.: A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **51**(6), 3163–3185 (2013)
13. Chen, Y., Gottlieb, S., Maday, Y.: Parametric analytical preconditioning and its applications to the reduced collocation methods. *C. R. Math.* **352**(7/8), 661–666 (2014). ArXiv e-prints
14. Chevreuil, M., Lebrun, R., Nouy, A., Rai, P.: A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA J. Uncertain. Quantif.* **3**(1), 897–921 (2015)
15. Cohen, A., DeVore, R.: Approximation of high-dimensional parametric PDEs. *Acta Numer.* **24**, 1–159 (2015)
16. Cohen, A., DeVore, R.: Kolmogorov widths under holomorphic mappings. *IMA J. Numer. Anal.* (2015)
17. Defant, A., Floret, K.: Tensor Norms and Operator Ideals. North-Holland, Amsterdam/New York (1993)
18. DeVore, R., Petrova, G., Wojtaszczyk, P.: Greedy algorithms for reduced bases in banach spaces. *Constr. Approx.* **37**(3), 455–466 (2013)
19. Dolgov, S., Khoromskij, B.N., Litvinenko, A., Matthies, H. G.: Polynomial chaos expansion of random coefficients and the solution of stochastic partial differential equations in the tensor train format. *SIAM/ASA J. Uncertain. Quantif.* **3**(1), 1109–1135 (2015)
20. Doostan, A., Ghanem, R., Red-Horse, J.: Stochastic model reductions for chaos representations. *Comput. Methods Appl. Mech. Eng.* **196**(37–40), 3951–3966 (2007)
21. Doostan, A., Validi, A., Iaccarino, G.: Non-intrusive low-rank separated approximation of high-dimensional stochastic models. *Comput. Methods Appl. Mech. Eng.* **263**(0), 42–55 (2013)
22. Espig, M., Grasedyck, L., Hackbusch, W.: Black box low tensor-rank approximation using fiber-crosses. *Constr. Approx.* **30**, 557–597 (2009)
23. Espig, M., Hackbusch, W., Khachatryan, A.: On the convergence of alternating least squares optimisation in tensor format representations (May 2015). ArXiv e-prints 1506.00062
24. Falcó, A., Nouy, A.: Proper generalized decomposition for nonlinear convex problems in tensor banach spaces. *Numerische Mathematik* **121**, 503–530 (2012)
25. Falcó, A., Hackbusch, W., Nouy, A.: Geometric structures in tensor representations. *Found. Comput. Math.* (Submitted)
26. Giraldi, L., Liu, D., Matthies, H.G., Nouy, A.: To be or not to be intrusive? The solution of parametric and stochastic equations—proper generalized decomposition. *SIAM J. Sci. Comput.* **37**(1), A347–A368 (2015)
27. Grasedyck, L.: Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.* **31**, 2029–2054 (2010)

28. Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* **36**(1), 53–78 (2013)
29. Hackbusch, W.: *Tensor Spaces and Numerical Tensor Calculus*. Volume 42 of Springer Series in Computational Mathematics. Springer, Heidelberg (2012)
30. Hackbusch, W., Kuhn, S.: A new scheme for the tensor representation. *J. Fourier Anal. Appl.* **15**(5), 706–722 (2009)
31. Hackbusch, W., Khoromskij, B., Tyrtyshnikov, E.: Approximate iterations for structured matrices. *Numerische Mathematik* **109**, 365–383 (2008). 10.1007/s00211-008-0143-0.
32. Holtz, S., Rohwedder, T., Schneider, R.: On manifolds of tensors of fixed tt-rank. *Numerische Mathematik* **120**(4), 701–731 (2012)
33. Kahlbacher, M., Volkwein, S.: Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discuss. Math.: Differ. Incl. Control Optim.* **27**, 95–117 (2007)
34. Khoromskij, B.: Tensors-structured numerical methods in scientific computing: survey on recent advances. *Chemom. Intell. Lab. Syst.* **110**(1), 1–19 (2012)
35. Khoromskij, B.B., Schwab, C.: Tensor-structured Galerkin approximation of parametric and stochastic elliptic pdes. *SIAM J. Sci. Comput.* **33**(1), 364–385 (2011)
36. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009)
37. Kressner, D., Tobler, C.: Low-rank tensor krylov subspace methods for parametrized linear systems. *SIAM J. Matrix Anal. Appl.* **32**(4), 1288–1316 (2011)
38. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: Generalized reduced basis methods and n-width estimates for the approximation of the solution manifold of parametric pdes. In: Brezzi, F., Colli Franzone, P., Gianazza, U., Gilardi, G. (eds.) *Analysis and Numerics of Partial Differential Equations*. Volume 4 of Springer INdAM Series, pp. 307–329. Springer, Milan (2013)
39. Lubich, C., Rohwedder, T., Schneider, R., Vandereycken, B.: Dynamical approximation by hierarchical tucker and tensor-train tensors. *SIAM J. Matrix Anal. Appl.* **34**(2), 470–494 (2013)
40. Maday, Y., Mula, O.: A generalized empirical interpolation method: application of reduced basis techniques to data assimilation. In: Brezzi, F., Colli Franzone, P., Gianazza, U., Gilardi, G. (eds.) *Analysis and Numerics of Partial Differential Equations*. Volume 4 of Springer INdAM Series, pP. 221–235. Springer, Milan (2013)
41. Maday, Y., Nguyen, N.C., Patera, A.T., Pau, G.S.H.: A general multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.* **8**(1), 383–404 (2009)
42. Matthies, H.G., Keese, A.: Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **194**(12–16), 1295–1331 (2005)
43. Matthies, H.G., Zander, E.: Solving stochastic systems with low-rank tensor compression. *Linear Algebra Appl.* **436**(10), 3819–3838 (2012)
44. Nouy, A.: A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Comput. Methods Appl. Mech. Eng.* **196**(45–48), 4521–4537 (2007)
45. Nouy, A.: Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms. *Comput. Methods Appl. Mech. Eng.* **197**, 4718–4736 (2008)
46. Nouy, A.: Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Eng.* **17**(4), 403–434 (2010)
47. Oseledets, I.: Tensor-train decomposition. *SIAM J. Sci. Comput.* **33**(5), 2295–2317 (2011)
48. Oseledets, I., Tyrtyshnikov, E.: Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM J. Sci. Comput.* **31**(5), 3744–3759 (2009)
49. Oseledets, I., Tyrtyshnikov, E.: TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* **432**(1), 70–88 (2010)

-
- 50. Patera, A.T., Rozza, G.: Reduced Basis Approximation and A-Posteriori Error Estimation for Parametrized PDEs. MIT-Pappalardo Graduate Monographs in Mechanical Engineering. Massachusetts Institute of Technology, Cambridge (2007)
 - 51. Pietsch, A.: Eigenvalues and s-Numbers. Cambridge University Press, Cambridge/New York (1987)
 - 52. Prud'homme, C., Rovas, D., Veroy, K., Maday, Y., Patera, A.T., Turinici, G.: Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods. *J. Fluids Eng.* **124**(1), 70–80 (2002)
 - 53. Quarteroni, A., Rozza, G., Manzoni, A.: Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.* **1**(1), 1–49 (2011)
 - 54. Rauhut, H., Schneider, R., Stojanac, Z.: Tensor completion in hierarchical tensor representations (Apr 2014). ArXiv e-prints
 - 55. Schneider, R., Uschmajew, A.: Approximation rates for the hierarchical tensor format in periodic sobolev spaces. *J. Complex.* **30**(2), 56–71 (2014) Dagstuhl 2012
 - 56. Temlyakov, V.: Greedy approximation in convex optimization (June 2012). ArXiv e-prints
 - 57. Uschmajew, A., Vandereycken, B.: The geometry of algorithms using hierarchical tensors. *Linear Algebra Appl.* **439**(1), 133–166 (2013)
 - 58. Zahm, O., Nouy, A.: Interpolation of inverse operators for preconditioning parameter-dependent equations (April 2015). ArXiv e-prints

Random Vectors and Random Fields in High Dimension: Parametric Model-Based Representation, Identification from Data, and Inverse Problems

Christian Soize

Abstract

The statistical inverse problem for the experimental identification of a non-Gaussian matrix-valued random field, that is, the model parameter of a boundary value problem, using some partial and limited experimental data related to a model observation, is a very difficult and challenging problem. A complete advanced methodology and the associated tools are presented for solving such a problem in the following framework: the random field that must be identified is a non-Gaussian matrix-valued random field and is not simply a real-valued random field; this non-Gaussian random field is in high stochastic dimension and is identified in a general class of random fields; some fundamental algebraic properties of this non-Gaussian random field must be satisfied such as symmetry, positiveness, invertibility in mean square, boundedness, symmetry class, spatial-correlation lengths, etc.; and the available experimental data sets correspond only to partial and limited data for a model observation of the boundary value problem.

The developments presented are mainly related to the elasticity framework, but the methodology is general and can be used in many areas of computational sciences and engineering. The developments are organized as follows. The first part is devoted to the definition of the statistical inverse problem that has to be solved in high stochastic dimension and is focussed on stochastic elliptic operators such that the ones that are encountered in the boundary value problems of the linear elasticity. The second one deals with the construction of two possible parameterized representations for a non-Gaussian positive-definite matrix-valued random field that models the model parameter of a boundary value problem. A parametric model-based representation is then constructed in introducing a statistical reduced model and a polynomial chaos expansion, first with deterministic coefficients and after with random coefficients. This

C. Soize (✉)

Laboratoire Modélisation et Simulation Multi Echelle (MSME), Université Paris-Est,

Marne-la-Vallée, France

e-mail: christian.soize@univ-paris-est.fr

parametric model-based representation is directly used for solving the statistical inverse problem. The third part is devoted to the description of all the steps of the methodology allowing the statistical inverse problem to be solved in high stochastic dimension. These steps are based on the identification of a prior stochastic model of the Non-Gaussian random field by using the maximum likelihood method and then, on the identification of a posterior stochastic model of the Non-Gaussian random field by using the Bayes method. The fourth part presents the construction of an algebraic prior stochastic model of the model parameter of the boundary value problem, for a non-Gaussian matrix-valued random field. The generator of realizations for such an algebraic prior stochastic model for a non-Gaussian matrix-valued random field is presented.

Keywords

Random vector • Random field • Random matrix • High dimension • High stochastic dimension • Non-Gaussian • Non-Gaussian random field • Representation of random fields • Polynomial chaos expansion • Generator • Maximum entropy principle • Prior model • Maximum likelihood method • Bayesian method • Identification • Inverse problem • Statistical inverse problem • Random media • Heterogeneous microstructure • Composite materials • Porous media

Contents

1	Introduction	886
2	Notions on the High Stochastic Dimension and on the Parametric Model-Based Representations for Random Fields	886
2.1	What Is a Random Vector or a Random Field with a High Stochastic Dimension?	886
2.2	What Is a Parametric Model-Based Representation for the Statistical Identification of a Random Model Parameter from Experimental Data?	887
3	Brief History	888
3.1	Classical Methods for Statistical Inverse Problems	888
3.2	Case of a Gaussian Random Model Parameter	889
3.3	Case for Which the Model Parameter Is a Non-Gaussian Second-Order Random Field	889
3.4	Finite-Dimension Approximation of the BVP and Finite-Dimension Parameterization of the Random Field	890
3.5	Parameterization of the Non-Gaussian Second-Order Random Vector η	890
3.6	Statistical Inverse Problem for Identifying a Non-Gaussian Random Field as a Model Parameter of a BVP, Using Polynomial Chaos Expansion	891
3.7	Algebraic Prior Stochastic Models of the Model Parameters of BVP	892
4	Overview	893
5	Notations	894
5.1	Euclidean Space	894
5.2	Sets of Matrices	894
5.3	Kronecker Symbol, Unit Matrix, and Indicator Function	894
5.4	Norms and Usual Operators	894

5.5	Order Relation in the Set of All the Positive-Definite Real Matrices	895
5.6	Probability Space, Mathematical Expectation, and Space of Second-Order Random Vectors	895
6	Setting the Statistical Inverse Problem to be Solved in High Stochastic Dimension	895
6.1	Stochastic Elliptic Operator and Boundary Value Problem	895
6.2	Stochastic Finite Element Approximation of the Stochastic Boundary Value Problem	897
6.3	Experimental Data Sets	898
6.4	Statistical Inverse Problem to be Solved	898
7	Parametric Model-Based Representation for the Model Parameters and Model Observations	899
7.1	Introduction a Class of Lower-Bounded Random Fields for $[K]$ and Normalization	899
7.2	Construction of the Nonlinear Transformation \mathcal{G}	900
7.3	Truncated Reduced Representation of Second-Order Random Field $[G]$ and Its Polynomial Chaos Expansion	902
7.4	Parameterization of Compact Stiefel Manifold $V_m(\mathbb{R}^N)$	904
7.5	Parameterized Representation for Non-Gaussian Random Field $[K]$	904
7.6	Parametric Model-Based Representation of Random Observation Model U	905
8	Methodology for Solving the Statistical Inverse Problem in High Stochastic Dimension	905
8.1	Step 1: Introduction of a Family $\{[K^{APSM}(x; s)], x \in \Omega\}$ of Algebraic Prior Stochastic Models (APSM) for Non-Gaussian Random Field $[K]$	905
8.2	Step 2: Identification of an Optimal Algebraic Prior Stochastic Model (OAPSM) for Non-Gaussian Random Field $[K]$	906
8.3	Step 3: Choice of an Adapted Representation for Non-Gaussian Random Field $[K]$ and Optimal Algebraic Prior Stochastic Model for Non-Gaussian Random Field $[G]$	907
8.4	Step 4: Construction of a Truncated Reduced Representation of Second-Order Random Field $[G^{OAPSM}]$	907
8.5	Step 5: Construction of a Truncated Polynomial Chaos Expansion of η^{OAPSM} and Representation of Random Field $[K^{OAPSM}]$	908
8.6	Step 6: Identification of the Prior Stochastic Model $[K^{prior}]$ of $[K]$ in the General Class of the Non-Gaussian Random Fields	912
8.7	Step 7: Identification of a Posterior Stochastic Model $[K^{post}]$ of $[K]$	913
9	Construction of a Family of Algebraic Prior Stochastic Models	914
9.1	General Properties of the Non-Gaussian Random Field $[K]$ with a Lower Bound	915
9.2	Algebraic Prior Stochastic Model for the Case of Anisotropic Statistical Fluctuations	915
9.3	Algebraic Prior Stochastic Model for the Case of Dominant Statistical Fluctuations in a Symmetry Class with Some Anisotropic Statistical Fluctuations	919
10	Key Research Findings and Applications	931
10.1	Additional Ingredients for Statistical Reduced Models, Symmetry Properties, and Generators for High Stochastic Dimension	931
10.2	Tensor-Valued Random Fields and Continuum Mechanics of Heterogenous Materials	931
11	Conclusions	931
	References	932

1 Introduction

The statistical inverse problem for the experimental identification of a non-Gaussian matrix-valued random field that is the model parameter of a boundary value problem, using some partial and limited experimental data related to a model observation, is a very difficult and challenging problem. The classical methodologies that are very efficient for Gaussian random fields cannot be used for non-Gaussian matrix-valued random fields in high stochastic dimension, in particular under the assumption that only partial and limited experimental data are available for the statistical inverse problem that has to be solved for identifying the non-Gaussian random field through a boundary value problem. This means that experimental data must be enriched in introducing adapted informative prior stochastic models for the non-Gaussian matrix-valued random fields in order to take into account fundamental algebraic properties such as symmetry, positiveness, invertibility in mean square, boundedness, symmetry class, spatial-correlation lengths, etc. The objective is then to present a complete advanced methodology and the associated tools for solving such a statistical inverse problem in high stochastic dimension and related to non-Gaussian matrix-valued random fields.

2 Notions on the High Stochastic Dimension and on the Parametric Model-Based Representations for Random Fields

2.1 What Is a Random Vector or a Random Field with a High Stochastic Dimension?

The *stochastic dimension* of a random vector or a random field is an important notion that allows for evaluating the level of complexity of a statistical inverse problem related to the identification of a random model parameter (random vector, random field) of a stochastic boundary value problem (for instance, the coefficients of a partial differential equation) using experimental data related to a random model observation (random variable, random vector, random field) of this boundary value problem.

Let us consider a random vector \mathbf{U} with values in \mathbb{R}^{N_U} in which N_U is an integer. The stochastic dimension of \mathbf{U} is not, in general, the value of integer N_U . For instance, if \mathbf{U} is written as $\mathbf{U} = \eta \mathbf{b}$, in which η is a real-valued random variable and where \mathbf{b} is a deterministic vector given in \mathbb{R}^{N_U} , then the stochastic dimension of \mathbf{U} is 1 for any value of integer N_U . If \mathbf{U} is written as $\mathbf{U} = \sum_{i=1}^m \eta_i \mathbf{b}^i$ with $m \leq N_U$, in which η_1, \dots, η_m are m independent real-valued random variables and where $\mathbf{b}^1, \dots, \mathbf{b}^m$ are m algebraically independent vectors given in \mathbb{R}^{N_U} , then the stochastic dimension of \mathbf{U} is m , and \mathbf{U} is in high stochastic dimension if m is large. If \mathbf{U} is a second-order random vector whose covariance matrix is known, then the use of the principal component analysis allows the reduced representation $\mathbf{U}^{(m)} = \sum_{i=1}^m \eta_i \sqrt{\lambda_i} \mathbf{b}^i$ of \mathbf{U} to be constructed with $m < N_U$ and where m is calculated

in order that the mean-square error of $\mathbf{U} - \mathbf{U}^{(m)}$ is sufficiently small. It can thus be written $\mathbf{U} \sim \mathbf{U}^{(m)}$ (in mean square). In such a reduced representation, $\lambda_1 \geq \dots > \lambda_m > 0$ are the dominant eigenvalues of the covariance matrix of \mathbf{U} and $\mathbf{b}^1, \dots, \mathbf{b}^m$ are the associated orthonormal eigenvectors in \mathbb{R}^{N_U} . The components η_1, \dots, η_m are m centered and uncorrelated real-valued random variables. If random vector \mathbf{U} is a Gaussian random vector, then η_1, \dots, η_m are m independent Gaussian real-valued random variables, and, for this particular Gaussian case, the stochastic dimension of \mathbf{U} is m . However, for the general case, \mathbf{U} is a non-Gaussian random vector, and consequently, the real-valued random variables η_1, \dots, η_m (that are centered and uncorrelated) are not independent but are statistically dependent. In such a case, m is not the stochastic dimension of \mathbf{U} , but clearly the stochastic dimension is less or equal to m (the equality is obtained for the Gaussian case). Let us assume that there exists a deterministic nonlinear mapping \mathcal{Y} from \mathbb{R}^{N_g} into \mathbb{R}^m such that the random vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_m)$ can be written as $\boldsymbol{\eta} = \mathcal{Y}(\mathcal{E}_1, \dots, \mathcal{E}_{N_g})$ in which $N_g < m$ and where $\mathcal{E}_1, \dots, \mathcal{E}_{N_g}$ are N_g independent real-valued random variables (for instance, \mathcal{Y} can be constructed using the polynomial chaos expansion of the second-order random vector $\boldsymbol{\eta}$). In such a case, the stochastic dimension of \mathbf{U} is less or equal to N_g . If among all the possible nonlinear mappings and all the possible integers N_g such that $1 \leq N_g < m$, the mapping \mathcal{Y} and the integer N_g correspond to the smallest possible value of N_g such as $\boldsymbol{\eta} = \mathcal{Y}(\mathcal{E}_1, \dots, \mathcal{E}_{N_g})$, then N_g is the stochastic dimension of \mathbf{U} , and \mathbf{U} has a high stochastic dimension if N_g is large.

If $\{\mathbf{u}(\mathbf{x}), \mathbf{x} \in \Omega\}$ is a second-order random field indexed by $\Omega \subset \mathbb{R}^d$ with values in \mathbb{R}^{N_u} , for which its cross covariance function is square integrable $\Omega \times \Omega$, then a reduced representation $\mathbf{u}^{(m)}(\mathbf{x}) = \sum_{i=1}^m \eta_i \sqrt{\lambda_i} \mathbf{b}^i(\mathbf{x})$ of \mathbf{u} can be constructed using the Karhunen-Loève expansion of \mathbf{u} , in which m is calculated in order that the mean-square error of $\mathbf{u} - \mathbf{u}^{(m)}$ is sufficiently small. Therefore, the explanations given before can be applied to the random vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_m)$ in order to estimate the stochastic dimension of random field \mathbf{u} .

2.2 What Is a Parametric Model-Based Representation for the Statistical Identification of a Random Model Parameter from Experimental Data?

In order to simply explain what is a parametric model-based representation for the statistical identification of a random model parameter from experimental data, let us consider the stochastic elliptic boundary value problem formulated for a real-valued random field $u(\mathbf{x})$ indexed by $\mathbf{x} = (x_1, \dots, x_d)$ belonging to a subset Ω of \mathbb{R}^d and which is assumed to have a unique second-order stochastic solution u . The stochastic elliptic operator of the boundary value problem is written as $-\sum_{j=1}^d \frac{\partial}{\partial x_j} \{K(\mathbf{x}) \frac{\partial}{\partial x_j} u(\mathbf{x})\}$ in which the random field $K = \{K(\mathbf{x}), \mathbf{x} \in \Omega\}$, indexed by Ω , with values in $\mathbb{R}^+ = [0, +\infty[$, is defined as the *model parameter* of the boundary value problem. Let \mathbf{U} be a random *model observation* that is assumed to be a random vector with values in \mathbb{R}^{N_U} , which is deduced from random field u by a

deterministic observation operator \mathcal{O} , such that $\mathbf{U} = \mathcal{O}(u)$. Consequently, random model observation \mathbf{U} can be written as $\mathbf{U} = \mathcal{H}(K)$ in which \mathcal{H} is a deterministic nonlinear functional of K . For all \mathbf{x} in Ω , a representation of K is assumed to be written as $K(\mathbf{x}) = \mathcal{G}(G(\mathbf{x}))$ with $G(\mathbf{x}) = G_0(\mathbf{x}) + \sum_{i=1}^m \eta_i \sqrt{\lambda_i} G_i(\mathbf{x})$. The deterministic nonlinear mapping \mathcal{G} is independent of \mathbf{x} and is assumed to be from \mathbb{R} into \mathbb{R}^+ . With the introduction of such a deterministic mapping \mathcal{G} , for all \mathbf{x} fixed in Ω , the support of the probability distribution of the random variable $G(\mathbf{x})$ is \mathbb{R} instead of \mathbb{R}^+ for $K(\mathbf{x})$. In the reduced representation of the random field G indexed by Ω , with values in \mathbb{R} , the quantities $G_0(\mathbf{x})$, λ_i , and $G_i(\mathbf{x})$ are some real numbers. The random vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_m)$ is written as $\boldsymbol{\eta} = \mathcal{Y}(\boldsymbol{\Xi}; [z])$ in which $\boldsymbol{\Xi} = (\Xi_1, \dots, \Xi_{N_g})$ is a given vector-valued random variable, where \mathcal{Y} is a deterministic nonlinear mapping representing the truncated polynomial chaos expansion of $\boldsymbol{\eta}$ with respect to $\boldsymbol{\Xi}$ and where $[z]$ is the real matrix of the \mathbb{R}^m -valued coefficients of the truncated polynomial chaos expansion of $\boldsymbol{\eta}$. It can then be deduced that random model observation \mathbf{U} can be rewritten as $\mathbf{U} = \mathcal{B}(\boldsymbol{\Xi}, [z])$ in which \mathcal{B} is a deterministic nonlinear mapping depending on \mathcal{H} , \mathcal{G} , and \mathcal{Y} . This last representation is defined as a *parametric model-based representation* of the random model observation \mathbf{U} in which the real matrix $[z]$ is the hyperparameter of the representation. Let us assume that some experimental data $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},n_{\text{exp}}}$ related to random model observation \mathbf{U} are available. The *identification of the model parameter* K using the *experimental data* consists in identifying the real matrix $[z]$ using the parametric model-based representation $\mathbf{U} = \mathcal{B}(\boldsymbol{\Xi}, [z])$ of the random model observation and the corresponding experimental data.

3 Brief History

3.1 Classical Methods for Statistical Inverse Problems

The problem related to the identification of a model parameter (scalar, vector, field) of a boundary value problem (BVP) (for instance, the coefficients of a partial differential equation) using experimental data related to a model observation (scalar, vector, field) of this BVP is a problem for which there exists a rich literature, including numerous textbooks. In general and in the deterministic context, there is not a unique solution because the function, which maps the model parameter (that belongs to an admissible set) to the model observation (that belongs to another admissible set) is not a one-to-one mapping and, consequently, cannot be inverted. It is an ill-posed problem. However, such a problem can be reformulated in terms of an optimization problem consisting in calculating an optimal value of the model parameter, which minimizes a certain distance between the observed experimental data and the model observation that is computed with the BVP and that depends on the model parameter (see, for instance, [76] for an overview concerning the general methodologies and [36] for some mathematical aspects related to the inverse problems for partial differential equations). In many cases, the analysis of such an inverse problem can have a unique solution in the framework of statistics,

that is to say when the model parameter is modeled by a random quantity, with or without external noise on the model observation (observed output). In such a case, the random model observation is completely defined by its probability distribution (in finite or in infinite dimension) that is the unique transformation of the probability distribution of the random model parameter. This transformation is defined by the functional that maps the model parameter to the model observation. Such a formulation is constructed for obtaining a well-posed problem that has a unique solution in the probability theory framework. We refer the reader to [38] and [72] for an overview concerning the general methodologies for statistical and computational inverse problems, including general least-square inversion and the maximum likelihood method [54, 67] and including the Bayesian approach [8, 9, 67, 68].

3.2 Case of a Gaussian Random Model Parameter

A Gaussian second-order random vector is completely defined by its second-order moments, that is to say, by its mean vector and by its covariance matrix. Similarly, a Gaussian second-order random field is completely defined by its mean function and by its cross covariance function or, if the random field is homogeneous (stationary) and mean-square continuous, by its spectral measure [17]. If the model parameter is Gaussian (random vector or random field), then the statistical inverse problem (identification of the system parameter using experimental data related to the model observation of the system) consists in identifying the second-order moments, which is relatively easy for a low or a high stochastic dimension. Concerning the description of the Gaussian random fields, we refer the reader to the abundant existing literature (see, for instance, [41, 53, 74]).

3.3 Case for Which the Model Parameter Is a Non-Gaussian Second-Order Random Field

A non-Gaussian second-order random field is completely defined by its system of marginal probability distributions, which is an uncountable family of probability distributions on sets of finite dimension and not only by its mean function and its covariance function as for a Gaussian random field. The experimental identification of such a non-Gaussian random field then requires the introduction of an adapted representation in order to be in capability to solve the statistical inverse problem. For any non-Gaussian second-order random field, an important type of representation is based on the use of the polynomial chaos expansion [7], for which the development and the use in computational sciences and engineering have been pioneered by Roger Ghanem in 1990–1991 [23]. An efficient construction is proposed, which consists in combining a Karhunen-Loëve expansion (that allows using a statistical reduced model) with a polynomial chaos expansion of the statistical reduced model. This type of construction has then been reanalyzed and used for solving boundary value problems using the spectral approach (see, for instance, [13, 18, 22, 24, 25, 42,

[46, 47, 52]). The polynomial chaos expansion has also been extended for an arbitrary probability measure [20, 43, 44, 57, 77, 78] and for sparse representation [5]. New algorithms have been proposed for obtaining a robust computation of realizations of high degrees of polynomial chaos [49, 65]. This type of representation has also been extended for the case of the polynomial chaos expansion with random coefficients [66], for the construction of a basis adaptation in homogeneous chaos spaces [73], and for an arbitrary multimodal multidimensional probability distribution [64].

3.4 Finite-Dimension Approximation of the BVP and Finite-Dimension Parameterization of the Random Field

A finite-dimension parameterized representation of the non-Gaussian random field must be constructed in order to be able to solve the statistical inverse problem. In addition and in general, an explicit solution of the BVP cannot be obtained and consequently, a finite-dimension approximation of the solution of the BVP must also be constructed (using, for instance, the finite element method), accompanied by a convergence analysis. The combination of these two approximations leads us to introduce a non-Gaussian second-order random vector η with values in \mathbb{R}^m , which is the finite-dimension parameterized representation of the random model parameter of the system. Consequently, the statistical inverse problem consists in identifying the non-Gaussian second-order random vector η that is completely defined by its probability distribution on \mathbb{R}^m . Nevertheless, as η corresponds to a finite-dimension parameterization of the finite discretization of a random field, it is necessary to construct, first, a good mathematical representation of the random field and of its finite-dimension parameterization, before performing its spatial discretization.

3.5 Parameterization of the Non-Gaussian Second-Order Random Vector η

Since it is assumed that the experimental data that are available for the statistical inverse problem are partial and limited, the parametric statistics must be used instead of the nonparametric statistics that cannot be used. This implies that a parameterized representation of the non-Gaussian second-order random vector η must be constructed. There are two main methods for constructing such a parameterization.

- (i) The first one is a direct approach that consists in constructing a algebraic prior representation of the non-Gaussian probability distribution of η in using the maximum entropy principle (MaxEnt) [37, 63] under the constraints defined by the available information. A general computational methodology, for the problems in high stochastic dimension, is proposed in [4, 59] and is synthesized in Sect. 11 of ▶ Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification. Such a construction allows a low-dimension

hyperparameterization to be obtained for the non-Gaussian probability distribution on \mathbb{R}^m . Therefore, the parametric statistics [54, 67, 76] can be used for solving the statistical inverse problem consisting in identifying the vector-valued hyperparameter of the probability distribution constructed with the MaxEnt. In counterpart, the “distance” between the observed experimental data and the random model observation cannot be, in general, reduced to zero. A residual error exists. If there are a sufficient amount of experimental data, this error can be reduced by identifying a posterior probability distribution of η using the Bayesian approach [8, 9, 67].

- (ii) The second method is an indirect approach which consists in introducing a representation $\eta = \mathcal{Y}(\Xi)$ in which \mathcal{Y} is an unknown deterministic nonlinear (measurable) mapping from \mathbb{R}^{N_g} into \mathbb{R}^m (which has to be constructed) and where Ξ is a given random vector with values in \mathbb{R}^{N_g} , for which its probability distribution is known (for instance, a normalized Gaussian random vector). The statistical inverse problem then consists in identifying the nonlinear mapping \mathcal{Y} . Consequently, a parameterization of mapping \mathcal{Y} must be introduced in order to use parametric statistics, and there are two main approaches.
 1. The first one corresponds to the truncated polynomial chaos expansion of second-order random vector η with respect to the normalized Gaussian measure. In this case, Ξ is a normalized Gaussian random vector, and the orthogonal polynomials are the normalized Hermite polynomials [23]). If an arbitrary probability measure is used instead of the normalized Gaussian measure, then Ξ is a normalized random vector with this arbitrary probability distribution, and the orthogonal polynomials are constructed with respect to this arbitrary probability distribution [49, 57, 64, 77, 78]. Such a polynomial expansion defines a parameterization, noted as $\mathcal{Y}(\Xi, [z])$, of mapping \mathcal{Y} , in which the real matrix $[z]^T$ represents the \mathbb{R}^m -valued coefficients of the polynomial chaos expansion of η , and the identification of \mathcal{Y} is replaced by the identification of the hyperparameter $[z]$.
 2. The second approach consists in introducing an algebraic prior representation $\eta = \mathcal{Y}(\Xi, s)$ in which s is a vector-valued hyperparameter that has a small dimension and which must be identified using parametric statistics [54, 67, 76]. Similar to the method (i) presented before, if there is a sufficient amount of experimental data, the prior model can be updated in constructing a posterior probability distribution using the Bayesian approach [22, 45].

3.6 Statistical Inverse Problem for Identifying a Non-Gaussian Random Field as a Model Parameter of a BVP, Using Polynomial Chaos Expansion

The use of the polynomial chaos expansion for constructing a parameterized representation of a non-Gaussian random field that models the model parameter of a boundary value problem, in order to identify it using a statistical inverse method, has been initialized in [14, 15], used in [30], and revisited in [12]. In [11], the

construction of the probability model of the random coefficients of the polynomial chaos expansion is proposed by using the asymptotic sampling Gaussian distribution constructed with the Fisher information matrix and is used for model validation [24]. This work has been developed for statistical inverse problems that are rather in low stochastic dimension, and new ingredients have been introduced in [49, 61, 63] for statistical inverse problems in high stochastic dimension. In using the reduced chaos decomposition with random coefficients of random fields [66], a Bayesian approach for identifying the posterior probability model of the random coefficients of the polynomial chaos expansion of the model parameter of the BVP has been proposed in [2] for the low stochastic dimension and in [62] for the high stochastic dimension. The experimental identification of a non-Gaussian positive matrix-valued random field in high stochastic dimension, using partial and limited experimental data for a model observation related to the random solution of a stochastic BVP, is a difficult problem that requires both adapted representations and methodologies [48, 61–63].

3.7 Algebraic Prior Stochastic Models of the Model Parameters of BVP

In the methodology devoted to the identification of a non-Gaussian random field in high stochastic dimension, an important step is the construction of a parameterized representation for which the number of hyperparameters (in the parameterized representation) is generally very large due to the high stochastic dimension. In the framework of hypotheses for which only partial and limited data are available, such an identification is difficult if there is no information concerning the region of the admissible set (in high dimension), in which the optimal values of these hyperparameters must be searched. The optimization process, related to the statistical inverse problem, requires to localize the region in which the algorithms must search for an optimal value. The method consists in previously identifying the “center” of such a region, which corresponds to the value of the hyperparameters of the parameterized representation using a set of realizations generated with an algebraic prior stochastic model (APSM) that is specifically constructed on the basis of the available information associated with all the mathematical properties of the non-Gaussian random field that has to be identified. This APSM allows for enriching the information in order to overcome the lack of experimental data (since only partial experimental data are assumed to be available). This is particularly crucial for the identification of the non-Gaussian matrix-valued random field encountered, for instance, in three-dimensional linear elasticity, for which some works have been performed in order to introduce the symmetry, the positiveness and invertibility properties [56, 58, 60], the boundedness [27, 32], a capability of the prior stochastic model to exhibit a capability to generate simultaneously anisotropic statistical fluctuations and some statistical fluctuations in a symmetry class such as isotropic, cubic, transversely isotropic, orthotropic, etc. [26, 28, 29, 33, 69], and to develop the corresponding generators of realizations [26, 28, 29, 58, 61].

4 Overview

A complete methodology and the associated tools are presented for the experimental identification of a non-Gaussian matrix-valued random field that is the model parameter of a boundary value problem, using some experimental data related to a model observation. The difficulties of the statistical inverse problem that are presented are due to the following chosen framework that corresponds to many practical situations in computational sciences and engineering:

- A non-Gaussian matrix-valued random field must be identified, not simply a real-valued random field.
- The non-Gaussian random field that has to be identified is in high stochastic dimension and must be identified in a general class of random fields.
- Some fundamental algebraic properties of the non-Gaussian random field must be satisfied such as symmetry, positiveness, invertibility in mean square, boundedness, symmetry class, spatial-correlation lengths, etc.
- The available experimental data sets correspond only to partial and limited data for a model observation of the boundary value problem.

For such a statistical inverse problem, the above framework implies the use of an adapted and advanced methodology. The developments presented hereinafter are mainly related to the elasticity framework, but the methodology is general and can be used in many areas of computational sciences and engineering. The developments are organized as follows.

- The first one is devoted to the definition of the statistical inverse problem that has to be solved in high stochastic dimension and is focused on stochastic elliptic operators such as the ones that are encountered in the boundary value problems of the linear elasticity.
- The second one deals with the construction of two possible parameterized representations for a non-Gaussian positive-definite matrix-valued random field that models the model parameter of a boundary value problem. A parametric model-based representation is then constructed in introducing a statistical reduced model and a polynomial chaos expansion, first with deterministic coefficients and after with random coefficients. This parametric model-based representation is directly used for solving the statistical inverse problem.
- The third part is devoted to the description of all the steps of the methodology allowing the statistical inverse problem to be solved in high stochastic dimension. This methodology corresponds to the work initialized in [61], extended in [62] for constructing a posterior stochastic model using the Bayesian approach, and revisited in [48, 49].
- The fourth part presents the construction of an algebraic prior stochastic model of the model parameter of the boundary value problem, for a non-Gaussian matrix-valued random field. This construction is based on the works [27, 28, 58, 60] and reuses the formalism and the results introduced in the developments presented in

Sect. 12 of ▶ Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification. The generator of realizations for such an algebraic prior stochastic model for a non-Gaussian matrix-valued random field is presented [28, 58, 61].

5 Notations

The following algebraic notations are used.

5.1 Euclidean Space

Let $\mathbf{x} = (x_1, \dots, x_n)$ be a vector in \mathbb{R}^n . The Euclidean space \mathbb{R}^n is equipped with the usual inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^n x_j y_j$ and the associated norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

5.2 Sets of Matrices

Let $\mathbb{M}_{n,m}(\mathbb{R})$ be the set of all the $(n \times m)$ real matrices,

$\mathbb{M}_n(\mathbb{R}) = \mathbb{M}_{n,n}(\mathbb{R})$ the square matrices,

$\mathbb{M}_n^S(\mathbb{R})$ be the set of all the symmetric $(n \times n)$ real matrices,

$\mathbb{M}_n^U(\mathbb{R})$ be the set of all the upper triangular $(n \times n)$ real matrices with positive diagonal entries,

$\mathbb{M}_n^+(\mathbb{R})$ be the set of all the positive-definite symmetric $(n \times n)$ real matrices.

The ensembles of real matrices are such that

$$\mathbb{M}_n^+(\mathbb{R}) \subset \mathbb{M}_n^S(\mathbb{R}) \subset \mathbb{M}_n(\mathbb{R}).$$

5.3 Kronecker Symbol, Unit Matrix, and Indicator Function

The Kronecker symbol is denoted by δ_{jk} and is such that $\delta_{jk} = 0$ if $j \neq k$ and $\delta_{jj} = 1$. The unit (or identity) matrix in $\mathbb{M}_n(\mathbb{R})$ is denoted by $[I_n]$ and is such that $[I_n]_{jk} = \delta_{jk}$. Let \mathbb{S} be any subset of any set \mathbb{M} , possibly with $\mathbb{S} = \mathbb{M}$. The indicator function $M \mapsto \mathbb{1}_{\mathbb{S}}(M)$ defined on set \mathbb{M} is such that $\mathbb{1}_{\mathbb{S}}(M) = 1$ if $M \in \mathbb{S} \subset \mathbb{M}$, and $\mathbb{1}_{\mathbb{S}}(M) = 0$ if $M \notin \mathbb{S}$.

5.4 Norms and Usual Operators

- (i) The determinant of a matrix $[G]$ in $\mathbb{M}_n(\mathbb{R})$ is denoted by $\det[G]$, and its trace is denoted by $\text{tr}[G] = \sum_{j=1}^n G_{jj}$.
- (ii) The transpose of a matrix $[G]$ in $\mathbb{M}_{n,m}(\mathbb{R})$ is denoted by $[G]^T$, which is in $\mathbb{M}_{m,n}(\mathbb{R})$.

- (iii) The operator norm of a matrix $[G]$ in $\mathbb{M}_{n,m}(\mathbb{R})$ is denoted by $\|G\| = \sup_{\|\mathbf{x}\| \leq 1} \| [G] \mathbf{x} \|$ for all \mathbf{x} in \mathbb{R}^m , which is such that $\| [G] \mathbf{x} \| \leq \|G\| \|\mathbf{x}\|$ for all \mathbf{x} in \mathbb{R}^m .
- (iv) For $[G]$ and $[H]$ in $\mathbb{M}_{n,m}(\mathbb{R})$, we denote $\ll [G], [H] \gg = \text{tr}\{[G]^T [H]\}$ and the Frobenius norm (or Hilbert-Schmidt norm) $\|G\|_F$ of $[G]$ is such that $\|G\|_F^2 = \ll [G], [G] \gg = \text{tr}\{[G]^T [G]\} = \sum_{j=1}^n \sum_{k=1}^m G_{jk}^2$, which is such that $\|G\| \leq \|G\|_F \leq \sqrt{n} \|G\|$.
- (v) The gradient $\nabla_{\mathbf{x}} u(\mathbf{x})$ at point \mathbf{x} in \mathbb{R}^n of the real-valued function $\mathbf{x} \mapsto u(\mathbf{x})$ is the vector in \mathbb{R}^n such that $\{\nabla_{\mathbf{x}} u(\mathbf{x})\}_j = \partial u(\mathbf{x}) / \partial x_j$ for $j = 1, \dots, n$. The divergence $\text{div}_{\mathbf{x}}(\mathbf{u}(\mathbf{x}))$ at point \mathbf{x} in \mathbb{R}^n of the \mathbb{R}^n -valued function $\mathbf{x} \mapsto \mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), \dots, u_n(\mathbf{x}))$ is the real number such that $\text{div}_{\mathbf{x}}(\mathbf{u}(\mathbf{x})) = \sum_{j=1}^n \partial u_j(\mathbf{x}) / \partial x_j$.

5.5 Order Relation in the Set of All the Positive-Definite Real Matrices

Let $[G]$ and $[H]$ be two matrices in $\mathbb{M}_n^+(\mathbb{R})$. The notation $[G] > [H]$ means that the matrix $[G] - [H]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$.

5.6 Probability Space, Mathematical Expectation, and Space of Second-Order Random Vectors

The mathematical expectation relative to a probability space (Θ, \mathcal{T}, P) is denoted by E . The space of all the second-order random variables, defined on (Θ, \mathcal{T}, P) , with values in \mathbb{R}^n , equipped with the inner product $((\mathbf{X}, \mathbf{Y})) = E\{\langle \mathbf{X}, \mathbf{Y} \rangle\}$ and with the associated norm $|||\mathbf{X}||| = ((\mathbf{X}, \mathbf{X}))^{1/2}$, is a Hilbert space denoted by \mathcal{L}_n^2 .

6 Setting the Statistical Inverse Problem to be Solved in High Stochastic Dimension

Let d be an integer such that $1 \leq d \leq 3$. Let n be another finite integer such that $n \geq 1$, and let N_u be an integer such that $1 \leq N_u \leq n$. Let Ω be a bounded open domain of \mathbb{R}^d , with generic point $\mathbf{x} = (x_1, \dots, x_d)$, with boundary $\partial\Omega$, and let be $\bar{\Omega} = \Omega \cup \partial\Omega$.

6.1 Stochastic Elliptic Operator and Boundary Value Problem

Let $[\mathbf{K}] = \{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \bar{\Omega}\}$ be a non-Gaussian random field, in high stochastic dimension, defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by $\bar{\Omega}$, with values in $\mathbb{M}_n^+(\mathbb{R})$. It should be noted that random field $[\mathbf{K}]$ being with values in $\mathbb{M}_n^+(\mathbb{R})$,

random field $[\mathbf{K}]$ cannot be a Gaussian field. Such a random field $[\mathbf{K}]$ allows for constructing the coefficients of a given stochastic elliptic operator $\mathbf{u} \mapsto \mathcal{D}_{\mathbf{x}}(\mathbf{u})$ that applies to the random field $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), \dots, u_{N_u}(\mathbf{x}))$, indexed by Ω , with values in \mathbb{R}^{N_u} .

The boundary value problem that is formulated in \mathbf{u} involves the stochastic elliptic operator $\mathcal{D}_{\mathbf{x}}$, and some Dirichlet and Neumann boundary conditions are given on $\partial\Omega$ that is written as the union of three parts, $\partial\Omega = \Gamma_0 \cup \Gamma \cup \Gamma_1$. On the part Γ_0 , a Dirichlet condition is given. The part Γ corresponds to the part of the boundary on which there is a zero Neumann condition and on which experimental data are available for \mathbf{u} . On the part Γ_1 , a Neumann condition is given. The boundary value problems, involving such a stochastic elliptic operator $\mathcal{D}_{\mathbf{x}}$, are encountered in many problems of computational sciences and engineering.

■ *Examples of stochastic elliptic operators.*

- (i) For a three-dimensional anisotropic diffusion problem, the stochastic elliptic differential operator $\mathcal{D}_{\mathbf{x}}$ relative to the density u of the diffusing medium is written as

$$\{\mathcal{D}_{\mathbf{x}}(u)\}(\mathbf{x}) = -\operatorname{div}_{\mathbf{x}}([\mathbf{K}(\mathbf{x})] \nabla_{\mathbf{x}} u(\mathbf{x})), \quad \mathbf{x} \in \Omega, \quad (26.1)$$

in which $d = n = 3$ and $N_u = 1$ and where $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is the $\mathbb{M}_n^+(\mathbb{R})$ -valued random field of the medium.

- (ii) For the wave propagation inside a three-dimensional random heterogeneous anisotropic linear elastic medium, we have $d = 3$, $n = 6$, and $N_u = 3$, and the stochastic elliptic differential operator $\mathcal{D}_{\mathbf{x}}$ relative to the displacement field \mathbf{u} is written as

$$\{\mathcal{D}_{\mathbf{x}}(\mathbf{u})\}(\mathbf{x}) = -[D_{\mathbf{x}}]^T [\mathbf{K}(\mathbf{x})] [D_{\mathbf{x}}] \mathbf{u}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (26.2)$$

in which $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is the $\mathbb{M}_n^+(\mathbb{R})$ -valued elasticity random field of the medium deduced from the fourth-order tensor-valued elasticity field $\{\mathbf{C}_{ijkl}(\mathbf{x}), \mathbf{x} \in \Omega\}$ by the following equation,

$$[\mathbf{K}] = \begin{bmatrix} \mathbf{C}_{1111} & \mathbf{C}_{1122} & \mathbf{C}_{1133} & \sqrt{2}\mathbf{C}_{1112} & \sqrt{2}\mathbf{C}_{1113} & \sqrt{2}\mathbf{C}_{1123} \\ \mathbf{C}_{2211} & \mathbf{C}_{2222} & \mathbf{C}_{2233} & \sqrt{2}\mathbf{C}_{2212} & \sqrt{2}\mathbf{C}_{2213} & \sqrt{2}\mathbf{C}_{2223} \\ \mathbf{C}_{3311} & \mathbf{C}_{3322} & \mathbf{C}_{3333} & \sqrt{2}\mathbf{C}_{3312} & \sqrt{2}\mathbf{C}_{3313} & \sqrt{2}\mathbf{C}_{3323} \\ \sqrt{2}\mathbf{C}_{1211} & \sqrt{2}\mathbf{C}_{1222} & \sqrt{2}\mathbf{C}_{1233} & 2\mathbf{C}_{1212} & 2\mathbf{C}_{1213} & 2\mathbf{C}_{1223} \\ \sqrt{2}\mathbf{C}_{1311} & \sqrt{2}\mathbf{C}_{1322} & \sqrt{2}\mathbf{C}_{1333} & 2\mathbf{C}_{1312} & 2\mathbf{C}_{1313} & 2\mathbf{C}_{1323} \\ \sqrt{2}\mathbf{C}_{2311} & \sqrt{2}\mathbf{C}_{2322} & \sqrt{2}\mathbf{C}_{2333} & 2\mathbf{C}_{2312} & 2\mathbf{C}_{2313} & 2\mathbf{C}_{2323} \end{bmatrix}, \quad (26.3)$$

in which $[D_{\mathbf{x}}]$ is the differential operator,

$$[D_{\mathbf{x}}] = [M^{(1)}] \frac{\partial}{\partial x_1} + [M^{(2)}] \frac{\partial}{\partial x_2} + [M^{(3)}] \frac{\partial}{\partial x_3}, \quad (26.4)$$

where $[M^{(1)}]$, $[M^{(2)}]$ and $[M^{(3)}]$ are the $(n \times N_u)$ real matrices defined by

$$[M^{(1)}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \end{bmatrix}, [M^{(2)}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}, [M^{(3)}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}. \quad (26.5)$$

■ *Example of a time-independent stochastic boundary value problem in linear elasticity.*

Let be $d = 3$, $n = 6$, and $N_u = 3$. Let us consider the boundary value problem related to the linear elastostatic deformation of a three-dimensional random heterogeneous anisotropic linear elastic medium occupying domain Ω , for which an experimental displacement field $\mathbf{u}^{\text{exp},\ell}$ is measured on Γ . Let $\mathbf{n}(\mathbf{x}) = (n_1(\mathbf{x}), n_2(\mathbf{x}), n_3(\mathbf{x}))$ be the unit normal to $\partial\Omega$, exterior to Ω . The stochastic boundary value problem is written as

$$\mathcal{D}_{\mathbf{x}}(\mathbf{u}) = \mathbf{0} \quad \text{in } \Omega, \quad (26.6)$$

in which the stochastic operator $\mathcal{D}_{\mathbf{x}}$ is defined by Eq. (26.2), where the Dirichlet condition is

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_0, \quad (26.7)$$

and where the Neumann condition is written as

$$[\mathcal{M}_{\mathbf{n}}(\mathbf{x})]^T [\mathbf{K}(\mathbf{x})][D_{\mathbf{x}}] \mathbf{u}(\mathbf{x}) = \mathbf{0} \quad \text{on } \Gamma, \text{ and } = \mathbf{f}_{\Gamma_1} \quad \text{on } \Gamma_1, \quad (26.8)$$

in which $[\mathcal{M}_{\mathbf{n}}(\mathbf{x})] = [M^{(1)}] n_1(\mathbf{x}) + [M^{(2)}] n_2(\mathbf{x}) + [M^{(3)}] n_3(\mathbf{x})$ and where \mathbf{f}_{Γ_1} is a given surface force field applied to Γ_1 . The boundary value problem defined by Eqs. (26.6), (26.7) and (26.8) is typically the one for which the random field $\{\mathbf{K}(\mathbf{x}), \mathbf{x} \in \Omega\}$ has to be identified by solving a statistical inverse problem in high stochastic dimension with the partial and limited experimental data $\{\mathbf{u}^{\text{exp},\ell}, \ell = 1, \dots, v_{\text{exp}}\}$.

6.2 Stochastic Finite Element Approximation of the Stochastic Boundary Value Problem

Let us assume that the weak formulation of the stochastic boundary value problem involving stochastic elliptic operator $\mathcal{D}_{\mathbf{x}}$ is discretized by using the finite element method. Let $\mathcal{I} = \{\mathbf{x}^1, \dots, \mathbf{x}^{N_p}\} \subset \Omega$ be the finite subset of Ω made up of all the

integrating points in the numerical integration formulae for the finite elements [79] used in the mesh of Ω . Let $\mathbf{U} = (U_1, \dots, U_{N_U})$ be the random model observation with values in \mathbb{R}^{N_U} , constituted of the N_U observed degrees of freedom for which there are available experimental data (corresponding to some degrees of freedom of the nodal values of \mathbf{u} at all the nodes in Γ). The random observation vector \mathbf{U} is the unique deterministic nonlinear transformation of the finite family of the N_p dependent random matrices $[\mathbf{K}(\mathbf{x}^1)], \dots, [\mathbf{K}(\mathbf{x}^{N_p})]$ such that

$$\mathbf{U} = \mathbf{h}([\mathbf{K}(\mathbf{x}^1)], \dots, [\mathbf{K}(\mathbf{x}^{N_p})]), \quad (26.9)$$

in which

$$([K^1], \dots, [K^{N_p}]) \mapsto \mathbf{h}([K^1], \dots, [K^{N_p}]) : \mathbb{M}_n^+(\mathbb{R}) \times \dots \times \mathbb{M}_n^+(\mathbb{R}) \longrightarrow \mathbb{R}^{N_U}, \quad (26.10)$$

is a deterministic nonlinear transformation that is constructed by solving the discretized boundary value problem.

6.3 Experimental Data Sets

It is assumed that v_{exp} experimental data sets are available for the random observation vector \mathbf{U} . Each experimental data set corresponds to partial experimental data (only some degrees of freedom of the nodal values of the displacement field on Γ are observed) with a limited length (v_{exp} is relatively small). These v_{exp} experimental data sets correspond to measurements of v_{exp} experimental configurations associated with the same boundary value problem. For configuration ℓ , with $\ell = 1, \dots, v_{\text{exp}}$, the observation vector (corresponding to \mathbf{U} for the computational model) is denoted by $\mathbf{u}^{\text{exp},\ell}$ and belongs to \mathbb{R}^{N_U} . Therefore, the available data are made up of the v_{exp} vectors $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},v_{\text{exp}}}$ in \mathbb{R}^{N_U} . It is assumed that $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},v_{\text{exp}}}$ correspond to v_{exp} independent realizations of a random vector \mathbf{U}^{exp} defined on a probability space $(\Theta^{\text{exp}}, \mathcal{T}^{\text{exp}}, \mathcal{P}^{\text{exp}})$ and correspond to random observation vector \mathbf{U} of the stochastic computational model (random vectors \mathbf{U}^{exp} and \mathbf{U} are not defined on the same probability space). It should be noted that the experimental data do not correspond to a field measurement in $\overline{\Omega}$ but only to a field measurement on the part Γ of the boundary $\partial\Omega$ of domain Ω . This is the reason why the experimental data are called “partial”.

6.4 Statistical Inverse Problem to be Solved

The problem that must be solved is the identification of non-Gaussian matrix-valued random field $[\mathbf{K}]$, using the partial and limited experimental data $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},v_{\text{exp}}}$ relative to the random observation vector \mathbf{U} of the stochastic computational model and defined by Eq. (26.9).

7 Parametric Model-Based Representation for the Model Parameters and Model Observations

As explained in the previous paragraph entitled Sect. 2.2, a parametric model-based representation $\mathbf{U} = \mathcal{B}(\boldsymbol{\Xi}, [z])$ must be constructed in order to be able to solve the statistical inverse problem allowing random model parameter $[\mathbf{K}]$ to be identified using the experimental data sets. For that, it is needed to introduce:

- a representation of the non-Gaussian positive-definite matrix-valued random field $[\mathbf{K}]$ that is expressed as a transformation \mathcal{G} of a non-Gaussian second-order symmetric matrix-valued random field $[\mathbf{G}]$, such that for all \mathbf{x} in Ω , $[\mathbf{K}(\mathbf{x})] = \mathcal{G}([\mathbf{G}(\mathbf{x})])$, where \mathcal{G} is independent of \mathbf{x} (in fact, two types of representation are proposed),
- a truncated reduced representation of random field $[\mathbf{G}]$,
- a parameterized representation for non-Gaussian random field $[\mathbf{K}]$,
- the parametric model-based representation $\mathbf{U} = \mathcal{B}(\boldsymbol{\Xi}, [z])$.

7.1 Introduction a Class of Lower-Bounded Random Fields for $[\mathbf{K}]$ and Normalization

In order to normalize random field $[\mathbf{K}]$, a deterministic function $\mathbf{x} \mapsto [\underline{K}(\mathbf{x})]$ from Ω into $\mathbb{M}_n^+(\mathbb{R})$ is introduced such that, for all \mathbf{x} in Ω and for all \mathbf{z} in \mathbb{R}^n , $\langle [\underline{K}(\mathbf{x})] \mathbf{z}, \mathbf{z} \rangle \geq \underline{k}_0 \|\mathbf{z}\|^2$ and $\langle [\underline{K}(\mathbf{x})] \mathbf{z}, \mathbf{z} \rangle \leq \underline{k}_1 \|\mathbf{z}\|^2$ in which \underline{k}_0 and \underline{k}_1 are positive real constants, independent of \mathbf{x} , such that $0 < \underline{k}_0 < \underline{k}_1 < +\infty$. These two technical inequalities correspond to the mathematical hypotheses that are required for obtaining a uniform deterministic elliptic operator whose coefficient is $[\underline{K}]$.

We introduce the following class of non-Gaussian positive-definite matrix-valued random fields $[\mathbf{K}]$, which admit a positive-definite matrix-valued lower bound, defined by

$$[\mathbf{K}(\mathbf{x})] = \frac{1}{1+\varepsilon} [\underline{L}(\mathbf{x})]^T \{ \varepsilon[I_n] + [\mathbf{K}_0(\mathbf{x})] \} [\underline{L}(\mathbf{x})], \quad \forall \mathbf{x} \in \Omega, \quad (26.11)$$

in which $\varepsilon > 0$ is any fixed positive real number, where $[\underline{L}(\mathbf{x})]$ is the upper triangular $(n \times n)$ real matrix such that $[\underline{K}(\mathbf{x})] = [\underline{L}(\mathbf{x})]^T [\underline{L}(\mathbf{x})]$ and where $[\mathbf{K}_0] = \{[\mathbf{K}_0(\mathbf{x})], \mathbf{x} \in \Omega\}$ is any random field indexed by Ω , with values in $\mathbb{M}_n^+(\mathbb{R})$. Equation (26.11) can be inverted,

$$[\mathbf{K}_0(\mathbf{x})] = (1+\varepsilon)[\underline{L}(\mathbf{x})]^{-T} [\mathbf{K}(\mathbf{x})] [\underline{L}(\mathbf{x})]^{-1} - \varepsilon[I_n], \quad \forall \mathbf{x} \in \Omega. \quad (26.12)$$

We have the following important properties for the class defined:

- Random field $[\mathbf{K}]$ is effective with values in $\mathbb{M}_n^+(\mathbb{R})$. For all \mathbf{x} fixed in Ω , the lower bound is the matrix belonging to $\mathbb{M}_n^+(\mathbb{R})$ defined by $[K_\varepsilon(\mathbf{x})] = \frac{\varepsilon}{1+\varepsilon} [\underline{K}(\mathbf{x})]$,

and for all random matrix $[\mathbf{K}_0(\mathbf{x})]$ with values in $\mathbb{M}_n^+(\mathbb{R})$, $[\mathbf{K}(\mathbf{x})]$, defined by Eq. (26.12), is a random matrix with values in a subset of $\mathbb{M}_n^+(\mathbb{R})$ such that $[\mathbf{K}(\mathbf{x})] \geq [K_\varepsilon(\mathbf{x})]$ almost surely.

- For all integer $p \geq 1$, $\{[\mathbf{K}(\mathbf{x})]^{-1}, \mathbf{x} \in \Omega\}$ is a p -order random field with values in $\mathbb{M}_n^+(\mathbb{R})$, i.e., for all \mathbf{x} in Ω , $E\{\|[\mathbf{K}(\mathbf{x})]^{-1}\|_F^p\} < +\infty$ and, in particular, is a second-order random field.
- If $[\mathbf{K}_0]$ is a second-order random field, i.e., for all \mathbf{x} in Ω , $E\{\|\mathbf{K}_0(\mathbf{x})\|_F^2\} < +\infty$, then $[\mathbf{K}]$ is a second-order random field, i.e., for all \mathbf{x} in Ω , $E\{\|\mathbf{K}(\mathbf{x})\|_F^2\} < +\infty$.
- If function $[K]$ is chosen as the mean function of random field $[\mathbf{K}]$, i.e., $[K(\mathbf{x})] = E\{[\mathbf{K}(\mathbf{x})]\}$ for all \mathbf{x} , then $E\{[\mathbf{K}_0(\mathbf{x})]\}$ is equal to $[I_n]$, which shows that random field $[\mathbf{K}_0]$ is normalized.
- The class of random fields defined by Eq. (26.11) yields a uniform stochastic elliptic operator $\mathcal{D}_{\mathbf{x}}$ that allows for studying the existence and uniqueness of a second-order random solution of a stochastic boundary value problem involving $\mathcal{D}_{\mathbf{x}}$.

7.2 Construction of the Nonlinear Transformation \mathcal{G}

Two types of representation of random field $[\mathbf{K}_0]$ is proposed hereinafter: An “exponential-type representation” and a “square-type representation”.

■ *Exponential-type representation of random field $[\mathbf{K}_0]$.*

For all second-order random field $[\mathbf{G}] = \{[\mathbf{G}(\mathbf{x})], \mathbf{x} \in \Omega\}$ with values in $\mathbb{M}_n^S(\mathbb{R})$, which is not assumed to be Gaussian, the random field $[\mathbf{K}_0]$ defined by

$$[\mathbf{K}_0(\mathbf{x})] = \exp_{\mathbb{M}}([\mathbf{G}(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad (26.13)$$

in which $\exp_{\mathbb{M}}$ denotes the exponential of symmetric square real matrices, is a random field with values in $\mathbb{M}_n^+(\mathbb{R})$. If $[\mathbf{K}_0]$ is any random field with values in $\mathbb{M}_n^+(\mathbb{R})$, then there exists a unique random field $[\mathbf{G}]$ with values in $\mathbb{M}_n^S(\mathbb{R})$ such that

$$[\mathbf{G}(\mathbf{x})] = \log_{\mathbb{M}}([\mathbf{K}_0(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad (26.14)$$

in which $\log_{\mathbb{M}}$ is the reciprocity mapping of $\exp_{\mathbb{M}}$, which is defined on $\mathbb{M}_n^+(\mathbb{R})$ with values in $\mathbb{M}_n^S(\mathbb{R})$, but in general, random field $[\mathbf{G}]$ is not a second-order random field. If $[\mathbf{G}]$ is any second-order random field with values in $\mathbb{M}_n^+(\mathbb{R})$, in general, the random field $[\mathbf{K}_0] = \exp_{\mathbb{M}}([\mathbf{G}])$ is not a second-order random field. Nevertheless, it can be proved that, if $[\mathbf{K}_0]$ and $[\mathbf{K}_0]^{-1}$ are second-order random fields with values in $\mathbb{M}_n^+(\mathbb{R})$, then there exists a second-order random field $[\mathbf{G}]$ with values in $\mathbb{M}_n^S(\mathbb{R})$ such that $[\mathbf{K}_0] = \exp_{\mathbb{M}}([\mathbf{G}])$.

■ *Square-type representation of random field $[\mathbf{K}_0]$.*

Let $g \mapsto h(g; a)$ be a given function from \mathbb{R} in \mathbb{R}^+ , depending on one positive real parameter a . For all fixed a , it is assumed that:

- $h(\cdot; a)$ is a strictly monotonically increasing function on \mathbb{R} , which means that $h(g; a) < h(g'; a)$ if $-\infty < g < g' < +\infty$;

- (ii) there are real numbers $0 < c_h < +\infty$ and $0 < c_a < +\infty$, such that, for all g in \mathbb{R} , we have $h(g; a) \leq c_a + c_h g^2$.

The introduced hypotheses imply that, for all $a > 0$, $g \mapsto h(g; a)$ is a one-to-one mapping from \mathbb{R} onto \mathbb{R}^+ and consequently, the reciprocity mapping, $v \mapsto h^{-1}(v; a)$, is a strictly monotonically increasing function from \mathbb{R}^+ onto \mathbb{R} . The square-type representation of random field $[\mathbf{K}_0]$, indexed by Ω , with values in $\mathbb{M}_n^+(\mathbb{R})$, is defined by

$$[\mathbf{K}_0(\mathbf{x})] = \mathbb{L}([\mathbf{G}(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad (26.15)$$

in which $[\mathbf{G}] = \{[\mathbf{G}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is a second-order random field with values in $\mathbb{M}_n^S(\mathbb{R})$ and where $[G] \mapsto \mathbb{L}([G])$ is a measurable mapping from $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{M}_n^+(\mathbb{R})$ which is defined as follows. The matrix $[K_0] = \mathbb{L}([G]) \in \mathbb{M}_n^+(\mathbb{R})$ is written as $[K_0] = [L]^T [L]$ in which $[L]$ belongs to $\mathbb{M}_n^U(\mathbb{R})$, which is written as $[L] = \mathcal{L}([G])$ where $[G] \mapsto \mathcal{L}([G])$ is the measurable mapping from $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{M}_n^U(\mathbb{R})$ defined by

$$[\mathcal{L}([G])]_{jk} = [G]_{jk}, \quad 1 \leq j < k \leq n, \quad [\mathcal{L}([G])]_{jj} = \sqrt{h([G]_{jj}; a_j)}, \quad 1 \leq j \leq n, \quad (26.16)$$

in which a_1, \dots, a_n are positive real numbers. If $[\mathbf{K}_0]$ is any random field indexed by Ω with values in $\mathbb{M}_n^+(\mathbb{R})$, then there exists a unique random field $[\mathbf{G}]$ with values in $\mathbb{M}_n^S(\mathbb{R})$ such that

$$[\mathbf{G}(\mathbf{x})] = \mathbb{L}^{-1}([\mathbf{K}_0(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad (26.17)$$

in which \mathbb{L}^{-1} is the reciprocity function of \mathbb{L} , from $\mathbb{M}_n^+(\mathbb{R})$ into $\mathbb{M}_n^S(\mathbb{R})$, which is explicitly defined as follows. For all $1 \leq j \leq k \leq n$,

$$[\mathbf{G}(\mathbf{x})]_{jk} = [\mathbb{L}^{-1}([\mathbf{L}(\mathbf{x})])]_{jk}, \quad [\mathbf{G}(\mathbf{x})]_{kj} = [\mathbf{G}(\mathbf{x})]_{jk}, \quad (26.18)$$

in which $[L] \mapsto \mathbb{L}^{-1}([L])$ is the unique reciprocity mapping of \mathbb{L} (due to the existence of $v \mapsto h^{-1}(v; a)$) defined on $\mathbb{M}_n^U(\mathbb{R})$ and where $[\mathbf{L}(\mathbf{x})]$ follows from the Cholesky factorization of random matrix $[\mathbf{K}_0(\mathbf{x})] = [\mathbf{L}(\mathbf{x})]^T [\mathbf{L}(\mathbf{x})]$ (see Eq. (26.15)).

Example of function h . An example of such a function is given in *An algebraic prior stochastic model $[\mathbf{K}^{\text{APSM}}]$ for the case of anisotropic statistical fluctuations* of the present section. Nevertheless, for the sake of clarity, we detail it hereinafter. Let $h = h^{\text{APSM}}$ be the function h^{APSM} defined in [58] as follows. Let be $s = \delta/\sqrt{n+1}$ in which δ is a parameter such that $0 < \delta < \sqrt{(n+1)/(n-1)}$ and which allows the statistical fluctuations level to be controlled. Let be $a_j = 1/(2s^2) + (1-j)/2 > 0$ and $h^{\text{APSM}}(g; a) = 2s^2 F_{\Gamma_a}^{-1}(F_W(g/s))$ with $F_W(\tilde{w}) = \int_{-\infty}^{\tilde{w}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}w^2) dw$ and $F_{\Gamma_a}^{-1}(u) = \gamma$ the reciprocal function such that $F_{\Gamma_a}(\gamma) = u$ with $F_{\Gamma_a}(\gamma) = \int_0^\gamma \frac{1}{\Gamma(a)} t^{a-1} e^{-t} dt$ and $\Gamma(a) = \int_0^{+\infty} t^{a-1} e^{-t} dt$. Then, for all $j = 1, \dots, n$, it can be proved that $g \mapsto h^{\text{APSM}}(g; a_j)$ is a strictly monotonically increasing function

from \mathbb{R} into \mathbb{R}^+ and there are positive real numbers c_h and c_{a_j} such that, for all g in \mathbb{R} , we have $h^{\text{APSM}}(g; a_j) \leq c_{a_j} + c_h g^2$. In addition, it can easily be seen that the reciprocity function is written as $h^{\text{APSM}^{-1}}(v; a) = s F_W^{-1}(F_{\Gamma_a}(v/(2s^2)))$.

■ *Construction of the transformation \mathcal{G} and its inverse \mathcal{G}^{-1} .*

For the *exponential-type representation*, the transformation \mathcal{G} is defined by Eq. (26.11) with Eq. (26.13), and its inverse \mathcal{G}^{-1} is defined by Eq. (26.14) with Eq. (26.12), and is such that, for all \mathbf{x} in Ω ,

$$[\mathbf{K}(\mathbf{x})] = \mathcal{G}([\mathbf{G}(\mathbf{x})]) := \frac{1}{1 + \varepsilon} [\underline{L}(\mathbf{x})]^T \{ \varepsilon [I_n] + \exp_{\mathbb{M}}([\mathbf{G}(\mathbf{x})]) \} [\underline{L}(\mathbf{x})], \quad (26.19)$$

$$[\mathbf{G}(\mathbf{x})] = \mathcal{G}^{-1}([\mathbf{K}(\mathbf{x})]) := \log_{\mathbb{M}}\{ (1 + \varepsilon)[\underline{L}(\mathbf{x})]^{-T} [\mathbf{K}(\mathbf{x})] [\underline{L}(\mathbf{x})]^{-1} - \varepsilon [I_n] \}. \quad (26.20)$$

For the *square-type representation*, the transformation \mathcal{G} is defined by Eq. (26.11) with Eq. (26.15), and its inverse \mathcal{G}^{-1} is defined by Eq. (26.17) with Eq. (26.12), and is such that, for all \mathbf{x} in Ω ,

$$[\mathbf{K}(\mathbf{x})] = \mathcal{G}([\mathbf{G}(\mathbf{x})]) := \frac{1}{1 + \varepsilon} [\underline{L}(\mathbf{x})]^T \{ \varepsilon [I_n] + [\mathcal{L}([\mathbf{G}(\mathbf{x})])]^T [\mathcal{L}([\mathbf{G}(\mathbf{x})])] \} [\underline{L}(\mathbf{x})], \quad (26.21)$$

$$[\mathbf{G}(\mathbf{x})] = \mathcal{G}^{-1}([\mathbf{K}(\mathbf{x})]) := \mathbb{L}^{-1}\{ (1 + \varepsilon)[\underline{L}(\mathbf{x})]^{-T} [\mathbf{K}(\mathbf{x})] [\underline{L}(\mathbf{x})]^{-1} - \varepsilon [I_n] \}. \quad (26.22)$$

Let $\mathbb{M}_n^{+b}(\mathbb{R})$ be the subset of $\mathbb{M}_n^+(\mathbb{R})$, constituted of all the positive-definite matrices $[K]$ such that, for all \mathbf{x} in Ω , the matrix $[K] - [K_\varepsilon(\mathbf{x})] > 0$. Transformation \mathcal{G} maps $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{M}_n^{+b}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$ and \mathcal{G}^{-1} maps $\mathbb{M}_n^{+b}(\mathbb{R})$ into $\mathbb{M}_n^S(\mathbb{R})$.

7.3 Truncated Reduced Representation of Second-Order Random Field $[\mathbf{G}]$ and Its Polynomial Chaos Expansion

Two versions of the nonlinear transformation \mathcal{G} from $\mathbb{M}_n^S(\mathbb{R})$ into $\mathbb{M}_n^+(\mathbb{R})$ are defined by Eqs. (26.19) and (26.21). For the statistical inverse problem, $[\mathbf{G}]$ is chosen in the class of the second-order random field indexed by Ω with values in $\mathbb{M}_n^S(\mathbb{R})$, which is reduced using its truncated Karhunen-Loève decomposition in which the random coordinates are represented using a truncated polynomial Gaussian chaos. Consequently, the approximation $[\mathbf{G}^{(m,N,N_g)}]$ of the non-Gaussian second-order random field $[\mathbf{G}]$ is introduced such that

$$[\mathbf{G}^{(m,N,N_g)}(\mathbf{x})] = [G_0(\mathbf{x})] + \sum_{i=1}^m \sqrt{\lambda_i} [G_i(\mathbf{x})] \eta_i, \quad (26.23)$$

$$\eta_i = \sum_{j=1}^N y_i^j \Psi_j(\Xi), \quad (26.24)$$

in which

- $\lambda_1 \geq \dots \geq \lambda_m > 0$ are the dominant eigenvalues and $[G_1], \dots, [G_m]$ are the corresponding orthonormal eigenfunctions of the covariance operator Cov_G of random field $[G]$. The kernel of this covariance operator is the tensor-valued cross covariance function $C_G(\mathbf{x}, \mathbf{x}')$ of $[G]$, which is assumed to be square integrable on $\Omega \times \Omega$,
- $\{\Psi_j\}_{j=1}^N$ only depends on a random vector $\boldsymbol{\Xi} = (\Xi_1, \dots, \Xi_{N_g})$ of $N_g \leq m$ independent normalized Gaussian random variables Ξ_1, \dots, Ξ_{N_g} defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$,
- $\{\Psi_j\}_{j=1}^N$ are the polynomial Gaussian chaos that are written as $\Psi_j(\boldsymbol{\Xi}) = \Phi_{\alpha_1}(\Xi_1) \times \dots \times \Phi_{\alpha_{N_g}}(\Xi_{N_g})$, in which j is the index associated with the multi-index $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{N_g})$ in \mathbb{N}^{N_g} , the degree of $\Psi_j(\boldsymbol{\Xi})$ is $\alpha_1 + \dots + \alpha_{N_g} \leq N_d$ and where $\Phi_{\alpha_k}(\Xi_k)$ is the normalized univariate Hermite polynomial on \mathbb{R} . Consequently, $\{\Psi_j\}_{j=1}^N$ are composed of the normalized multivariate Hermite polynomials such that $E\{\Psi_j(\boldsymbol{\Xi}) \Psi_{j'}(\boldsymbol{\Xi})\} = \delta_{jj'}$,
- the constant Hermite polynomial $\Psi_0(\boldsymbol{\Xi}) = 1$ with index $j = 0$ (corresponding to the zero multi-index $(0, \dots, 0)$) is not included in Eq. (26.24). Consequently, the integer N is such that $N = (N_d + N_g)! / (N_d! N_g!) - 1$ where N_d is the maximum degree of the normalized multivariate Hermite polynomials,
- y_i^j are the coefficients that are supposed to verify $\sum_{j=1}^N y_i^j y_{i'}^j = \delta_{ii'}$, which ensures that the random variables, $\{\eta_i\}_{i=1}^m$, are uncorrelated centered random variables with unit variance, which means that $E\{\eta_i \eta_{i'}\} = \delta_{ii'}$. The relation between the coefficients can be rewritten as

$$[z]^T [z] = [I_m], \quad (26.25)$$

in which $[z] \in \mathbb{M}_{N,m}(\mathbb{R})$ is such that

$$[z]_{ji} = y_i^j, \quad 1 \leq i \leq m, \quad 1 \leq j \leq N. \quad (26.26)$$

Introducing the random vectors $\boldsymbol{\eta} = (\eta_1, \dots, \eta_m)$ and $\boldsymbol{\Psi}(\boldsymbol{\Xi}) = (\Psi_1(\boldsymbol{\Xi}), \dots, \Psi_N(\boldsymbol{\Xi}))$, Eq. (26.24) can be rewritten as

$$\boldsymbol{\eta} = [z]^T \boldsymbol{\Psi}(\boldsymbol{\Xi}). \quad (26.27)$$

Equation (26.25) means that $[z]$ belongs to the compact Stiefel manifold

$$\mathbb{V}_m(\mathbb{R}^N) = \{[z] \in \mathbb{M}_{N,m}(\mathbb{R}); [z]^T [z] = [I_m]\}. \quad (26.28)$$

7.4 Parameterization of Compact Stiefel Manifold $\mathbb{V}_m(\mathbb{R}^N)$

A parametrization of $\mathbb{V}_m(\mathbb{R}^N)$ defined by Eq. (26.28) is given hereinafter. For all $[z_0]$ fixed in $\mathbb{V}_m(\mathbb{R}^N)$, let $T_{[z_0]}$ be the tangent vector space to $\mathbb{V}_m(\mathbb{R}^N)$ at $[z_0]$. The objective is to construct a mapping $[w] \mapsto [z] = \mathcal{R}_{[z_0]}([w])$ from $T_{[z_0]}$ onto $\mathbb{V}_m(\mathbb{R}^N)$ such that $\mathcal{R}_{[z_0]}([0]) = [z_0]$ and such that, if $[w]$ belongs to a subset of $T_{[z_0]}$, this subset being centered in $[w] = [0]$ and having a sufficiently small diameter, then $[z] = \mathcal{R}_{[z_0]}([w])$ belongs to a subset of $\mathbb{V}_m(\mathbb{R}^N)$, approximatively centered in $[z] = [z_0]$. There are several possibilities for constructing such a parameterization (see, for instance, [1, 19]). For instance, a parameterization can be constructed as described in [48] using the geometry of algorithms with orthogonality constraints [19]. Hereinafter, we present the construction proposed in [1] for which the algorithm has a small complexity with respect to the other possible possibilities. Let us assume that $N > m$ that is generally the case. For $[z_0]$ fixed in $\mathbb{V}_m(\mathbb{R}^N)$, the mapping $\mathcal{R}_{[z_0]}$ is defined by

$$[z] = \mathcal{R}_{[z_0]}([w]) := \text{qr}([z_0] + \sigma [w]), \quad [w] \in T_{[z_0]}, \quad (26.29)$$

in which qr is the mapping that corresponds to the QR economy-size decomposition of matrix $[z_0] + \sigma [w]$, for which only the first m columns of matrix $[q]$ such that $[z_0] + \sigma [w] = [q] [r]$ are computed and such that $[z]^T [z] = [I_m]$. In Eq. (26.29), σ allows the diameter of the subset of $T_{[z_0]}$ centered in $[0]$ to be controlled.

7.5 Parameterized Representation for Non-Gaussian Random Field $[\mathbf{K}]$

Let $\{[\mathbf{G}^{(m,N,N_g)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ be defined by Eqs. (26.23) and (26.24), and let \mathcal{G} be defined by Eq. (26.19) for the *exponential-type representation* and by Eq. (26.21) for the *square-type representation*. The corresponding parameterized representation for non-Gaussian positive-definite matrix-valued random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is denoted by $\{[\mathbf{K}^{(m,N,N_g)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ and is rewritten, for all \mathbf{x} in Ω , as

$$[\mathbf{K}^{(m,N,N_g)}(\mathbf{x})] = \mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\Xi}, [z]), \quad (26.30)$$

in which $(\mathbf{x}, \boldsymbol{\xi}, [z]) \mapsto \mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\xi}, [z])$ is a deterministic mapping defined on $\Omega \times \mathbb{R}^{N_g} \times \mathbb{V}_m(\mathbb{R}^N)$ with values in $\mathbb{M}_n^+(\mathbb{R})$ such that

$$\mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\xi}, [z]) = \mathcal{G}([G_0(\mathbf{x})] + \sum_{i=1}^m \sqrt{\lambda_i} [G_i(\mathbf{x})] \{[z]^T \Psi(\boldsymbol{\xi})\}_i). \quad (26.31)$$

7.6 Parametric Model-Based Representation of Random Observation Model \mathbf{U}

From Eqs. (26.9) and (26.30), the parametric model-based representation of random model observation \mathbf{U} with values in \mathbb{R}^{N_U} , corresponding to the representation $\{[\mathbf{K}^{(m,N,N_g)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$, is denoted by $\mathbf{U}^{(m,N,N_g)}$ and is written as

$$\mathbf{U}^{(m,N,N_g)} = \mathcal{B}^{(m,N,N_g)}(\boldsymbol{\Xi}, [z]), \quad (26.32)$$

in which $(\xi, [z]) \mapsto \mathcal{B}^{(m,N,N_g)}(\xi, [z])$ is a deterministic mapping defined on $\mathbb{R}^{N_g} \times \mathbb{V}_m(\mathbb{R}^N)$ with values in \mathbb{R}^{N_U} such that

$$\mathcal{B}^{(m,N,N_g)}(\xi, [z]) = \mathbf{h}(\mathcal{K}^{(m,N,N_g)}(\mathbf{x}^1, \xi, [z]), \dots, \mathcal{K}^{(m,N,N_g)}(\mathbf{x}^{N_p}, \xi, [z])). \quad (26.33)$$

For N_p fixed, the sequence $\{\mathbf{U}^{(m,N,N_g)}\}_{m,N,N_g}$ of \mathbb{R}^{N_U} -valued random variables converge to \mathbf{U} in \mathcal{L}_{N_U} ².

8 Methodology for Solving the Statistical Inverse Problem in High Stochastic Dimension

A general methodology is presented for solving the statistical inverse problem defined in the previous section entitled Sect. 6. The steps of the identification procedure are defined hereinafter.

8.1 Step 1: Introduction of a Family $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x}; \mathbf{s})]; \mathbf{x} \in \Omega\}$ of Algebraic Prior Stochastic Models (APSM) for Non-Gaussian Random Field $[\mathbf{K}]$

The first step consists in introducing a family $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x}; \mathbf{s})], \mathbf{x} \in \Omega\}$ of algebraic prior stochastic models (APSM) for the non-Gaussian second-order random field $[\mathbf{K}]$, defined on $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by Ω , with values in $\mathbb{M}_n^+(\mathbb{R})$, which has been introduced in the previous paragraph entitled Sect. 6.1. This family depends on an unknown hyperparameter \mathbf{s} belonging to an admissible set \mathcal{C}_s that is a subset of \mathbb{R}^{N_s} , for which the dimension, N_s , is assumed to be relatively small, while the stochastic dimension of $[\mathbf{K}^{\text{APSM}}]$ is high. For instance, \mathbf{s} can be made up of the mean function, a matrix-valued lower bound, some spatial-correlation lengths, some parameters controlling the statistical fluctuations and the shape of the tensor-valued correlation function. For \mathbf{s} fixed in \mathcal{C}_s , the probability distribution (i.e., the system of marginal probability distributions) of random field $[\mathbf{K}^{\text{APSM}}]$ and the corresponding generator of independent realizations are assumed to have been constructed and, consequently, are assumed to be known.

An example of such a construction is explicitly given in the next section entitled Sect. 9.

As it has been explained in the previous paragraph entitled Sect. 3.7 of Sect. 3, Step 1 is a fundamental step of the methodology. The real capability to correctly solve the statistical inverse problem in high stochastic dimension is directly related to the pertinence and to the quality of the constructed APSM that allows for enriching the information in order to overcome the lack of experimental data (only partial experimental data are assumed to be available). Such a construction must be carried out using the MaxEnt principle of Information Theory, under the constraints defined by the available information such as the symmetries, the positiveness, the invertibility in mean square, the boundedness, the capability of the APSM to exhibit simultaneously anisotropic statistical fluctuations and some statistical fluctuations in a given symmetry class such as isotropic, cubic, transversely isotropic, orthotropic, etc. In addition, the corresponding generators of realizations must be developed. For the MaxEnt principle and the construction of generators, we refer the reader to ► Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” section in part II of the present Handbook on Uncertainty Quantification.

8.2 Step 2: Identification of an Optimal Algebraic Prior Stochastic Model (OAPSM) for Non-Gaussian Random Field [K]

The second step consists in identifying an optimal value \mathbf{s}^{opt} in \mathcal{C}_s of hyperparameter \mathbf{s} using experimental data sets $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},v_{\text{exp}}}$ relative to the random model observation \mathbf{U} of the stochastic computational model, which is written, taking into account Eq. (26.9), as

$$\mathbf{U} = \mathbf{h}([\mathbf{K}^{\text{APSM}}(\mathbf{x}^1; \mathbf{s})], \dots, [\mathbf{K}^{\text{APSM}}(\mathbf{x}^{N_p}; \mathbf{s})]), \quad (26.34)$$

The calculation of \mathbf{s}^{opt} in \mathcal{C}_s can be carried out by using the maximum likelihood method:

$$\mathbf{s}^{\text{opt}} = \arg \max_{\mathbf{s} \in \mathcal{C}_s} \sum_{\ell=1}^{v_{\text{exp}}} \log p_{\mathbf{U}}(\mathbf{u}^{\text{exp},\ell}; \mathbf{s}), \quad (26.35)$$

in which $p_{\mathbf{U}}(\mathbf{u}^{\text{exp},\ell}; \mathbf{s})$ is the value, in $\mathbf{u} = \mathbf{u}^{\text{exp},\ell}$, of the probability density function $p_{\mathbf{U}}(\mathbf{u}; \mathbf{s})$ of the random vector \mathbf{U} defined by Eq. (26.34) and depending on \mathbf{s} . The optimal algebraic prior model $\{[\mathbf{K}^{\text{OAPSM}}(\mathbf{x})], \mathbf{x} \in \Omega\} := \{[\mathbf{K}^{\text{APSM}}(\mathbf{x}; \mathbf{s}^{\text{opt}})], \mathbf{x} \in \Omega\}$ is then obtained. Using the generator of realizations of the optimal APSM, v_{KL} independent realizations $[K^{(1)}], \dots, [K^{(v_{\text{KL}})}]$ can be computed such that, for $\ell = 1, \dots, v_{\text{KL}}$ and $\theta_\ell \in \Theta$, the deterministic field $[K^{(\ell)}] := \{[K^{(\ell)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is such that

$$[K^{(\ell)}] = \{[\mathbf{K}^{\text{OAPSM}}(\mathbf{x}; \theta_\ell)], \mathbf{x} \in \Omega\}. \quad (26.36)$$

These realizations can be generated at points $\mathbf{x}^1, \dots, \mathbf{x}^{N_p}$ (or at any other points), with v_{KL} as large as it is desired without inducing a significant computational cost.

8.3 Step 3: Choice of an Adapted Representation for Non-Gaussian Random Field [K] and Optimal Algebraic Prior Stochastic Model for Non-Gaussian Random Field [G]

For a fixed choice of the type of representation for random field [K] given by Eq. (26.19) (exponential type) or Eq. (26.21) (square type), the corresponding optimal algebraic prior model $\{[G^{OAPSM}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[G(\mathbf{x})], \mathbf{x} \in \Omega\}$ is written as

$$[G^{OAPSM}(\mathbf{x})] = \mathcal{G}^{-1}([K^{OAPSM}(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad (26.37)$$

in which \mathcal{G}^{-1} is defined by Eq. (26.20) (exponential type) or by Eq. (26.22) (square type). It is assumed that random field $[G^{OAPSM}]$ is a second-order random field. From the v_{KL} independent realizations $[K^{(1)}], \dots, [K^{(v_{KL})}]$ of random field $[K^{OAPSM}]$ (see Eq. (26.36)), it can be deduced the v_{KL} independent realizations $[G^{(1)}], \dots, [G^{(v_{KL})}]$ of random field $[G^{OAPSM}]$ such that,

$$[G^{(\ell)}(\mathbf{x})] = \mathcal{G}^{-1}([K^{(\ell)}(\mathbf{x})]), \quad \forall \mathbf{x} \in \Omega, \quad \ell = 1, \dots, v_{KL}. \quad (26.38)$$

8.4 Step 4: Construction of a Truncated Reduced Representation of Second-Order Random Field $[G^{OAPSM}]$

The v_{KL} independent realizations $[G^{(1)}], \dots, [G^{(v_{KL})}]$ of random field $[G^{OAPSM}]$ (computed with Eq. (26.38)) are used to calculate, for random field $[G^{OAPSM}]$, an estimation, $[G_0]$, of the mean function and an estimation, $\text{Cov}_{G^{OAPSM}}$, of the covariance operator whose kernel is the tensor-valued cross covariance function $C_{G^{OAPSM}}(\mathbf{x}, \mathbf{x}')$ that is assumed to be square integrable on $\Omega \times \Omega$. The first m eigenvalues $\lambda_1 \geq \dots \geq \lambda_m$ and the corresponding orthonormal eigenfunctions $[G_1], \dots, [G_m]$ of covariance operator $\text{Cov}_{G^{OAPSM}}$ are then computed. For a given convergence tolerance, the optimal value of m is calculated, and the truncated reduced representation $\{[G^{OAPSM(m)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of the second-order random field $\{[G^{OAPSM}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is written (see Eq. (26.23)) as

$$[G^{OAPSM(m)}(\mathbf{x})] = [G_0(\mathbf{x})] + \sum_{i=1}^m \sqrt{\lambda_i} [G_i(\mathbf{x})] \eta_i^{OAPSM}, \quad \forall \mathbf{x} \in \Omega. \quad (26.39)$$

Using the v_{KL} independent realizations $[G^{(1)}], \dots, [G^{(v_{KL})}]$ of random field $[G^{OAPSM}]$ calculated with Eq. (26.38), v_{KL} independent realizations $\eta^{(1)}, \dots, \eta^{(v_{KL})}$ of the

random vector $\boldsymbol{\eta}^{\text{OAPSM}} = (\eta_1^{\text{OAPSM}}, \dots, \eta_m^{\text{OAPSM}})$ are calculated, for $i = 1, \dots, m$ and for $\ell = 1, \dots, v_{\text{KL}}$, by

$$\eta_i^{(\ell)} = \frac{1}{\sqrt{\lambda_i}} \int_{\Omega} \ll [G^{(\ell)}(\mathbf{x})] - [G_0(\mathbf{x})], [G_i(\mathbf{x})] \gg d\mathbf{x}. \quad (26.40)$$

8.5 Step 5: Construction of a Truncated Polynomial Chaos Expansion of $\boldsymbol{\eta}^{\text{OAPSM}}$ and Representation of Random Field $[\mathbf{K}^{\text{OAPSM}}]$

Using independent realizations $\boldsymbol{\eta}^{(1)}, \dots, \boldsymbol{\eta}^{(v_{\text{KL}})}$ of random vector $\boldsymbol{\eta}^{\text{OAPSM}}$ (see Eq. (26.40)), this step consists in constructing the approximation $\boldsymbol{\eta}^{\text{chaos}}(N_d, N_g) = (\eta_1^{\text{chaos}}(N_d, N_g), \dots, \eta_m^{\text{chaos}}(N_d, N_g))$ of $\boldsymbol{\eta}^{\text{OAPSM}}$ using Eq. (26.27), for which the matrix $[z]$ in $\mathbb{M}_{N,m}(\mathbb{R})$ of the coefficients verifies $[z]^T [z] = [I_m]$,

$$\boldsymbol{\eta}^{\text{OAPSM}} \simeq \boldsymbol{\eta}^{\text{chaos}}(N_d, N_g), \quad \boldsymbol{\eta}^{\text{chaos}}(N_d, N_g) = [z]^T \Psi(\boldsymbol{\Xi}), \quad (26.41)$$

in which the integer N is defined by

$$N = \mathbb{h}(N_d, N_g) := (N_d + N_g)! / (N_d! N_g!) - 1, \quad (26.42)$$

where the integer N_d is the maximum degree of the normalized multivariate Hermite polynomials and N_g the dimension of random vector $\boldsymbol{\Xi}$. In Eq. (26.41), the symbol “ \simeq ” means that the mean-square convergence is reached for N_d and N_g (with $N_g \leq m$) sufficiently large.

■ Identification of an optimal value $[z_0(N_d, N_g)]$ of $[z]$ for a fixed value of N_d and N_g .

For a fixed value of N_d and N_g such that $N_d \geq 1$ and $1 \leq N_g \leq m$, the identification of $[z]$ is performed using the maximum likelihood method. The log-likelihood function is written as

$$\mathcal{L}([z]) = \sum_{\ell=1}^{v_{\text{KL}}} \log p_{\boldsymbol{\eta}^{\text{chaos}}(N_d, N_g)}(\boldsymbol{\eta}^{(\ell)}; [z]), \quad (26.43)$$

and the optimal value $[z_0(N_d, N_g)]$ of $[z]$ is given by

$$[z_0(N_d, N_g)] = \arg \max_{[z] \in \mathbb{V}_m(\mathbb{R}^N)} \mathcal{L}([z]), \quad (26.44)$$

in which $\mathbb{V}_m(\mathbb{R}^N)$ is defined by Eq. (26.28).

- (i) For $[z]$ fixed in $\mathbb{V}_m(\mathbb{R}^N)$, the probability density function $\mathbf{e} \mapsto p_{\boldsymbol{\eta}^{\text{chaos}}(N_d, N_g)}(\mathbf{e}; [z])$ of random variable $\boldsymbol{\eta}^{\text{chaos}}(N_d, N_g)$ is estimated by the multidimensional

kernel density estimation method using v_{chaos} independent realizations $\boldsymbol{\eta}^{\text{chaos}(1)}, \dots, \boldsymbol{\eta}^{\text{chaos}(v_{\text{chaos}})}$ of random vector $\boldsymbol{\eta}^{\text{chaos}}(N_d, N_g)$, which is such that $\boldsymbol{\eta}^{\text{chaos}(\ell)} = [z]^T \boldsymbol{\Psi}(\boldsymbol{\Xi}^{(\ell)})$ in which $\boldsymbol{\Xi}^{(1)}, \dots, \boldsymbol{\Xi}^{(v_{\text{chaos}})}$ are v_{chaos} independent realizations of $\boldsymbol{\Xi}$.

- (ii) For the high-dimension case, i.e., for $m \times N$ very large, the optimization problem defined by Eq. (26.44) must be solved with adapted and robust algorithms:

- The first one is required for generating the independent realizations $\boldsymbol{\Psi}_j(\boldsymbol{\Xi}^{(\ell)})$ of $\boldsymbol{\Psi}_j(\boldsymbol{\Xi})$ in preserving the orthogonality condition for any high values of N_g and N_d . An efficient algorithm is presented hereinafter.
- The second one requires an advanced algorithm to optimize the trials for solving the high-dimension optimization problem defined by Eq. (26.44), the constraint $[z]^T[z] = [I_m]$ being automatically and exactly satisfied as described in [61].

■ *Efficient algorithm for generating realizations of the multivariate polynomial chaos in high dimension and for an arbitrary probability measure.*

Let $\boldsymbol{\Psi}(\boldsymbol{\Xi}) = (\boldsymbol{\Psi}_1(\boldsymbol{\Xi}), \dots, \boldsymbol{\Psi}_N(\boldsymbol{\Xi}))$ be the \mathbb{R}^N -valued random vector in which $\{\boldsymbol{\Psi}_j(\boldsymbol{\Xi})\}_{j=1}^N$ are the normalized multivariate Hermite polynomials. The objective is to compute the $(N \times v_{\text{chaos}})$ real matrix $[\boldsymbol{\Psi}] = [\boldsymbol{\Psi}(\boldsymbol{\Xi}^{(1)}) \dots \boldsymbol{\Psi}(\boldsymbol{\Xi}^{(v_{\text{chaos}})})]$,

$$[\boldsymbol{\Psi}] = \begin{bmatrix} \boldsymbol{\Psi}_1(\boldsymbol{\Xi}^{(1)}) & \dots & \boldsymbol{\Psi}_1(\boldsymbol{\Xi}^{(v_{\text{chaos}})}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Psi}_N(\boldsymbol{\Xi}^{(1)}) & \dots & \boldsymbol{\Psi}_N(\boldsymbol{\Xi}^{(v_{\text{chaos}})}) \end{bmatrix}, \quad (26.45)$$

of the v_{chaos} independent realizations $\boldsymbol{\Psi}(\boldsymbol{\Xi}^{(1)}), \dots, \boldsymbol{\Psi}(\boldsymbol{\Xi}^{(v_{\text{chaos}})})$, in preserving the orthogonality properties

$$\lim_{v_{\text{chaos}} \rightarrow +\infty} \frac{1}{v_{\text{chaos}}} [\boldsymbol{\Psi}] [\boldsymbol{\Psi}]^T = [I_N]. \quad (26.46)$$

It should be noted that the algorithm, which is used for the Gaussian chaos $\boldsymbol{\Psi}_j(\boldsymbol{\Xi}) = \Phi_{\alpha_1}(\boldsymbol{\Xi}_1) \times \dots \times \Phi_{\alpha_{N_g}}(\boldsymbol{\Xi}_{N_g})$ for $j = 1, \dots, N$, can also be used for an arbitrary non-separable probability distribution $p_{\boldsymbol{\Xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}$ on \mathbb{R}^{N_g} without any modification, but in such a case, the multivariate polynomials $\{\boldsymbol{\Psi}_j(\boldsymbol{\Xi})\}_{j=1}^N$, which verify the orthogonality property, $E\{\boldsymbol{\Psi}_j(\boldsymbol{\Xi}) \boldsymbol{\Psi}_{j'}(\boldsymbol{\Xi})\} = \int_{\mathbb{R}^{N_g}} \boldsymbol{\Psi}_j(\boldsymbol{\xi}) \boldsymbol{\Psi}_{j'}(\boldsymbol{\xi}) p_{\boldsymbol{\Xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \delta_{jj'}$, are not written as a tensorial product of univariate polynomials (we have not $\boldsymbol{\Psi}_j(\boldsymbol{\Xi}) = \Phi_{\alpha_1}(\boldsymbol{\Xi}_1) \times \dots \times \Phi_{\alpha_{N_g}}(\boldsymbol{\Xi}_{N_g})$). It has been proved in [65] that, for the usual probability measure, the use of the explicit algebraic formula (constructed with a symbolic Toolbox) or the use of the computational recurrence relation with respect to the degree, induces important numerical noise, and the orthogonality property is lost. In addition, if a global orthogonalization was done to correct this loss of orthogonality, then the independence of the realizations would be lost. A robust computational method has been proposed in [49, 65] to preserve the orthogonality

properties and the independence of the realizations. The two main steps are the following.

- (i) Using a generator of independent realizations of Ξ whose probability distribution is $p_{\Xi}(\xi) d\xi$, the realizations $\mathcal{M}_j(\Xi^{(1)}), \dots, \mathcal{M}_j(\Xi^{(v_{\text{chaos}})})$ of the multivariate monomials $\mathcal{M}_j(\Xi) = \Xi_1^{j_1} \times \dots \times \Xi_{N_g}^{j_{N_g}}$ are computed, in which $j = 1, \dots, N$ is the index associated with the multi-index (j_1, \dots, j_{N_g}) . Let $\mathcal{M}(\Xi) = (\mathcal{M}_1(\Xi), \dots, \mathcal{M}_N(\Xi))$ be the \mathbb{R}^N -valued random variable and let $[M]$ be the $(N \times v_{\text{chaos}})$ real matrix such that

$$[M] = [\mathcal{M}(\Xi^{(1)}) \dots \mathcal{M}(\Xi^{(v_{\text{chaos}})})] = \begin{bmatrix} \mathcal{M}_1(\Xi^{(1)}) & \dots & \mathcal{M}_1(\Xi^{(v_{\text{chaos}})}) \\ \vdots & \ddots & \vdots \\ \mathcal{M}_N(\Xi^{(1)}) & \dots & \mathcal{M}_N(\Xi^{(v_{\text{chaos}})}) \end{bmatrix}. \quad (26.47)$$

- (ii) An orthogonalization of the realizations of the multivariate monomials is carried out using an algorithm (that is different from the Gram-Schmidt orthogonalization algorithm, which is not stable in high dimension) based on the fact that:
(a) the matrix $[\Psi]$, defined by Eq. (26.45), can be written as $[\Psi] = [A][M]$ in which $[A]$ is an invertible $(N \times N)$ real matrix and where $[M]$ is defined by Eq. (26.47), and (b) the matrix $[R] = E\{\mathcal{M}(\Xi)\mathcal{M}(\Xi)^T\}$ is written as $[R] = \lim_{v_{\text{chaos}} \rightarrow +\infty} \frac{1}{v_{\text{chaos}}} [M][M]^T = [A]^{-1}[A]^{-T}$. The algorithm is summarized as follows:

- Computing matrix $[M]$ and then $[R] \simeq \frac{1}{v_{\text{chaos}}} [M][M]^T$ for v_{chaos} sufficiently high.
- Computing $[A]^{-T}$ that corresponds to the Cholesky decomposition of $[R]$.
- Computing the lower triangular matrix $[A]$.
- Computing $[\Psi] = [A][M]$.

■ *Identification of truncation parameters N_d and N_g .*

The quantification of the mean-square convergence of $\eta^{\text{chaos}}(N_d, N_g) = [z_0(N_d, N_g)]^T \Psi(\Xi)$ toward η^{OAPSM} with respect to N_d and N_g , in which $[z_0(N_d, N_g)]$ is given by Eq. (26.44), is carried out using the L^1 -log error function introduced in [61], which allows for measuring the errors of the small values of the probability density function (the tails of the pdf).

- (i) For a fixed value of $N_d \leq m$ and N_g , and for $i = 1, \dots, m$:
- Let $e \mapsto p_{\eta_i^{\text{OAPSM}}}(e)$ be the pdf of random variable η_i^{OAPSM} , which is estimated with the one-dimensional kernel density estimation method using the independent realizations $\eta^{(1)}, \dots, \eta^{(v_{\text{KL}})}$ of the random vector η^{OAPSM} .
 - Let $e \mapsto p_{\eta_i^{\text{chaos}}(N_d, N_g)}(e; [z_0(N_d, N_g)])$ be the pdf of random variable $\eta_i^{\text{chaos}}(N_d, N_g)$, which is estimated with the one-dimensional kernel density estimation method using v_{chaos} independent realizations,

$\eta^{\text{chaos}^{(1)}}(N_d, N_g), \dots, \eta^{\text{chaos}^{(v_{\text{chaos}})}}(N_d, N_g)$, of random vector $\eta^{\text{chaos}}(N_d, N_g)$, which are such that $\eta^{\text{chaos}^{(\ell)}}(N_d, N_g) = [z_0(N_d, N_g)]^T \Psi(\Xi^{(\ell)})$ in which $\Xi^{(1)}, \dots, \Xi^{(v_{\text{chaos}})}$ are v_{chaos} independent realizations of Ξ .

- The L^1 -log error is introduced as described in [61]:

$$\text{err}_i(N_d, N_g) = \int_{\text{BI}_i} |\log_{10} p_{\eta_i^{\text{OAPSM}}}(e) - \log_{10} p_{\eta_i^{\text{chaos}}(N_d, N_g)}(e; [z_0(N_d, N_g)])| de, \quad (26.48)$$

in which BI_i is a bounded interval of the real line, which is defined as the support of the one-dimensional kernel density estimator of random variable η_i^{OAPSM} and which is then adapted to independent realizations $\eta^{(1)}, \dots, \eta^{(v_{\text{KL}})}$ of η^{OAPSM} .

- (ii) For random vector $\eta^{\text{chaos}}(N_d, N_g)$, the L^1 -log error function is denoted by $\text{err}(N_d, N_g)$ and is defined by

$$\text{err}(N_g, N_d) = \frac{1}{m} \sum_{i=1}^m \text{err}_i(N_d, N_g). \quad (26.49)$$

- (iii) The optimal values N_d^{opt} and N_g^{opt} of the truncation parameters N_d and N_g are determined for minimizing the error function $\text{err}(N_d, N_g)$ in taking into account the admissible set for the values of N_d and N_g as described in [49]. Let \mathcal{C}_{N_d, N_g} be the admissible set for the values of N_d and N_g , which is defined by

$$\mathcal{C}_{N_d, N_g} = \{(N_d, N_g) \in \mathbb{N}^2 \mid N_g \leq m, (N_d + N_g)! / (N_d! N_g!) - 1 \geq m\}.$$

It should be noted the more the values of N_d and N_g are high, the bigger is the matrix $[z_0(N_d, N_g)]$, and thus, the more difficult it is to perform the numerical identification. Rather than directly minimizing error function $\text{err}(N_d, N_g)$, it is more accurate to search for the optimal values of N_d and N_g that minimize the dimension of the projection basis, $(N_d + N_g)! / (N_d! N_g!)$. For a given error threshold ε , we then introduce the admissible set \mathcal{C}_ε such that

$$\mathcal{C}_\varepsilon = \{(N_d, N_g) \in \mathcal{C}_{N_d, N_g} \mid \text{err}(N_g, N_d) \leq \varepsilon\},$$

and the optimal values N_d^{opt} and N_g^{opt} are given as the solution of the optimization problem,

$$(N_d^{\text{opt}}, N_g^{\text{opt}}) = \arg \min_{(N_d, N_g) \in \mathcal{C}_\varepsilon} (N_d + N_g)! / (N_d! N_g!), \quad N^{\text{opt}} = h(N_d^{\text{opt}}, N_g^{\text{opt}}).$$

■ *Changing the notation.*

Until the end of Step 5 and in Steps 6 and 7, in order to simplify the notations, N_d^{opt} , N_g^{opt} , N^{opt} , and $[z_0(N_d^{\text{opt}}, N_g^{\text{opt}})]$ are simply rewritten as N_d , N_g , N , and $[z_0]$.

■ *Representation of random field $[\mathbf{K}^{\text{OAPSM}}]$.*

It can then be deduced that the optimal representation $\{[\mathbf{K}^{\text{OAPSM}}(m, N, N_g)(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[\mathbf{K}^{\text{OAPSM}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is written as

$$[\mathbf{K}^{\text{OAPSM}}(m, N, N_g)(\mathbf{x})] = \mathcal{K}^{(m, N, N_g)}(\mathbf{x}, \boldsymbol{\Xi}, [z_0]), \quad \forall \mathbf{x} \in \Omega, \quad (26.50)$$

in which $\mathcal{K}^{(m, N, N_g)}(\mathbf{x}, \boldsymbol{\Xi}, [z_0])$ is defined by Eq. (26.30) with $N_d = N_d^{\text{opt}}$, $N_g = N_g^{\text{opt}}$, and $[z] = [z_0(N_d^{\text{opt}}, N_g^{\text{opt}})]$.

8.6 Step 6: Identification of the Prior Stochastic Model $[\mathbf{K}^{\text{prior}}]$ of $[\mathbf{K}]$ in the General Class of the Non-Gaussian Random Fields

This step consists in identifying the prior stochastic model $\{[\mathbf{K}^{\text{prior}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$, using the maximum likelihood method and the experimental data sets $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},v_{\text{exp}}}$ relative to the random model observation \mathbf{U} of the stochastic computational model (see Eq. (26.9)) and using the parametric model-based representation of random observation model \mathbf{U} (see Eq. (26.32)). We thus have to identify the value $[z^{\text{prior}}]$ in $\mathbb{V}_m(\mathbb{R}^{N^{\text{opt}}})$ of $[z]$ such that

$$[z^{\text{prior}}] = \arg \max_{[z] \in \mathbb{V}_m(\mathbb{R}^N)} \sum_{\ell=1}^{v_{\text{exp}}} \log p_{\mathbf{U}^{(m, N, N_g)}}(\mathbf{u}^{\text{exp},\ell}; [z]), \quad (26.51)$$

in which $p_{\mathbf{U}^{(m, N, N_g)}}(\mathbf{u}^{\text{exp},\ell}; [z])$ is the value, in $\mathbf{u} = \mathbf{u}^{\text{exp},\ell}$, of the pdf $p_{\mathbf{U}^{(m, N, N_g)}}(\mathbf{u}; [z])$ of the random vector $\mathbf{U}^{(m, N, N_g)}$ given (see Eq. (26.32)) by

$$\mathbf{U}^{(m, N, N_g)} = \mathcal{B}^{(m, N, N_g)}(\boldsymbol{\Xi}, [z]), \quad (26.52)$$

where $(\boldsymbol{\xi}, [z]) \mapsto \mathcal{B}^{(m, N, N_g)}(\boldsymbol{\xi}, [z])$ is the deterministic mapping from $\mathbb{R}^{N_g} \times \mathbb{V}_m(\mathbb{R}^N)$ into \mathbb{R}^{N_U} defined by Eq. (26.33) with Eq. (26.31) in which $[G_0(\mathbf{x})]$, λ_i , and $[G_i(\mathbf{x})]$, for $i = 1, \dots, m$, are the quantities computed in Step 4.

- (i) For $[z]$ fixed in $\mathbb{V}_m(\mathbb{R}^N)$, pdf $\mathbf{u} \mapsto p_{\mathbf{U}^{(m, N, N_g)}}(\mathbf{u}; [z])$ of random variable $\mathbf{U}^{(m, N, N_g)}$ is estimated by the multidimensional kernel density estimation method using v_{chaos} independent realizations $\boldsymbol{\Xi}^{(1)}, \dots, \boldsymbol{\Xi}^{(v_{\text{chaos}})}$ of $\boldsymbol{\Xi}$.
- (ii) Let us assume that $N > m$ is generally the case. The parameterization $[z] = \mathcal{R}_{[z_0]}([w])$ defined by Eq. (26.29) is used for exploring, with a random search algorithm, the subset of $\mathbb{V}_m(\mathbb{R}^N)$, centered in $[z_0] := [z_0(N_d, N_g)] \in \mathbb{V}_m(\mathbb{R}^N)$ computed in Step 5. The optimization problem defined by Eq. (26.51) is replaced by $[z^{\text{prior}}] = \mathcal{R}_{[z_0]}([w^{\text{prior}}])$ with

$$[w^{\text{prior}}] = \arg \max_{[w] \in T_{[z_0]}} \sum_{\ell=1}^{v_{\text{exp}}} \log p_{\mathbf{U}^{(m,N,N_g)}}(\mathbf{u}^{\text{exp},\ell}; \mathcal{R}_{[z_0]}([w])). \quad (26.53)$$

For solving the high-dimension optimization problem defined by Eq. (26.53), a random search algorithm is used for which $[w]$ is modeled by a random matrix $[\mathbf{W}] = \text{Proj}_{T_{[z_0]}}([\Lambda])$ with values in $T_{[z_0]}$, which is the projection on $T_{[z_0]}$ of a random matrix $[\Lambda]$ with values in $\mathbb{M}_{N,m}(\mathbb{R})$ whose entries are independent normalized Gaussian real-valued random variables, i.e., $E\{[\Lambda]_{ji}\} = 0$ and $E\{[\Lambda]_{ji}^2\} = 1$. The positive parameter σ introduced in Eq. (26.29) allows for controlling the “diameter” of the subset (centered in $[z_0]$) that is explored by the random search algorithm.

- (iii) The representation of the prior stochastic model $\{[\mathbf{K}^{\text{prior}}(m,N,N_g)(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is given by Eqs. (26.30) and (26.31) that are rewritten as

$$[\mathbf{K}^{\text{prior}}(m,N,N_g)(\mathbf{x})] = \mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\Xi}, [z^{\text{prior}}]), \quad \forall \mathbf{x} \in \Omega, \quad (26.54)$$

in which $[z^{\text{prior}}]$ is given by Eq. (26.51) and where $\mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\xi}, [z^{\text{prior}}])$ is defined by Eq. (26.31) with $[z] = [z^{\text{prior}}]$.

8.7 Step 7: Identification of a Posterior Stochastic Model $[\mathbf{K}^{\text{post}}]$ of $[\mathbf{K}]$

- (i) A posterior stochastic model $\{[\mathbf{K}^{\text{post}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ can be constructed using the Bayesian method. In such a framework, the coefficients $[z]$ of the polynomial chaos expansion $\eta^{\text{chaos}}(N_d, N_g) = [z]^T \boldsymbol{\Psi}(\boldsymbol{\Xi})$ (see Eq. (26.41)) are modeled by a random matrix $[\mathbf{Z}]$ (see [66]) as proposed in [62] and consequently, $[z]$ is modeled by a $\mathbb{V}_m(\mathbb{R}^N)$ -valued random variable $[\mathbf{Z}]$. The prior model $[\mathbf{Z}^{\text{prior}}]$ of $[\mathbf{Z}]$ is chosen as

$$[\mathbf{Z}^{\text{prior}}] = \mathcal{R}_{[z^{\text{prior}}]}([\mathbf{W}^{\text{prior}}]), \quad (26.55)$$

in which $\mathcal{R}_{[z^{\text{prior}}]}$ is the mapping defined by Eq. (26.29), where $[z^{\text{prior}}]$ has been calculated in Step 6 and where $[\mathbf{W}^{\text{prior}}] = \text{Proj}_{T_{[z^{\text{prior}}]}}([\Lambda^{\text{prior}}])$ is a random matrix with values in $T_{[z^{\text{prior}}]}$, which is the projection on $T_{[z^{\text{prior}}]}$ of a random matrix $[\Lambda^{\text{prior}}]$ with values in $\mathbb{M}_{N,m}(\mathbb{R})$ whose entries are independent normalized Gaussian real-valued random variables, i.e. $E\{[\Lambda]_{ji}\} = 0$ and $E\{[\Lambda]_{ji}^2\} = 1$. For a sufficiently small value of σ , the statistical fluctuations of the $\mathbb{V}_m(\mathbb{R}^{N^{\text{opt}}})$ -valued random matrix $[\mathbf{Z}^{\text{prior}}]$ are approximatively centered around $[z^{\text{prior}}]$. The Bayesian update allows the posterior distribution of the random matrix $[\mathbf{W}^{\text{post}}]$ with values in $T_{[z^{\text{prior}}]}$ to be estimated using the stochastic

solution $\mathbf{U}^{(m,N,N_g)} = \mathcal{B}^{(m,N,N_g)}(\boldsymbol{\Xi}, \mathcal{R}_{[\varepsilon^{\text{prior}}]}([\mathbf{W}^{\text{prior}}]))$ and the experimental data set $\mathbf{u}^{\text{exp},1}, \dots, \mathbf{u}^{\text{exp},N_{\text{exp}}}$.

- (ii) The representation of the posterior stochastic model $\{[\mathbf{K}^{\text{post}(m,N,N_g)}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is given by Eqs. (26.30) and (26.31) that are rewritten as

$$[\mathbf{K}^{\text{post}(m,N,N_g)}(\mathbf{x})] = \mathcal{K}^{(m,N,N_g)}(\mathbf{x}, \boldsymbol{\Xi}, \mathcal{R}_{[\varepsilon^{\text{prior}}]}([\mathbf{W}^{\text{post}}])), \quad \forall \mathbf{x} \in \Omega, \quad (26.56)$$

in which $\mathcal{K}^{(m,N^{\text{opt}},N_g^{\text{opt}})}$ is defined by Eq. (26.31).

- (iii) Once the probability distribution of $[\mathbf{W}^{\text{post}}]$ has been estimated by Step 7, ν_{KL} independent realizations can be calculated for the random field $[\mathbf{G}^{\text{post}}(\mathbf{x})] = [G_0(\mathbf{x})] + \sum_{i=1}^m \sqrt{\sigma_i} [G_i(\mathbf{x})] \eta_i^{\text{post}}$ in which $\eta^{\text{post}} = [\mathbf{Z}^{\text{post}}]^T \boldsymbol{\Psi}(\boldsymbol{\Xi})$ and where $[\mathbf{Z}^{\text{post}}] = \mathcal{R}_{[\varepsilon^{\text{prior}}]}([\mathbf{W}^{\text{post}}])$. The identification procedure can then be restarted from Step 4 replacing $[\mathbf{G}^{\text{OAPSM}}]$ by $[\mathbf{G}^{\text{post}}]$.

9 Construction of a Family of Algebraic Prior Stochastic Models

We present an explicit construction of a family $\{[\mathbf{K}^{\text{APS}}(\mathbf{x}; \mathbf{s})], \mathbf{x} \in \Omega\}$ of algebraic prior stochastic models for the non-Gaussian second-order random field $[\mathbf{K}]$ indexed by Ω , with values in $\mathbb{M}_n^+(\mathbb{R})$, which has been introduced in Step 1 of Sect. 8. This family depends on a hyperparameter \mathbf{s} belonging to the admissible set \mathcal{C}_s that is a subset of \mathbb{R}^{N_s} , for which the dimension, N_s , is assumed to be relatively small, while the stochastic dimension of $[\mathbf{K}^{\text{APS}}]$ is high. For \mathbf{s} fixed in \mathcal{C}_s , we give a construction of the random field $[\mathbf{K}^{\text{APS}}]$ and the corresponding generator of its realizations. In order to simplify the notations, \mathbf{s} will be omitted as long as no confusion is possible. The formalism and the results, presented in Sect. 12 of ▶ Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification, are reused. Two prior algebraic stochastic models $[\mathbf{K}^{\text{APS}}]$ are presented hereinafter.

- The first one is the algebraic prior stochastic model $[\mathbf{K}^{\text{APS}}]$ for the non-Gaussian positive-definite matrix-valued random field $[\mathbf{K}]$ that exhibits anisotropic statistical fluctuations (initially introduced in [56, 58]) and for which there is a parameterization with a maximum of $d \times n(n+1)/2$ spatial-correlation lengths and for which a positive-definite lower bound is given [60, 63]. An extension of this model can be found in [32] for the case for which some positive-definite lower and upper bounds are introduced as constraints.
- The second one is the algebraic prior stochastic model $[\mathbf{K}^{\text{APS}}]$ described in [28, 29] for the non-Gaussian positive-definite matrix-valued random field $[\mathbf{K}]$ that exhibits (i) dominant statistical fluctuations in a symmetry class $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$ of dimension N (isotropic, cubic, transversal isotropic, tetragonal,

trigonal, orthotropic, monoclinic) for which there is a parameterization with $d N$ spatial-correlation lengths, (ii) anisotropic statistical fluctuations for which there is a parameterization with a maximum of $d \times n(n+1)/2$ spatial-correlation lengths, and (iii) a positive-definite lower bound.

9.1 General Properties of the Non-Gaussian Random Field $[K]$ with a Lower Bound

Let $\{[K(x)], x \in \Omega\}$ be a non-Gaussian random defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by $\Omega \subset \mathbb{R}^d$ with $1 \leq d \leq 3$, with values in $\mathbb{M}_n^+(\mathbb{R})$ with $n = 6$, homogeneous on \mathbb{R}^d , and of second-order, $E\{\|[K(x)]\|_F^2\} < +\infty$ for all x in Ω . Let $[K] \in \mathbb{M}_n^+(\mathbb{R})$ be its mean value that is independent of x (homogeneous random field) and let $[C_\ell] \in \mathbb{M}_n^+(\mathbb{R})$ be its positive-definite lower bound that is also assumed to be independent of x . For all x in Ω ,

$$[K] = E\{[K(x)]\}, \quad [K(x)] - [C_\ell] > 0 \quad a.s. \quad (26.57)$$

9.2 Algebraic Prior Stochastic Model for the Case of Anisotropic Statistical Fluctuations

We consider the case for which the random field exhibits anisotropic statistical fluctuations.

9.2.1 Introduction of an Adapted Representation

The prior stochastic model $\{[K^{APSM}(x)], x \in \Omega\}$ of the random field $\{[K(x)], x \in \Omega\}$ is defined on $(\Theta, \mathcal{T}, \mathcal{P})$, is indexed by $\Omega \subset \mathbb{R}^d$, is with values in $\mathbb{M}_n^+(\mathbb{R})$, is homogeneous on \mathbb{R}^d , and is a second-order random field that is written as

$$[K^{APSM}(x)] = [C_\ell] + [\underline{C}]^{1/2} [G_0(x)] [\underline{C}]^{1/2}, \quad \forall x \in \Omega, \quad (26.58)$$

where $[\underline{C}]^{1/2}$ is the square root of the matrix $[\underline{C}]$ in $\mathbb{M}_n^+(\mathbb{R})$, independent of x , defined by

$$[\underline{C}] = [K] - [C_\ell] \in \mathbb{M}_n^+(\mathbb{R}). \quad (26.59)$$

In Eq. (26.58), $\{[G_0(x)], x \in \mathbb{R}^d\}$ is a random field defined on $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by \mathbb{R}^d , with values in $\mathbb{M}_n^+(\mathbb{R})$, homogeneous on \mathbb{R}^d , second order such that, for all x in \mathbb{R}^d ,

$$E\{[G_0(x)]\} = [I_n], \quad [G_0(x)] > 0 \quad a.s. \quad (26.60)$$

It can then be deduced that, for all x in Ω ,

$$E\{[K^{APSM}(x)]\} = [K], \quad [K^{APSM}(x)] - [C_\ell] > 0 \quad a.s. \quad (26.61)$$

9.2.2 Construction of Random Field $[\mathbf{G}_0]$ and Its Generator of Realizations

■ *Random fields \mathcal{U}_{jk} as the stochastic germs of the random field $[\mathbf{G}_0]$.*

Random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$ is constructed as a nonlinear transformation of $n(n+1)/2$ independent second-order, centered, homogeneous, Gaussian, and normalized random fields $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{1 \leq j \leq k \leq n}$, defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by \mathbb{R}^d , with values in \mathbb{R} , and named the *stochastic germs* of the non-Gaussian random field $[\mathbf{G}_0]$. We then have

$$E\{\mathcal{U}_{jk}(\mathbf{x})\} = 0, \quad E\{\mathcal{U}_{jk}(\mathbf{x})^2\} = 1. \quad (26.62)$$

Consequently, the random fields $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{1 \leq j \leq k \leq n}$ are completely and uniquely defined by the $n(n+1)/2$ autocorrelation functions $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_d) \mapsto R_{\mathcal{U}_{jk}}(\boldsymbol{\zeta}) = E\{\mathcal{U}_{jk}(\mathbf{x} + \boldsymbol{\zeta})\mathcal{U}_{jk}(\mathbf{x})\}$ from \mathbb{R}^d into \mathbb{R} , such that $R_{\mathcal{U}_{jk}}(0) = 1$. The spatial-correlation lengths $L_1^{jk}, \dots, L_d^{jk}$ of random field $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ are defined by

$$L_\alpha^{jk} = \int_0^{+\infty} |R_{\mathcal{U}_{jk}}(0, \dots, \zeta_\alpha, \dots, 0)| d\zeta_\alpha, \quad \alpha = 1, \dots, d, \quad (26.63)$$

and are generally chosen as parameters for the parameterization.

Example of parameterization for autocorrelation function $R_{\mathcal{U}_{jk}}$. The autocorrelation function (corresponding to a minimal parameterization) is written as

$$R_{\mathcal{U}_{jk}}(\boldsymbol{\zeta}) = \rho_1^{jk}(\zeta_1) \times \dots \times \rho_d^{jk}(\zeta_d), \quad (26.64)$$

in which, for all $\alpha = 1, \dots, d$, $\rho_\alpha^{jk}(0) = 1$, and for all $\zeta_\alpha \neq 0$,

$$\rho_\alpha^{jk}(\zeta_\alpha) = 4(L_\alpha^{jk})^2 / (\pi^2 \zeta_\alpha^2) \sin^2(\pi \zeta_\alpha / (2L_\alpha^{jk})), \quad (26.65)$$

where $L_1^{jk}, \dots, L_d^{jk}$ are positive real numbers. Each random field \mathcal{U}_{jk} is mean-square continuous on \mathbb{R}^d and its power spectral density function defined on \mathbb{R}^d has a compact support, $[-\pi/L_1^{jk}, \pi/L_1^{jk}] \times \dots \times [-\pi/L_d^{jk}, \pi/L_d^{jk}]$. Such a model has $d n(n+1)/2$ real parameters $\{L_1^{jk}, \dots, L_d^{jk}\}_{1 \leq j \leq k \leq n}$ that represent the spatial-correlation lengths of the stochastic germs $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{1 \leq j \leq k \leq n}$, because

$$\int_0^{+\infty} |R_{\mathcal{U}_{jk}}(0, \dots, \zeta_\alpha, \dots, 0)| d\zeta_\alpha = L_\alpha^{jk}. \quad (26.66)$$

■ *Defining an adapted family of functions for the nonlinear transformation.*

Let $\{u \mapsto h(u; a)\}_{a > 0}$ be the adapted family of functions from \mathbb{R} into $]0, +\infty[$, in which a is a positive real number, such that $\Gamma_a = h(\mathcal{U}; a)$ is a gamma random

variable with parameter a , while \mathcal{U} is a normalized Gaussian random variable ($E\{\mathcal{U}\} = 0$ and $E\{\mathcal{U}^2\} = 1$). Consequently, for all u in \mathbb{R} , we have

$$h(u; a) = F_{\Gamma_a}^{-1}(F_{\mathcal{U}}(u)), \quad (26.67)$$

in which $u \mapsto F_{\mathcal{U}}(u) = \int_{-\infty}^u \frac{1}{\sqrt{2\pi}} e^{-v^2/2} dv$ is the cumulative distribution function of the normalized Gaussian random variable \mathcal{U} . The function $p \mapsto F_{\Gamma_a}^{-1}(p)$, from $]0, 1[$ into $]0, +\infty[$, is the reciprocal function of the cumulative distribution function $\gamma \mapsto F_{\Gamma_a}(\gamma) = \int_0^\gamma \frac{1}{\Gamma(a)} t^{a-1} e^{-t} dt$ of the gamma random variable Γ_a with parameter a , in which $\Gamma(a)$ is the gamma function defined by $\Gamma(a) = \int_0^{+\infty} t^{a-1} e^{-t} dt$.

■ Defining the random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$ and its generator of realizations.

For all \mathbf{x} fixed in \mathbb{R}^d , the available information is defined by Eq. (26.60) and by the constraint $|E\{\log(\det[\mathbf{G}_0(\mathbf{x})])\}| < +\infty$, which is introduced in order that the zero matrix be a repulsive value for the random matrix $[\mathbf{G}_0(\mathbf{x})]$. The use of the maximum entropy principle under the constraints defined by this available information leads to taking the random matrix $[\mathbf{G}_0(\mathbf{x})]$ in ensemble SG_0^+ defined in Sect. 8.1 of ►Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification. Taking into account the algebraic representation of any random matrix belonging to ensemble SG_0^+ , the spatial-correlation structure of random field $[\mathbf{G}_0]$ is then introduced in replacing the Gaussian random variables U_{jk} by the Gaussian real-valued random fields $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ defined above, for which the spatial-correlation structure is defined by the spatial-correlation lengths $\{L_\alpha^{jk}\}_{\alpha=1,\dots,d}$. Consequently, the random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$, defined on probability space $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by \mathbb{R}^d , with values in $\mathbb{M}_n^+(\mathbb{R})$, is constructed as follows:

- (i) Let $\{\mathcal{U}_{jk}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{1 \leq j \leq k \leq n}$ be the $n(n+1)/2$ independent random fields introduced above. Consequently, for all \mathbf{x} in \mathbb{R}^d ,

$$E\{\mathcal{U}_{jk}(\mathbf{x})\} = 0, \quad E\{\mathcal{U}_{jk}(\mathbf{x})^2\} = 1, \quad 1 \leq j \leq k \leq n. \quad (26.68)$$

- (ii) Let δ be the real number, independent of \mathbf{x} , such that

$$0 < \delta < \sqrt{(n+1)/(n+5)} < 1. \quad (26.69)$$

The parameter δ allows for controlling the statistical fluctuations (dispersion) of the random field $[\mathbf{G}_0]$.

- (iii) For all \mathbf{x} fixed in \mathbb{R}^d , the random matrix $[\mathbf{G}_0(\mathbf{x})]$ is written as

$$[\mathbf{G}_0(\mathbf{x})] = [\mathbf{L}(\mathbf{x})]^T [\mathbf{L}(\mathbf{x})], \quad (26.70)$$

in which $[\mathbf{L}(\mathbf{x})]$ is the upper $(n \times n)$ real triangular random matrix defined as follows:

- For $1 \leq j \leq k \leq n$, the $n(n+1)/2$ random fields $\{[\mathbf{L}(\mathbf{x})]_{jk}, \mathbf{x} \in \Omega\}$ are independent.
 - For $j < k$, the real-valued random field $\{[\mathbf{L}(\mathbf{x})]_{jk}, \mathbf{x} \in \Omega\}$ is defined by $[\mathbf{L}(\mathbf{x})]_{jk} = \sigma_n \mathcal{U}_{jk}(\mathbf{x})$ in which σ_n is such that $\sigma_n = \delta/\sqrt{n+1}$.
 - For $j = k$, the positive-valued random field $\{[\mathbf{L}(\mathbf{x})]_{jj}, \mathbf{x} \in \Omega\}$ is defined by $[\mathbf{L}(\mathbf{x})]_{jj} = \sigma_n \sqrt{2 h(\mathcal{U}_{jj}(\mathbf{x}), a_j)}$ in which $a_j = (n+1)/(2\delta^2) + (1-j)/2$.
- (iv) The representation of random field $[\mathbf{G}_0]$ defined by Eq.(26.70) allows for computing realizations of the family of dependent random matrices $\{[\mathbf{G}_0(\mathbf{x}^1)], \dots, [\mathbf{G}_0(\mathbf{x}^{N_p})]\}$ in which $\mathbf{x}^1, \dots, \mathbf{x}^{N_p}$ are N_p given points in Ω , which are expressed using the realizations of $\{\mathcal{U}_{jk}(\mathbf{x}^1), \dots, \mathcal{U}_{jk}(\mathbf{x}^{N_p})\}_{1 \leq j \leq k \leq n}$ that are simulated using either the representation adapted to a large value of N_p , or another one adapted to a small or moderate value of N_p (see [58]).

■ *A few basic properties of random field $[\mathbf{G}_0]$.*

The random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \Omega\}$, defined on $(\Theta, \mathcal{T}, \mathcal{P})$, indexed by \mathbb{R}^d , with values in $\mathbb{M}_n^+(\mathbb{R})$, is a homogeneous, second-order, and mean-square continuous random field. For all \mathbf{x} in \mathbb{R}^d ,

$$E\{\|\mathbf{G}_0(\mathbf{x})\|_F^2\} < +\infty, \quad E\{[\mathbf{G}_0(\mathbf{x})]\} = [I_n]. \quad (26.71)$$

It can be proved that the introduced dispersion parameter corresponds to the following definition

$$\delta = \left\{ \frac{1}{n} E\{\|\mathbf{G}_0(\mathbf{x}) - [I_n]\|_F^2\} \right\}^{1/2}, \quad (26.72)$$

which shows that

$$E\{\|\mathbf{G}_0(\mathbf{x})\|_F^2\} = n(\delta^2 + 1), \quad (26.73)$$

in which δ is independent of \mathbf{x} . For all \mathbf{x} fixed in \mathbb{R}^d , the probability density function with respect to the measure $d^S G = 2^{n(n-1)/4} \prod_{1 \leq j \leq k \leq n} dG_{jk}$ of random matrix $[\mathbf{G}_0(\mathbf{x})]$ is independent of \mathbf{x} and is written as

$$P_{[\mathbf{G}_0(\mathbf{x})]}([G]) = \mathbb{1}_{\mathbb{M}_n^+(\mathbb{R})}([G]) \times C_{\mathbf{G}_0} \times (\det[G])^{\frac{(n+1)(1-\delta^2)}{2\delta^2}} \times \exp\left\{-\frac{(n+1)}{2\delta^2} \text{tr}[G]\right\}, \quad (26.74)$$

where $C_{\mathbf{G}_0}$ is the positive constant of normalization. For all \mathbf{x} fixed in \mathbb{R}^d , the random variables $\{[\mathbf{G}_0(\mathbf{x})]_{jk}, 1 \leq j \leq k \leq 6\}$ are mutually dependent. In addition, the system of the marginal probability distributions of random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \Omega\}$

is completely defined and is not Gaussian. There exists a positive constant b_G independent of \mathbf{x} , but depending on δ , such that for all $\mathbf{x} \in \mathbb{R}^d$,

$$E\{\|[\mathbf{G}_0(\mathbf{x})]^{-1}\|^2\} \leq b_G < +\infty. \quad (26.75)$$

Since $[\mathbf{G}_0(\mathbf{x})]$ is a random matrix with values in $\mathbb{M}_n^+(\mathbb{R})$, then $[\mathbf{G}_0(\mathbf{x})]^{-1}$ exists (almost surely). However, since almost sure convergence does not imply mean-square convergence, Eq. (26.75) cannot simply be deduced. Let $\overline{\Omega} = \Omega \cup \partial\Omega$ be the closure of the bounded set Ω . We then have

$$E\left\{\left(\sup_{\mathbf{x} \in \overline{\Omega}} \|[\mathbf{G}_0(\mathbf{x})]^{-1}\|\right)^2\right\} = c_G^2 < +\infty, \quad (26.76)$$

in which sup is the supremum and where $0 < c_G < +\infty$ is a finite positive constant.

9.2.3 Definition of the Hyperparameter s

The hyperparameter $\mathbf{s} \in \mathcal{C}_s \subset \mathbb{R}^{N_s}$ of the algebraic prior stochastic model $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x}; \mathbf{s})], \mathbf{x} \in \Omega\}$ that has been constructed for the anisotropic statistical fluctuations is constituted of:

- the reshaping of $[C_\ell] \in \mathbb{M}_n^+(\mathbb{R})$ (the lower bound) and $[\underline{K}] \in \mathbb{M}_n^+(\mathbb{R})$ (the mean value),
- the $d n(n+1)/2$ positive real numbers, $\{L_1^{jk}, \dots, L_d^{jk}\}_{1 \leq j \leq k \leq n}$ (the spatial-correlation lengths, for the parameterization given in the example) and δ (the dispersion) such that $0 < \delta < \sqrt{(n+1)/(n+5)}$.

9.3 Algebraic Prior Stochastic Model for the Case of Dominant Statistical Fluctuations in a Symmetry Class with Some Anisotropic Statistical Fluctuations

We now consider the case for which the random field exhibits dominant statistical fluctuations in a symmetry class and some anisotropic statistical fluctuations.

9.3.1 Positive-Definite Matrices Belonging to a Symmetry Class

A given symmetry class is defined by a subset $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ of $\mathbb{M}_n^+(\mathbb{R})$ such that any matrix $[M]$ belonging to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ is written as

$$[M] = \sum_{j=1}^N m_j [E_j^{\text{sym}}], \quad \mathbf{m} = (m_1, \dots, m_N) \in \mathcal{C}_{\mathbf{m}} \subset \mathbb{R}^N, \quad [E_j^{\text{sym}}] \in \mathbb{M}_n^S(\mathbb{R}), \quad (26.77)$$

in which $\{[E_j^{\text{sym}}], j = 1, \dots, N\}$ is the matrix basis of $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (Walpole's tensor basis [75] in the framework of the elasticity theory) and where the admissible subset $\mathcal{C}_{\mathbf{m}}$ of \mathbb{R}^N is defined by

$$\mathcal{C}_{\mathbf{m}} = \{\mathbf{m} \in \mathbb{R}^N \mid \sum_{j=1}^N m_j [E_j^{\text{sym}}] \in \mathbb{M}_n^+(\mathbb{R})\}. \quad (26.78)$$

It should be noted that matrices $[E_1^{\text{sym}}], \dots, [E_N^{\text{sym}}]$ are symmetric but are not positive definite. For the usual material symmetry classes, the possible values of N are the following: 2 for isotropic, 3 for cubic, 5 for transversely isotropic, 6 or 7 for tetragonal, 6 or 7 for trigonal, 9 for orthotropic, 13 for monoclinic, and 21 for anisotropic. The following properties are proved (see [34, 75]):

- (i) If $[M]$ and $[M']$ belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$, then for all a and b in \mathbb{R} , $a[M] + b[M'] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R})$, and

$$[M][M'] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}), \quad [M]^{-1} \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}), \quad [M]^{1/2} \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}). \quad (26.79)$$

- (ii) Any matrix $[N]$ belonging to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ can be written as

$$[N] = \exp_{\mathbb{M}}([\mathcal{N}]), \quad [\mathcal{N}] = \sum_{j=1}^N y_j [E_j^{\text{sym}}], \quad \mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N, \quad (26.80)$$

in which $\exp_{\mathbb{M}}$ is the exponential of symmetric real matrices. It should be noted that matrix $[\mathcal{N}]$ is a symmetric real matrix but does not belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (because \mathbf{y} is in \mathbb{R}^N and not in $\mathcal{C}_{\mathbf{m}}$ and therefore, $[\mathcal{N}]$ is not positive definite).

- (iii) From Eqs. (26.77) and (26.80), it can be deduced that

$$\exp_{\mathbb{M}}\left(\sum_{j=1}^N y_j [E_j^{\text{sym}}]\right) = \sum_{j=1}^N m_j(\mathbf{y}) [E_j^{\text{sym}}], \quad \forall \mathbf{y} \in \mathbb{R}^N, \quad (26.81)$$

in which $\mathbf{m}(\mathbf{y}) = (m_1(\mathbf{y}), \dots, m_N(\mathbf{y}))$ belongs to $\mathcal{C}_{\mathbf{m}(\mathbf{y})}$ that is defined by Eq. (26.78). Let $[\mathcal{E}]$ be the matrix in $\mathbb{M}_N^S(\mathbb{R})$ such that $[\mathcal{E}]_{kj} = \ll [E_j^{\text{sym}}], [E_k^{\text{sym}}] \gg$ and let $\mathcal{F}(\mathbf{y}) = (\mathcal{F}_1(\mathbf{y}), \dots, \mathcal{F}_N(\mathbf{y}))$ be the vector in \mathbb{R}^N such that $\mathcal{F}_k(\mathbf{y}) = \ll \exp_{\mathbb{M}}(\sum_{j=1}^N y_j [E_j^{\text{sym}}]), [E_k^{\text{sym}}] \gg$. For all \mathbf{y} fixed in \mathbb{R}^N , $\mathbf{m}(\mathbf{y})$ is the unique solution in $\mathcal{C}_{\mathbf{m}(\mathbf{y})}$ of the linear system,

$$[\mathcal{E}] \mathbf{m}(\mathbf{y}) = \mathcal{F}(\mathbf{y}). \quad (26.82)$$

It should be noted that, in the purely computational framework that is proposed in the previous Sect. 8, an explicit calculation of $\mathcal{F}(\mathbf{y})$ is not required. For each numerical value of vector \mathbf{y} , vector $\mathbf{m}(\mathbf{y})$ is computed by solving the linear equation defined by Eq. (26.82) in which $\mathcal{F}(\mathbf{y})$ is numerically calculated.

9.3.2 Introduction of the Matrices $[\underline{C}]$, $[\underline{S}]$, and $[\underline{A}]$ Related to the Mean Value of the Matrix-Valued Random Field

Let $[\underline{C}]$ be the matrix in $\mathbb{M}_n^+(\mathbb{R})$, independent of \mathbf{x} , representing the mean value of the random matrix $[\mathbf{C}(\mathbf{x})] = [\mathbf{K}(\mathbf{x})] - [C_\ell]$. From Eq. (26.57), it can then be deduced that

$$[\underline{C}] = [\underline{K}] - [C_\ell] \in \mathbb{M}_n^+(\mathbb{R}). \quad (26.83)$$

Let $[\underline{A}]$ be the deterministic matrix in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$, independent of \mathbf{x} , representing the projection of the mean matrix $[\underline{C}]$ on the symmetry class $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$,

$$[\underline{A}] = P^{\text{sym}}([\underline{C}]) \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}), \quad (26.84)$$

in which $[\underline{C}]$ is defined by Eq. (26.83) and where P^{sym} is the projection operator from $\mathbb{M}_n^+(\mathbb{R})$ onto $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$.

- (i) For a random field with values in a given symmetry class with $N < 21$ (there are no anisotropic statistical fluctuations), the matrices $[\underline{K}]$ and $[C_\ell]$ belong to the symmetry class and consequently, $[\underline{C}]$ must belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$, and thus, $[\underline{A}]$ is equal to $[\underline{C}]$.
- (ii) If the class of symmetry is anisotropic (thus, $N = 21$), then $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ coincides with $\mathbb{M}_n^+(\mathbb{R})$ and again, $[\underline{A}]$ is equal to the mean matrix $[\underline{C}]$ that belongs to $\mathbb{M}_n^+(\mathbb{R})$.
- (iii) In general, for a given symmetry class with $N < 21$, and due to the presence of anisotropic statistical fluctuations, the mean value $[\underline{C}]$ of the random matrix $[\mathbf{C}(\mathbf{x})] = [\mathbf{K}(\mathbf{x})] - [C_\ell]$ belongs to $\mathbb{M}_n^+(\mathbb{R})$ but does not belong to $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. For this case, an invertible deterministic $(n \times n)$ real matrix $[\underline{S}]$ is introduced such that

$$[\underline{C}] = [\underline{S}]^T [\underline{A}] [\underline{S}]. \quad (26.85)$$

The construction of $[\underline{S}]$ is performed as follows. Let $[\underline{L}_{\underline{C}}]$ and $[\underline{L}_{\underline{A}}]$ be the upper triangular real matrices with positive diagonal entries resulting from the Cholesky factorization of matrices $[\underline{C}]$ and $[\underline{A}]$,

$$[\underline{C}] = [\underline{L}_{\underline{C}}]^T [\underline{L}_{\underline{C}}], \quad [\underline{A}] = [\underline{L}_{\underline{A}}]^T [\underline{L}_{\underline{A}}]. \quad (26.86)$$

Therefore, the matrix $[\underline{S}]$ is defined by

$$[\underline{S}] = [\underline{L}_{\underline{A}}]^{-1} [\underline{L}_{\underline{C}}]. \quad (26.87)$$

It should be noted that for cases (i) and (ii) above, Eq. (26.85) shows that $[\underline{S}] = [I_n]$.

9.3.3 Introduction of an Adapted Representation for the Random Field

The prior stochastic model $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ of the second-order random field $\{[\mathbf{K}(\mathbf{x})], \mathbf{x} \in \Omega\}$, indexed by $\Omega \subset \mathbb{R}^d$, with values in $\mathbb{M}_n^+(\mathbb{R})$, is written as

$$[\mathbf{K}^{\text{APSM}}(\mathbf{x})] = [C_t] + [\underline{S}]^T [\mathbf{A}(\mathbf{x})]^{1/2} [\mathbf{G}_0(\mathbf{x})] [\mathbf{A}(\mathbf{x})]^{1/2} [\underline{S}], \quad \forall \mathbf{x} \in \Omega. \quad (26.88)$$

in which the deterministic $(n \times n)$ real matrix $[\underline{S}]$ is defined by Eq. (26.85) and where $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \Omega\}$ and $\{[\mathbf{A}(\mathbf{x})], \mathbf{x} \in \Omega\}$ are random fields indexed by \mathbb{R}^d and homogeneous on \mathbb{R}^d . Consequently, the random field $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ that is indexed by Ω is the restriction to $\Omega \subset \mathbb{R}^d$ of a homogeneous random field.

- *Anisotropic statistical fluctuations described by $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$.*

The random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$ models the anisotropic statistical fluctuations. This random field and its generator of realizations are constructed in the previous paragraph *Construction of random field $[\mathbf{G}_0]$ and its generator of realizations of Algebraic prior stochastic model for the case of anisotropic statistical fluctuations* section (see Eq. (26.70)). The random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \Omega\}$ is defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$; is indexed by \mathbb{R}^d , with values in $\mathbb{M}_n^+(\mathbb{R})$; and is non-Gaussian, homogeneous, second-order, and mean-square continuous on \mathbb{R}^d .

- For all \mathbf{x} in \mathbb{R}^d , the mean value of random matrix $[\mathbf{G}_0(\mathbf{x})]$ is matrix $[I_n]$ (see Eq. (26.71)).
- The level of the anisotropic statistical fluctuations is controlled by the dispersion parameter δ (independent of \mathbf{x}) such that $0 < \delta < \sqrt{(n+1)/(n+5)}$ (see Eq. (26.72)).
- The hyperparameter s_{G_0} of random field $[\mathbf{G}_0]$ is constituted of the dispersion parameter δ and of the spatial-correlation lengths $\{L_1^{jk}, \dots, L_d^{jk}\}_{1 \leq j \leq k \leq n}$ that are positive real numbers (see Eq. (26.63)).

- *Statistical fluctuations in the given symmetry class described by $\{[\mathbf{A}(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$.*

The random field $\{[\mathbf{A}(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$ models the statistical fluctuations belonging to the given symmetry class $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. This random field, defined on the probability space $(\Theta', \mathcal{T}', \mathcal{P}')$, is statistically independent of random field $\{[\mathbf{G}_0(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$; is indexed by \mathbb{R}^d , with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$; and is non-Gaussian, homogeneous, second-order, and mean-square continuous on \mathbb{R}^d . In Eq. (26.88), for all \mathbf{x} fixed in \mathbb{R}^d , the random matrix $[\mathbf{A}(\mathbf{x})]^{1/2}$ is the square root of random matrix $[\mathbf{A}(\mathbf{x})]$ and, due to Eq. (26.79), is with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$.

- For all \mathbf{x} in \mathbb{R}^d , the mean value of random matrix $[\mathbf{A}(\mathbf{x})]$ is the matrix $[\underline{A}]$ (independent of \mathbf{x} and defined by Eq. (26.84)) such that

$$E\{[\mathbf{A}(\mathbf{x})]\} = [\underline{A}] \in \mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R}). \quad (26.89)$$

- In order that, for all \mathbf{x} in \mathbb{R}^d , the zero matrix be a repulsive value for random matrix $[\mathbf{A}(\mathbf{x})]$, the following constraint is introduced,

$$E\{\log(\det[\mathbf{A}(\mathbf{x})])\} = c_A, \quad |c_A| < +\infty, \quad (26.90)$$

in which real constant c_A is independent of \mathbf{x} .

- The level of the statistical fluctuations belonging to the given symmetry class is controlled by the dispersion parameter δ_A (independent of \mathbf{x}) defined by

$$\delta_A = \sqrt{\frac{E\{\|\mathbf{A}(\mathbf{x}) - \underline{A}\|_F^2\}}{\|\underline{A}\|_F^2}} = \sqrt{\frac{E\{\|\mathbf{A}(\mathbf{x})\|_F^2\}}{\|\underline{A}\|_F^2} - 1}. \quad (26.91)$$

- Due to the statistical independence of $[\mathbf{A}(\mathbf{x})]$ and $[\mathbf{G}_0(\mathbf{x})]$, taking the mathematical expectation of the two members of Eq. (26.88), and from Eqs. (26.83) and (26.85), it can be deduced that, for all \mathbf{x} in Ω ,

$$E\{[\mathbf{K}^{\text{APSM}}(\mathbf{x})]\} = [\underline{K}], \quad [\mathbf{K}^{\text{APSM}}(\mathbf{x})] - [C_\ell] > 0 \quad a.s. \quad (26.92)$$

9.3.4 Remarks Concerning the Control of the Statistical Fluctuations and the Limit Cases

- Anisotropic statistical fluctuations going to zero ($\delta \rightarrow 0$).

For a given symmetry class with $N < 21$, if the level of anisotropic statistical fluctuations goes to zero, i.e., if $\delta \rightarrow 0$, which implies that, for all \mathbf{x} in \mathbb{R}^d , random matrix $[\mathbf{G}_0(\mathbf{x})]$ goes to $[I_n]$ (in probability distribution) and implies that $[\underline{A}]$ goes to $[\underline{C}]$ and thus $[\underline{S}]$ goes to $[I_n]$, then Eq. (26.86) shows that $[\mathbf{K}^{\text{APSM}}(\mathbf{x})] - [C_\ell]$ goes to $[\mathbf{A}(\mathbf{x})]$ (in probability distribution), which is a random matrix with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$. Consequently, if there are no anisotropic statistical fluctuations ($\delta = 0$), then Eq. (26.88) becomes

$$[\mathbf{K}^{\text{APSM}}(\mathbf{x})] = [C_\ell] + [\mathbf{A}(\mathbf{x})], \quad \forall \mathbf{x} \in \Omega, \quad (26.93)$$

and $\{[\mathbf{K}^{\text{APSM}}(\mathbf{x})], \mathbf{x} \in \Omega\}$ is a random field indexed by Ω with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$.

- Statistical fluctuations in the symmetry class going to zero ($\delta_A \rightarrow 0$).

If the given symmetry class is anisotropic ($N = 21$) and if $\delta_A \rightarrow 0$, then $[\underline{A}]$ goes to the mean matrix $[\underline{C}]$ and $[\underline{S}]$ goes to $[I_n]$, and Eq. (26.88) shows that $[\mathbf{K}^{\text{APSM}}(\mathbf{x})] - [C_\ell]$ goes to $[\underline{C}]^{1/2} [\mathbf{G}_0(\mathbf{x})] [\underline{C}]^{1/2}$ (in probability distribution), which is a random matrix with values in $\mathbb{M}_n^+(\mathbb{R})$. Consequently, if there are no statistical fluctuations in the symmetry class ($\delta_A = 0$), then Eq. (26.86) becomes

$$[\mathbf{K}^{\text{APSM}}(\mathbf{x})] = [C_\ell] + [\underline{C}]^{1/2} [\mathbf{G}_0(\mathbf{x})] [\underline{C}]^{1/2}, \quad \forall \mathbf{x} \in \Omega, \quad (26.94)$$

which is Eq. (26.58).

9.3.5 Parameterization of Random Field $\{[\mathbf{A}(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$

Random field $\{[\mathbf{A}(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$, with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$, is written as

$$[\mathbf{A}(\mathbf{x})] = [\underline{A}]^{1/2} [\mathbf{N}(\mathbf{x})] [\underline{A}]^{1/2}, \quad \forall \mathbf{x} \in \mathbb{R}^d, \quad (26.95)$$

in which $\{[\mathbf{N}(\mathbf{x})], \mathbf{x} \in \mathbb{R}^d\}$ is the random field indexed by \mathbb{R}^d with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$,

$$[\mathbf{N}(\mathbf{x})] = \exp_{\mathbb{M}} \left(\sum_{j=1}^N Y_j(\mathbf{x}) [E_j^{\text{sym}}] \right), \quad \forall \mathbf{x} \in \mathbb{R}^d, \quad (26.96)$$

in which $\exp_{\mathbb{M}}$ denotes the exponential of the symmetric real matrices, where $\mathbf{Y}(\mathbf{x}) = (Y_1(\mathbf{x}), \dots, Y_N(\mathbf{x}))$ and where $\{\mathbf{Y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ is a non-Gaussian random field defined on $(\Theta', \mathcal{T}', \mathcal{P}')$, indexed by \mathbb{R}^d with values in \mathbb{R}^N , homogeneous, second-order, mean-square continuous on \mathbb{R}^d . Using the change of representation defined by Eqs. (26.81) and (26.82), random matrix $[\mathbf{N}(\mathbf{x})]$ defined by Eq. (26.96) can be rewritten as

$$[\mathbf{N}(\mathbf{x})] = \sum_{j=1}^N m_j(\mathbf{Y}(\mathbf{x})) [E_j^{\text{sym}}]. \quad (26.97)$$

- *Remark concerning the set of the values of random matrix $[\mathbf{A}(\mathbf{x})]$.*

For all \mathbf{x} fixed in \mathbb{R}^d , $[\mathbf{N}(\mathbf{x})]$ is a random matrix with values in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (see Eq. (26.96)) and $[\underline{A}]$ is in $\mathbb{M}_n^{\text{sym}}(\mathbb{R})$ (see Eq. (26.89)). From Eqs. (26.79) and (26.95), it can be deduced that random matrix $[\mathbf{A}(\mathbf{x})]$ is in $\mathbb{M}_n^{\text{sym}}(\mathbb{R}) \subset \mathbb{M}_n^+(\mathbb{R})$.

- *Available information for random matrix $[\mathbf{N}(\mathbf{x})]$.*

For all \mathbf{x} fixed in \mathbb{R}^d , substituting the representation of $[\mathbf{A}(\mathbf{x})]$ defined by Eq. (26.95) into Eqs. (26.89) and (26.90) yields the following available information for random matrix $[\mathbf{N}(\mathbf{x})]$,

$$E\{[\mathbf{N}(\mathbf{x})]\} = [I_n], \quad (26.98)$$

$$E\{\log(\det[\mathbf{N}(\mathbf{x})])\} = c_N, \quad |c_N| < +\infty, \quad (26.99)$$

in which real constant c_N is independent of \mathbf{x} .

- *Available information for random matrix $\mathbf{Y}(\mathbf{x})$.*

Substituting the representation of $[\mathbf{N}(\mathbf{x})]$ defined by Eq. (26.96) into the constraint defined by Eq. (26.99) yields the following constraint for $\mathbf{Y}(\mathbf{x})$,

$$E \left\{ \sum_{j=1}^N Y_j(\mathbf{x}) \text{tr}[E_j^{\text{sym}}] \right\} = c_N, \quad |c_N| < +\infty, \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad (26.100)$$

Substituting the representation of $[N(\mathbf{x})]$ defined by Eq. (26.97) into the constraint defined by Eq. (26.98) yields $E\{\sum_{j=1}^N m_j(\mathbf{Y}(\mathbf{x})) [E_j^{\text{sym}}]\} = [I_n]$. Performing the projection of this equation on the basis $\{[E_k^{\text{sym}}], k = 1, \dots, N\}$ yields (similarly to Eq. (26.82)), $[\mathcal{E}] E\{\mathbf{m}(\mathbf{Y}(\mathbf{x}))\} = \mathcal{I}$ in which $\mathcal{I} = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ is the vector in \mathbb{R}^N such that $\mathcal{I}_k = \ll [I_n, [E_k^{\text{sym}}]] \gg$. The constraint on $[N(\mathbf{x})]$ defined by Eq. (26.98) is transferred in the following constraint on $\mathbf{Y}(\mathbf{x})$,

$$E\{\mathbf{m}(\mathbf{Y}(\mathbf{x}))\} = [\mathcal{E}]^{-1} \mathcal{I} \quad \text{on } \mathbb{R}^N, \quad (26.101)$$

The constraints defined by Eqs. (26.100) and (26.101) are globally rewritten as

$$E\{\mathbf{g}(\mathbf{Y}(\mathbf{x}))\} = \mathbf{f} \quad \text{on } \mathbb{R}^{1+N}, \quad (26.102)$$

in which

- $\mathbf{y} \mapsto \mathbf{g}(\mathbf{y}) = (g_1(\mathbf{y}), \dots, g_{1+N}(\mathbf{y}))$ is the mapping from \mathbb{R}^N into \mathbb{R}^{1+N} such that $g_1(\mathbf{y}) = \sum_{j=1}^N y_j \text{tr}[E_j^{\text{sym}}]$ and $g_{1+j}(\mathbf{y}) = m_j(\mathbf{y})$ for $j = 1, \dots, N$.
- $\mathbf{f} = (f_1, \dots, f_{1+N})$ is the vector in \mathbb{R}^{1+N} such that $f_1 = c_N$ and $f_{1+j} = \{[\mathcal{E}]^{-1} \mathcal{I}\}_j$ for $j = 1, \dots, N$.

9.3.6 Construction of the pdf for Random Vector $\mathbf{Y}(\mathbf{x})$ Using the MaxEnt Principle

For all \mathbf{x} fixed in \mathbb{R}^d , the probability density function $\mathbf{y} \mapsto p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y})$ from \mathbb{R}^N into \mathbb{R}^+ of the \mathbb{R}^N -valued random vector $\mathbf{Y}(\mathbf{x})$ is independent of \mathbf{x} (\mathbf{Y} is homogeneous). This pdf is constructed using the maximum entropy principle presented in Sect. 11 of ▶ Chap. 8, “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification, under the constraints defined by the normalization condition $\int_{\mathbb{R}^N} p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y}) d\mathbf{y} = 1$ and by Eq. (26.102). For all \mathbf{y} in \mathbb{R}^N , the pdf is written as

$$p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y}) = c_0(\boldsymbol{\lambda}^{\text{sol}}) \exp(- < \boldsymbol{\lambda}^{\text{sol}}, \mathbf{g}(\mathbf{y}) >), \quad \forall \mathbf{y} \in \mathbb{R}^N, \quad (26.103)$$

in which $c_0(\boldsymbol{\lambda})$ is defined by

$$c_0(\boldsymbol{\lambda}) = \left\{ \int_{\mathbb{R}^N} \exp(- < \boldsymbol{\lambda}, \mathbf{g}(\mathbf{y}) >) d\mathbf{y} \right\}^{-1}, \quad \boldsymbol{\lambda} \in \mathbb{R}^{1+N}, \quad (26.104)$$

where the Lagrange multiplier $\boldsymbol{\lambda}^{\text{sol}} = (\lambda_1^{\text{sol}}, \dots, \lambda_{1+N}^{\text{sol}})$ belongs to an admissible set $\mathcal{C}_{\boldsymbol{\lambda}} \subset \mathbb{R}^{1+N}$ and is calculated for satisfying Eq. (26.102) by using the efficient numerical method presented in Sect. 11.2 with the MCMC generator presented

in Sect. 11.3 of ▶ Chap. 8 “Random Matrix Models and Nonparametric Method for Uncertainty Quantification” in part II of the present Handbook on Uncertainty Quantification.

Remark. In pdf $p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y})$ constructed with Eq. (26.103), the Lagrange multiplier λ^{sol} depends only on one real parameter that is c_N . Such a parameter has no physical meaning and must be expressed as a function, κ , of the coefficient of variation δ_A defined by Eq. (26.91), such that $c_N = \kappa(\delta_A)$. This means that the family of the pdf constructed with Eq. (26.103) is reparameterized as a function of the dispersion parameter δ_A using $c_N = \kappa(\delta_A)$. An explicit expression of function κ cannot be obtained and is constructed numerically in using Eq. (26.91) in which $E\{\|\mathbf{A}(\mathbf{x})\|_F^2\} = \sum_{j=1}^N \sum_{k=1}^N \ll [E_j^{\text{sym}}][A], [A][E_k^{\text{sym}}] \gg \gg \int_{\mathbb{R}^N} m_j(\mathbf{y}) m_k(\mathbf{y}) p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y}) d\mathbf{y}$.

9.3.7 Constructing a Spatial-Correlation Structure for Random Field $\{\mathbf{Y}(\mathbf{x}); \mathbf{x} \in \mathbb{R}^d\}$ and Its Generator

A spatial-correlation structure is introduced as proposed in [28] for the non-Gaussian second-order homogeneous random field $\{\mathbf{Y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ with values in \mathbb{R}^N , for which its first-order marginal probability density function $\mathbf{y} \mapsto p_{\mathbf{Y}(\mathbf{x})}(\mathbf{y})$ (see Eq. (26.103)) is imposed. This pdf is independent of \mathbf{x} and depends on dispersion parameter δ_A . Such a spatial-correlation structure for random field $\{\mathbf{Y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ is transferred to random field $\{\mathbf{A}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$, thanks to the transformation defined by Eqs. (26.95) and (26.96), which is written, for all \mathbf{x} in \mathbb{R}^d , as $[\mathbf{A}(\mathbf{x})] = [A]^{1/2} \exp_{\mathbb{M}}(\sum_{j=1}^N Y_j(\mathbf{x}) [E_j^{\text{sym}}]) [A]^{1/2}$.

■ *Introduction of a Gaussian random field $\{\mathbf{B}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ that defines the spatial-correlation structure.*

- (i) Let $\mathbf{B} = (B_1, \dots, B_N)$ be a random field defined on the probability space $(\Theta', \mathcal{T}', \mathcal{P}')$, indexed by \mathbb{R}^d , with values in \mathbb{R}^N , such that the components B_1, \dots, B_N are N independent real-valued second-order random fields that are Gaussian, homogeneous, centered, normalized, and mean-square continuous. The continuous autocorrelation function $\xi \mapsto [R_{\mathbf{B}}(\xi)] = E\{\mathbf{B}(\mathbf{x} + \xi) \mathbf{B}(\mathbf{x})^T\}$ from \mathbb{R}^d into $\mathbb{M}_N(\mathbb{R})$ is thus diagonal,

$$[R_{\mathbf{B}}(\xi)]_{jk} = \delta_{jk} R_j(\xi), \quad [R_{\mathbf{B}}(\mathbf{0})] = [I_N], \quad (26.105)$$

in which $\xi \mapsto R_j(\xi) = E\{B_j(\mathbf{x} + \xi) B_j(\mathbf{x})\}$, from \mathbb{R}^d into \mathbb{R} , is the autocorrelation function of the centered random field $\{B_j(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$. For all fixed j , since the second-order random field $\{B_j(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ is Gaussian and centered, this random field is completely and uniquely defined by its autocorrelation function $R_j(\xi) = E\{B_j(\mathbf{x} + \xi) B_j(\mathbf{x})\}$ defined for all $\xi = (\xi_1, \dots, \xi_d)$ in \mathbb{R}^d and such that $R_j(0) = 1$. The spatial-correlation lengths $\mathbb{L}_1^j, \dots, \mathbb{L}_d^j$ of random field $\{B_j(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ are defined by

$$\mathbb{L}_\alpha^j = \int_0^{+\infty} |R_j(0, \dots, \zeta_\alpha, \dots, 0)| d\zeta_\alpha.$$

In the parameterization of each autocorrelation function R_j , the parameters $\mathbb{L}_1^j, \dots, \mathbb{L}_d^j$ are generally chosen as hyperparameters.

Example of parameterization for autocorrelation function R_j . A minimal parameterization can be defined as $R_j(\xi) = \rho_1^j(\xi_1) \times \dots \times \rho_d^j(\xi_d)$ in which, for all $\alpha = 1, \dots, d$, $\rho_\alpha^j(0) = 1$ and where, for $\xi_\alpha \neq 0$, $\rho_\alpha^j(\xi_\alpha) = 4(\mathbb{L}_\alpha^j)^2 / (\pi^2 \xi_\alpha^2) \sin^2(\pi \xi_\alpha / (2\mathbb{L}_\alpha^j))$, in which $\mathbb{L}_1^j, \dots, \mathbb{L}_d^j$ are positive real numbers. Each random field B_j is mean-square continuous on \mathbb{R}^d and its power spectral density function defined on \mathbb{R}^d has a compact support, $[-\pi/\mathbb{L}_1^j, \pi/\mathbb{L}_1^j] \times \dots \times [-\pi/\mathbb{L}_d^j, \pi/\mathbb{L}_d^j]$. The parameters, $\mathbb{L}_1^j, \dots, \mathbb{L}_d^j$, represent the spatial-correlation lengths of the stochastic germ $\{B_j(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$.

- (ii) For all countable ordered subsets $0 \leq r_1 < \dots < r_k < r_{k+1} < \dots$ of \mathbb{R}^+ , the sequence of random fields $\{\mathbf{B}^{r_k r_{k+1}}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{k \in \mathbb{N}}$
 - is mutually independent random fields,
 - is such that, $\forall k \in \mathbb{N}$, $\{\mathbf{B}^{r_k r_{k+1}}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ is an independent copy of $\{\mathbf{B}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$, which implies that $E\{\mathbf{B}^{r_k r_{k+1}}(\mathbf{x})\} = E\{\mathbf{B}(\mathbf{x})\} = \mathbf{0}$ and that

$$E\{\mathbf{B}^{r_k r_{k+1}}(\mathbf{x}) (\mathbf{B}^{r_k r_{k+1}}(\mathbf{x}))^T\} = E\{\mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T\} = [R_{\mathbf{B}}(\mathbf{0})] = [I_N]. \quad (26.106)$$

■ Defining an \mathbf{x} -dependent family of normalized Wiener stochastic processes $\{\mathbf{W}_{\mathbf{x}}(r), r \geq 0\}$ containing the spatial-correlation structure.

Let $\{\mathbf{W}_{\mathbf{x}}(r), r \geq 0\}$ be the \mathbf{x} -dependent family of stochastic processes defined on probability space $(\mathcal{O}', \mathcal{T}', \mathcal{P}')$, indexed by $r \geq 0$, with values in \mathbb{R}^N , such that $\mathbf{W}_{\mathbf{x}}(0) = \mathbf{0}$ almost surely and, for all \mathbf{x} fixed \mathbb{R}^d and for all $0 \leq s < r < +\infty$, the increment $\Delta \mathbf{W}_{\mathbf{x}}^{sr} := \mathbf{W}_{\mathbf{x}}(r) - \mathbf{W}_{\mathbf{x}}(s)$ is written as

$$\Delta \mathbf{W}_{\mathbf{x}}^{sr} = \sqrt{r-s} \mathbf{B}^{sr}(\mathbf{x}). \quad (26.107)$$

From the properties of random field $\{\mathbf{B}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ and of the family of random fields $\{\mathbf{B}^{r_k r_{k+1}}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{k \in \mathbb{N}}$ for all countable ordered subsets $0 \leq r_1 < \dots < r_k < r_{k+1} < \dots$, it is deduced that, for all \mathbf{x} fixed in \mathbb{R}^d ,

- (i) the components $W_{\mathbf{x}}^{(1)}, \dots, W_{\mathbf{x}}^{(N)}$ of $\mathbf{W}_{\mathbf{x}}$ are mutually independent real-valued stochastic processes,
- (ii) $\{\mathbf{W}_{\mathbf{x}}(r), r \geq 0\}$ is a stochastic process with independent increments,
- (iii) For all $0 \leq s < r < +\infty$, the increment $\Delta \mathbf{W}_{\mathbf{x}}^{sr} = \mathbf{W}_{\mathbf{x}}(r) - \mathbf{W}_{\mathbf{x}}(s)$ is a \mathbb{R}^N -valued second-order random variable which is Gaussian, centered, and with a covariance matrix that is written as $[C_{\Delta \mathbf{W}_{\mathbf{x}}^{sr}}] = E\{\Delta \mathbf{W}_{\mathbf{x}}^{sr} (\Delta \mathbf{W}_{\mathbf{x}}^{sr})^T\} = (r-s) [I_N]$.
- (iv) Since $\mathbf{W}_{\mathbf{x}}(0) = \mathbf{0}$, and from (i), (ii), and (iii), it can be deduced that $\{\mathbf{W}_{\mathbf{x}}(r), r \geq 0\}$ is a \mathbb{R}^N -valued normalized Wiener process.

■ *Constructing random field $\{\mathbf{Y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ and its generator.*

The construction of random field $\{\mathbf{Y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$ is carried out by introducing a family (indexed by \mathbf{x} in \mathbb{R}^d) of Itô stochastic differential equations (ISDE),

- for which the Wiener process is the family $\{\mathbf{W}_x(r), r \geq 0\}$ that contains the imposed spatial-correlation structure defined by Eq. (26.105),
- that admits the same unique invariant measure (independent of \mathbf{x}), which is defined by the pdf $p_{\mathbf{Y}(\mathbf{x})}$ given by Eqs. (26.103) and (26.104).

Taking into account Eq. (26.103), the potential $\mathbf{u} \mapsto \Phi(\mathbf{u})$, from \mathbb{R}^N into \mathbb{R} , is defined by

$$\Phi(\mathbf{u}) = <\lambda^{\text{sol}}, \mathbf{g}(\mathbf{u})>. \quad (26.108)$$

For all \mathbf{x} fixed in \mathbb{R}^d , let $\{(\mathbf{U}_x(r), \mathbf{V}_x(r)), r \geq 0\}$ be the Markov stochastic process defined on the probability space $(\Theta', \mathcal{T}', \mathcal{P}')$, indexed by $r \geq 0$, with values in $\mathbb{R}^N \times \mathbb{R}^N$, satisfying, for all $r > 0$, the following ISDE,

$$d\mathbf{U}_x(r) = \mathbf{V}_x(r) dr, \quad (26.109)$$

$$d\mathbf{V}_x(r) = -\nabla_{\mathbf{u}}\Phi(\mathbf{U}_x(r)) dr - \frac{1}{2}f_0\mathbf{V}_x(r) dr + \sqrt{f_0} d\mathbf{W}_x(r), \quad (26.110)$$

with the initial conditions,

$$\mathbf{U}_x(0) = \mathbf{u}_0, \quad \mathbf{V}_x(0) = \mathbf{v}_0 \quad a.s., \quad (26.111)$$

in which \mathbf{u}_0 and \mathbf{v}_0 are given vectors in \mathbb{R}^N (that are generally taken as zero in the applications) and $f_0 > 0$ is a free parameter whose usefulness is explained below. From Eqs. (26.82) and (26.102), it can be deduced that function $\mathbf{u} \mapsto \Phi(\mathbf{u})$: (i) is continuous on \mathbb{R}^N and (ii) is such that $\mathbf{u} \mapsto \|\nabla_{\mathbf{u}}\Phi(\mathbf{u})\|$ is a locally bounded function on \mathbb{R}^N (i.e., is bounded on all compact sets in \mathbb{R}^N). In addition the Lagrange multiplier λ^{sol} , which belongs to $\mathcal{C}_{\lambda} \subset \mathbb{R}^{1+N}$, is such that

$$\inf_{\|\mathbf{u}\| > R} \Phi(\mathbf{u}) \rightarrow +\infty \quad \text{if} \quad R \rightarrow +\infty, \quad (26.112)$$

$$\inf_{\mathbf{u} \in \mathbb{R}^n} \Phi(\mathbf{u}) = \Phi_{\min} \quad \text{with} \quad \Phi_{\min} \in \mathbb{R}, \quad (26.113)$$

$$\int_{\mathbb{R}^n} \|\nabla_{\mathbf{u}}\Phi(\mathbf{u})\| e^{-\Phi(\mathbf{u})} d\mathbf{u} < +\infty. \quad (26.114)$$

Taking into account (i), (ii), and Eqs. (26.112), (26.113), and (26.114), using Theorems 4–7 in pages 211–216 of Ref. [55] for which the Hamiltonian is taken as $\mathbb{H}(\mathbf{u}, \mathbf{v}) = \|\mathbf{v}\|^2/2 + \Phi(\mathbf{u})$, and using [17, 39] for the ergodic property, it can be deduced that the problem defined by Eqs. (26.109), (26.110), and (26.111) admits a unique solution. For all \mathbf{x} fixed in \mathbb{R}^d , this solution is a second-order diffusion

stochastic process $\{(\mathbf{U}_x(r), \mathbf{V}_x(r)), r \geq 0\}$, which converges to a stationary and ergodic diffusion stochastic process $\{(\mathbf{U}_x^{\text{st}}(r_{\text{st}}), \mathbf{V}_x^{\text{st}}(r_{\text{st}})), r_{\text{st}} \geq 0\}$, when r goes to infinity, associated with the invariant probability measure $P_{\text{st}}(d\mathbf{u}, d\mathbf{v}) = \rho_{\text{st}}(\mathbf{u}, \mathbf{v}) d\mathbf{u} d\mathbf{v}$ (that is independent of x). The probability density function $(\mathbf{u}, \mathbf{v}) \mapsto \rho_{\text{st}}(\mathbf{u}, \mathbf{v})$ on $\mathbb{R}^N \times \mathbb{R}^N$ is the unique solution of the steady-state Fokker-Planck equation associated with Eqs. (26.109) and (26.110) and is written (see pp. 120–123 in [55]) as

$$\rho_{\text{st}}(\mathbf{u}, \mathbf{v}) = c_N \exp\left\{-\frac{1}{2}\|\mathbf{v}\|^2 - \Phi(\mathbf{u})\right\}, \quad (26.115)$$

in which c_N is the constant of normalization. Equations (26.103), (26.108), and (26.115) yield

$$p_{Y(x)}(\mathbf{y}) = \int_{\mathbb{R}^N} \rho_{\text{st}}(\mathbf{y}, \mathbf{v}) d\mathbf{v}, \quad \forall \mathbf{y} \in \mathbb{R}^N. \quad (26.116)$$

Random variable $Y(x)$ (for which the pdf $p_{Y(x)}$ is defined by Eq. (26.103)) can then be written, for all fixed positive value of r_{st} , as

$$Y(x) = \mathbf{U}_x^{\text{st}}(r_{\text{st}}) = \lim_{r \rightarrow +\infty} \mathbf{U}_x(r) \quad \text{in probability distribution.} \quad (26.117)$$

The free parameter $f_0 > 0$ introduced in Eq. (26.110) allows a dissipation term to be introduced in the nonlinear second-order dynamical system (formulated in the Hamiltonian form with an additional dissipative term) for obtaining more rapidly the asymptotic behavior corresponding to the stationary and ergodic solution associated with the invariant measure. Using Eq. (26.117) and the ergodic property of stationary stochastic process \mathbf{U}_x^{st} , it should be noted that, if w is any mapping from \mathbb{R}^N into an Euclidean space such that $E\{w(Y(x))\} = \int_{\mathbb{R}^N} w(\mathbf{y}) p_{Y(x)} d\mathbf{y}$ is finite, then

$$E\{w(Y(x))\} = \lim_{R \rightarrow +\infty} \frac{1}{R} \int_0^R w(\mathbf{U}_x(r, \theta')) dr, \quad (26.118)$$

in which, for $\theta' \in \Theta'$, $\mathbf{U}_x(\cdot, \theta')$ is any realization of \mathbf{U}_x .

9.3.8 Discretization Scheme of the Family of ISDE

A discretization scheme must be used for numerically solving Eqs. (26.109), (26.110), and (26.111). For general surveys on discretization schemes for ISDE, we refer the reader to [40, 70] (among others). The present case, related to a Hamiltonian dynamical system, has also been analyzed using an implicit Euler scheme in [71]. Hereinafter, we present the Störmer-Verlet scheme (see [28, 29]), which is an efficient scheme that preserves energy for nondissipative Hamiltonian dynamical systems (see [35] for reviews about this scheme in the deterministic case, and see [6] and the therein for the stochastic case).

Let $\mu \geq 1$ be an integer. For all \mathbf{x} in \mathbb{R}^d , the ISDE defined by Eqs. (26.109), (26.110), and (26.111) is solved on the finite interval $[0, (\mu - 1) \Delta r]$, in which Δr is the sampling step of the continuous index parameter r . The integration scheme is based on the use of the μ sampling points $r_k = (k - 1) \Delta r$ for $k = 1, \dots, \mu$, and the following notations are used: $\mathbf{U}_\mathbf{x}^k = \mathbf{U}_\mathbf{x}(r_k)$, $\mathbf{V}_\mathbf{x}^k = \mathbf{V}_\mathbf{x}(r_k)$, and $\mathbf{W}_\mathbf{x}^k = \mathbf{W}_\mathbf{x}(r_k)$, with $\mathbf{U}_\mathbf{x}^1 = \mathbf{u}_0$, $\mathbf{V}_\mathbf{x}^1 = \mathbf{v}_0$, and $\mathbf{W}_\mathbf{x}^1 = \mathbf{W}_\mathbf{x}(0) = \mathbf{0}$. From Eq. (26.107) and for $k = 1, \dots, \mu - 1$, the increment $\Delta \mathbf{W}_\mathbf{x}^{k+1} = \mathbf{W}_\mathbf{x}^{k+1} - \mathbf{W}_\mathbf{x}^k$ is written as

$$\Delta \mathbf{W}_\mathbf{x}^{k+1} = \sqrt{\Delta r} \mathbf{B}^{k+1}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^d, \quad (26.119)$$

in which the $\mu - 1$ random fields $\{\mathbf{B}^{k+1}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}_{k=1,\dots,\mu-1}$ are independent copies of random field $\{\mathbf{B}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$. For $k = 1, \dots, \mu - 1$, the Störmer-Verlet scheme is written as

$$\mathbf{U}_\mathbf{x}^{k+\frac{1}{2}} = \mathbf{U}_\mathbf{x}^k + \frac{\Delta r}{2} \mathbf{V}_\mathbf{x}^k, \quad (26.120)$$

$$\mathbf{V}_\mathbf{x}^{k+1} = \frac{1-b}{1+b} \mathbf{V}_\mathbf{x}^k + \frac{\Delta r}{1+b} \mathbf{L}_\mathbf{x}^{k+\frac{1}{2}} + \frac{\sqrt{f_0}}{1+b} \Delta \mathbf{W}_\mathbf{x}^{k+1}, \quad (26.121)$$

$$\mathbf{U}_\mathbf{x}^{k+1} = \mathbf{U}_\mathbf{x}^{k+\frac{1}{2}} + \frac{\Delta r}{2} \mathbf{V}_\mathbf{x}^{k+1}, \quad (26.122)$$

where $b = f_0 \Delta r / 4$, and where $\mathbf{L}_\mathbf{x}^{k+\frac{1}{2}}$ is the \mathbb{R}^N -valued random variable such that $\mathbf{L}_\mathbf{x}^{k+\frac{1}{2}} = -\{\nabla_{\mathbf{u}} \Phi(\mathbf{u})\}_{\mathbf{u}=\mathbf{U}_\mathbf{x}^{k+\frac{1}{2}}}$. For a given realization θ' in Θ' , the sequence $\{\mathbf{U}_\mathbf{x}^k(\theta'), k = 1, \dots, \mu\}$ is constructed using Eqs. (26.120), (26.121), and (26.122). The discretization of Eq. (26.118) yields the following estimation of the mathematical expectation,

$$E\{w(\mathbf{Y}(\mathbf{x}))\} = \lim_{\mu \rightarrow +\infty} \hat{w}_\mu(\mathbf{x}), \quad \hat{w}_\mu(\mathbf{x}) = \frac{1}{\mu - \mu_0 + 1} \sum_{k=\mu_0}^{\mu} w(\mathbf{U}_\mathbf{x}^k(\theta')), \quad (26.123)$$

in which, for f_0 fixed, the integer $\mu_0 > 1$ is chosen to remove the transient part of the response induced by the initial condition. For details concerning the optimal choice of the numerical parameters, such as μ_0 , μ , f_0 , Δr , \mathbf{u}_0 , and \mathbf{v}_0 , we refer the reader to [29, 34, 59].

9.3.9 Definition of the Hyperparameter \mathbf{s}

The hyperparameter parameter $\mathbf{s} \in \mathcal{C}_s \subset \mathbb{R}^{N_s}$ of the algebraic prior stochastic model $\{\mathbf{K}^{\text{APSM}}(\mathbf{x}; \mathbf{s}), \mathbf{x} \in \Omega\}$, which has been constructed for the dominant statistical fluctuations belonging to a given symmetry class of dimension n , with some anisotropic statistical fluctuations, are constituted of the quantities summarized hereinafter:

- the reshaping of $[C_\ell] \in \mathbb{M}_n^+(\mathbb{R})$ (the lower bound) and $[K] \in \mathbb{M}_n^+(\mathbb{R})$ (the mean value),

- for the control of the anisotropic statistical fluctuations (modeled by random field $[\mathbf{G}_0]$), the $d n(n+1)/2$ positive real numbers, $\{L_1^{jk}, \dots, L_d^{jk}\}_{1 \leq j \leq k \leq n}$ (the spatial-correlation lengths, for the parameterization given in the example), and δ (the dispersion) such that $0 < \delta < \sqrt{(n+1)/(n+5)}$,
- for the control of the statistical fluctuations belonging to a symmetry class (modeled by random field $[\mathbf{A}]$), the $d N$ positive real numbers, $\{\mathbb{L}_1^j, \dots, \mathbb{L}_d^j\}_{1 \leq j \leq N}$ (the spatial-correlation lengths, for the parameterization given in the example), and δ_A (the dispersion) such that $0 < \delta_A$.

10 Key Research Findings and Applications

10.1 Additional Ingredients for Statistical Reduced Models, Symmetry Properties, and Generators for High Stochastic Dimension

- Karhunen-Loëve's expansion revisited for vector-valued random fields and identification from a set of realizations: scaling [50], a posteriori error, and optimal reduced basis [51].
- Construction of a basis adaptation in homogeneous chaos spaces [73].
- ISDE-based generator for a class of non-Gaussian vector-valued random fields in uncertainty quantification [28, 29].
- Random elasticity tensors of materials exhibiting symmetry properties [26–28] and stochastic boundedness constraints [10, 27, 32].
- Random field representations and robust algorithms for the identification of polynomial chaos representations in high dimension from a set of realizations [3, 48, 49, 51, 61, 62, 64].

10.2 Tensor-Valued Random Fields and Continuum Mechanics of Heterogenous Materials

- Composites reinforced with fibers with experimental identification [30, 31].
- Polycrystalline microstructures [32].
- Porous materials with anisotropic permeability tensor random field [33] and with interphases [34].
- Human cortical bone with mechanical alterations in ultrasonic range [16].

11 Conclusions

A complete advanced methodology and the associated tools have been presented for solving the challenging statistical inverse problem related to the experimental identification of a non-Gaussian matrix-valued random field that is the model

parameter of a boundary value problem, using some partial and limited experimental data related to a model observation. Many applications and validation of this methodology can be found in the given references.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2000)
2. Arnst, M., Ghanem, R., Soize, C.: Identification of Bayesian posteriors for coefficients of chaos expansions. *J. Comput. Phys.* **229**(9), 3134–3154 (2010)
3. Batou, A., Soize, C.: Stochastic modeling and identification of an uncertain computational dynamical model with random fields properties and model uncertainties. *Arch. Appl. Mech.* **83**(6), 831–848 (2013)
4. Batou, A., Soize, C.: Calculation of Lagrange multipliers in the construction of maximum entropy distributions in high stochastic dimension. *SIAM/ASA J. Uncertain. Quantif.* **1**(1), 431–451 (2013)
5. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys.* **230**(6), 2345–2367 (2011)
6. Burrage, K., Lenane, I., Lythe, G.: Numerical methods for second-order stochastic differential equations. *SIAM J. Sci. Comput.* **29**, 245–264 (2007)
7. Cameron, R.H., Martin, W.T.: The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Ann. Math. Second Ser.* **48**(2), 385–392 (1947)
8. Carlin, B.P., Louis, T.A.: Bayesian Methods for Data Analysis, 3rd edn. Chapman & Hall/CRC Press, Boca Raton (2009)
9. Congdon, P.: Bayesian Statistical Modelling, 2nd edn. Wiley, Chichester (2007)
10. Das, S., Ghanem, R.: A bounded random matrix approach for stochastic upscaling. *Multiscale Model. Simul.* **8**(1), 296–325 (2009)
11. Das, S., Ghanem, R., Spall, J.C.: Asymptotic sampling distribution for polynomial chaos representation from data: a maximum entropy and fisher information approach. *SIAM J. Sci. Comput.* **30**(5), 2207–2234 (2008)
12. Das, S., Ghanem, R., Finette, S.: Polynomial chaos representation of spatio-temporal random field from experimental measurements. *J. Comput. Phys.* **228**, 8726–8751 (2009)
13. Debusschere, B.J., Najim, H.N., Pebay, P.P., Knio, O.M., Ghanem, R., Le Maître, O.: Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.* **26**(2), 698–719 (2004)
14. Desceliers, C., Ghanem, R., Soize, C.: Maximum likelihood estimation of stochastic chaos representations from experimental data. *Int. J. Numer. Methods Eng.* **66**(6), 978–1001 (2006)
15. Desceliers, C., Soize, C., Ghanem, R.: Identification of chaos representations of elastic properties of random media using experimental vibration tests. *Comput. Mech.* **39**(6), 831–838 (2007)
16. Desceliers, C., Soize, C., Naili, S., Haiat, G.: Probabilistic model of the human cortical bone with mechanical alterations in ultrasonic range. *Mech. Syst. Signal Process.* **32**, 170–177 (2012)
17. Doob, J.L.: Stochastic Processes. Wiley, New York (1990)
18. Doostan, A., Ghanem, R., Red-Horse, J.: Stochastic model reduction for chaos representations. *Comput. Methods Appl. Mech. Eng.* **196**(37–40), 3951–3966 (2007)
19. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* **20**(2), 303–353 (1998)
20. Ernst, O.G., Mugler, A., Starkloff, H.J., Ullmann, E.: On the convergence of generalized polynomial chaos expansions. *ESAIM Math. Model. Numer. Anal.* **46**(2), 317–339 (2012)

21. Ghanem, R., Dham, S.: Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transp. Porous Media* **32**, 239–262 (1998)
22. Ghanem, R., Doostan, R.: Characterization of stochastic system parameters from experimental data: a Bayesian inference approach. *J. Comput. Phys.* **217**(1), 63–81 (2006)
23. Ghanem, R., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991). See also the revised edition, Dover Publications, New York (2003)
24. Ghanem, R., Doostan, R., Red-Horse, J.: A probability construction of model validation. *Comput. Methods Appl. Mech. Eng.* **197**(29–32), 2585–2595 (2008)
25. Ghosh, D., Ghanem, R.: Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions. *Int. J. Numer. Methods Eng.* **73**(2), 162–184 (2008)
26. Guilleminot, J., Soize, C.: Non-Gaussian positive-definite matrix-valued random fields with constrained eigenvalues: application to random elasticity tensors with uncertain material symmetries. *Int. J. Numer. Methods Eng.* **88**(11), 1128–1151 (2011)
27. Guilleminot, J., Soize, C.: Probabilistic modeling of apparent tensors in elastostatics: a MaxEnt approach under material symmetry and stochastic boundedness constraints. *Probab. Eng. Mech.* **28**, 118–124 (2012)
28. Guilleminot, J., Soize, C.: Stochastic model and generator for random fields with symmetry properties: application to the mesoscopic modeling of elastic random media. *Multiscale Model. Simul. (SIAM Interdiscip. J.)* **11**(3), 840–870 (2013)
29. Guilleminot, J., Soize, C.: Itô SDE-based generator for a class of non-Gaussian vector-valued random fields in uncertainty quantification. *SIAM J. Sci. Comput.* **36**(6), A2763–A2786 (2014)
30. Guilleminot, J., Soize, C., Kondo, D., Binetruy, C.: Theoretical framework and experimental procedure for modelling volume fraction stochastic fluctuations in fiber reinforced composites. *Int. J. Solids Struct.* **45**(21), 5567–5583 (2008)
31. Guilleminot, J., Soize, C., Kondo, D.: Mesoscale probabilistic models for the elasticity tensor of fiber reinforced composites: experimental identification and numerical aspects. *Mech. Mater.* **41**(12), 1309–1322 (2009)
32. Guilleminot, J., Noshadravan, A., Soize, C., Ghanem, R.G.: A probabilistic model for bounded elasticity tensor random fields with application to polycrystalline microstructures. *Comput. Methods Appl. Mech. Eng.* **200**, 1637–1648 (2011)
33. Guilleminot, J., Soize, C., Ghanem, R.: Stochastic representation for anisotropic permeability tensor random fields. *Int. J. Numer. Anal. Methods Geom.* **36**(13), 1592–1608 (2012)
34. Guilleminot, J., Le, T.T., Soize, C.: Stochastic framework for modeling the linear apparent behavior of complex materials: application to random porous materials with interphases. *Acta Mech. Sinica* **29**(6), 773–782 (2013)
35. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, Heidelberg (2002)
36. Isakov, V.: *Inverse Problems for Partial Differential Equations*. Springer, New York (2006)
37. Jaynes, E.T.: Information theory and statistical mechanics. *Phys. Rev.* **106**(4), 620–630; **108**(2), 171–190 (1957)
38. Kaipio, J., Somersalo, E.: *Statistical and Computational Inverse Problems*. Springer, New York (2005)
39. Khasminskii, R.: *Stochastic Stability of Differential Equations*, 2nd edn. Springer, Heidelberg (2012)
40. Kloeden, P.E., Platen, E.: *Numerical Solution of Stochastic Differential Equations*. Springer, Heidelberg (1992)
41. Krée, P., Soize, C.: *Mathematics of Random Phenomena*. Reidel, Dordrecht (1986)
42. Le Maître, O.P., Knio, O.M.: *Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics*. Springer, Heidelberg (2010)
43. Le Maître, O.P., Knio, O.M., Najm, H.N.: Uncertainty propagation using Wiener-Haar expansions. *J. Comput. Phys.* **197**(1), 28–57 (2004)
44. Lucor, D., Su, C.H., Karniadakis, G.E.: Generalized polynomial chaos and random oscillators. *Int. J. Numer. Methods Eng.* **60**(3), 571–596 (2004)

45. Marzouk, Y.M., Najm, H.N.: Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *J. Comput. Phys.* **228**(6), 1862–1902 (2009)
46. Najm, H.H.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **41**, 35–52 (2009)
47. Nouy, A.: Proper generalized decomposition and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Eng.* **16**(3), 403–434 (2010)
48. Nouy, A., Soize, C.: Random fields representations for stochastic elliptic boundary value problems and statistical inverse problems. *Eur. J. Appl. Math.* **25**(3), 339–373 (2014)
49. Perrin, G., Soize, C., Duhamel, D., Funfschilling, C.: Identification of polynomial chaos representations in high dimension from a set of realizations. *SIAM J. Sci. Comput.* **34**(6), A2917–A2945 (2012)
50. Perrin, G., Soize, C., Duhamel, D., Funfschilling, C.: Karhunen-Loève expansion revisited for vector-valued random fields: scaling, errors and optimal basis. *J. Comput. Phys.* **242**(1), 607–622 (2013)
51. Perrin, G., Soize, C., Duhamel, D., Funfschilling, C.: A posterior error and optimal reduced basis for stochastic processes defined by a set of realizations. *SIAM/ASA J. Uncertain. Quantif.* **2**, 745–762 (2014)
52. Puig, B., Poirion, F., Soize, C.: Non-Gaussian simulation using Hermite polynomial expansion: convergences and algorithms. *Probab. Eng. Mech.* **17**(3), 253–264 (2002)
53. Rozanov, Y.A.: *Random Fields and Stochastic Partial Differential Equations*. Kluwer Academic, Dordrecht (1998)
54. Serfling, R.J.: *Approximation Theorems of Mathematical Statistics*. Wiley, New York (1980)
55. Soize, C.: The Fokker-Planck Equation for Stochastic Dynamical Systems and Its Explicit Steady State Solutions. World Scientific, Singapore (1994)
56. Soize, C.: Random-field model for the elasticity tensor of anisotropic random media. *Comptes Rendus Mecanique* **332**, 1007–1012 (2004)
57. Soize, C., Ghanem, R.: Physical systems with random uncertainties: chaos representation with arbitrary probability measure. *SIAM J. Sci. Comput.* **26**(2), 395–410 (2004)
58. Soize, C.: Non Gaussian positive-definite matrix-valued random fields for elliptic stochastic partial differential operators. *Comput. Methods Appl. Mech. Eng.* **195**(1–3), 26–64 (2006)
59. Soize, C.: Construction of probability distributions in high dimension using the maximum entropy principle. Applications to stochastic processes, random fields and random matrices. *Int. J. Numer. Methods Eng.* **76**(10), 1583–1611 (2008)
60. Soize, C.: Tensor-valued random fields for meso-scale stochastic model of anisotropic elastic microstructure and probabilistic analysis of representative volume element size. *Probab. Eng. Mech.* **23**(2–3), 307–323 (2008)
61. Soize, C.: Identification of high-dimension polynomial chaos expansions with random coefficients for non-Gaussian tensor-valued random fields using partial and limited experimental data. *Comput. Methods Appl. Mech. Eng.* **199**(33–36), 2150–2164 (2010)
62. Soize, C.: A computational inverse method for identification of non-Gaussian random fields using the Bayesian approach in very high dimension. *Comput. Methods Appl. Mech. Eng.* **200**(45–46), 3083–3099 (2011)
63. Soize, C.: *Stochastic Models of Uncertainties in Computational Mechanics*. American Society of Civil Engineers (ASCE), Reston (2012)
64. Soize, C.: Polynomial chaos expansion of a multimodal random vector. *SIAM/ASA J. Uncertain. Quantif.* **3**(1), 34–60 (2015)
65. Soize, C., Desceliers, C.: Computational aspects for constructing realizations of polynomial chaos in high dimension. *SIAM J. Sci. Comput.* **32**(5), 2820–2831 (2010)
66. Soize, C., Ghanem, R.: Reduced chaos decomposition with random coefficients of vector-valued random variables and random fields. *Comput. Methods Appl. Mech. Eng.* **198**(21–26), 1926–1934 (2009)
67. Spall, J.C.: *Introduction to Stochastic Search and Optimization*. Wiley, Hoboken (2003)
68. Stuart, A.M.: Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559 (2010)

69. Ta, Q.A., Clouet, D., Cottereau, R.: Modeling of random anisotropic elastic media and impact on wave propagation. *Eur. J. Comput. Mech.* **19**(1–2–3), 241–253 (2010)
70. Talay, D.: Simulation and numerical analysis of stochastic differential systems. In: Kree, P., Wedig, W. (eds.) *Probabilistic Methods in Applied Physics*. Lecture Notes in Physics, vol. 451, pp. 54–96. Springer, Heidelberg (1995)
71. Talay, D.: Stochastic Hamiltonian system: exponential convergence to the invariant measure and discretization by the implicit Euler scheme. *Markov Process. Relat. Fields* **8**, 163–198 (2002)
72. Tarantola, A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia (2005)
73. Tipireddy, R., Ghanem, R.: Basis adaptation in homogeneous chaos spaces. *J. Comput. Phys.* **259**, 304–317 (2014)
74. Vanmarcke, E.: *Random Fields, Analysis and Synthesis*, Revised and Expanded New edn. World Scientific, Singapore (2010)
75. Walpole, L.J.: Elastic behavior of composite materials: theoretical foundations. *Adv. Appl. Mech.* **21**, 169–242 (1981)
76. Walter, E., Pronzato, L.: *Identification of Parametric Models from Experimental Data*. Springer, Berlin (1997)
77. Wan, X.L., Karniadakis, G.E.: Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.* **28**(3), 901–928 (2006)
78. Xiu, D.B., Karniadakis, G.E.: Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
79. Zienkiewicz, O.C., Taylor, R.L.: *The Finite Element Method for Solid and Structural Mechanics*, 6th edn. Elsevier/Butterworth-Heinemann, Amsterdam (2005)

Peng Chen and Christoph Schwab

Abstract

This work surveys formulation and algorithms for model order reduction (MOR for short) techniques in accelerating computational forward and inverse UQ. Operator equations (comprising elliptic and parabolic partial differential equations (PDEs for short) and boundary integral equations (BIEs for short)) with distributed uncertain input, being an element of an infinite-dimensional, separable Banach space X , are admitted. Using an unconditional basis of X , computational UQ for these equations is reduced to numerical solution of countably parametric operator equations with smooth parameter dependence.

In computational forward UQ, efficiency of MOR is based on recent sparsity results for countably parametric solutions which imply upper bounds on Kolmogorov N -widths of the manifold of (countably) parametric solutions and quantities of interest (QoI for short) with dimension-independent convergence rates. Subspace sequences which realize the N -width convergence rates are obtained by greedy search algorithms in the solution manifold. Heuristic search strategies in parameter space based on finite searches over anisotropic sparse grids render greedy searches in reduced basis construction feasible. Instances of the parametric forward problems which arise in the greedy searches are assumed to be discretized by abstract classes of Petrov–Galerkin (PG for short) discretizations of the parametric operator equation, covering most conforming primal, dual, and mixed finite element methods (FEMs), as well as certain space-time Galerkin schemes for the application problem of interest. Based on the PG discretization, MOR for both linear and nonlinear and affine and nonaffine parametric problems are presented.

P. Chen (✉)

Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, USA

e-mail: peng@ices.utexas.edu

C. Schwab

Department Mathematik, Seminar for Applied Mathematics, ETH Zurich, Switzerland

e-mail: christoph.schwab@sam.math.ethz.ch

Computational inverse UQ for the mentioned operator equations is considered in the Bayesian setting of [M. Dashti and A.M. Stuart: Inverse problems a Bayesian perspective, arXiv:1302.6989v3, this Handbook]. The (countably) parametric Bayesian posterior density inherits, in the absence of concentration effects for small observation noise covariance, the sparsity and N -width bounds of the (countably) parametric manifolds of solution and QoI. This allows, in turn, for the deployment of MOR techniques for the parsimonious approximation of the parametric Bayesian posterior density, with convergence rates which are only limited by the sparsity of the uncertain inputs in the forward model.

Keywords

Uncertainty quantification • Sparse grid • Reduced basis • Empirical interpolation • Greedy algorithm • High fidelity • Petrov–Galerkin • A posteriori error estimate • A priori error estimate • Bayesian inversion

Contents

1	Introduction	938
2	Forward UQ	942
2.1	A Class of Forward Problems with Uncertain Input Data	942
2.2	Uncertainty Parametrization	946
2.3	Parameter Sparsity in Forward UQ	951
3	Model Order Reduction	955
3.1	High-Fidelity Approximation	956
3.2	Reduced Basis Compression	958
3.3	Reduced Basis Construction	960
3.4	Linear and Affine-Parametric Problems	963
3.5	Nonlinear and Nonaffine-Parametric Problems	968
3.6	Sparse Grid RB Construction	975
4	Inverse UQ	976
4.1	Bayesian Inverse Problems for Parametric Operator Equations	976
4.2	Parametric Bayesian Posterior	978
4.3	Well-Posedness and Approximation	980
4.4	Reduced Basis Acceleration of MCMC	982
4.5	Dimension and Order Adaptive, Deterministic Quadrature	983
4.6	Quasi-Monte Carlo Quadrature	984
5	Software	984
6	Conclusion	985
7	Glossary	986
	References	987

1 Introduction

A core task in computational forward and inverse UQ in computational science is the numerical solution of parametric numerical models for the system of interest. Uncertainty in numerical solutions obtained from the computational model can be crudely classified as follows:

(i) *Modeling error*: the mathematical model under consideration does not correctly describe the physical phenomena of interest: its exact solution does not predict the QoI properly. (ii) *Discretization error*: the discretization of the mathematical model (the substitution of the continuous mathematical model by a discrete, finite-dimensional approximation which, in principle, is solvable to any prescribed numerical accuracy on computers in float point arithmetic), whose computational realization is used in forward UQ, introduces a mismatch between the “true response” (understood as exact solution of the mathematical model) and the “computed response” obtained from the computational model, *for a given (set of) input data*, and for a *prescribed quantity of interest (QoI)*. Discretization errors comprise replacing mathematical continuum models by finite difference, finite volume, or finite element models and inexact solution of the finite-dimensional problems which result from discretization, continuous time models by discrete timestepping, random variables by Monte Carlo samples, and their realization by random number generators. The classical paradigm of numerical analysis requires discretizations to be *stable* and *consistent*. Particular issues for discretizations in the context of UQ are *uniform stability and consistency*, with respect to all instances of the uncertain input u . (iii) *Computational error*: the discretized model for the numerical computation of the QoI which is obtained from a mathematical model and its subsequent discretization is not numerically solvable *for the given input data* with the computer resources at hand. This could be, for example, due to CPU limitations and imprecise float point arithmetic (rounding) but also due to runtime failures of hardware components, partial loss of data at runtime, etc.

Under the (strong) assumptions that modeling error (i) and computational error (iii) are negligible, which is to say that epistemic uncertainty is absent, i.e., the mathematical model under consideration is well posed and accounts in principle for all phenomena of interest, and the assumption that float point computations are reliable, a *first key task in computational forward UQ is the efficient computational prediction of the QoI of a mathematical model for any given instance of uncertain input u from a space of admissible input data X* . Computational challenges arise when the space X of uncertain inputs is infinite-dimensional. For example, for *distributed uncertain input* (such as material properties in inhomogeneous solids or fluids, shapes obtained from noisy imaging, random forcing, etc.), X is a (subset of a) function space. Computational UQ then involves *uncertainty parametrization* via a basis of X (such as, for example, a Fourier basis in $X = L^2$), resulting in a *parametric forward problem depending on sequences $y = (y_j)_{j \geq 1}$ of uncertain input parameters*. Probability measures on X which encode information on aleatoric uncertainty can be introduced via countable products of probability measures on sequence space. Computational UQ procedures access the parametric response or forward solutions *nonintrusively*, i.e., by numerical solution of the forward model for instances of the parameter sequence y . Due to the possibly infinite dimension of space of uncertain inputs y and due to the possibly high cost of each forward solve, two *key issues in computational UQ* are (a) efficient computational sampling of uncertain inputs from high (possibly infinite) dimensional parameter spaces and (b) efficient numerical solution of parametric forward models.

Two key methodologies are reviewed, which address issue (a) and issue (b): *sparsity* of the parametric response map and *model order reduction (MOR)*. Sparsity refers to the possibility to represent responses of the parametric forward model with user-specified accuracy $0 < \varepsilon \ll 1$ with responses at $O(\varepsilon^{-1/s})$ parameter instances where the sparsity parameter $s > 0$ and $O()$ are independent of the dimension of the parameter space. MOR refers to replacing the parametric response from the mathematical model for all admissible parameters by one parsimonious, the so-called reduced order model, respectively, *surrogate model* which allows fast response evaluation with certified accuracy ε for any parameter instance y . The accuracy which can be achieved by MOR with n basis functions, for a given set of solutions, is determined by the so-called *n -width of the solution set in the space of all possible solutions*.

Recent results on sparsity of parametric forward models are reviewed, covering in particular uncertain coefficients, loadings and domains of definition in partial differential equation models which arise in engineering and in the sciences. *Sparsity adapted interpolation schemes in parameter space* allow to build *polynomial chaos surrogates of the parametric forward response maps*. The corresponding point sets in parameter space are (*generalized*) *sparse grids*; they are based on dimension- and order-adaptive interpolation processes described here. As Monte Carlo sampling, the performance of collocation in these deterministic point sets is independent of the dimension of the parameter space, and their convergence rates can exceed $1/2$, provided the parametric response exhibits sufficient sparsity.

Sampling of high-dimensional parameter spaces on generalized sparse grids can be used in so-called stochastic collocation for building polynomial surrogates of the parametric forward maps, in the “training” phase of MOR and for deterministic Smolyak quadrature over all uncertainties to evaluate response statistics in forward UQ and Bayesian estimates in inverse UQ, for example.

The chapter’s focus is therefore on model order reduction and sparse, adaptive collocation methods in high-dimensional parameter spaces in computational forward and inverse UQ. All concepts are developed on abstract classes of forward problems with parametric *distributed* uncertain input data, which is to say that the uncertain data may take values in an infinite-dimensional function space. Mathematical forward models which are considered here are specified in terms of (linear or nonlinear) PDEs, of elliptic or parabolic type, with smooth nonlinearities. Key steps in the approach are:

- Uncertainty parametrization: upon choosing a basis Ψ of the space X of uncertain input data, the forward problem turns into a parametric problem. The choice of basis Ψ is, in general, highly nonunique. To ensure stability in float point realizations of the parametric forward problem, care must be taken to choose *well-conditioned* bases Ψ of X . For example, in Hilbert spaces X , ideally orthonormal bases Ψ should be chosen or at least so-called Riesz bases with good Riesz constants.
- Upon choosing a (well-conditioned) basis of X , a (countably) parametric family of parametric, deterministic problems are obtained; these are referred to in the

literature typically as “ y -PDE”. For distributed uncertain inputs u , these are, usually, *infinite-dimensional parameter sequences* y .

- The (minimal, for computational UQ) requirements of well-posedness of the forward problem for all admissible instances of the uncertain input u and for continuous dependence of the solution on the input imply that *the set of all parametric solutions* $\mathcal{M} := \{q(y) : y \in U\} \subset \mathcal{X}$ form a submanifold of the solution space \mathcal{X} . For smooth (linear or nonlinear) elliptic and parabolic problems, the manifold \mathcal{M} is in fact an analytic manifold; see [28] and the references therein.
- “Smooth” (analytic) dependence of PDE on uncertain inputs, resp. parameters implies algebraic (exponential) smallness of the n -width $\mathcal{M} \subset \mathcal{X}$. This enables, in principle, *accelerated forward solves with work which scales algebraic (logarithmic) in accuracy* ε .
- Determining (near-)optimal subspaces which realize Kolmogorov n -width approximation bounds is feasible using so-called reduced basis methods. Section 3 provides key elements of the corresponding algorithms and the related theoretical results, with particular attention to the countably parametric problems resulting from forward problems with distributed, uncertain input data. Detailed references to the literature on these techniques are provided.
- The availability of suitably compressed approximations of forward models with uncertain inputs implies, in particular, dramatic accelerations of any algorithm which involves numerous, repeated approximate evaluations of these forward models, for instance, optimization problems under uncertainty [14, 19], PDE-constrained optimization algorithms, and the corresponding inverse problems. An application to *inverse uncertainty quantification by Bayesian inversion* is presented in Sect. 4. Bayes’ theorem provides an expression for the “most likely” expected output in a quantity of interest (QoI), conditional on a set of given, noisy observations of (functionals of) the forward response. The numerical realization becomes, with the mentioned uncertainty parametrization, an *infinite-dimensional integration problem* against a probability measure which is only known up to a normalization constant. Current computational approaches to deal with this problem are various variants of Monte Carlo methods, such as Markov chain Monte Carlo (MCMC), sequential Monte Carlo, etc. Due to their generally slow convergence, numerous evaluations of the forward models are necessary, for a large number of proposals (generated by the corresponding samplers) of the uncertain input data. In this setting, running the Markov chains on a reduced basis surrogate of the forward model can afford dramatic reductions in CPU time; as shown in Sect. 4, the resulting computational Bayesian estimates of the expected QoI will inherit an error which is of the order of the error incurred in the MOR.

The propagation of MOR error bounds translates one-to-one to other recently developed computational methods for Bayesian inversion which circumvent the use of MC sampling. These methods rather tackle the infinite-dimensional, parametric integrals obtained by inserting the uncertainty parametrization into Bayes’ formula, for example, by adaptive Smolyak or higher-order quasi-Monte Carlo integration.

2 Forward UQ

By forward uncertainty quantification (“forward UQ” for short), we denote the efficient computational realization of the uncertainty-to-solution map. The present section specifies an abstract class of smooth, possibly nonlinear, operator equations with distributed, uncertain input data which allow for efficient computational forward UQ. The common feature of this class of problems is based on *holomorphic extension* of the parametric uncertainty-to-solution maps, as described in [28] and the references there.

2.1 A Class of Forward Problems with Uncertain Input Data

By \mathcal{X} and \mathcal{Y} , we denote separable Hilbert spaces with duals \mathcal{X}' and \mathcal{Y}' , respectively.

They are used for the formulation of the mathematical model of the forward problem: system responses q (such as temperature, concentration, displacements, electric fields, etc.) take values in \mathcal{X} , whereas loads and source terms are understood as objects in \mathcal{Y}' , i.e., they are assumed to act on “test functions” $v \in \mathcal{Y}$.

Admissible mathematical models take form of a *residual map* \mathcal{R} which associates, for a given uncertain input $u \in X$, to each state $q \in \mathcal{X}$ a response $\mathcal{R} : q \mapsto \mathcal{R}(u; q) \in \mathcal{Y}'$. With the space X of uncertain parameters $u \in X$ being infinite-dimensional, we speak about $u \in X$ as *distributed, uncertain parameters*. Three prototypical examples are presented: diffusion with uncertain diffusion coefficient, smooth nonlinear elliptic problem in a parametric domain, and a parabolic problem.

Example 1 (Linear diffusion problem with uncertain diffusion coefficient). The mathematical model is set in a bounded domain $D \subset \mathbb{R}^d$ (assumed certain), with diffusion coefficient $u(x) \in L^\infty(D)$ (assumed uncertain) and a source term $f(x) \in L^2(D)$ (assumed certain), find a concentration $q(x) \in H_0^1(D)$ such that

$$\mathcal{R}(q; u) := \operatorname{div}(u(x)\operatorname{grad}q(x)) + f(x) = 0 \quad \text{in } H^{-1}(D). \quad (27.1)$$

Here the spaces for the system response q are $\mathcal{X} = \mathcal{Y} = H_0^1(D)$, the space for the uncertain input is $X = L^\infty(D)$, and $\mathcal{Y}' = (H_0^1(D))^* = H^{-1}(D)$. Note that (27.1) is to hold in the weak or variational sense of $\mathcal{Y}' = H^{-1}(D)$; this gives rise to the *variational form of the residual equation*: find $q \in \mathcal{X}$ such that

$$0 = {}_{\mathcal{Y}}\langle v, \mathcal{R}(q; u) \rangle_{\mathcal{Y}'} = \int_D \operatorname{grad}v \cdot u(x)\operatorname{grad}q(x)dx - \int_D v(x)f(x)dx \quad (27.2)$$

for all $v \in \mathcal{Y} = H_0^1(D)$.

Here, and in what follows, ${}_{\mathcal{Y}}\langle \cdot, \cdot \rangle_{\mathcal{Y}'}$ denotes the $\mathcal{Y} \times \mathcal{Y}'$ -duality.

Analogous formulations arise for any second-order, linear elliptic PDE in divergence form, such as Helmholtz equations for time-harmonic wave propagation

in random media or in (displacement formulations) of boundary value problems in computational mechanics in solids with uncertain material properties.

Example 2. In a bounded domain $D \subset \mathbb{R}^n$, and in the time interval $I = (0, T)$ for a time-horizon $0 < T < \infty$, and for the affine-parametric, elliptic operator $A(\mathbf{y})q = \operatorname{div}(u(x)\operatorname{grad}q(x))$ as in (27.1), for given $f(x, t)$ and for given $u_0 \in L_2(D)$, the parametric, linear parabolic evolution problem is considered

$$B(\mathbf{y})q := \partial_t q - A(\mathbf{y})q = f, \quad q(\cdot, t)|_{\partial D} = 0 \quad \text{in } (0, T) \times D, \quad q(\cdot, 0) = q_0. \quad (27.3)$$

The parabolic, parametric evolution operator $B(\mathbf{y})$ in (27.3) allows for a weak residual formulation analogous to (27.1) with the Bochner spaces $\mathcal{X} = L_2(I; V) \cap H^1(I; V')$, $\mathcal{Y} = L_2(I; V) \times H$, $V = H_0^1(D)$ and $H = L_2(D)$. Here, the parametric bilinear form $B(\mathbf{y}; w, v)$ is defined, for $v = (v_1, v_2) \in L_2(I; V) \times H$, by

$$B(\mathbf{y}; w, v) := \int_I \langle \frac{dw}{dt}(t), v_1(t) \rangle_H + \int_D u(x, \mathbf{y}) \nabla w \cdot \nabla v_1 dx dt + \langle w(0), v_2 \rangle_H,$$

where $u(x, \mathbf{y})$ denotes the linear-parametric, isotropic diffusion coefficient as in (27.13).

An abstract setting is considered which accommodates both examples (and more general models) in a unified fashion. For a distributed, uncertain parameter $u \in X$, one considers a “forward” operator $\mathcal{R}(q; u)$ depending on u and acting on $q \in \mathcal{X}$. Assuming at our disposal, a “nominal parameter instance” $\langle u \rangle \in X$ (such as, for example, the expectation of an X -valued random field u) and for $u \in B_X(\langle u \rangle; R)$, an open ball of sufficiently small radius $R > 0$ in X centered at a nominal input instance $\langle u \rangle \in X$, the nonlinear operator equation is considered

$$\text{given } u \in B_X(\langle u \rangle; R), \text{ find } q \in \mathcal{X} \quad \text{s.t.} \quad {}_{\mathcal{Y}'} \langle \mathcal{R}(q; u), v \rangle_{\mathcal{Y}} = 0 \quad \forall v \in \mathcal{Y}. \quad (27.4)$$

Given $u \in B_X(\langle u \rangle; R)$, a solution q_0 of (27.4) is called *regular at u* if and only if $\mathcal{R}(\cdot; u)$ is differentiable with respect to q and if the differential $D_q \mathcal{R}(q_0; u) \in \mathcal{L}(\mathcal{X}; \mathcal{Y}')$ is an isomorphism. For the well-posedness of operator equations involving $\mathcal{R}(q; u)$, one assumes the map $\mathcal{R}(\cdot; u) : \mathcal{X} \mapsto \mathcal{Y}'$ admits a family of regular solutions *locally, in an open neighborhood of the nominal parameter instance $\langle u \rangle \in X$* .

Assumption 1. *The structural conditions*

$$\mathcal{R}(q; u) = A(q; u) - F(u) \quad \text{in } \mathcal{Y}', \quad (27.5)$$

hold, and for all u in a sufficiently small, closed neighborhood $\tilde{X} \subseteq X$ of $\langle u \rangle \in X$ the parametric forward problem: for every $u \in \tilde{X} \subseteq X$, given $F(u) \in \mathcal{Y}'$,

find $q(u) \in \mathcal{X}$ such that the residual Eq. (27.4) is well-posed. I.e., for every fixed $u \in \tilde{X} \subset X$, and for every $F(u) \in \mathcal{Y}'$, there exists a unique solution $q(u)$ of (27.4) which depends continuously on u .

The set $\{(q(u), u) : u \in \tilde{X}\} \subset \mathcal{X} \times X$ is called a *regular branch of solutions* of (27.5) if

$$\begin{aligned} \tilde{X} &\ni u \mapsto q(u) \text{ is continuous as mapping from } X \mapsto \mathcal{X}, \\ \mathcal{R}(q(u); u) &= 0 \quad \text{in } \mathcal{Y}'. \end{aligned} \tag{27.6}$$

The solutions is called in the regular branch (27.6) *nonsingular* if, in addition, the differential

$$(D_q \mathcal{R})(q(u); u) \in \mathcal{L}(\mathcal{X}, \mathcal{Y}') \text{ is an isomorphism from } \mathcal{X} \text{ onto } \mathcal{Y}', \text{ for all } u \in \tilde{X}. \tag{27.7}$$

The following proposition collects well-known sufficient conditions for well-posedness of (27.5). For regular branches of nonsingular solutions given by (27.5), (27.6), and (27.7), the mathematical model is well posed if the differential $D_q \mathcal{R}$ satisfies the so-called inf-sup conditions. In UQ, these classical (e.g., [5, 60]) conditions are to hold *uniformly with respect to the uncertain input data* $u \in \tilde{X} \subseteq X$.

Proposition 1. Assume that \mathcal{Y} is reflexive and that, for some nominal value $\langle u \rangle \in X$ of the uncertain input data, the operator equation (27.5) admits a regular branch of solutions (27.6). Then the differential $D_q \mathcal{R}$ at $(\langle u \rangle, q_0)$ given by the bilinear map

$$\mathcal{X} \times \mathcal{Y} \ni (\varphi, \psi) \mapsto {}_{\mathcal{Y}'} \langle D_q \mathcal{R}(q_0; \langle u \rangle) \varphi, \psi \rangle_{\mathcal{Y}}$$

is boundedly invertible, uniformly with respect to $u \in \tilde{X}$ where $\tilde{X} \subset X$ is an open neighborhood of the nominal instance $\langle u \rangle \in X$ of the uncertain parameter. In particular, there exists a constant $\beta > 0$ such that there holds

$$\forall u \in \tilde{X} : \begin{aligned} \inf_{0 \neq \varphi \in \mathcal{X}} \sup_{0 \neq \psi \in \mathcal{Y}} \frac{{}_{\mathcal{Y}'} \langle (D_q \mathcal{R})(q_0; u) \varphi, \psi \rangle_{\mathcal{Y}}}{\|\varphi\|_{\mathcal{X}} \|\psi\|_{\mathcal{Y}}} &\geq \beta > 0, \\ \inf_{0 \neq \psi \in \mathcal{Y}} \sup_{0 \neq \varphi \in \mathcal{X}} \frac{{}_{\mathcal{Y}'} \langle (D_q \mathcal{R})(q_0; u) \varphi, \psi \rangle_{\mathcal{Y}}}{\|\varphi\|_{\mathcal{X}} \|\psi\|_{\mathcal{Y}}} &\geq \beta > 0 \end{aligned} \tag{27.8}$$

and

$$\forall u \in \tilde{X} : \| (D_q \mathcal{R})(q_0, u) \|_{\mathcal{L}(\mathcal{X}, \mathcal{Y}')} = \sup_{0 \neq \varphi \in \mathcal{X}} \sup_{0 \neq \psi \in \mathcal{Y}} \frac{{}_{\mathcal{Y}'} \langle (D_q \mathcal{R})(q_0; u) \varphi, \psi \rangle_{\mathcal{Y}}}{\|\varphi\|_{\mathcal{X}} \|\psi\|_{\mathcal{Y}}} \leq \beta^{-1}. \tag{27.9}$$

The inf-sup conditions (27.8) and (27.9) are implied, for linear, self-adjoint PDEs such as the diffusion equation in Example 1, by the more familiar concept of *coercivity*.

Example 3. In the context of Example 1, one verifies with $\mathcal{Y} = \mathcal{X} = H_0^1(D)$ that

$$\gamma' \langle (D_q \mathcal{R})(q_0; u)\varphi, \psi \rangle_{\mathcal{Y}} = \int_D \nabla \psi \cdot u(x) \nabla \varphi(x) dx.$$

In particular, for linear operator equations, the residual $\mathcal{R}(q; u)$ is linear with respect to q and the differential $(D_q \mathcal{R})(q_0; u)$ in (27.8) does not depend on q_0 . Note that *uniform validity of (27.8) for all realizations of the uncertain input u* implies a constraint on the set $\tilde{X} \subseteq X$ of admissible data: one may choose, for example, $X = L^\infty(D)$ and require u to take values on

$$\tilde{X} = \{u \in X : \text{essinf}(u) \geq c_0 > 0\}.$$

Then (27.8) (and thus also (27.7)) are implied by the *coercivity of $(D_q \mathcal{R})(q_0; u)$ in $\mathcal{X} = H_0^1(D)$*

$$\begin{aligned} \gamma' \langle (D_q \mathcal{R})(q_0; u)\varphi, \varphi \rangle_{\mathcal{Y}} \int_D \nabla \varphi \cdot u(x) \nabla \varphi(x) dx &\geq c_0 \int_D |\nabla \varphi(x)|^2 dx \\ &\geq c_0 \frac{1}{2} (1 + C_P) \|\varphi\|_{\mathcal{X}}^2, \end{aligned}$$

where $C_P(D) > 0$ denotes the constant in the *Poincaré-inequality* $\|\nabla \varphi\|_{L^2(D)}^2 \geq C_P \|\varphi\|_{L^2(D)}^2$, valid in \mathcal{X} uniformly for all inputs $u \in \tilde{X}$. The saddle point stability conditions (27.8) and the possibility of different trial and test function spaces are not necessary here.

Remark 1. The saddle point stability conditions (27.8) are, however, indispensable for indefinite variational problems such as the space-time formulation of the parabolic evolution problem (27.3). For (27.3), the inf-sup conditions (27.8) have been verified in [70, Appendix]. Consider, for further illustration, the *Helmholtz Equations* for the propagation of time-harmonic pressure amplitude $\pi(x, t)$ in an uncertain, linearly elastic medium, in a bounded, certain domain $D \subset \mathbb{R}^d$, with homogeneous Dirichlet boundary conditions on ∂D .

Separation of variables $\pi(x, t) = \exp(i\omega t)q(x)$ implies the Helmholtz equation,

$$-\operatorname{div}(u(x) \nabla q(x)) - \omega^2 q(x) = f(x) \quad \text{in } D, \quad q|_{\partial D} = 0. \quad (27.10)$$

One chooses again $\mathcal{X} = \mathcal{Y} = H_0^1(D)$ and $u \in \tilde{X}$ as in Example 3. Then, for large frequency $\omega > 0$, the saddle point stability conditions (27.8) hold only if $\omega^2 \notin \bigcup_{u \in \tilde{X}} \Sigma(A(\cdot; u))$, where $\Sigma(A(\cdot; u)) \subset \mathbb{R}_{>0}$ denotes the (discrete and countable)

spectrum of the second order elliptic operator $A(q; u) := -\operatorname{div}(u(x)\nabla q(x)) \in \mathcal{L}(\mathcal{X}; \mathcal{X}')$ as is revealed by an straightforward eigenfunction argument. The stability constant β in (27.8) is $\beta = \inf_{\lambda \in \Sigma(A(\cdot; u)), u \in \tilde{X}} \{|\lambda - \omega^2|\}$.

Under conditions (27.8) and (27.9), for every $u \in \tilde{X} \subseteq X$, there exists a unique, regular solution $q(u)$ of (27.5) which is uniformly bounded with respect to $u \in \tilde{X}$ in the sense that there exists a constant $C(F, \tilde{X}) > 0$ such that

$$\sup_{u \in \tilde{X}} \|q(u)\|_{\mathcal{X}} \leq C(F, \tilde{X}). \quad (27.11)$$

For (27.8), (27.9), (27.10), and (27.11) being valid, we shall say that the set $\{(q(u), u) : u \in \tilde{X}\} \subset \mathcal{X} \times \tilde{X}$ forms a *regular branch of nonsingular solutions*.

If the data-to-solution map $\tilde{X} \ni u \mapsto q(u)$ is also Fréchet differentiable with respect to u at every point of the regular branch $\{(q(u); u) : u \in \tilde{X}\} \subset \mathcal{X} \times \tilde{X}$, the dependence of the “forward map,” i.e., the mapping relating u to $q(u)$ with the branch of nonsingular solutions, is locally Lipschitz on \tilde{X} : there exists a Lipschitz constant $L(F, \tilde{X})$ such that

$$\forall u, v \in \tilde{X} : \|q(u) - q(v)\|_{\mathcal{X}} \leq L(F, \tilde{X}) \|u - v\|_X. \quad (27.12)$$

This follows from the identity $(D_u q)(u) = -(D_q \mathcal{R})^{-1}(D_u \mathcal{R})$ and from the isomorphism property $(D_u \mathcal{R}_q)(q_0; \langle u \rangle) \in \mathcal{L}_{iso}(\mathcal{X}, \mathcal{Y}')$ which is implied by (27.8) and (27.9) and from the continuity of the differential $D_q \mathcal{R}$ on the regular branch.

In what follows, the abstract setting (27.4) is considered with uniformly continuously differentiable mapping $\mathcal{R}(q; u)$ in a product of neighborhoods $B_X(\langle u \rangle; R) \times B_{\mathcal{X}}(q(\langle u \rangle); R) \subset X \times \mathcal{X}$ of sufficiently small radius $R > 0$, satisfying the structural assumption (27.5). In Proposition 1 and throughout what follows, $q(\langle u \rangle) \in \mathcal{X}$ denotes the unique regular solution of (27.5) at the nominal input $\langle u \rangle \in X$.

2.2 Uncertainty Parametrization

As mentioned in the introduction, a key step in computational UQ is the *parametrization of the uncertain input data* $u \in X$ in terms of a (possibly infinite) sequence $\mathbf{y} = (y_j)_{j \geq 1}$ of parameters, taking values in a *parameter domain* U .

In the particular case where $u \in X$ is a random variable taking values in (a subset \tilde{X} of) the Banach space X , probabilistic UQ involves probability measures on the space X of uncertain input data.

In *uncertainty parametrization*, X is assumed to be separable. This is guaranteed in particular when X is finite-dimensional, i.e., when *the uncertain input data consists of a finite number of parameters*.

In the case of *distributed uncertain input* $u \in X$, the data space X is infinite-dimensional; to facilitate uncertainty parametrization, X is assumed to admit an

unconditional Schauder basis Ψ : $X = \text{span}\{\psi_j : j \geq 1\}$. This is in particular the case for separable Hilbert spaces X . Then, every $u \in \tilde{X} \subset X$ can be parametrized *linearly* in this basis, i.e., it admits a *parametric representation with linear dependence on the parameters y_j* :

$$u = u(\mathbf{y}) := \langle u \rangle + \sum_{j \geq 1} y_j \psi_j \quad \text{for some } \mathbf{y} = (y_j)_{j \geq 1} \in U. \quad (27.13)$$

Some examples of linear uncertainty parametrizations (27.13) are: (i) Karhunen–Loëve expansions which arise, in particular, from a numerical PCA of a random field model of uncertain input u (see, e.g., [32, 69, 72, 73]), (ii) unconditional Schauder bases (see, e.g., [27]), (iii) wavelet or trigonometric bases, and (iv) isogeometric geometry parametrizations from computer aided design (see, e.g., [3] and the references there).

The representation (27.13) is not unique: rescaling y_j and ψ_j will not change u . One assumes, therefore, throughout what follows that *the basis sequence $\{\psi_j\}_{j \geq 1}$ is normalized such that the parameter domain is $U = [-1, 1]^{\mathbb{N}}$* . However, for *Gaussian random field inputs*, the assumption of bounded parameter ranges is not satisfied: see [46, 69] for parametric formulations in this case.

Note that often, in applications, the dependence of u on the parameters y_j is not granted; one distinguishes:

- (a) *Linear parametrization* (27.13)
- (b) *Affine or separable parametrization*: the uncertain/parametric heat conductivity is given by [59]

$$u(x, \mathbf{y}) = \sum_{j \geq 1} \theta_j(\mathbf{y}) \psi_j(x), \quad \mathbf{y} \in U. \quad (27.14)$$

where we emphasize that for every $j \geq 1$, each coordinate function $\theta_j(\mathbf{y})$ may depend on *all* coordinates $y_j \in \mathbf{y}$. A typical example of a separable uncertainty parametrization is the so-called thermal fin problem (cp. [59])

$$u(x, \mathbf{y}) = \sum_{j=1}^J \chi_{D_j}(x) 10^{y_j}, \quad (27.15)$$

where $D := \bigcup_{j=1}^J D_j$ is decomposed into J nonoverlapping subdomains D_j , $j = 1, \dots, J$; χ_{D_j} is a characteristic function such that $\chi_{D_j}(x) = 1$ if $x \in D_j$ and 0 otherwise; $y_j \in [-1, 1]$, $j = 1, \dots, J$.

- (c) *Nonlinear transformation of an affine parametrization*. This case occurs, for example, in log Gaussian models with a positivity constraint, such as a linear diffusion equation with a log Gaussian permeability; for illustration, consider the Dirichlet problem:

$$-\nabla \cdot (u(x, \omega) \nabla q(x, \omega)) = f(x) \quad \text{in } D, \quad q|_{\partial D} = 0. \quad (27.16)$$

Here, the uncertain input is a log Gaussian random field, i.e.,

$$u(x, \omega) = \exp(g(x, \omega)), \quad \text{where} \quad g(x, \omega) = \sum_{j \geq 1} Y_j(\omega) \psi_j(x), \quad Y_j \sim N(0, 1). \quad (27.17)$$

Due to Y_j taking values in all of \mathbb{R} with positive probability, in (27.17), the normalization of the terms is effected by requiring that the standard deviation of Y_j be one.

- (d) *Nonseparable, nonlinear parametric operator equations.* Examples of this class typically arise in problems of *domain uncertainty*: upon diffeomorphic transformation of the problem to a fixed nominal domain, we obtain a parametric problem with uncertain operator whose coefficients are rational functions of the parameters. We refer the reader to Example 4 ahead for illustration.

Bases in the uncertain input space X are, in general, not unique, even when fixing the scaling of the coordinates y_j in (27.13). Being bases of X , they are mathematically equivalent. *In the context of computational UQ, the concrete choice of basis can have a significant impact on the numerical stability of UQ algorithms.* For illustration we mention the (textbook) example $X = L^2(-1, 1)$, for which two bases are given by $\Psi_1 = \{1, x, x^2, \dots\}$ and $\Psi_2 = \{P_j(x) : j = 0, 1, 2, \dots\}$ denoting P_j the classical Legendre polynomial of degree $j \geq 0$, with normalization $P_j(1) = 1$. Both bases, Ψ_1 and Ψ_2 , as well as the trigonometric functions constituting a Karhunen–Loëve basis of $X = L^2(-1, 1)$, are *global*, meaning that their elements are supported in the entire domain $[-1, 1]$. Alternative bases of $X = L^2(-1, 1)$ with local supports are spline wavelet bases, such as the Haar wavelet basis. Most localized bases have an intrinsic limit on the approximation order which can be reached for sufficiently smooth uncertain input data. Uncertainty parametrization with localized bases can, however, substantially increase sparsity in the parametric forward map.

Norm-convergence of the series (27.13) in X is implied by the *summability condition*

$$\sum_{j \geq 1} \|\psi_j\|_X < \infty, \quad (27.18)$$

“Uncertain input data” can also signify *domain uncertainty*, i.e., the shape of the physical domain D in which the boundary value problem is considered is uncertain. Upon suitable domain parametrization, such problems also are covered by the ensuing parametric, variational formulation; domain uncertainty in the physical domain can be reduced by domain mapping to a parametric problem in a fixed, nominal domain D_0 . This was considered for (27.2) with parametric coefficient u depending on the shape of the [28] for a particular example, and the following, smooth and nonlinear elliptic problem from [23]; see also [52] for applications to artery variability in Hemodynamics, to [45] for application of reduced basis techniques in electromagnetic scattering.

Example 4 (Domain uncertainty). A nonlinear operator equation in a random domain [28] is considered. The basic approach to dealing with domain uncertainty consists in *domain mapping* to a fixed reference domain, and in transforming the mathematical model from the physical domain to a fixed reference domain which is assumed known. Note that in certain classes of mathematical models, such as, for example, elastic deformation of a continuous medium, such reference domains arise naturally; in mathematical elasticity, the reference domain would be referred to as *reference configuration*. Note also that the reference domain may not necessarily be attained by concrete realizations of the uncertain input. Also, in stochastic domain modelling, the reference domain need not coincide with the nominal domain.

In transforming to a fixed reference domain, parametric domain uncertainty is transferred to the differential operator on the reference domain. Due to the smooth, but highly nonlinear nature of the domain transformations, the resulting parametric differential operators on the reference domain exhibit, as a rule, highly nonlinear, rational dependence w.r. to the parameter sequence \mathbf{y} , even if the mathematical model is linear.

Here we consider in addition a mathematical model which is nonlinear, also w.r. to the state variable q taking values in the function space \mathcal{X} .

The mathematical model in the physical domain reads: given $\mathbf{y} \in U$, find $q(\mathbf{y}) : D_{u(\mathbf{y})} \rightarrow \mathbb{R}$ such that

$$-\Delta q(\mathbf{y}) + q^3(\mathbf{y}) = f \quad \text{in } D_{u(\mathbf{y})}, \quad q(\mathbf{y}) = 0 \quad \text{on } \partial D_{u(\mathbf{y})}, \quad (27.19)$$

where the random domain $D_{u(\mathbf{y})}$ is homeomorphic to the unit disc, and explicitly given by

$$D_{u(\mathbf{y})} := \{x = (r \cos(\theta), r \sin(\theta)) : 0 \leq r < u(\mathbf{y}; \theta), 0 \leq \theta < 2\pi\}, \quad \mathbf{y} \in U. \quad (27.20)$$

Here, the random radius $u(\mathbf{y})$, as defined in (27.13), is given explicitly by

$$u(0) = \langle u \rangle = 1 \text{ and } \psi_j = \frac{0.5}{j^\alpha} \sin(j\theta) \quad j \geq 1, \text{ where } \alpha > 2. \quad (27.21)$$

By T_u we denote a transformation map from the nominal domain $D_{\langle u \rangle}$, the unit disk of \mathbb{R}^2 centered at the origin, to the parametric domain D_u , $T_u(r \cos(\theta), r \sin(\theta)) := (u(\mathbf{y})r \cos(\theta), u(\mathbf{y})r \sin(\theta))$. The nonlinear operator equation (27.19) in the parametric, uncertain physical domain becomes a parametric, nonlinear equation in the fixed nominal domain, which reads as: given a parameter sequence $\mathbf{y} \in U$, find a parametric response $q(\mathbf{y}) : D_{\langle u \rangle} \rightarrow \mathbb{R}$ such that

$$\begin{cases} -\operatorname{div}(M(\mathbf{y})\nabla q(\mathbf{y})) + q^3(\mathbf{y})d(\mathbf{y}) = f d(\mathbf{y}) & \text{in } D_{\langle u \rangle}, \\ q(\mathbf{y}) = 0 & \text{on } \partial D_{\langle u \rangle}, \end{cases} \quad (27.22)$$

where $d(\mathbf{y})$ denotes the determinant of the Jacobian dT_u of the map T_u , given as $d(\mathbf{y}) = (u(\mathbf{y}))^2$;

$$M(\mathbf{y}) := d(\mathbf{y})dT_u^{-1}dT_u^{-\top} = \begin{pmatrix} 1 + (b(\mathbf{y}))^2 & -b(\mathbf{y}) \\ -b(\mathbf{y}) & 1 \end{pmatrix} \quad \text{where } b(\mathbf{y}) := \frac{\partial_\theta u(\mathbf{y})}{u(\mathbf{y})}. \quad (27.23)$$

This example fits into the abstract setting of Sect. 2.1 as follows: the uncertain datum $\mathbf{u} = u(\mathbf{y}; \cdot) \in X_t = C_{\text{per}}^t([0, 2\pi))$ where the degree of smoothness $t = t(\alpha)$ depends on the exponent $\alpha > 2$ in (27.21). The spaces \mathcal{X} and \mathcal{Y} then are function spaces on the nominal domain, and chosen as $\mathcal{X} = \mathcal{Y} = H_0^1(D_{\langle u \rangle})$.

We note that uncertain inputs \mathbf{u} with “higher regularity” (when measured in a smoothness scale $\{X_t\}_{t \geq 0}$ with $X = X_0 \supset X_1 \supset X_2 \supset \dots$ on the admissible input data) correspond to stronger decay of ψ_j : for $\mathbf{u} \in X_t \subset X$, in (27.13) the $\{\psi_j\}_{j \geq 1}$ are assumed scaled such that

$$\mathbf{b} := \{\|\psi_j\|_X\}_{j \geq 1} \in \ell^p(\mathbb{N}) \quad \text{for some } 0 < p = p(t) < 1, \quad (27.24)$$

where the sequence $\mathbf{b} = (b_j)_{j \geq 1}$ is given by $b_j := \|\psi_j\|_X$. We also introduce the subset

$$U = \left\{ \mathbf{y} \in [-1, 1]^\mathbb{N} : u(\mathbf{y}) := \langle \mathbf{u} \rangle + \sum_{j \geq 1} y_j \psi_j \in \tilde{X} \right\}. \quad (27.25)$$

See (27.21) in Example 4, where the exponent $\alpha > 2$ determines the (Hölder) smoothness of the domain transformation, and where the spaces X_t correspond to Hölder spaces of 2π -periodic functions.

Once an unconditional basis $\{\psi_j\}_{j \geq 1}$ of X has been chosen, every realization $\mathbf{u} \in X$ can be identified in a one-to-one fashion with the pair $(\langle \mathbf{u} \rangle, \mathbf{y})$ where $\langle \mathbf{u} \rangle$ denotes the *nominal instance* of the uncertain datum \mathbf{u} and \mathbf{y} is the coordinate vector in representation (27.13). Inserting (27.13) into (27.4), we obtain under Assumption 1 the *equivalent, countably-parametric form*: given $F : U \rightarrow \mathcal{Y}'$,

$$\text{find } q(\mathbf{y}; F) \in \mathcal{X} : \forall \mathbf{y} \in U : \mathcal{R}(q; \mathbf{y}) := A(q; \mathbf{y}) - F(\mathbf{y}) = 0 \quad \text{in } \mathcal{Y}'. \quad (27.26)$$

Remark 2. In what follows, by a slight abuse of notation, one identifies the subset U in (27.25) with the countable set of parameters from the infinite-dimensional parameter domain $U \subseteq \mathbb{R}^\mathbb{N}$ without explicitly writing so. The operator $A(q; u)$ in (27.5) then becomes, via the parametric dependence $u = u(\mathbf{y})$, a parametric operator family $A(q; u(\mathbf{y}))$ which one denotes (with slight abuse of notation) by $\{A(q; \mathbf{y}) : \mathbf{y} \in U\}$, with the parameter set $U = [-1, 1]^\mathbb{N}$ (again, one uses in what follows this definition in place of the set U as defined in (27.25)). If $A(q; \mathbf{y})$ in (27.5) is linear, one has $A(q; \mathbf{y}) = A(\mathbf{y})q$ with $A(\mathbf{y}) \in \mathcal{L}(\mathcal{X}, \mathcal{Y}')$. One does not assume,

however, that the maps $q \mapsto A(q; \mathbf{y})$ are linear in what follows, unless explicitly stated.

With this understanding, and under the assumptions (27.11) and (27.12), the operator equation (27.5) will admit, for every $\mathbf{y} \in U$, a unique solution $q(\mathbf{y}; F)$ which is, due to (27.11) and (27.12), uniformly bounded and depends Lipschitz continuously on the parameter sequence $\mathbf{y} \in U$: there holds

$$\sup_{\mathbf{y} \in U} \|q(\mathbf{y}; F)\|_{\mathcal{X}} \leq C(F, U), \quad (27.27)$$

and, if the local Lipschitz condition (27.12) holds, there exists a Lipschitz constant $L > 0$ such that

$$\|q(\mathbf{y}; F) - q(\mathbf{y}'; F)\|_{\mathcal{X}} \leq L(F, U) \|u(\mathbf{y}) - u(\mathbf{y}')\|_{\mathcal{X}}. \quad (27.28)$$

The Lipschitz constant in (27.28) is not, in general, equal to $L(F, \tilde{X})$ in (27.12): it depends on the nominal instance $\langle u \rangle \in X$ and on the choice of basis $\{\psi_j\}_{j \geq 1}$.

Unless explicitly stated otherwise, throughout what follows, we identify $q_0 = q(\mathbf{0}; F) \in \mathcal{X}$ in Proposition 1 with the solution of (27.4) at the nominal input $\langle u \rangle \in X$.

2.3 Parameter Sparsity in Forward UQ

In forward UQ for problems with distributed, uncertain input data, upon uncertainty parametrizations such as (27.13), the solution of the forward problem becomes, as functions of the parameters y_j in the sequence \mathbf{y} , a countably parametric map from the parameter space U to the solutions' state space \mathcal{X} . Efficient computational UQ for such problems is crucially related to the approximation of such countably parametric maps with *convergence rates which are independent of the dimension*, i.e., independent of the number of coordinates which are active in the approximation. Mathematical results are reviewed which allow to establish such approximation results, with a key insight being that the attainable convergence rate is independent of the dimension of the space of active parameters, and depends only on the “sparsity” of the parametric map which is to be approximated. One technique to verify parametric sparsity for a broad class of parametric problems is to verify the existence of suitable *holomorphic extensions* of the parameter to solution map into the complex domain. The existence of such extensions is closely related to polynomial approximations of the parametric maps. The version presented here applies the holomorphic extension of countably parametric maps for which polynomial approximations take the form of so-called generalized polynomial chaos expansions which will be described in detail below. The results presented in this section are detailed in [28, 29] and the references there.

2.3.1 (b, p)-Holomorphy

For $s > 1$, introduce the Bernstein ellipse in the complex plane

$$\mathcal{E}_s := \left\{ \frac{w + w^{-1}}{2} : 1 \leq |w| \leq s \right\} \subset \mathbb{C}, \quad (27.29)$$

which has semi axes of length $\frac{s+s^{-1}}{2} > 1$ and $\frac{s-s^{-1}}{2} > 0$ and denote

$$\mathcal{E}_\rho := \bigotimes_{j \geq 1} \mathcal{E}_{\rho_j} \subset \mathbb{C}^{\mathbb{N}}, \quad (27.30)$$

the tensorized poly-ellipse when $\rho := (\rho_j)_{j \geq 1}$ is a sequence of semi-axis sums $\rho_j > 1$. With the convention $\mathcal{E}_1 = [-1, 1]$, one also admits $\rho_j = 1$ in (27.30), so that $U \subseteq \mathcal{E}_\rho$.

Sparsity analysis of parametric maps $q : U \mapsto \mathcal{X} : y \mapsto q(y)$ as in [28] and the references there relies on holomorphic extensions of parametric solutions q from U to \mathcal{E}_ρ .

Definition 1. For a positive sequence $\mathbf{b} = (b_j)_{j \geq 1} \in \ell^p(\mathbb{N})$ for some $0 < p < 1$, a parametric mapping $U \ni y \mapsto q(y) \in X$ satisfies the (\mathbf{b}, p) -holomorphy assumption in the Hilbert space \mathcal{X} if and only if

1. For each $y \in U$, there exists a unique $q(y) \in X$ and the map $y \mapsto q(y)$ from U to \mathcal{X} is uniformly bounded, i.e.,

$$\sup_{y \in U} \|q(y)\|_{\mathcal{X}} \leq C_0, \quad (27.31)$$

for some finite constant $C_0 > 0$.

2. For some $0 < \varepsilon < 1$ there exists a constant $C_\varepsilon \geq C_0 > 0$ such that for any sequence $\rho := (\rho_j)_{j \geq 1}$ of semiaxis sums ρ_j strictly larger than 1 that is $(\mathbf{b}, \varepsilon)$ -admissible, i.e.,

$$\sum_{j=1}^{\infty} (\rho_j - 1) b_j \leq \varepsilon, \quad (27.32)$$

the parametric map $y \mapsto q(y) \in X$ admits a complex extension $z \mapsto q(z)$ (taking values in the complexification of the space \mathcal{X}) that is a holomorphic mapping with respect to each variable z_j on a set of the form $\mathcal{O}_\rho := \bigotimes_{j \geq 1} \mathcal{O}_{\rho_j}$, $\mathcal{O}_{\rho_j} \subset \mathbb{C}$ is an open set containing \mathcal{E}_{ρ_j} , and the modulus $\|q(z)\|_{\mathcal{X}}$ of this extension is bounded on \mathcal{E}_ρ in (27.30) according to

$$\sup_{z \in \mathcal{E}_\rho} \|q(z)\|_{\mathcal{X}} \leq C_\varepsilon. \quad (27.33)$$

The significance of (\mathbf{b}, p) holomorphy lies in the following facts: (a) solution of well-posed, countably possibly nonlinear parametric operator equations with (\mathbf{b}, p) holomorphic, parametric operator families are (\mathbf{b}, p) holomorphic; (b) (\mathbf{b}, p) holomorphic parametric solution maps $\{q(\mathbf{y}) : \mathbf{y} \in U\} \subset \mathcal{X}$ allow for tensorized so-called “polynomial chaos” approximations with dimension-independent N -term approximation rates which depend only on the summability exponent p of the sequence \mathbf{b} ; (c) (\mathbf{b}, p) holomorphic parametric solution maps $\{q(\mathbf{y}) : \mathbf{y} \in U\} \subset \mathcal{X}$ can also be constructively approximated by sparse, Smolyak-type interpolation methods; see [25, 26]; and (d) (\mathbf{b}, p) holomorphy is preserved under composition with holomorphic maps, in particular, for example, in the context of *Bayesian inverse problems*; see [66, 68] for details. (e) (\mathbf{b}, p) holomorphic parametric solution maps $\{q(\mathbf{y}) : \mathbf{y} \in U\} \subset \mathcal{X}$ allow for *low-parametric, reduced basis surrogates*. Points (b) and (c) are explained next, detailing in particular computational approximation strategies for the efficient computation of sparse approximations of countably-parametric solution families.

2.3.2 Sparse Polynomial Approximation

(\mathbf{b}, p) -holomorphy ensures ℓ^p summability of gpc Legendre coefficients of (\mathbf{b}, p) holomorphic parametric solution maps $\{q(\mathbf{y}) : \mathbf{y} \in U\} \subset \mathcal{X}$. To state the result, from [26], for any coefficient bound sequence $\mathbf{c} := (c_\nu)_{\nu \in \mathcal{F}} \subset \mathbb{R}$, one associates its downward closed envelope $\mathbf{c}^* := (c_\nu^*)_{\nu \in \mathcal{F}}$ defined by

$$\mathbf{c}_\nu^* := \sup_{\mu \geq \nu} |c_\mu^*|, \quad \nu \in \mathcal{F}, \quad (27.34)$$

where $\mu \geq \nu$ means that $\mu_j \geq \nu_j$ for all j . An index set $\Lambda \subset \mathcal{F}$ is downward closed if and only if

$$\nu \in \Lambda \quad \text{and} \quad \mu \leq \nu \Rightarrow \mu \in \Lambda. \quad (27.35)$$

For a summability exponent $p > 0$, one introduces the space $\ell_m^p(\mathcal{F})$ of sequences that have their downward closed envelope in $\ell^p(\mathcal{F})$. One approximates the parametric responses by truncating the tensorized Legendre (“generalized polynomial chaos”) series

$$q(\mathbf{y}) = \sum_{\nu \in \mathcal{F}} q_\nu P_\nu(\mathbf{y}), \quad (27.36)$$

where the convergence is understood to be unconditional (in particular, the limit exists and is independent of the particular enumeration of \mathcal{F}) and where the tensorized Legendre polynomials $P_\nu(\mathbf{y})$ are given by $P_\nu(\mathbf{y}) := \prod_{j \geq 1} P_{\nu_j}(y_j)$, with P_n denoting the univariate Legendre polynomial of degree n for the interval $[-1, 1]$ with the classical normalization $\|P_n\|_{L^\infty([-1, 1])} = |P_n(1)| = 1$. The series (27.36) may be rewritten as

$$q(\mathbf{y}) = \sum_{\nu \in \mathcal{F}} v_\nu L_\nu(\mathbf{y}), \quad (27.37)$$

where $L_v(\mathbf{y}) := \prod_{j \geq 1} L_{v_j}(y_j)$, with L_n denoting the version of P_n normalized in $L^2([-1, 1], \frac{dt}{2})$, i.e.,

$$q_v = \left(\prod_{j \geq 1} (1 + 2v_j) \right)^{1/2} v_v. \quad (27.38)$$

Theorem 1 ([25]). *For a (\mathbf{b}, p) -holomorphic, parametric map $U \ni \mathbf{y} \rightarrow q(\mathbf{y}) \in \mathcal{X}$ in a Hilbert space \mathcal{X} , the sequences $(\|q_v\|_{\mathcal{X}})_{v \in \mathcal{F}}$ and $(\|v_v\|_{\mathcal{X}})_{v \in \mathcal{F}}$ of (norms of) the tensorized Legendre coefficients belong to $\ell_m^p(\mathcal{F})$, and*

$$q(\mathbf{y}) = \sum_{v \in \mathcal{F}} q_v P_v = \sum_{v \in \mathcal{F}} v_v L_v, \quad (27.39)$$

holds in the sense of unconditional convergence in $L^\infty(U, \mathcal{X})$.

There exists a sequence $(\Lambda_N)_{N \geq 1}$, with $\#(\Lambda_N) = N$ of nested downward closed sets such that

$$\inf_{w \in \mathcal{X}_{\Lambda_N}} \|q - w\|_{L^\infty(U, \mathcal{X})} \leq C(N+1)^{-s}, \quad s = \frac{1}{p} - 1, \quad (27.40)$$

where for any finite set $\Lambda \subset \mathcal{F}$ one defines

$$\mathcal{X}_\Lambda := \text{span} \left\{ \sum_{v \in \Lambda} w_v \mathbf{y}^v : w_v \in \mathcal{X} \right\}. \quad (27.41)$$

2.3.3 Sparse Grid Interpolation

Polynomial interpolation processes on the spaces \mathcal{X}_Λ for general downward closed sets Λ of multiindices have been introduced and studied in [26]. Given $z := (z_j)_{j \geq 1}$, a sequence of pairwise distinct points of $[-1, 1]$, one associates with any finite subset $\Lambda \subset \mathcal{F}$ the following *sparse interpolation grid* in U :

$$\Gamma_\Lambda := \{z_v : v \in \Lambda\} \quad \text{where} \quad z_v := (z_{v_j})_{j \geq 1}. \quad (27.42)$$

If $\Lambda \subset \mathcal{F}$ is downward closed, then the sparse grid Γ_Λ is unisolvant for \mathbb{P}_Λ : for any function g defined in Γ_Λ and taking values in \mathcal{X} , there exists a unique sparse grid interpolation polynomial $I_\Lambda g$ in \mathbb{P}_Λ that coincides with g on Γ_Λ . The interpolation polynomial $I_\Lambda g \in \mathbb{P}_\Lambda \otimes \mathcal{X}$ can be computed recursively: if $\Lambda := \{v^1, \dots, v^N\}$ such that for any $k = 1 \dots, N$, $\Lambda_k := \{v^1, \dots, v^k\}$ is downward closed, then

$$I_\Lambda g = \sum_{i=1}^N g_{v^i} H_{v^i}, \quad (27.43)$$

where the polynomials $(H_v)_{v \in \Lambda}$ are a hierarchical basis of \mathbb{P}_Λ given by

$$H_v(\mathbf{y}) := \prod_{j \geq 1} h_{v_j}(y_j) \quad \text{where } h_0(t) = 1 \text{ and } h_k(t) = \prod_{j=0}^{k-1} \frac{t - z_j}{z_k - z_j}, \quad k \geq 1, \quad (27.44)$$

and where the coefficients g_{v^k} are recursively defined by

$$g_{v^1} := g(z_0), \quad g_{v^k+1} := g(z_{v^k+1}) - I_{\Lambda_k} g(z_{v^k+1}) = g(z_{v^k+1}) - \sum_{i=1}^k g_{v^i} H_{v^i}(z_{v^k+1}). \quad (27.45)$$

The sparse grid $\Gamma_\Lambda \subset U$ is unisolvant for the space \mathcal{X}_Λ of multivariate polynomials with coefficients in \mathcal{X} . The interpolation operator that maps functions defined on U with values in \mathcal{X} into \mathcal{X}_Λ can be computed by the recursion (27.43) if one admits $g_v \in \mathcal{X}$. Naturally, in this case, the coefficients g_v being elements of a function space cannot be exactly represented and must be additionally approximated, e.g., by a finite element or a collocation approximation in a finite-dimensional subspace $\mathcal{X}_h \subset \mathcal{X}$.

The following result recovers the best N -term approximation rate $\mathcal{O}(N^{-s})$ in (27.40) for the interpolation in \mathbb{P}_Λ different choice of downward closed sets Λ . See [25] for a proof.

Theorem 2. *For any (b, p) -holomorphic, \mathcal{X} -valued parametric map $\mathbf{y} \mapsto q(\mathbf{y})$ there exists a constant $C > 0$ and a nested sequence of downward closed sets $(\Lambda_n)_{N \geq 1}$ with $\#(\Lambda_N) = N$ for which*

$$\|q - I_{\Lambda_N} q\|_{L^\infty(U, \mathcal{X})} \leq C(N + 1)^{-s}, \quad s = \frac{1}{p} - 1. \quad (27.46)$$

Remark 3. The above theorem guarantees the *existence of some sparse grid interpolant* with dimension-independent error convergence rate. However, practical construction of a sparse grid is not a trivial task, which depends on specific problems. A dimension-adaptive algorithm has been proposed in [42] and further developed in [15, 54, 66]. This algorithm has been found to perform well in a host of examples from forward and inverse UQ. Its convergence and (quasi)optimality are, however, not yet justified mathematically.

3 Model Order Reduction

Given any sample $\mathbf{y} \in U$, an accurate solution of the forward PDE model (27.26) relies on a stable and consistent numerical solver with high precision, which typically requires a high-fidelity discretization of the PDE model and a computationally expensive solving of the corresponding algebraic system. Such a large-scale

computation for a large number of samples is the most critical challenge in UQ problems. This section outlines *model order reduction* (MOR for short) methods in order to effectively alleviate the computational burden while facilitating certified accuracy of the parametric solution as well as its related quantities of interest. The material in this section is related to developments during the past decade. Our presentation is therefore synoptic, and the reader is referred to the surveys [44, 50, 51] and the references there for more detailed elaboration, and further references.

3.1 High-Fidelity Approximation

At first, a stable and consistent high-fidelity approximation of the solution of the parametric problem (27.26) following [23] is presented. To guarantee the stability of the HiFi approximation at any given $\mathbf{y} \in U$, one considers the Petrov–Galerkin (PG) discretization in the one-parameter family of pairs of subspaces $\mathcal{X}_h \subset \mathcal{X}$ and $\mathcal{Y}_h \subset \mathcal{Y}$ with equal dimensions, i.e., $N_h = \dim(\mathcal{X}_h) = \dim(\mathcal{Y}_h) < \infty$, where h represents a discretization parameter, for instance, the meshwidth of a PG finite element discretization. To ensure the convergence of the HiFi PG solution $q_h \in \mathcal{X}_h$ to the exact solution $q \in \mathcal{X}$ as $h \rightarrow 0$, one assumes the subspace families \mathcal{X}_h and \mathcal{Y}_h to be dense in \mathcal{X} and \mathcal{Y} as the discretization parameter (being, e.g., a meshwidth or an inverse spectral order) $h \rightarrow 0$, i.e.,

$$\forall w \in \mathcal{X} : \lim_{h \rightarrow 0} \inf_{w_h \in \mathcal{X}_h} \|w - w_h\|_{\mathcal{X}} = 0, \text{ and } \forall v \in \mathcal{Y} : \lim_{h \rightarrow 0} \inf_{v_h \in \mathcal{Y}_h} \|v - v_h\|_{\mathcal{Y}} = 0. \quad (27.47)$$

Moreover, to quantify the convergence rate of the discrete approximation, one introduces a *scale of smoothness spaces* $\mathcal{X}^s \subset \mathcal{X} = \mathcal{X}^0$ and $\mathcal{Y}^s \subset \mathcal{Y} = \mathcal{Y}^0$ indexed by the smoothness parameter $s > 0$. Here, one has in mind for example spaces of functions with s extra derivatives in Sobolev or Besov spaces. Then, for appropriate choices of the subspaces \mathcal{X}_h and \mathcal{Y}_h hold the approximation properties: there exist constants $C_s > 0$ such that for all $0 < h \leq 1$ holds

$$\begin{aligned} \forall w \in \mathcal{X}^s : \inf_{w_h \in \mathcal{X}_h} \|w - w_h\|_{\mathcal{X}} &\leq C_s h^s \|w\|_{\mathcal{X}^s} \\ \text{and } \forall v \in \mathcal{Y}^s : \inf_{v_h \in \mathcal{Y}_h} \|v - v_h\|_{\mathcal{Y}} &\leq C_s h^s \|v\|_{\mathcal{Y}^s}. \end{aligned} \quad (27.48)$$

Here, the constant C_s is assumed independent of the discretization parameter h but may depend on the smoothness parameter s . For small values of h and/or if s is large, the PG discretization produces high-fidelity (HiFi) approximations $q_h \in \mathcal{X}_h$ of the true solution $q \in \mathcal{X}$ by solving

$$\text{given } \mathbf{y} \in U, \text{ find } q_h(\mathbf{y}) \in \mathcal{X}_h : \quad \mathbf{y}' \langle \mathcal{R}(q_h(\mathbf{y}); \mathbf{y}), v_h \rangle_{\mathcal{Y}} = 0 \quad \forall v_h \in \mathcal{Y}_h. \quad (27.49)$$

A globally convergent Newton iteration method can be applied to solve the nonlinear, parametric HiFi-PG approximation problem (27.49) numerically; see [23, 33] for details.

To establish the well-posedness of the HiFi-PG approximation problem (27.49) as well as the a priori and a posteriori error estimates for the approximate solution q_h , the following assumptions are imposed.

Assumption 2. Let $a(\cdot, \cdot; y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ denote the parametric bilinear form for each $y \in U$ associated with the Fréchet derivative of \mathcal{R} at q , i.e.

$$a(w, v; y) := {}_{\mathcal{Y}'} \langle D_q \mathcal{R}(q(y); y)(w), v \rangle_{\mathcal{Y}} \quad \forall w \in \mathcal{X}, \forall v \in \mathcal{Y}. \quad (27.50)$$

The following conditions are assumed to hold

A1 stability: the parametric bilinear form a satisfies the discrete HiFi-PG inf-sup condition

$$\forall y \in U : \inf_{0 \neq w_h \in \mathcal{X}_h} \sup_{0 \neq v_h \in \mathcal{Y}_h} \frac{a(w_h, v_h; y)}{\|w_h\|_{\mathcal{X}} \|v_h\|_{\mathcal{Y}}} =: \beta_h(y) \geq \beta_h > 0, \quad (27.51)$$

where the inf-sup constant $\beta_h(y)$ depends on h and on y and may vanish $\beta_h(y) \rightarrow 0$ as $h \rightarrow 0$.

A2 consistency: the best approximation satisfies the consistent approximation property

$$\forall y \in U : \lim_{h \rightarrow 0} \frac{1}{\beta_h^2(y)} \inf_{w_h \in \mathcal{X}_h} \|q(y) - w_h\|_{\mathcal{X}} = 0. \quad (27.52)$$

In view of the convergence rate in (27.48), (27.52) amounts to require $h^s/\beta_h^2(y) \rightarrow 0$ as $h \rightarrow 0$.

A3 local Lipschitz continuity: there exists ϵ_0 and $L > 0$ such that for all $w \in \mathcal{X}$ with $\|q(y) - w\|_{\mathcal{X}} \leq \epsilon_0$, there holds

$$\forall y \in U : \|D_q \mathcal{R}(q(y); y) - D_q \mathcal{R}(w; y)\|_{\mathcal{L}(\mathcal{X}, \mathcal{Y}')} \leq L \|q(y) - w\|_{\mathcal{X}}. \quad (27.53)$$

Assumption 2 is sufficient to guarantee the existence of a solution $q_h(y) \in \mathcal{X}_h$ of the HiFi-PG approximation problem (27.49) for any $y \in U$, which is locally unique and satisfies a priori error estimate. The results are presented in the following theorem, whose proof follows that in [60].

Theorem 3. Under Assumption 2, there exists $h_0 > 0$ and $\eta_0 > 0$ such that for $0 < h \leq h_0$, there exists a solution $q_h(y) \in \mathcal{X}_h$ of the HiFi-PG approximation

problem (27.49), which is unique in $\mathcal{B}_{\mathcal{X}}(q(\mathbf{y}); \eta_0 \beta_h(\mathbf{y}))$. Moreover, for $0 < h \leq h_0$, there holds the *a priori* error estimate

$$\|q(\mathbf{y}) - q_h(\mathbf{y})\|_{\mathcal{X}} \leq 2 \frac{\|a(\mathbf{y})\|}{\beta(\mathbf{y})} \left(1 + \frac{\|a(\mathbf{y})\|}{\beta_h(\mathbf{y})}\right) \inf_{w_h \in \mathcal{X}_h} \|q(\mathbf{y}) - w_h\|_{\mathcal{X}}, \quad (27.54)$$

where $\|a(\mathbf{y})\| := \|D_q \mathcal{R}(q(\mathbf{y}); \mathbf{y})\|_{\mathcal{L}(\mathcal{X}, \mathcal{Y})}$. Depending on the smoothness parameter $s > 0$ (see (27.48)) and the polynomial degree $r \geq 1$ of the Finite Element space, one has

$$\inf_{w_h \in \mathcal{X}_h} \|q(\mathbf{y}) - w_h\|_{\mathcal{X}} \leq Ch^k \|q(\mathbf{y})\|_{\mathcal{X}^s}, \quad k = \min\{s, r\}, \quad (27.55)$$

where C is independent of the mesh size h and uniformly bounded w.r.t. \mathbf{y} . Moreover, one has the *a posteriori* error estimate

$$\|q(\mathbf{y}) - q_h(\mathbf{y})\|_{\mathcal{X}} \leq \frac{4}{\beta(\mathbf{y})} \|\mathcal{R}(q_h(\mathbf{y}); \mathbf{y})\|_{\mathcal{Y}}. \quad (27.56)$$

In many (but not all) practical applications in UQ, the lower bound of the stability constants $\beta(\mathbf{y})$ and $\beta_h(\mathbf{y})$ in Assumption 2 are independent of \mathbf{y} and of h : consider specifically the parametric elliptic diffusion problem in Example 3. In this example, one has that for $\mathcal{X} = \mathcal{Y} = H_0^1(D)$ holds, for every $\mathbf{y} \in U$, that $\beta_h(\mathbf{y}) \geq \beta(\mathbf{y}) \geq c_0(1 + C_P)/2$.

3.2 Reduced Basis Compression

In order to avoid too many computationally expensive numerical forward solutions of the HiFi-PG problem (27.49) at a large number of required samples $\mathbf{y} \in U$, one computes surrogate solutions with certified accuracy at low cost by applying reduced basis (RB) compression techniques [4, 61, 63]. The rationale for this lies in that the intrinsic dimension of the solution manifold $\mathcal{M}_h := \{q_h(\mathbf{y}), \mathbf{y} \in U\}$ is often low, even if the dimension of parameter space is high or infinite, so that the parametric solution can be compressed into a low-dimensional subspace of the HiFi space.

One assumes available a pair of N -dimensional subspaces $\mathcal{X}_N \subset \mathcal{X}_h$ and $\mathcal{Y}_N \subset \mathcal{Y}_h$ with $N \ll N_h$, which are known as RB (trial and test) spaces, whose construction is detailed in the next section. The compressed RB-PG discretization of the HiFi-PG approximation (27.26) takes the following form.

$$\text{given } \mathbf{y} \in U, \text{ find } q_N(\mathbf{y}) \in \mathcal{X}_N : \quad \mathbf{y}' \langle \mathcal{R}(q_N(\mathbf{y}); \mathbf{y}), v_N \rangle_{\mathcal{Y}} = 0 \quad \forall v_N \in \mathcal{Y}_N, \quad (27.57)$$

which can be solved by a Newton iteration [23]. Note that the RB-PG problem (27.57) has a structure which is identical to that of the HiFi-PG problem (27.49),

except for the trial and test spaces, which indicate that the RB solution $q_N(\mathbf{y})$ is a PG compression/projection of the HiFi solution $q_h(\mathbf{y})$; specifically, a PG projection from the HiFi space into the RB space. To ensure the well-posedness of the RB solution, one makes the following assumptions.

Assumption 3. *Holding Assumption 2, with the same notation of the bilinear form $a(\cdot, \cdot; \mathbf{y}) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ defined as in (27.50), one makes the further assumptions that*

A1 stability: *the parametric bilinear form a satisfies the discrete RB-PG inf-sup condition: there holds*

$$\forall \mathbf{y} \in U : \inf_{0 \neq w_N \in \mathcal{X}_N} \sup_{0 \neq v_N \in \mathcal{Y}_N} \frac{a(w_N, v_N; \mathbf{y})}{\|w_N\|_{\mathcal{X}} \|v_N\|_{\mathcal{Y}}} =: \beta_N(\mathbf{y}) \geq \beta_N > 0, \quad (27.58)$$

where β_N is a lower bound of the inf-sup constant $\beta_N(\mathbf{y})$, which depends on N and on \mathbf{y} and may converge to the HiFi inf-sup constant $\beta_N(\mathbf{y}) \rightarrow \beta_h(\mathbf{y})$ as $N \rightarrow N_h$.

A2 consistency: *the best approximation satisfies the consistent approximation property*

$$\forall \mathbf{y} \in U : \lim_{N \rightarrow N_h} \frac{1}{\beta_N^2(\mathbf{y})} \inf_{w_N \in \mathcal{X}_N} \|q_h(\mathbf{y}) - w_N\|_{\mathcal{X}} = 0. \quad (27.59)$$

Proceeding as in [60], one can establish the following error estimates for the RB solution (see [23])

Theorem 4. *Under Assumption 3, there exist $N_0 > 0$ and $\eta'_0 > 0$ such that for $N \geq N_0$, there exists a solution $q_N(\mathbf{y}) \in \mathcal{X}_N$ of the RB-PG compression problem (27.57), which is unique in $\mathcal{B}_{\mathcal{X}}(q_h(\mathbf{y}); \eta'_0 \beta_N(\mathbf{y}))$. Moreover, for any $N \geq N_0$, there holds the a priori error estimate*

$$\|q_h(\mathbf{y}) - q_N(\mathbf{y})\|_{\mathcal{X}} \leq 2 \frac{\|a(\mathbf{y})\|}{\beta_h(\mathbf{y})} \left(1 + \frac{\|a(\mathbf{y})\|}{\beta_N(\mathbf{y})} \right) \inf_{w_N \in \mathcal{X}_N} \|q_h(\mathbf{y}) - w_N\|_{\mathcal{X}}. \quad (27.60)$$

Moreover, one has the a-posteriori error estimate

$$\|q_h(\mathbf{y}) - q_N(\mathbf{y})\|_{\mathcal{X}} \leq \frac{4}{\beta_h(\mathbf{y})} \|\mathcal{R}(q_N(\mathbf{y}); \mathbf{y})\|_{\mathcal{Y}}. \quad (27.61)$$

Remark 4. Note that both the a priori and the a posteriori error estimates of the RB solution turn out to be the same as those of the HiFi solution with different stability constants and different approximation spaces. These results are obtained as

a consequence of the fact that the RB-PG problem (27.57) is nothing different from the HiFi-PG problem (27.49) except in different approximation spaces.

3.3 Reduced Basis Construction

As the computational cost for the solution of the RB-PG problem (27.57) critically depends on the RB degrees of freedom (dof) N , one needs to construct the optimal RB space \mathcal{X}_N that is most “representative” for all the parametric solutions with required approximation accuracy, such that N is as small as possible. However, it is computationally unfeasible to obtain such a optimal subspace \mathcal{X}_N as it is an infinite dimensional optimization problem involving expensive HiFi solutions. In the following, two practical algorithms are presented that allow in practice quasi-optimal construction of RB trial spaces \mathcal{X}_N . Construction of the RB test space \mathcal{Y}_N is deferred to the next sections.

3.3.1 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) [11], also known as principle component analysis (PCA for short) in statistics or Karhunen–Loëve (KL for short) decomposition in stochastic analysis, aims to extract the maximum information/energy/variance from a finite number of available solution “snapshots”. Such solution snapshots could be, e.g., solutions at a finite set of parameter values in our context. In practice, the POD is determined from a finite training set $\Xi_t = \{\mathbf{y}^n \in U, n = 1, \dots, N_t\}$ with N_t random samples, and the corresponding HiFi solutions $q_h(\mathbf{y})$, $\mathbf{y} \in \Xi_t$. The POD basis functions are defined as follows [62]: let \mathbb{C} denote the correlation matrix with rank $N_r \leq N_t$, which is given by

$$\mathbb{C}_{mn} = (q_h(\mathbf{y}^m), q_h(\mathbf{y}^n))_{\mathcal{X}}, \quad m, n = 1, \dots, N_t; \quad (27.62)$$

let $(\lambda_n, \boldsymbol{\psi}_n)_{n=1}^{N_r}$ denote the eigenpairs of the correlation matrix \mathbb{C} , i.e.

$$\mathbb{C}\boldsymbol{\psi}_n = \lambda_n \boldsymbol{\psi}_n, \quad n = 1, \dots, N_r. \quad (27.63)$$

Then the POD basis functions are given by

$$\xi_h^n = \sum_{m=1}^{N_t} \frac{1}{\sqrt{\lambda_n}} \boldsymbol{\psi}_n^{(m)} q_h(\mathbf{y}^m), \quad n = 1, \dots, N_r. \quad (27.64)$$

In common practice, instead of assembling the large correlation matrix \mathbb{C} and compute its eigenpairs, one may apply singular value decomposition (SVD) method or its reduced version such as thin SVD [10, 62] in order to speed up the computation of the POD basis functions.

The POD basis functions are optimal in the “average” sense [62].

Proposition 2. Let $W = \{w_h^1, \dots, w_h^N\}$ denote any N -dimensional ($N \leq N_r$) orthonormal functions in \mathcal{X}_h , i.e. $(w_h^m, w_h^n)_\mathcal{X} = \delta_{mn}$, $m, n = 1, \dots, N$; let P_N^W denote the \mathcal{X} -projection operator on W , i.e.

$$P_N^W w_h = \sum_{n=1}^N (w_h, w_h^n)_\mathcal{X} w_h^n \quad \forall w_h \in \mathcal{X}_h. \quad (27.65)$$

Then POD basis functions $W_{\text{pod}} = \{\xi_h^1, \dots, \xi_h^N\}$ given by (27.64) are orthonormal and satisfy

$$W_{\text{pod}} = \underset{W \subset \mathcal{X}_h}{\operatorname{argmin}} \sum_{n=1}^{N_t} \|q_h(\mathbf{y}^n) - P_N^W q_h(\mathbf{y}^n)\|_\mathcal{X}^2. \quad (27.66)$$

Moreover,

$$\sum_{n=1}^{N_t} \|q_h(\mathbf{y}^n) - P_N^{W_{\text{pod}}} q_h(\mathbf{y}^n)\|_\mathcal{X}^2 = \sum_{n=N+1}^{N_r} \lambda_n. \quad (27.67)$$

Remark 5. Proposition 2 implies that the POD basis functions achieve the optimal compression measured in the ensemble of square \mathcal{X} -norm of the orthogonal projection error. Moreover, the ensemble of the projection errors can be bounded explicitly according to (27.67), which can serve as an error indicator to choose the suitable number of POD basis functions given certain requirement of accuracy. Due to its optimality, POD has been widely used for reduced basis construction in Hilbert spaces [7, 10, 75].

Remark 6. However, to compute the POD basis functions, one needs to compute the HiFi solution at a sufficiently large number of properly chosen random samples. The possibly large training set could be prohibitive for the given computational budget, especially for high-dimensional problems that require numerous samples.

3.3.2 Greedy Algorithm

In order to avoid solving too many HiFi-PG problems for the construction of the RB spaces with a relatively much smaller number of basis functions, one turns to a greedy algorithm [4, 6, 58, 61, 63], which only requires the same number of HiFi solutions as that of the RB basis functions. An abstract formulation of the greedy search algorithm reads: choose the first sample \mathbf{y}^1 such that

$$\mathbf{y}^1 := \underset{\mathbf{y} \in U}{\operatorname{argsup}} \|q_h(\mathbf{y})\|_\mathcal{X}, \quad (27.68)$$

at which one constructs the first RB space $\mathcal{X}_1 = \text{span}\{q_h(\mathbf{y}^1)\}$. Then, for $N = 1, 2, \dots$, one seeks the next sample \mathbf{y}^{N+1} such that

$$\mathbf{y}^{N+1} := \underset{\mathbf{y} \in U}{\operatorname{argsup}} \|q_h(\mathbf{y}) - q_N(\mathbf{y})\|_\mathcal{X}, \quad (27.69)$$

where $q_N(\mathbf{y})$ is the RB solution, and construct the new RB space $\mathcal{X}_{N+1} = \mathcal{X}_N \oplus \text{span}\{q_h(\mathbf{y}^{N+1})\}$. However, both (27.68) and (27.69) are infinite dimensional optimization problems and necessitate many HiFi solutions for the evaluation of the RB errors. In order to tackle this challenge, the true error (27.69) is replaced ideally by a tight error bound $\Delta_N(\mathbf{y})$ [61, 63], i.e.

$$c \Delta_N(\mathbf{y}) \leq \|q_h(\mathbf{y}) - q_N(\mathbf{y})\|_{\mathcal{X}} \leq C \Delta_N(\mathbf{y}) \quad \forall \mathbf{y} \in U, \quad (27.70)$$

with constants $0 < c \leq C < \infty$ possibly depending on \mathbf{y} , and preferably $\gamma := c/C \approx 1$. Meanwhile, one can relax the first sample such that $\|q_h(\mathbf{y}^1)\|_{\mathcal{X}} \geq \gamma \sup_{\mathbf{y} \in U} \|q_h(\mathbf{y})\|_{\mathcal{X}}$. It is crucial that the cost for the evaluation of the error bound $\Delta_N(\mathbf{y})$ should be so small that its evaluation at a large number of training samples remains feasible, i.e., the cost at each sample is effectively independent of the HiFi dof. The relaxation of the true error to an effective error bound leads to the development of the so-called weak greedy algorithm for which an a priori error estimate of the error incurred by RB compression is established in the following theorem.

Theorem 5. Let $d_N(\mathcal{M}_h, \mathcal{X}_h)$ denote the Kolmogorov N -width, i.e., the worst-case scenario error of the \mathcal{X} -projection of the HiFi solution $q_h(\mathbf{y}) \in \mathcal{M}_h$ in the optimal among all possible N -dimensional subspaces $\mathcal{Z}_N \subset \mathcal{X}_h$. Specifically,

$$d_N(\mathcal{M}_h, \mathcal{X}_h) := \inf_{\mathcal{Z}_N \subset \mathcal{X}_h} \sup_{\mathbf{y} \in U} \inf_{w_N \in \mathcal{Z}_N} \|q_h(\mathbf{y}) - w_N\|_{\mathcal{X}}. \quad (27.71)$$

Let σ_N denote the worst-case scenario RB compression error, i.e.

$$\sigma_N := \sup_{\mathbf{y} \in U} \|q_h(\mathbf{y}) - q_N(\mathbf{y})\|_{\mathcal{X}}. \quad (27.72)$$

Then the following results hold for the convergence rates of the RB compression error [4, 34]:

- If $d_N \leq C_0 N^{-\alpha}$ for some $C_0 > 0$ and $\alpha > 0$, and any $N = 1, 2, \dots$, then $\sigma_N \leq C_1 N^{-\alpha}$ for all $N = 1, 2, \dots$, where $C_1 := 2^{5\alpha+1} \gamma^{-2} C_0$;
- If $d_N \leq C_0 e^{-c_0 N^\alpha}$ for some $C_0 > 0$, $c_0 > 0$, $\alpha > 0$, and any $N = 1, 2, \dots$, then $\sigma_N \leq C_1 e^{-c_1 N^\alpha}$ for all $N = 1, 2, \dots$, where $C_1 := \sqrt{2C_0} \gamma^{-1}$ and $c_1 := 2^{-1-2\alpha} c_0$.

Proof. The proof of the results in the finite dimensional approximation spaces \mathcal{X}_h and \mathcal{M}_h follows those in [34] where \mathcal{X} is a Hilbert space and the solution manifold \mathcal{M} is a compact set in \mathcal{X} .

Remark 7. The above results indicate that the RB compression by the (weak) greedy algorithm achieves optimal convergence rates in comparison with the Kolmogorov width, in the case of both algebraic rate and exponential rate. However,

the Kolmogorov width is typically not available for general parametric problems. In our setting, i.e., for smooth parameter dependence, it can be bounded from above by the sparse interpolation error estimate in (27.46), i.e., with algebraic convergence rate N^{-s} . Exponential convergence rates are shown in [16] for a one-dimensional parametric problem whose solution is analytic w.r.t. the parameter.

Remark 8. Construction of an N -dimensional RB space only requires N HiFi solutions by the greedy algorithm, which dramatically reduces the computational cost for the RB construction as long as evaluation of the error bound is inexpensive with operation independent of the HiFi dof N_h .

3.4 Linear and Affine-Parametric Problems

To illustrate the reduction in complexity which can be achieved by RB compression, linear and affine problems (e.g., Examples 1 and 3) with the uncertain parametrization given in Sect. 2.2 are first considered, for which one assumes the terms in (27.26) can be written more explicitly as

$$A(q; \mathbf{y}) = A(\mathbf{y})q = \sum_{j \geq 0} y_j A_j q \quad \text{and} \quad F(\mathbf{y}) = \sum_{j \geq 0} y_j F_j, \quad (27.73)$$

where one sets $y_0 = 1$ for notational simplicity and where $A_j \in \mathcal{L}(\mathcal{X}, \mathcal{Y}')$, and $F_j \in \mathcal{Y}'$, $j \geq 0$. The parametrization (27.73) is sometime also called *linear parametrization* uncertainty, whereas the term *affine parametrization* refers to separable expansions of the form (27.73) with y_j replaced by functions $\theta_j(\mathbf{y})$ where $\theta_0 = 1$ and where the θ_j , $j \geq 1$ depend on several or all parameters $y_j \in \mathbf{y}$, but are independent of the physical coordinates.

In computational practice, one truncates the affine expansion up to $J + 1$ terms with $J \in \mathbb{N}$, depending on the required accuracy of the truncation. For notational convenience, one defines $\mathbb{J} = \{0, 1, \dots, J\}$. The ensuing development applies verbatim in the affine-parametric case, when the parameters y_j are replaced by $\theta_j(\mathbf{y})$ for $j \in \mathbb{J}$, with functions that are independent of the physical variable and where each $\theta_j(\mathbf{y})$ possibly depends on all coordinates $y_j \in \mathbf{y}$.

3.4.1 High-Fidelity Approximation

Under the linear and affine assumptions, the parametric HiFi-PG approximation problem (27.49) becomes

given $\mathbf{y} \in U$, find $q_h(\mathbf{y}) \in \mathcal{X}_h$:

$$\sum_{j \in \mathbb{J}} y_j \langle A_j q_h(\mathbf{y}), v_h \rangle_{\mathcal{Y}} = \sum_{j \in \mathbb{J}} y_j \langle F_j, v_h \rangle_{\mathcal{Y}} \quad \forall v_h \in \mathcal{Y}_h. \quad (27.74)$$

By $(w_h^n)_{n=1}^{N_h}$ and $(v_h^n)_{n=1}^{N_h}$ one denotes the basis functions of the HiFi trial and test spaces \mathcal{X}_h and \mathcal{Y}_h . Then, the parametric solution $q_h(\mathbf{y})$ can be written as

$$q_h(\mathbf{y}) = \sum_{n=1}^{N_h} q_h^n(\mathbf{y}) w_h^n, \quad (27.75)$$

where $\mathbf{q}_h(\mathbf{y}) = (q_h^1(\mathbf{y}), \dots, q_h^{N_h}(\mathbf{y}))^\top$ denotes the (parametric) coefficient vector of the HiFi PG solution $q_h(\mathbf{y})$. The algebraic formulation of (27.74) reads:

$$\text{given } \mathbf{y} \in U, \text{ find } \mathbf{q}_h(\mathbf{y}) \in \mathbb{R}^{N_h} : \quad \sum_{j \in \mathbb{J}} y_j \mathbb{A}_j^h \mathbf{q}_h(\mathbf{y}) = \sum_{j \in \mathbb{J}} y_j \mathbf{f}_j^h. \quad (27.76)$$

The HiFi matrix $\mathbb{A}_j^h \in \mathbb{R}^{N_h \times N_h}$ and the HiFi vector $\mathbf{f}_j^h \in \mathbb{R}^{N_h}$ can be assembled as

$$(\mathbb{A}_j^h)_{mn} =_{\mathcal{Y}'} \langle A_j w_h^n, v_h^m \rangle_{\mathcal{Y}} \text{ and } (\mathbf{f}_j^h)_m =_{\mathcal{Y}'} \langle F_j, v_h^m \rangle_{\mathcal{Y}} \quad m, n = 1, \dots, N_h, j \in \mathbb{J}. \quad (27.77)$$

3.4.2 Reduced Basis Compression

Analogously, by $(w_N^n)_{n=1}^N$ and $(v_N^n)_{n=1}^N$ one denotes the basis functions of the RB trial and test spaces \mathcal{X}_N and \mathcal{Y}_N , so that the RB solution $q_N(\mathbf{y})$ can be written as

$$q_N(\mathbf{y}) = \sum_{n=1}^N q_N^n(\mathbf{y}) w_N^n, \quad (27.78)$$

with the coefficient vector $\mathbf{q}_N(\mathbf{y}) = (q_N^1(\mathbf{y}), \dots, q_N^N(\mathbf{y}))^\top$. Then, the parametric RB-PG compression problem can be written in the algebraic formulation as

$$\text{given } \mathbf{y} \in U, \text{ find } \mathbf{q}_N(\mathbf{y}) \in \mathbb{R}^N : \quad \sum_{j \in \mathbb{J}} y_j \mathbb{A}_j^N \mathbf{q}_N(\mathbf{y}) = \sum_{j \in \mathbb{J}} y_j \mathbf{f}_j^N. \quad (27.79)$$

where the RB matrix $\mathbb{A}_j^N \in \mathbb{R}^{N \times N}$ and the RB vector $\mathbf{f}_j^N \in \mathbb{R}^N$ are obtained as

$$\mathbb{A}_j^N = \mathbb{V}^\top \mathbb{A}_j^h \mathbb{W} \text{ and } \mathbf{f}_j^N = \mathbb{V}^\top \mathbf{f}_j^h, \quad j \in \mathbb{J}, \quad (27.80)$$

where \mathbb{W} and \mathbb{V} are the transformation matrices between the HiFi and RB basis functions, i.e.,

$$w_N^n = \sum_{m=1}^{N_h} \mathbb{W}_{mn} w_h^m \text{ and } v_N^n = \sum_{m=1}^{N_h} \mathbb{V}_{mn} v_h^m, \quad n = 1, \dots, N. \quad (27.81)$$

Thanks to the linear and affine structure of the parametric terms in (27.73), one can assemble the RB matrices \mathbb{A}_j^N and the RB vectors \mathbf{f}_j^N , $j \in \mathbb{J}$, once and for all. For each given \mathbf{y} , one only needs to assemble and solve the RB algebraic system (27.79) with computational cost depends only on N as $O(N^2)$ for assembling and $O(N^3)$ for solving (27.79), which leads to considerable computational reduction as long as $N \ll N_h$.

3.4.3 Tight A Posteriori Error Bound

A tight and inexpensive error bound that facilitates the weak greedy algorithm for RB construction is designed based on Assumption 2, in particular **A1 stability**, where the bilinear form is defined as

$$\text{given any } \mathbf{y} \in U : \quad a(w, v; \mathbf{y}) :=_{\mathcal{Y}'} \langle A(\mathbf{y})w, v \rangle_{\mathcal{Y}}, \quad \forall w \in \mathcal{X}, \forall v \in \mathcal{Y}, \quad (27.82)$$

which satisfies the stability condition in the HiFi spaces \mathcal{X}_h and \mathcal{Y}_h as in (27.51). Let the linear form be defined as

$$f(v; \mathbf{y}) =_{\mathcal{Y}'} \langle F(\mathbf{y}), v \rangle_{\mathcal{Y}}, \quad \forall v \in \mathcal{Y}, \quad (27.83)$$

then the RB residual in the HiFi space is defined as

$$r(v_h; \mathbf{y}) = f(v_h; \mathbf{y}) - a(q_N(\mathbf{y}), v_h; \mathbf{y}), \quad \forall v_h \in \mathcal{Y}_h. \quad (27.84)$$

Let $e_N(\mathbf{y}) = q_h(\mathbf{y}) - q_N(\mathbf{y})$ denote the RB error, then by the stability condition (27.51) one has

$$\|e_N(\mathbf{y})\|_{\mathcal{X}} \leq \frac{|a(e_N(\mathbf{y}), v_h; \mathbf{y})|}{\beta_h \|v_h\|_{\mathcal{Y}}} = \frac{|r(v_h; \mathbf{y})|}{\beta_h \|v_h\|_{\mathcal{Y}}} \leq \frac{\|r(\cdot; \mathbf{y})\|_{\mathcal{Y}'}}{\beta_h} =: \Delta_N(\mathbf{y}), \quad (27.85)$$

which indicates that $\Delta_N(\mathbf{y})$ is a rigorous upper error bound for the RB error $e_N(\mathbf{y})$. On the other hand, to see the lower bound one defines the Riesz representation of the residual as $\hat{e}_N(\mathbf{y}) \in \mathcal{Y}_h$, i.e.,

$$(\hat{e}_N(\mathbf{y}), v_h)_{\mathcal{Y}} = r(v_h; \mathbf{y}), \quad \forall v_h \in \mathcal{Y}_h, \quad (27.86)$$

so that $\|\hat{e}_N(\mathbf{y})\|_{\mathcal{Y}} = \|r(\cdot; \mathbf{y})\|_{\mathcal{Y}'}$. By setting $v_h = \hat{e}_N(\mathbf{y})$ in (27.86), one has

$$\|\hat{e}_N(\mathbf{y})\|_{\mathcal{Y}}^2 = r(\hat{e}_N(\mathbf{y}); \mathbf{y}) = a(e_N(\mathbf{y}), \hat{e}_N(\mathbf{y}); \mathbf{y}) \leq \alpha_h \|e_N(\mathbf{y})\|_{\mathcal{X}} \|\hat{e}_N(\mathbf{y})\|_{\mathcal{Y}}, \quad (27.87)$$

where α_h is the continuity constant of the bilinear form a in $\mathcal{X}_h \times \mathcal{Y}_h$, which implies that

$$\frac{\beta_h}{\alpha_h} \Delta_N(\mathbf{y}) \leq \|e_N(\mathbf{y})\|_{\mathcal{X}}. \quad (27.88)$$

Therefore, the error bound $\Delta_N(\mathbf{y})$ is tight with the constants in (27.69) as $c = \beta_h/\alpha_h$ and $C = 1$, and $\gamma = c/C = \beta_h/\alpha_h$.

For the evaluation of $\Delta_N(\mathbf{y})$, one makes use of the affine structure (27.73) by computing the Riesz representation \mathcal{A}_j^n of the linear functional $a_j(w_N^n; \cdot) = {}_{\mathcal{Y}'} \langle A_j w_N^n, \cdot \rangle_{\mathcal{Y}} : \mathcal{Y}_h \rightarrow \mathbb{R}$ as the solution of

$$(\mathcal{A}_j^n, v_h)_{\mathcal{Y}} = a_j(w_N^n; v_h), \quad \forall v_h \in \mathcal{Y}_h, \quad j \in \mathbb{J}, n = 1, \dots, N, \quad (27.89)$$

where w_N^n is the n th RB basis function. Analogously, one computes the Riesz representation \mathcal{F}_j of the linear functional $f_j(\cdot) = {}_{\mathcal{Y}'} \langle F_j, \cdot \rangle_{\mathcal{Y}} : \mathcal{Y}_h \rightarrow \mathbb{R}$ as the solution of

$$(\mathcal{F}_j, v_h) = f_j(v_h), \quad \forall v_h \in \mathcal{Y}_h, \quad j \in \mathbb{J}. \quad (27.90)$$

Finally, one can compute the dual norm of the residual in the error bound $\Delta_N(\mathbf{y})$ by

$$\begin{aligned} \|\hat{e}_N(\mathbf{y})\|_{\mathcal{Y}}^2 &= \sum_{j, j' \in \mathbb{J}} y_j y_{j'} \left((\mathcal{F}_j, \mathcal{F}_{j'})_{\mathcal{Y}} - 2 \sum_{n=1}^N q_N^n(\mathbf{y}) (\mathcal{F}_j, \mathcal{A}_{j'}^n)_{\mathcal{Y}} \right. \\ &\quad \left. + \sum_{n, n'=1}^N q_N^n(\mathbf{y}) q_N^{n'}(\mathbf{y}) (A_j^n, A_{j'}^{n'})_{\mathcal{Y}} \right), \end{aligned}$$

where $(\mathcal{F}_j, \mathcal{F}_{j'})_{\mathcal{Y}}$, $(\mathcal{F}_j, \mathcal{A}_{j'}^n)_{\mathcal{Y}}$, and $(A_j^n, A_{j'}^{n'})_{\mathcal{Y}}$, $j, j' \in \mathbb{J}$, $n, n' = 1, \dots, N$, can be computed once and for all. Given any \mathbf{y} , one only need to assemble (27.91) whose cost depends on N as $O(N^2)$, not on N_h , which results in effective computational reduction as long as $N \ll N_h$.

The lower bound of the stability constant β_h in $\Delta_N(\mathbf{y})$ can be computed for once based on the specific structure of the parametrization (e.g., at extreme points $\mathbf{y} = \{y_j = \pm 1 : j = 1, 2, \dots\}$, or by a successive constraint method (SCM for short) [48, 49] for each \mathbf{y} , whose computational cost is independent of N_h .

3.4.4 Stable RB-PG Compression

Construction of the RB trial space \mathcal{X}_N by both POD and greedy algorithm ensures the consistency of the RB-PG compression. For its stability, a suitable RB test space \mathcal{Y}_N needs to be constructed depending on \mathcal{X}_N . In the case that $\mathcal{X} = \mathcal{Y}$, $\mathcal{X}_h = \mathcal{Y}_h$, and the linear problem with (27.73) is coercive, the choice $\mathcal{Y}_N := \mathcal{X}_N$ guarantees the coercivity (or stability) of the RB Galerkin compression.

In the case of saddle point variational formulations of the forward problem, such as time-harmonic acoustic or electromagnetic wave propagation, or when $\mathcal{X}_h \neq \mathcal{Y}_h$, MOR requires in addition to a reduction of the trial spaces also the numerical computation of a suitable inf-sup stable testfunction space. To this end, the so-called “supremizer” approach was proposed in [64], which is described as: denote by

$T_y : \mathcal{X}_h \rightarrow \mathcal{Y}_h$ a parameter dependent supremizer operator, which is defined by

$$(T_y w_h, v_h)_\mathcal{Y} = a(w_h, v_h; \mathbf{y}) \quad \forall v_h \in \mathcal{Y}_h. \quad (27.91)$$

This definition implies $\sup_{v_h \in \mathcal{Y}_h} |a(w_h, v_h; \mathbf{y})| = |a(w_h, T_y w_h; \mathbf{y})|$, i.e. $T_y w_h$ is the supremizer of w_h in \mathcal{Y}_h w.r.t. the bilinear form a . Then the \mathbf{y} -dependent RB test space \mathcal{Y}_N^y is constructed as

$$\mathcal{Y}_N^y = \text{span}\{T_y w_N, w_N \in \mathcal{X}_N\}. \quad (27.92)$$

For this construction, it holds that (see [20])

$$\beta_N(\mathbf{y}) := \inf_{w_N \in \mathcal{X}_N} \sup_{v_N \in \mathcal{Y}_N^y} \frac{a(w_N, v_N; \mathbf{y})}{\|w_N\|_{\mathcal{X}} \|v_N\|_{\mathcal{Y}}} \geq \beta_h(\mathbf{y}). \quad (27.93)$$

This implies that *inf-sup stability of the HiFi-PG discretization is inherited by the corresponding PG-RB trial and test spaces*: all RB-PG compression problems are inf-sup stable under the stability assumption of the HiFi-PG approximation: **A1 stability** in Assumption 2. In particular, if the HiFi-PG discretizations are inf-sup stable uniformly with respect to the uncertain input parameter u (resp. its parametrization in terms of \mathbf{y}), so is any PG-RB method obtained with PG trial space \mathcal{X}_N obtained by a greedy search, and the corresponding PG test space (27.92).

Due to the affine structure (27.73), one can compute $T_y w_N$ for each $\mathbf{y} \in U$ as

$$T_y w_N = \sum_{j \in \mathbb{J}} y_j T_j w_N, \quad \text{where } (T_j w_N, v_h)_\mathcal{Y} = a_j(w_N, v_h) \quad \forall v_h \in \mathcal{Y}_h, \quad (27.94)$$

where $T_j w_N, w_N \in \mathcal{X}_N, j \in \mathbb{J}$ needs to be computed only once; given any \mathbf{y} , $T_y w_N$ can be assembled in $O(N)$ operations, which is independent of the number N_h of degrees of freedom in the HiFi-PG discretization. The compressed RB-PG discretization (27.79) can be written more explicitly as

$$\text{given } \mathbf{y} \in U, \text{ find } \mathbf{q}_N(\mathbf{y}) \in \mathbb{R}^N : \quad \sum_{j, j' \in \mathbb{J}} y_j y_{j'} \mathbb{A}_{j, j'}^N \mathbf{q}_N(\mathbf{y}) = \sum_{j, j' \in \mathbb{J}} y_j y_{j'} \mathbf{f}_{j, j'}^N, \quad (27.95)$$

where the RB compressed stiffness matrix $\mathbb{A}_{j, j'}^N \in \mathbb{R}^{N \times N}$ and the RB compressed load vector $\mathbf{f}_{j, j'}^N \in \mathbb{R}^N$ are given by

$$\mathbb{A}_{j, j'}^N = \mathbb{W}^\top (\mathbb{A}_{j'}^h)^\top \mathbb{M}_h^{-1} \mathbb{A}_j^h \mathbb{W} \text{ and } \mathbf{f}_{j, j'}^N = \mathbb{W}^\top (\mathbb{A}_{j'}^h)^\top \mathbb{M}_h^{-1} \mathbf{f}_j^h, \quad j, j' \in \mathbb{J}, \quad (27.96)$$

where \mathbb{M}_h denotes the mass matrix with $(\mathbb{M}_h^{-1})_{n, n'} = (v_h^n, v_h^{n'})_\mathcal{Y}, n, n' = 1, \dots, N_h$. Since all these quantities are independent of \mathbf{y} , one can precompute these quantities once and for all.

Remark 9. The stable RB-PG compression is equivalent to a least-squares RB-PG compression presented in [10]; see also [62]. Alternatively, a minimum residual approach known as double greedy algorithm [31] can be applied for the construction of \mathcal{Y}_N to ensure inf-sup stability.

3.5 Nonlinear and Nonaffine-Parametric Problems

The linearity of the (integral or differential) operator in the forward model and the affine parameter dependence play a crucial role in effective decomposition of the parameter-dependent and parameter-independent quantities. This, in turn, leads to a massive reduction in computational work for the RB-PG compression. For more general problems that involve nonlinear terms w.r.t. the state variable q and/or nonaffine terms w.r.t. the parameter y , for instance, Example 4, it is necessary to obtain an affine approximation of these terms in order to retain the effective decomposition and RB reduction. In this section, such an affine approximation based on empirical interpolation [2, 12, 18, 40, 57] is presented.

3.5.1 High-Fidelity Approximation

To solve the nonlinear parametric HiFi-PG approximation problem (27.49), one applies a Newton iteration method based on the parametric tangent operator of the nonlinear residual [23]: given any $y \in U$ and an initial guess of the solution $q_h^{(1)}(y) \in \mathcal{X}_h$, for $k = 1, 2, \dots$, one finds $\delta q_h^{(k)}(y) \in \mathcal{X}_h$ such that

$$y' \langle D_q \mathcal{R}(q_h^{(k)}(y); y)(\delta q_h^{(k)}(y)), v_h \rangle_{\mathcal{Y}} = -y' \langle \mathcal{R}(q_h^{(k)}(y); y), v_h \rangle_{\mathcal{Y}} \quad \forall v_h \in \mathcal{Y}_h; \quad (27.97)$$

then the solution is updated according to

$$q_h^{(k+1)}(y) = q_h^{(k)}(y) + \eta^{(k)} \delta q_h^{(k)}(y), \quad (27.98)$$

where $\eta^{(k)}$ is a constant determined by a line search [33]. The Newton iteration is stopped once

$$\|\delta q_h^{(k)}(y)\|_{\mathcal{X}} \leq \varepsilon_{tol} \quad \text{or} \quad \|\mathcal{R}(q_h^{(k)}(y); y)\|_{\mathcal{Y}'} \leq \varepsilon_{tol}, \quad (27.99)$$

being ε_{tol} a tolerance; then one sets $q_h(y) = q_h^{(k+1)}(y)$.

With the bases $\{w_h^n\}_{n=1}^{N_h}$ and $\{v_h^n\}_{n=1}^{N_h}$ of \mathcal{X}_h and \mathcal{Y}_h , respectively, one can write

$$q_h^{(k)}(y) = \sum_{n=1}^{N_h} q_{h,n}^{(k)}(y) w_h^n \quad \text{and} \quad \delta q_h^{(k)}(y) = \sum_{n=1}^{N_h} \delta q_{h,n}^{(k)}(y) w_h^n, \quad (27.100)$$

with the coefficient vectors $\mathbf{q}_h^{(k)}(\mathbf{y}) = \left(q_{h,1}^{(k)}(\mathbf{y}), \dots, q_{h,N_h}^{(k)}(\mathbf{y})\right)^\top$ and $\delta\mathbf{q}_h^{(k)}(\mathbf{y}) = \left(\delta q_{h,1}^{(k)}(\mathbf{y}), \dots, \delta q_{h,N_h}^{(k)}(\mathbf{y})\right)^\top$ so that the algebraic formulation of the parametric HiFi-PG approximation problem (27.97) reads: find the coefficient vector $\delta\mathbf{q}_h^{(k)}(\mathbf{y}) := \left(\delta q_{h,1}^{(k)}(\mathbf{y}), \dots, \delta q_{h,N_h}^{(k)}(\mathbf{y})\right)^\top \in \mathbb{R}^{N_h}$ such that

$$\mathbb{J}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right) \delta\mathbf{q}_h^{(k)}(\mathbf{y}) = -\mathbf{r}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right), \quad (27.101)$$

where the Jacobian matrix $\mathbb{J}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right) \in \mathbb{R}^{N_h \times N_h}$ is given by

$$\left(\mathbb{J}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right) \right)_{nn'} = {}_{\mathcal{Y}'} \left\langle D_q \mathcal{R}(q_h^{(k)}(\mathbf{y}); \mathbf{y})(w_h^{n'}), v_h^n \right\rangle_{\mathcal{Y}}, \quad n, n' = 1, \dots, N_h, \quad (27.102)$$

and the residual vector $\mathbf{r}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right) \in \mathbb{R}^{N_h}$ takes the form

$$\left(\mathbf{r}_h \left(\mathbf{q}_h^{(k)}(\mathbf{y}); \mathbf{y} \right) \right)_n = {}_{\mathcal{Y}'} \left\langle \mathcal{R} \left(q_h^{(k)}(\mathbf{y}); \mathbf{y} \right), v_h^n \right\rangle_{\mathcal{Y}}, \quad n = 1, \dots, N_h. \quad (27.103)$$

3.5.2 Reduced Basis Compression

To solve the nonlinear parametric RB-PG compression problem, one applies the same Newton iteration method as for solving the HiFi-PG approximation problem. More specifically, starting from an initial guess of the solution $q_N^{(1)}(\mathbf{y}) \in \mathcal{X}_N$ for any given $\mathbf{y} \in U$, for $k = 1, 2, \dots$, one finds $\delta q_N^{(k)} \in \mathcal{X}_N$ such that

$$\begin{aligned} & {}_{\mathcal{Y}'} \langle D_q \mathcal{R} \left(q_N^{(k)}(\mathbf{y}); \mathbf{y} \right) \left(\delta q_N^{(k)}(\mathbf{y}) \right), v_N \rangle_{\mathcal{Y}} \\ &= -{}_{\mathcal{Y}'} \langle \mathcal{R} \left(q_N^{(k)}(\mathbf{y}); \mathbf{y} \right), v_N \rangle_{\mathcal{Y}} \quad \forall v_N \in \mathcal{Y}_N; \end{aligned} \quad (27.104)$$

then the RB solution is updated by

$$q_N^{(k+1)}(\mathbf{y}) = q_N^{(k)}(\mathbf{y}) + \eta^{(k)} \delta q_N^{(k)}(\mathbf{y}), \quad (27.105)$$

where again $\eta^{(k)}$ is a constant determined by a line search method [33]. The stopping criterion is

$$\|\delta q_N^{(k)}(\mathbf{y})\|_{\mathcal{X}} \leq \varepsilon_{\text{tol}} \quad \text{or} \quad \|\mathcal{R}(q_N^{(k)}(\mathbf{y}); \mathbf{y})\|_{\mathcal{Y}'} \leq \varepsilon_{\text{tol}}, \quad (27.106)$$

With the notation of the basis $(w_N^n)_{n=1}^N$ and $(v_N^n)_{n=1}^N$ for the RB trial and test spaces \mathcal{X}_N and \mathcal{Y}_N , one can expand the RB solution $q_N^{(k)}(\mathbf{y})$ and its update $\delta q_N^{(k)}$ as

$$q_N^{(k)}(\mathbf{y}) = \sum_{n=1}^N q_N^{(n,k)}(\mathbf{y}) w_N^n \text{ and } \delta q_N^{(k)}(\mathbf{y}) = \sum_{n=1}^N \delta q_N^{(n,k)}(\mathbf{y}) w_N^n \quad (27.107)$$

with the coefficient vectors $\mathbf{q}_N^{(k)} = (q_N^{(1,k)}(\mathbf{y}), \dots, q_N^{(N,k)}(\mathbf{y}))^\top$ and $\delta \mathbf{q}_N^{(k)} = (\delta q_N^{(1,k)}(\mathbf{y}), \dots, \delta q_N^{(N,k)}(\mathbf{y}))^\top$. Then the algebraic formulation of the RB-PG compression (27.104) reads: find $\delta \mathbf{q}_N^{(k)} \in \mathbb{R}^N$ such that

$$\mathbb{J}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \delta \mathbf{q}_N^{(k)}(\mathbf{y}) = -\mathbf{r}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}), \quad (27.108)$$

where the parametric RB Jacobian matrix $\mathbb{J}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \in \mathbb{R}^{N \times N}$ and the parametric RB residual vector $\mathbf{r}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \in \mathbb{R}^N$ are given (through the transformation matrix \mathbb{W} and \mathbb{V}) by

$$\begin{aligned} \mathbb{J}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) &= \mathbb{V}^\top \mathbb{J}_h(\mathbb{W} \mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \mathbb{W} \\ \text{and } \mathbf{r}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) &= \mathbb{V}^\top \mathbf{r}_h(\mathbb{W} \mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \end{aligned} \quad (27.109)$$

One can observe that, due to the nonlinearity (and/or nonaffinity) of the residual operator, neither the residual vector nor the Jacobian matrix allows affine decomposition into parameter-dependent and parameter-independent terms. This obstructs effective RB-PG compression.

3.5.3 Empirical Interpolation

The empirical interpolation method (EIM) was originally developed for affine decomposition of nonaffine parametric functions [2]. It was later applied to decompose nonaffine parametric discrete function [12] (known as discrete EIM) and nonaffine-parametric operator [40, 55]. In this presentation, it is applied to decompose the residual vector $\mathbf{r}_h^{(k)}$ and its tangent derivative in (27.103) and (27.102).

For notational simplicity, suppose one has M_t training (residual) vectors

$$\mathbf{r}_m^t \in \mathbb{R}^{N_h}, \quad m = 1, \dots, M_t, \quad (27.110)$$

for instance, collected from the residual vectors $\mathbf{r}_h(\mathbf{q}_h^{(k)}; \mathbf{y})$ at successive iteration steps enumerate with the index k and at different samples \mathbf{y} . A *greedy algorithm* is

applied to construct the empirical interpolation for the approximation of any given $\mathbf{r} \in \mathbb{R}^{N_h}$: one picks the first EI basis $\mathbf{r}_1 \in \mathbb{R}^{N_h}$ as

$$\mathbf{r}_1 = \mathbf{r}_{m^*}^t, \quad \text{where } m^* = \operatorname{argmax}_{1 \leq m \leq M_t} \|\mathbf{r}_m^t\|_\infty, \quad (27.111)$$

where $\|\cdot\|_\infty$ can also be replaced by $\|\cdot\|_2$; then one chooses the first index $n_1 \in \{1, \dots, N_h\}$, such that

$$n_1 = \operatorname{argmax}_{1 \leq n \leq N_h} |(\mathbf{r}_1)_n|, \quad (27.112)$$

where $(\mathbf{r}_1)_n$ is the n -th entry of the vector \mathbf{r}_1 . The number of EI basis is set as M , and $M = 1$ for the time being. For any $\mathbf{r} \in \mathbb{R}^{N_h}$, it is approximated by the (empirical) interpolation

$$\mathcal{I}_M \mathbf{r} = \sum_{m=1}^M c_m \mathbf{r}_m, \quad (27.113)$$

where the coefficient vector $\mathbf{c} = (c_1, \dots, c_M)^\top$ is obtained by solving the interpolation problem

$$(\mathbf{r})_{m'} = \sum_{m=1}^M c_m (\mathbf{r}_m)_{m'}, \quad m' = n_1, \dots, n_M. \quad (27.114)$$

More explicitly, let $\mathbb{P}_M \in \{0, 1\}^{M \times N_h}$ denote an index indicator matrix with nonzero entries $(\mathbb{P}_M)_{m,n_m} = 1, m = 1, \dots, M$; let $\mathbb{R}_M \in \mathbb{R}^{N_h \times M}$ denote the EI basis matrix whose m -th column is $\mathbf{r}_m, m = 1, \dots, M$. Then, the coefficient vector \mathbf{c} can be written as

$$\mathbf{c} = (\mathbb{P}_M \mathbb{R}_M)^{-1} (\mathbb{P}_M \mathbf{r}), \quad (27.115)$$

and the empirical interpolation becomes

$$\mathcal{I}_M \mathbf{r} = \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbf{r}. \quad (27.116)$$

For $M = 1, 2, \dots$, the next EI basis $\mathbf{r}_{M+1} \in \mathbb{R}^{N_h}$ is constructed as

$$\mathbf{r}_{M+1} = \frac{\mathbf{r}_{m^*}^t - \mathcal{I}_M \mathbf{r}_{m^*}^t}{\|\mathbf{r}_{m^*}^t - \mathcal{I}_M \mathbf{r}_{m^*}^t\|_\infty}, \quad \text{where } m^* = \operatorname{argmax}_{1 \leq m \leq M_t} \|\mathbf{r}_m^t - \mathcal{I}_M \mathbf{r}_m^t\|_\infty, \quad (27.117)$$

and find the next index n_{M+1} as

$$n_{M+1} = \operatorname{argmax}_{1 \leq n \leq N_h} |(\mathbf{r}_{M+1})_n|. \quad (27.118)$$

The greedy algorithm is terminated when $|(\mathbf{r}_{M+1})_{n_{M+1}}| \leq \varepsilon_{\text{tol}}$. The empirical interpolation is consistent in that when $M \rightarrow N_h$, one has $\mathbf{r}_M \rightarrow \mathbf{0}$ due to the interpolation property. Moreover, an a priori error analysis shows that the greedy algorithm for EI construction leads to the same result for the convergence of the EI compression error in comparison with the Kolmogorov width as the bound stated in Theorem 5, except for a Lebesgue constant depending on M , see [23, 56] for more details.

For any $\mathbf{y} \in U$, let $\mathbf{q}_{N,M}(\mathbf{y}) \in \mathbb{R}^N$ denote the coefficient of the solution $q_{N,M}(\mathbf{y}) \in \mathcal{X}_N$ of the RB-PG compression problem with the empirical interpolation. By the empirical interpolation of the HiFi residual vector in (27.109), one can approximate the RB residual vector as

$$\mathbf{r}_N(\mathbf{q}_N^{(k)}; \mathbf{y}) \approx \mathbf{r}_{N,M}(\mathbf{q}_{N,M}^{(k)}; \mathbf{y}) := \mathbb{V}^\top \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbf{r}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}), \quad (27.119)$$

where the \mathbf{y} -independent quantity $\mathbb{V}^\top \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \in \mathbb{R}^{M \times M}$ can be computed once and for all, and the \mathbf{y} -dependent quantity $\mathbb{P}_M \mathbf{r}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y})$ can be evaluated in $O(MN)$ operations as long as locally supported HiFi basis functions are used, e.g., Finite Element basis functions.

Similarly, one can approximate the HiFi Jacobian matrix in (27.109) as

$$\mathbb{J}_N(\mathbb{W} \mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) \approx \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}), \quad (27.120)$$

so that the RB Jacobian matrix in (27.109) can be approximated by

$$\begin{aligned} \mathbb{J}_N(\mathbf{q}_N^{(k)}(\mathbf{y}); \mathbf{y}) &\approx \mathbb{J}_{N,M}(\mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}) \\ &:= \mathbb{V}^\top \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}) \mathbb{W}, \end{aligned} \quad (27.121)$$

where the \mathbf{y} -dependent quantity $\mathbb{P}_M \mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}) \mathbb{W}$ can be computed efficiently with $O(M^2N)$ operations, as long as the Jacobian matrix $\mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y})$ is sparse. This is typically the case for PG PDE approximation with locally supported (e.g., Finite Element) basis functions. Direct approximation of the HiFi Jacobian matrix \mathbb{J}_h by empirical interpolation has also been studied in [10].

By the above EI compression, $\mathbf{q}_{N,M}(\mathbf{y})$ is the solution of the problem

$$\mathbb{V}^\top \mathcal{I}_M \mathbf{r}_h(\mathbb{W} \mathbf{q}_{N,M}(\mathbf{y}); \mathbf{y}) = 0. \quad (27.122)$$

One observes that the RB solution (with EI) $\mathbf{q}_N(\mathbf{y}) \neq \mathbf{q}_{N,M}(\mathbf{y})$ due to the empirical interpolation error. Moreover, the RB-EI solution $q_{N,M}(\mathbf{y})$ converges to the RB solution $q_N(\mathbf{y})$ as $M \rightarrow N_h$.

3.5.4 Computable a Posterior Error Indicator

For the derivation of a posterior error indicator of the RB-EI solution $q_{N,M}(\mathbf{y})$ at any \mathbf{y} , recall the HiFi-PG problem in the algebraic formulation with slight abuse of notation: given any $\mathbf{y} \in U$, find $q_h(\mathbf{y}) \in \mathcal{X}_h$, such that

$$\mathbf{r}_h(q_h(\mathbf{y}); \mathbf{y}) = 0. \quad (27.123)$$

Analogously, recall the RB-EI-PG problem with slight abuse of notation: given any $\mathbf{y} \in U$, find $q_{N,M}(\mathbf{y}) \in \mathcal{X}_N$, such that

$$\mathbb{V}^\top \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) = 0. \quad (27.124)$$

Subtracting (27.123) from (27.124), inserting two zero terms, one has by rearranging

$$\begin{aligned} \mathbf{r}_h(q_h(\mathbf{y}); \mathbf{y}) - \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) &= -(\mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})) \\ &\quad - (\mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathbb{V}^\top \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})). \end{aligned} \quad (27.125)$$

Taking a suitable vector norm $||\cdot||_*$ on both sides (preferably equivalent to the dual norm \mathcal{Y} which could be realized by Riesz representation, multilevel preconditioning or wavelet bases in the HiFi space)

$$\begin{aligned} ||\mathbf{r}_h(q_h(\mathbf{y}); \mathbf{y}) - \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})||_* &= ||D_q \mathbf{r}_h((q_h(\mathbf{y}); \mathbf{y})(q_h(\mathbf{y}) - q_{N,M}(\mathbf{y})))||_* \\ &\quad + o(||q_h(\mathbf{y}) - q_{N,M}(\mathbf{y})||_{\mathcal{X}}) \\ &\geq \tilde{\beta}_h ||q_h(\mathbf{y}) - q_{N,M}(\mathbf{y})||_{\mathcal{X}}. \end{aligned} \quad (27.126)$$

The constant $\tilde{\beta}_h$ can be estimated numerically, e.g., at some extreme realization $\mathbf{y} = -\mathbf{1}$ or $\mathbf{1}$, or by SCM [48, 49]. On the other hand, the right hand side (RHS) of (27.126) can be bounded by

$$\begin{aligned} \text{RHS} &\leq ||\mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})||_* \\ &\quad + ||\mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathbb{V}^\top \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})||_*, \end{aligned} \quad (27.127)$$

where the first term accounts for the empirical interpolation error, which can be approximated by

$$\begin{aligned} &||\mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})||_* \\ &\approx ||\mathcal{I}_{M+M'} \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})||_* \text{ (by (27.113))} \\ &= \left\| \sum_{m=M+1}^{M+M'} c_m \mathbf{r}_m \right\|_*, \end{aligned} \quad (27.128)$$

where one assumes that $\mathcal{I}_{M+M'}$ for some constant $M' \in \mathbb{N}$, e.g., $M' = 2$, is a more accurate EI compression operator for the residual, so that $(\mathcal{I} - \mathcal{I}_M)\mathbf{r}_h \approx (\mathcal{I}_{M+M'} - \mathcal{I}_M)\mathbf{r}_h$. For any $\mathbf{y} \in U$, the coefficients $c_m, m = M + 1, \dots, M + M'$, can be evaluated by (27.115) with $O(M + M')$ operations. The quantity (27.128) can be computed with operations depending only on M and M' as long as the HiFi terms are assembled for only once.

The second term of (27.127) represents the RB compression error, which can be evaluated as (by noting that (27.124) holds)

$$\begin{aligned} & \| \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathbb{V}^\top \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) \|_* = \| \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) \|_* \\ &= \| \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) \|_* \end{aligned} \quad (27.129)$$

where the HiFi terms can be computed for only once; given any $\mathbf{y} \in U$, evaluation of $\mathbb{P}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y})$ takes $O(MN)$ operations. Therefore, once the \mathbf{y} -independent quantities are (pre)computed, evaluation of the a posteriori error indicators for both the EI compression error and the RB compression error can be achieved efficiently, with cost depending only on N and M for each given $\mathbf{y} \in U$.

Finally, one can define the a posteriori error indicator of the RB-EI compression error as

$$\Delta_N(\mathbf{y}) := \frac{1}{\beta_h} (\| \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) - \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) \|_* + \| \mathcal{I}_M \mathbf{r}_h(q_{N,M}(\mathbf{y}); \mathbf{y}) \|_*), \quad (27.130)$$

which can be efficiently evaluated for each $\mathbf{y} \in U$ with cost independent of the HiFi dof N_h .

3.5.5 RB-EI-PG Compression

By following the same procedure as in the linear and affine case in Sect. 3.4.4, a stable RB-EI-PG compression problem can be obtained by least-squares formulation as: given $\mathbf{y} \in U$, with some initial solution $\mathbf{q}_{N,M}^{(1)} \in \mathbb{R}^N$, for $k = 1, 2$, find $\delta \mathbf{q}_{N,M}^{(k)}(\mathbf{y}) \in \mathbb{R}^N$, such that

$$\mathbb{J}_{N,M}^s \left(\mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right) \delta \mathbf{q}_{N,M} = -\mathbf{r}_h^s \left(\mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right), \quad (27.131)$$

where the Jacobian matrix $\mathbb{J}_{N,M}^s(\mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y})$ with stabilization is given by

$$\begin{aligned} & \left(\mathbb{P}_M \mathbb{J}_h \left(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right) \mathbb{W} \right)^\top (\mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1})^\top \mathbb{M}_h^{-1} \mathbb{R}_M (\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbb{J}_h \\ & \left(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right) \mathbb{W}, \end{aligned} \quad (27.132)$$

with $(\mathbb{R}_M(\mathbb{P}_M \mathbb{R}_M)^{-1})^\top \mathbb{M}_h^{-1} \mathbb{R}_M(\mathbb{P}_M \mathbb{R}_M)^{-1}$ evaluated once and for all and with $\mathbb{P}_M \mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}) \mathbb{W}$ evaluated in $O(M^2 N)$ operations. The stabilized RB-EI residual vector $\mathbf{r}_h^s \left(\mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right)$ is given by

$$\begin{aligned} & \left(\mathbb{P}_M \mathbb{J}_h(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y}) \mathbb{W} \right)^\top (\mathbb{R}_M(\mathbb{P}_M \mathbb{R}_M)^{-1})^\top \mathbb{M}_h^{-1} \mathbb{R}_M(\mathbb{P}_M \mathbb{R}_M)^{-1} \mathbb{P}_M \mathbf{r}_h \\ & \left(\mathbb{W} \mathbf{q}_{N,M}^{(k)}(\mathbf{y}); \mathbf{y} \right), \end{aligned} \quad (27.133)$$

which can also be efficiently evaluated for each given \mathbf{y} with an additional $O(MN)$ operations. Then until a suitable termination criterion is met, e.g., $\|\delta \mathbf{q}_{N,M}^{(k)}\|_2 \leq \varepsilon_{\text{tol}}$, for a prescribed tolerance ε_{tol} and for a suitable vector norm $\|\cdot\|_*$ (cp. Section 3.5.4) the solution is updated as $\mathbf{q}_{N,M}^{(k+1)} = \mathbf{q}_{N,M}^{(k)} + \eta^{(k)} \delta \mathbf{q}_{N,M}^{(k)}$ with suitable constant $\eta^{(k)}$ obtained by a line search method.

3.6 Sparse Grid RB Construction

For the construction of the RB space \mathcal{X}_N , one can directly solve the optimization problem (27.69) with the true error replaced by suitable a posteriori error estimate, for instance,

$$\mathbf{y}^{N+1} := \underset{\mathbf{y} \in U}{\operatorname{argsup}} \Delta_N(\mathbf{y}). \quad (27.134)$$

This approach has been adopted in [9] by solving model-constrained optimization problems with Lagrangian formulation, which requires both full and reduced solution of adjoint problems, leading to possibly many more expensive solution of HiFi problems than the number of reduced basis functions. In very high- or infinite-dimensional parameter space, the optimization problem is typically very difficult to solve as there might be many local maximal points.

A more common approach is to replace the parameter space U by a training set Ξ_t , which consists of a finite number of samples that are rich enough to construct the most representative RB space, yet should be limited due to the constraint of computational cost. Hence, it remains to seek the next sample according to

$$\mathbf{y}^{N+1} := \underset{\mathbf{y} \in \Xi_t}{\operatorname{argmax}} \Delta_N(\mathbf{y}). \quad (27.135)$$

To choose the training samples, random sampling methods have been mostly used in practice [63]; adaptive sampling with certain saturation criteria [43] has also been developed recently to remove and add samples from the training set. In the present setting of uncertainty parametrization as introduced in Sect. 2.2, one takes advantage of the sparsity of the parametric data-to-solution map which is implied

by (b, p) -holomorphy. This sparsity allows for dimension-independent convergence rates of adaptive sparse grid sampling based on an adaptive construction of a generalized/anisotropic sparse grid in the high-dimensional parameter space. The basic idea is to build the RB space \mathcal{X}_N (and EI basis for nonlinear and nonaffine problems) in tandem with the adaptive construction of the sparse grid; see [15, 17] for more details. The advantages of this approach are threefold: the first is that the training samples as well as the sparse grid nodes for RB construction are “the most representative ones”; the second is that the computational cost for the sparse grid construction is reduced by replacing the HiFi solution at each sparse grid node by its RB surrogate solution. This provides a new algorithm for fast sparse grid construction with certificated accuracy; third, one can obtain an explicitly computable a priori error estimate for the RB compression error based on a computable bound of the sparse grid interpolation error, as stated in Theorem 2. However, these advantages are less pronounced if the parameter dependence of the parametric solution family of the forward UQ problem is less sparse; specifically, if the sparsity parameter p being $0 < p < 1$ in the (b, p) -holomorphic property becomes large and close to 1.

4 Inverse UQ

The abstract, parametric problems which arise in forward UQ in Sect. 2 consisted in computing, for given, admissible uncertain input datum $u \in X$ (respective for any parameter sequence y in the parametrization (27.13) of u , an approximate response $q(u) \in X$, respectively a *Quantity of Interest (QoI for short)* $\phi(q) \in \mathcal{Z}$ where $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{Z}$ is a continuous mapping, and \mathcal{Z} denotes a suitable space containing realizations of the QoI. If, for example, solution values $q(u)$ are of interest, one chooses $\mathcal{Z} = \mathcal{X}$, if $\phi(\cdot) \in \mathcal{X}'$, one has $\mathcal{Z} = \mathbb{R}$. The sparsity results in Sect. 2, in particular the constructive interpolation approximation result Theorem 2 and the MOR results in

4.1 Bayesian Inverse Problems for Parametric Operator Equations

Following [32, 66, 68, 71, 73], one equips the space of uncertain inputs X and the space of solutions \mathcal{X} of the forward maps with norms $\|\cdot\|_X$ and with $\|\cdot\|_{\mathcal{X}}$, respectively. Consider the abstract (possibly nonlinear) operator equation (27.5) where the uncertain operator $A(\cdot; u) \in C^1(\mathcal{X}, \mathcal{Y}')$ is assumed to be boundedly invertible, at least locally for the uncertain input u sufficiently close to a nominal input $\langle u \rangle \in X$, i.e. for $\|u - \langle u \rangle\|_X$ sufficiently small so that, for such u , the response of the forward problem (27.5) is uniquely defined. Define the *forward response map*, which relates a given uncertain input u and a given forcing F to the response q in (27.5) by

$$X \ni u \mapsto q(u) := G(u; F(u)), \text{ where } G(u, F) : X \times \mathcal{Y} \mapsto \mathcal{X}. \quad (27.136)$$

To ease notation, one does not list the dependence of the response on F and simply denotes the dependence of the forward solution on the uncertain input as $q(u) = G(u)$. Assume given an observation functional $\mathcal{O}(\cdot) : \mathcal{X} \rightarrow Y$, which denotes a *bounded linear observation operator* on the space \mathcal{X} of observed system responses in Y . Throughout the remainder of this paper, one assumes that there is a finite number K of sensors, so that $Y = \mathbb{R}^K$ with $K < \infty$. Then $\mathcal{O} \in \mathcal{L}(\mathcal{X}; Y) \simeq (\mathcal{X}')^K$. One equips $Y = \mathbb{R}^K$ with the Euclidean norm, denoted by $|\cdot|$. For example, if $\mathcal{O}(\cdot)$ is a K -vector of observation functionals $\mathcal{O}(\cdot) = (o_k(\cdot))_{k=1}^K$.

In this setting, one wishes to predict *computationally* an expected (under the Bayesian posterior) system response of the QoI, conditional on given, noisy measurement data δ . Specifically, the data δ is assumed to consist of observations of system responses in the data space Y , corrupted by additive observation noise, e.g., by a realization of a random variable η taking values in Y with law \mathbb{Q}_0 . One assumes additive, centered Gaussian noise on the observed data $\delta \in Y$. That is, the data δ is composed of the observed system response and the additive noise η according to $\delta = \mathcal{O}(G(u)) + \eta \in Y$. One assumes that $Y = \mathbb{R}^K$ and η is Gaussian, i.e., a random vector $\eta \sim \mathbb{Q}_0 \sim \mathcal{N}(0, \Gamma)$ with a positive definite covariance Γ on $Y = \mathbb{R}^K$ (i.e., a symmetric, positive definite covariance matrix $\Gamma \in \mathbb{R}_{\text{sym}}^{K \times K}$ which is assumed to be known. The *uncertainty-to-observation map* of the system $\mathcal{G} : X \rightarrow Y = \mathbb{R}^K$ is $\mathcal{G} = \mathcal{O} \circ G$, so that

$$\delta = \mathcal{G}(u) + \eta = (\mathcal{O} \circ G)(u) + \eta \in Y, \quad (27.137)$$

where $Y = L_\Gamma^2(\mathbb{R}^K)$ denotes random vectors taking values in $Y = \mathbb{R}^K$ which are square integrable with respect to the Gaussian measure on $Y = \mathbb{R}^K$. Bayes' formula [32, 73] yields a density of the Bayesian posterior with respect to the prior whose negative log-likelihood equals the observation noise covariance-weighted, least-squares functional (also referred to as "potential" in what follows) $\Phi_\Gamma : X \times Y \rightarrow \mathbb{R}$ by $\Phi_\Gamma(u; \delta) = \frac{1}{2} |\delta - \mathcal{G}(u)|_\Gamma^2$, i.e.,

$$\Phi_\Gamma(u; \delta) = \frac{1}{2} |\delta - \mathcal{G}(u)|_\Gamma^2 := \frac{1}{2} ((\delta - \mathcal{G}(u))^\top \Gamma^{-1} (\delta - \mathcal{G}(u))). \quad (27.138)$$

In [32, 73], an infinite-dimensional version of Bayes' rule was shown to hold in the present setting. In particular, the local Lipschitz assumption (27.12) on the solutions' dependence on the data implies a corresponding Lipschitz dependence of the Bayesian potential (27.138) on $u \in X$. Specifically, there holds the following version of Bayes' theorem. Bayes' Theorem states that, under appropriate continuity conditions on the uncertainty-to-observation map $\mathcal{G} = (\mathcal{O} \circ G)(\cdot)$ and on the prior measure π_0 on $u \in X$, for positive observation noise covariance Γ in (27.138), the posterior π^δ of $u \in X$ given data $\delta \in Y$ is absolutely continuous with respect to the prior π_0 .

Theorem 6 ([32, Thm. 3.3]). Assume that the potential $\Phi_\Gamma : X \times Y \mapsto \mathbb{R}$ is, for given data $\delta \in Y$, π_0 measurable on $(X, \mathcal{B}(X))$ and that, for \mathbb{Q}_0 -a.e. data $\delta \in Y$ there holds

$$Z := \int_X \exp(-\Phi(u; \delta)) \pi_0(du) > 0.$$

Then the conditional distribution of $u|\delta$ exists and is denoted by π^δ . It is absolutely continuous with respect to π_0 and there holds

$$\frac{d\pi^\delta}{d\pi_0}(u) = \frac{1}{Z} \exp(-\Phi(u; \delta)). \quad (27.139)$$

In particular, then, the Radon-Nikodym derivative of the Bayesian posterior w.r.t. the prior measure admits a bounded density w.r.t. the prior π_0 which is denoted by Θ , and which is given by (27.139).

4.2 Parametric Bayesian Posterior

The uncertain datum u in the forward equation (27.5) is parametrized as in (27.13). Motivated by [66, 68], the basis for the presently proposed deterministic quadrature approaches for Bayesian estimation via the computational realization of Bayes' formula is a *parametric, deterministic representation* of the derivative of the posterior measure π^δ with respect to the *uniform prior measure π_0 on the set U of coordinates in the uncertainty parametrization* (27.25). The prior measure π_0 being uniform, one admits in (27.13) sequences y which take values in the parameter domain $U = [-1, 1]^\mathbb{J}$, with an index set $\mathbb{J} \subset \mathbb{N}$. Consider the countably-parametric, deterministic forward problem in the probability space

$$(U, \mathcal{B}, \pi_0). \quad (27.140)$$

To ease notation, one assumes throughout what follows that the prior measure π_0 on the uncertain input $u \in X$, parametrized in the form (27.13), is the uniform measure (the ensuing derivations are still applicable if π_0 is absolutely continuous with respect to the uniform measure, with a smooth and bounded density). Being π_0 a countable product probability measure, this assumption implies the statistical independence of the coordinates y_j in the parametrization (27.13). With the parameter domain U as in (27.140) the parametric uncertainty-to-observation map $\Xi : U \rightarrow Y = \mathbb{R}^K$ is given by

$$\Xi(y) = \mathcal{G}(u) \Big|_{u=(u) + \sum_{j \in \mathbb{J}} y_j \psi_j}. \quad (27.141)$$

Our reduced basis approach is based on a parametric version of Bayes' theorem 6, in terms of the uncertainty parametrization (27.13). To present it, one views U as the unit ball in $\ell^\infty(\mathbb{J})$, the Banach space of bounded sequences taking values in U .

Theorem 7. Assume that $\Xi : \bar{U} \rightarrow Y = \mathbb{R}^K$ is bounded and continuous. Then $\pi^\delta(d\mathbf{y})$, the distribution of $\mathbf{y} \in U$ given data $\delta \in Y$, is absolutely continuous with respect to $\pi_0(d\mathbf{y})$, i.e. there exists a parametric density $\Theta(\mathbf{y})$ such that

$$\frac{d\pi^\delta}{d\pi_0}(\mathbf{y}) = \frac{1}{Z} \Theta(\mathbf{y}) \quad (27.142)$$

with $\Theta(\mathbf{y})$ given by

$$\Theta(\mathbf{y}) = \exp(-\Phi_\Gamma(u; \delta)) \Big|_{u=\langle u \rangle + \sum_{j \in \mathbb{J}} y_j \psi_j}, \quad (27.143)$$

with Bayesian potential Φ_Γ as in (27.138) and with normalization constant Z given by

$$Z = \mathbb{E}^{\pi_0}[\Theta] = \int_U \Theta(\mathbf{y}) d\pi_0(\mathbf{y}) > 0. \quad (27.144)$$

Bayesian inversion is concerned with the approximation of a “most likely” system response $\phi : X \rightarrow \mathcal{Z}$ (sometimes also referred to as *quantity of interest (QoI)* which may take values in a Banach space \mathcal{Z}) of the QoI ϕ , conditional on given (noisy) observation data $\delta \in Y$. In particular the choice $\phi(u) = G(u)$ (with $\mathcal{Z} = \mathcal{X}$) facilitates computation of the “most likely” (as expectation under the posterior, given data δ) system response. With the QoI ϕ one associates the countably-parametric map

$$\Psi(\mathbf{y}) = \Theta(\mathbf{y}) \phi(u) \Big|_{u=\langle u \rangle + \sum_{j \in \mathbb{J}} y_j \psi_j} = \exp(-\Phi_\Gamma(u; \delta)) \phi(u) \Big|_{u=\langle u \rangle + \sum_{j \in \mathbb{J}} y_j \psi_j} : U \rightarrow \mathcal{Z}. \quad (27.145)$$

Then the Bayesian estimate of the QoI ϕ , given noisy observation data δ , reads

$$\mathbb{E}^{\pi^\delta}[\phi] = Z'/Z, \quad Z' := \int_{\mathbf{y} \in U} \Psi(\mathbf{y}) \pi_0(d\mathbf{y}), \quad Z := \int_{\mathbf{y} \in U} \Theta(\mathbf{y}) \pi_0(d\mathbf{y}). \quad (27.146)$$

The task in computational Bayesian estimation is therefore to approximate the ratio $Z'/Z \in \mathcal{Z}$ in (27.146). In the parametrization with respect to $\mathbf{y} \in U$, Z and Z' take the form of infinite-dimensional, iterated integrals with respect to the prior $\pi_0(d\mathbf{y})$.

4.3 Well-Posedness and Approximation

For the computational viability of Bayesian inversion, the quantity $\mathbb{E}^{\pi^\delta}[\phi]$ should be stable under perturbations of the data δ and under changes in the forward problem stemming, for example, from discretizations as considered in Sects. 3.1 and 3.2.

Unlike deterministic inverse problems where the data-to-solution maps can be severely ill-posed, for $\Gamma > 0$ the expectations (27.146) are Lipschitz continuous with respect to the data δ , *provided that the potential Φ_Γ in (27.138) is locally Lipschitz with respect to the data δ* in the following sense.

Assumption 4. *Let $\tilde{X} \subseteq X$ and assume $\Phi_\Gamma \in C(\tilde{X} \times Y; \mathbb{R})$ is Lipschitz on bounded sets. Assume also that there exist functions $M_i : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ (depending on $\Gamma > 0$) which are monotone, non-decreasing separately in each argument, such that for all $u \in \tilde{X}$, and for all $\delta, \delta_1, \delta_2 \in B_Y(0, r)$*

$$\Phi(u; \delta) \geq -M_1(r, \|u\|_X), \quad (27.147)$$

and

$$|\Phi_\Gamma(u; \delta_1) - \Phi_\Gamma(u; \delta_2)| \leq M_2(r, \|u\|_X) \|\delta_1 - \delta_2\|_Y. \quad (27.148)$$

Under Assumption 4, the expectation (27.146) depends Lipschitz on δ (see [32, Sec. 4.1] for a proof):

$$\forall \phi \in L^2(\pi^{\delta_1}, X; \mathbb{R}) \cap L^2(\pi^{\delta_2}, X; \mathbb{R}) \quad \|\mathbb{E}^{\pi^{\delta_1}}[\phi] - \mathbb{E}^{\pi^{\delta_2}}[\phi]\|_{\mathcal{Z}} \leq C(\Gamma, r) \|\delta_1 - \delta_2\|_Y. \quad (27.149)$$

Below, one shall be interested in the impact of approximation errors in the forward response of the system (e.g., due to discretization and approximate numerical solution of system responses) on the Bayesian predictions (27.146). For continuity of the expectations (27.146) w.r.t. changes in the potential, the following assumption is imposed.

Assumption 5. *Let $\tilde{X} \subseteq X$ and assume $\Phi \in C(\tilde{X} \times Y; \mathbb{R})$ is Lipschitz on bounded sets. Assume also that there exist functions $M_i : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ which are monotonically non-decreasing separately in each argument, such that for all $u \in \tilde{X}$, and all $\delta \in B_Y(0, r)$, Eq. (27.147) is satisfied and*

$$|\Phi_\Gamma(u; \delta) - \Phi_\Gamma^N(u; \delta)| \leq M_2(r, \|u\|_X) \|\delta\|_Y \psi(N) \quad (27.150)$$

where $\psi(N) \rightarrow 0$ as $N \rightarrow \infty$.

By π_N^δ one denotes the Bayesian posterior, given data $\delta \in Y$, with respect to Φ_Γ^N .

Proposition 3. Under Assumption 5, and the assumption that for $\tilde{X} \subseteq X$ and for some bounded $B \subset X$ one has $\pi_0(\tilde{X} \cap B) > 0$ and

$$X \ni u \mapsto \exp(M_1(\|u\|_X))(M_2(\|u\|_X))^2 \in L^1_{\pi_0}(X; \mathbb{R}),$$

there holds, for every QoI $\phi : X \rightarrow \mathcal{Z}$ such that, although the convergence rate s can be substantially higher than the rate $1/2$ afforded by MCMC methods (cp. Sect. 4.4 and [32, Sections 5.1, 5.2]) that $\phi \in L^2_{\pi^\delta}(X; \mathcal{Z}) \cap L^2_{\pi_N^\delta}(X; \mathcal{Z})$ uniformly w.r.t. N , that $Z > 0$ in (27.144) and

$$\|\mathbb{E}^{\pi^\delta}[\phi] - \mathbb{E}^{\pi_N^\delta}[\phi]\|_{\mathcal{Z}} \leq C(\Gamma, r)\|\delta\|_Y \psi(N). \quad (27.151)$$

For a proof of Proposition 3, see [32, Thm. 4.7, Rem. 4.8].

Below, concrete choices are presented for the convergence rate function $\psi(N)$ in estimates (27.150), (27.151) in terms of i) “dimension truncation” of the uncertainty parametrization (27.13), i.e., to a finite number of $s \geq 1$ terms in (27.13), and ii) Petrov–Galerkin discretization of the dimensionally truncated problem, and iii) generalized polynomial chaos (gpc) approximation of the dimensionally truncated problem for particular classes of forward problems. The verification of the consistency condition (27.150) in either of these cases will be based on (cf. [35]).

Proposition 4. Assume given a sequence $\{q^N\}_{N \geq 1}$ of approximations to the parametric forward response $X \ni u \mapsto q(u) \in \mathcal{X}$ such that, with the parametrization (27.13),

$$\sup_{y \in U} \|(q - q^N)(y)\|_{\mathcal{X}} \leq \psi(N) \quad (27.152)$$

with a consistency error bound $\psi \downarrow 0$ as $N \rightarrow \infty$ monotonically and uniformly w.r.t. $u \in \tilde{X}$ (resp. w.r.t. $y \in U$). By G^N one denotes the corresponding (Galerkin) approximations of the parametric forward maps. Then the approximate Bayesian potential

$$\Phi^N(u; \delta) = \frac{1}{2}(\delta - \mathcal{G}^N(u))^\top \Gamma^{-1}(\delta - \mathcal{G}^N(u)) : X \times Y \mapsto \mathbb{R}, \quad (27.153)$$

where $\mathcal{G}^N := \mathcal{O} \circ G^N$, satisfies (27.150).

The preceding result shows that the consistency condition (27.152) for the approximate forward map q^N ensures corresponding consistency of the Bayesian estimate $\mathbb{E}^{\pi_N^\delta}[\phi]$, due to (27.151). Note that so far, no specific assumption on the nature of approximation of the forward map has been made. Using a MOR surrogate of the parametric forward model, Theorem 5 allows to bound $\psi(N)$ in (27.152) by the

corresponding worst-case RB compression error σ_N in (27.72) which, is bound by the convergence rate of the corresponding N -width:

$$\psi(N) \leq \sigma_N \lesssim d_N(\mathcal{M}_h; \mathcal{X}_h) \quad (27.154)$$

in the various cases indicated in Theorem 5. Under Assumption 4, Proposition 4 ensures that Assumption 5 holds, with (27.154). One concludes in particular that replacing the forward model by a reduced basis surrogate will result in an error in the Bayesian estimate of the same asymptotic order of magnitude, as $N \rightarrow \infty$. This justifies, for example, running Markov chains on the surrogate forward model obtained from MOR. In doing this, however, care must be taken to account for the constants implied by \lesssim in (27.154): the constants do not depend on N , but large values of these constants can imply prohibitive errors for the (small) values N of the number of RB degrees of freedom employed in PG projections of MOR forward surrogates. In addition, it is pointed out that the estimate (27.151) depends on the observation noise covariance Γ , as well as on the size r of the observation data δ (measured in Y).

4.4 Reduced Basis Acceleration of MCMC

Markov chain Monte Carlo (MCMC) methods compute the expectation $\mathbb{E}^{\pi^\delta}[\phi]$ in (27.146) under the posterior by sampling from the posterior density. They proceed in approximation of the constant Z' in (27.146) by sample averages where, however, *the posterior distribution from which samples are to be drawn is itself to be determined during the course of the sampling process*. MCMC methods start by sampling from the (known) prior π_0 , and by updating, in the course of sampling, both numerator Z' as well as the normalizing constant Z in (27.146). Several variants exist; see [32, Sections 5.1,5.2] for a derivation of the *Metropolis-Hastings MCMC*, and to [32, Section 5.3] for *sequential Monte Carlo (sMC) methods*. A convergence theory in terms of certain *spectral gaps* is provided in [32, Thm. 5.13], resulting in the convergence rate $N^{-1/2}$ with N denoting the number of increments of the chain. In the context of the present paper, N denotes the number of (approximate) solves of the parametric forward problem (27.4), resp. of a discretization of it. Due to the low rate of convergence $1/2$ of the MCMC methods (and due to the high rejection rate of the samplers during burn-in), generally a very large number of samples is required. Moreover, successive updates of the MCMC samplers have an intrinsically serial structure which, in turn, foils massive parallelism to compensate for the slow convergence rate. It is therefore of high interest to examine the possibility of accelerating MCMC methods. In [47], the impact of various discretization and acceleration techniques for MCMC methods were analyzed for the computation of the expectation $\mathbb{E}^{\pi^\delta}[\phi]$ in (27.146); among them a generalized polynomial chaos (gpc) surrogate of the parametric forward map

$U \ni y \rightarrow q(y) \in \mathcal{X}$. The theory in [47] can be extended using the consistency error bound Assumption 5 in the Bayesian potential, and Proposition 4 for the RB error in the forward map. Practical application and implementation of RB with MCMC for (Bayesian) inverse problems can be found, for instance, in [30, 51].

4.5 Dimension and Order Adaptive, Deterministic Quadrature

The parametric, deterministic infinite-dimensional integrals Z' and Z in (27.146) are, in principle, accessible to any quadrature strategy which is able to deal efficiently with the high dimension of the integration domain, and which is able to exploit (b, p) sparsity of the parametric integrand functions.

Following [25, 42], a greedy strategy based on reduced sets of indices which are neighboring the currently active set Λ , defined by

$$\mathcal{N}(\Lambda) := \{\nu \notin \Lambda : \nu - e_j \in \Lambda, \forall j \in \mathbb{I}_\nu \text{ and } \nu_j = 0, \forall j > j(\Lambda) + 1\}$$

for any downward closed index set $\Lambda \subset \mathcal{F}$ of currently active gpc modes, where $j(\Lambda) := \max\{j : \nu_j > 0 \text{ for some } \nu \in \Lambda\}$. This heuristic approach aims at controlling the global approximation error by locally collecting indices of the current set of neighbors with the largest estimates error contributions. In the following, the resulting algorithm to recursively build the downward closed index set Λ in the Smolyak quadrature which is adapted to the posterior density (and, due to the explicit expression (27.143) from Bayes' formula, also to the observation data δ) is summarized. The reader is referred to [42, 66, 68] for details and numerical results. Development and analysis of the combination of RB, MOR and ASG for Bayesian inversion are described in depth in [21–23], for both linear and nonlinear, both affine and nonaffine parametric problems.

```

1: function ASG
2:   Set  $\Lambda_1 = \{0\}$ ,  $k = 1$  and compute  $\Delta_0(\Xi)$ .
3:   Determine the reduced set (27.155) of neighbors  $\mathcal{N}(\Lambda_1)$ .
4:   Compute  $\Delta_\nu(\Xi)$ ,  $\forall \nu \in \mathcal{N}(\Lambda_1)$ .
5:   while  $\sum_{\nu \in \mathcal{N}(\Lambda_k)} \|\Delta_\nu(\Xi)\|_{\mathcal{S}} > tol$  do
6:     Select  $\nu$  from  $\mathcal{N}(\Lambda_k)$  with largest  $\|\Delta_\nu\|_{\mathcal{S}}$  and set  $\Lambda_{k+1} = \Lambda_k \cup \{\nu\}$ .
7:     Determine the reduced set (27.155) of neighbors  $\mathcal{N}(\Lambda_{k+1})$ .
8:     Compute  $\Delta_\nu(\Xi)$ ,  $\forall \nu \in \mathcal{N}(\Lambda_{k+1})$ .
9:     Set  $k = k + 1$ .
10:   end while
11: end function
```

4.6 Quasi-Monte Carlo Quadrature

The adaptive, deterministic quadrature based on active multiindex sets determined by the algorithm ASG in Section 12 realizes, in practical experiments [66, 68], convergence rates $s = 1/p - 1$ which are determined only by the summability exponent p of the sequence \mathbf{b} of (X -norms of) the basis Ψ adopted for the space X , in order to parametrize the uncertain input data u as in (27.13). The downside/drawback of Algorithm 12 is that although the (dimension-independent) convergence rate s can be substantially higher than the rate $1/2$ afforded by MCMC methods (cp. Sect. 4.4), provided the summability exponent p is sufficiently small, it is intrinsically sequential in nature, due to the recursive construction of the active index sets; in this respect, it is analogous to MCMC methods which access the forward model (or a surrogate of it) through uncertainty instances produced by the sampler along the Markov chains. An alternative to these approaches which allows for dimension-independent convergence rate $s = 1/p$ in terms of the number of samples *and* which allows simultaneous, parallel access to the forward model in all instances of the uncertainty is the recently developed, higher-order quasi-Monte Carlo integration. It allows fully parallel evaluation of the integrals Z' and Z in the Bayesian estimate (27.146). The reader is referred to [38] for a general survey and numerous references. It has recently been shown that (\mathbf{b}, p) sparsity implies, indeed, the dimension-independent convergence rate $s = 1/p$ for certain types of higher order QMC integration; see [36] for the theory for linear, affine parametric operator equations $q \mapsto A(\mathbf{y})q$, [37] for multilevel extensions, and [39] for the verification of the convergence conditions in [36] implied by (\mathbf{b}, p) holomorphy. Computational construction of higher-order QMC integration rules on the bounded parameter domain U is described in [41]. There exist also QMC integration methods for unbounded parameter regions. Such arise typically for *Gaussian random field (GRF for short) inputs* u taking values in X . Upon uncertainty parametrization with, for example, a Karhunen–Loève expansion into eigenfunctions of the covariance operator of the GRF (27.17), there result parametric deterministic problems with unbounded parameter ranges (consisting, for GRF's, of countable cartesian products of real lines, i.e., $U = \mathbb{R}^{\mathbb{N}}$). In this case, the present theory still is applicable; however, *all stability and equivalence constants will depend, generally, on the parameter \mathbf{y} with the parametric dependence degenerating for “extremal events,” i.e., realizations of u whose parameters in the tail of the prior π_0 . This is particularly relevant for uncertain input data which involve a gaussian random field (27.17) in some form.*

5 Software

- **rbMIT:** The general algorithms of RB based on Finite Element are implemented in the software package rbMIT ©MIT in MATLAB. It is implemented mainly for demonstration and education. However, it is also friendly to use for development

and test of new algorithms. The code and an accompanying textbook [59] are available through the link:

http://augustine.mit.edu/methodology/methodology_rbMIT_System.htm

- **RBnICS:** An RB extension of the Finite Element software package FEniCS [53] is under development and public domain through the link:<http://mathlab.sissa.it/rbnics>. Implementation includes POD and greedy algorithm for coercive problems, which is suited for an introductory course of RB together with the book [44].
- **Dune-RB:** It is a module for the Dune (Distributed and Unified Numerics Environment) library in C++. Template classes are available for RB construction based on several HiFi discretizations, including Finite Element and Finite Volume. Parallelization is available for RB construction too. Tutorials and code are available at <http://www.dune-project.org/>.
- **pyMOR:** pyMOR is implemented in Python for MOR for parameterized PDE. It has friendly interfaces and proper integration with external high-dimensional PDE solvers. Finite element and finite volume discretizations implemented based on the library of NumPy/SciPy are available. For more information, see <http://pymor.org>.
- **AKSELLOS:** MOR remains the core technology for the startup company AKSELLOS in several engineering fields, such as port infrastructure and industrial machinery. Different components are included such as FEA and CAD. The HiFi solution in AKSELLOS is implemented with a HPC and cloud-based simulation platform and is available for commercial use. For more information see <http://www.akselos.com>.
- For further libraries/software packages, we refer to <http://www.ians.uni-stuttgart.de/MoRePaS/software/>.

6 Conclusion

In this work, both mathematical and computational foundations of model order reduction techniques for UQ problems with distributed uncertainties are surveyed. Based on the recent development of sparse polynomial approximation in infinite dimensions, the convergence property of MOR constructed by greedy algorithm is established. In particular, under the sparsity of the uncertainties and the holomorphy of the forward solution maps w.r.t. the parameters, the dimension-independent convergence rate of the RB compression error can be achieved. Details of the construction and the compression of MOR are provided for both affine and nonaffine and linear and nonlinear problems modelled by parametric operator equations. Stability of the HiFi approximation and the RB compression is fulfilled by Petrov–Galerkin formulation with suitably constructed test spaces. Efficient MOR construction is realized by a greedy search algorithm with sparse sampling scheme, which further leads to a fast method for sparse grid construction. The MOR techniques are applied for both forward and inverse UQ problems, leading

to considerable computational reduction in the *many-query* context for evaluation of statistics, and in the *real-time* context for fast Bayesian inversion.

MOR has been demonstrated to be very effective in reducing the computational cost for solving large-scale “smooth” problems, namely, the solution map depends rather smoothly on the input parameters, as characterized by parametric holomorphy Definition 1 in high- or even infinite dimensions. However, this smoothness is not a necessary condition for the effectiveness of MOR. As long as the solution lives in an intrinsically low-dimensional manifold, MOR can reasonably be expected to accelerate numerical forward solves. See, for instance, [13] where the solution is discontinuous w.r.t. the input parameter. As for more general problems where the solution is not contained in a low-dimensional manifold, as observed in some hyperbolic problems, it is a remarkable challenge to apply MOR. Development of MOR to deal with problems of this kind is an emerging and active research field [1, 24, 31, 40, 65, 74]. Another development of MOR in computational UQ is parallel sampling and construction of the reduced order model in order to take advantage of parallel computing, such as using quasi Monte Carlo sampling [35] and parallel construction of the sparse grid based MOR via a priori information. In particular, for inverse UQ problems, sampling according to the posterior distribution or the related Hessian information [8, 67] would lead to potentially much faster construction and better efficiency of MOR.

7 Glossary

List of used abbreviations and definition of technical terms:

- UQ: uncertainty quantification
- MOR: model order reduction
- PDE: partial differential equations
- POD: proper orthogonal decomposition
- RB: reduced basis
- EI: empirical interpolation
- SG: sparse grid
- FEM: finite element method
- PG: Petrov–Galerkin
- QoI: quantity of interest
- Fidelity (of a mathematical model): notion of quality of responses of a computational **surrogate** model for a given mathematical model
- HiFi: high fidelity
- SCM: successive constraint method
- Surrogate model: numerical model obtained by various numerical approximation of a mathematical model

References

1. Abgrall, R., Amsallem, D.: Robust model reduction by l-norm minimization and approximation via dictionaries: application to linear and nonlinear hyperbolic problems. Technical report (2015)
2. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique, Analyse Numérique* **339**(9), 667–672 (2004)
3. Beirão da Veiga, L., Buffa, A., Sangalli, G., Vázquez, R.: Mathematical analysis of variational isogeometric methods. *Acta Numer.* **23**, 157–287 (2014)
4. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P.: Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.* **43**(3), 1457–1472 (2011)
5. Brezzi, F., Rappaz, J., Raviart, P.-A.: Finite-dimensional approximation of nonlinear problems. I. Branches of nonsingular solutions. *Numer. Math.* **36**(1), 1–25 (1980/1981)
6. Buffa, A., Maday, Y., Patera, A.T., Prudhomme, C., and Turinici, G.: A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Math. Modell. Numer. Anal.* **46**(03), 595–603 (2012)
7. Bui-Thanh, T., Damodaran, M., Willcox, K.E.: Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J.* **42**(8), 1505–1516 (2004)
8. Bui-Thanh, T., Ghattas, O., Martin, J., Stadler, G.: A computational framework for infinite-dimensional Bayesian inverse problems Part I: the linearized case, with application to global seismic inversion. *SIAM J. Sci. Comput.* **35**(6), A2494–A2523 (2013)
9. Bui-Thanh, T., Willcox, K., Ghattas, O.: Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J. Sci. Comput.* **30**(6), 3270–3288 (2008)
10. Carlberg, K., Bou-Mosleh, C., Farhat, C.: Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Int. J. Numer. Methods Eng.* **86**(2), 155–181 (2011)
11. Chatterjee, A.: An introduction to the proper orthogonal decomposition. *Curr. Sci.* **78**(7), 808–817 (2000)
12. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**(5), 2737–2764 (2010)
13. Chen, P., Quarteroni, A.: Accurate and efficient evaluation of failure probability for partial differential equations with random input data. *Comput. Methods Appl. Mech. Eng.* **267**(0), 233–260 (2013)
14. Chen, P., Quarteroni, A.: Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraints. *SIAM/ASA J. Uncertain. Quantif.* **2**(1), 364–396 (2014)
15. Chen, P., Quarteroni, A.: A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods. *J. Comput. Phys.* **298**, 176–193 (2015)
16. Chen, P., Quarteroni, A., Rozza, G.: A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **51**(6), 3163–3185 (2013)
17. Chen, P., Quarteroni, A., Rozza, G.: Comparison of reduced basis and stochastic collocation methods for elliptic problems. *J. Sci. Comput.* **59**, 187–216 (2014)
18. Chen, P., Quarteroni, A., Rozza, G.: A weighted empirical interpolation method: a priori convergence analysis and applications. *ESAIM: Math. Modell. Numer. Anal.* **48**, 943–953, 7 (2014)
19. Chen, P., Quarteroni, A., Rozza, G.: Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations. *Numerische Mathematik* **133**(1), 67–102 (2015)

20. Chen, P., Quarteroni, A., Rozza, G.: Reduced order methods for uncertainty quantification problems. Report 2015-03, Seminar for Applied Mathematics, ETH Zürich (2015, Submitted)
21. Chen, P., Schwab, Ch.: Sparse-grid, reduced-basis Bayesian inversion. *Comput. Methods Appl. Mech. Eng.* **297**, 84–115 (2015)
22. Chen, P., Schwab, Ch.: Adaptive sparse grid model order reduction for fast Bayesian estimation and inversion. In: Gärcke, J., Pflüger, D. (eds.) *Sparse Grids and Applications – Stuttgart 2014*, pp. 1–27. Springer, Cham (2016)
23. Chen, P., Schwab, Ch.: Sparse-grid, reduced-basis Bayesian inversion: nonaffine-parametric nonlinear equations. *J. Comput. Phys.* **316**, 470–503 (2016)
24. Cheng, M., Hou, T.Y., Zhang, Z.: A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations i: derivation and algorithms. *J. Comput. Phys.* **242**, 843–868 (2013)
25. Chkifa, A., Cohen, A., DeVore, R., Schwab, Ch.: Adaptive algorithms for sparse polynomial approximation of parametric and stochastic elliptic pdes. *M2AN Math. Mod. Num. Anal.* **47**(1), 253–280 (2013)
26. Chkifa, A., Cohen, A., Schwab, Ch.: High-dimensional adaptive sparse polynomial interpolation and applications to parametric pdes. *J. Found. Comput. Math.* **14**(4), 601–633 (2013)
27. Ciesielski, Z., Domsta, J.: Construction of an orthonormal basis in $C^m(I^d)$ and $W_p^m(I^d)$. *Studia Math.* **41**, 211–224 (1972)
28. Cohen, A., Chkifa, A., Schwab, Ch.: Breaking the curse of dimensionality in sparse polynomial approximation of parametric pdes. *J. Math. Pures et Appliquées* **103**(2), 400–428 (2015)
29. Cohen, A., DeVore, R.: Kolmogorov widths under holomorphic mappings. *IMA J. Numer. Anal.* (2015). doi:[doi:dr066v1-dru066](https://doi.org/10.1093/imanum/dru066)
30. Cui, T., Marzouk, Y.M., Willcox, K.E.: Data-driven model reduction for the Bayesian solution of inverse problems (2014). arXiv preprint [arXiv:1403.4290](https://arxiv.org/abs/1403.4290)
31. Dahmen, W., Plesken, C., Welper, G.: Double greedy algorithms: reduced basis methods for transport dominated problems. *ESAIM: Math. Modell. Numer. Anal.* **48**(03), 623–663 (2014)
32. Dashti, M., Stuart, A.M.: The Bayesian approach to inverse problems. In: Ghanem, R., et al. (eds.) *Handbook of UQ* (2016). <http://www.springer.com/us/book/9783319123844>
33. Deuflhard, P.: *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, vol. 35. Springer, Berlin/New York (2011)
34. DeVore, R., Petrova, G., Wojtaszczyk, P.: Greedy algorithms for reduced bases in banach spaces. *Constr. Approx.* **37**(3), 455–466 (2013)
35. Dick, J., Gantner, R., LeGia, Q.T., Schwab, Ch.: Higher order Quasi Monte Carlo integration for Bayesian inversion of holomorphic, parametric operator equations. Technical report, Seminar for Applied Mathematics, ETH Zürich (2015)
36. Dick, J., Kuo, F.Y., Le Gia, Q.T., Nuyens, D., Schwab, Ch.: Higher order QMC Petrov-Galerkin discretization for affine parametric operator equations with random field inputs. *SIAM J. Numer. Anal.* **52**(6), 2676–2702 (2014)
37. Dick, J., Kuo, F.Y., Le Gia, Q.T., Schwab, C.: Multi-level higher order QMC Galerkin discretization for affine parametric operator equations. *SIAM J. Numer. Anal.* (2016, to appear)
38. Dick, J., Kuo, F.Y., Sloan, I.H.: High-dimensional integration: the Quasi-Monte Carlo way. *Acta Numer.* **22**, 133–288 (2013)
39. Dick, J., LeGia, Q.T., Schwab, Ch.: Higher order Quasi Monte Carlo integration for holomorphic, parametric operator equations. *SIAM/ASA J. Uncertain. Quantif.* **4**(1), 48–79 (2016)
40. Drohmann, M., Haasdonk, B., Ohlberger, M.: Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.* **34**(2), A937–A969 (2012)
41. Gantner, R.N., Schwab, Ch.: Computational higher order quasi-monte carlo integration. Technical report 2014-25, Seminar for Applied Mathematics, ETH Zürich (2014)
42. Gerstner, T., Griebel, M.: Dimension-adaptive tensor-product quadrature. *Computing* **71**(1), 65–87 (2003)

43. Hesthaven, J., Stamm, B., Zhang, S.: Efficient Greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM: Math. Modell. Numer. Anal.* **48**(1), 259–283 (2011)
44. Hesthaven, J.S., Rozza, G., Stamm, B.: Certified Reduced Basis Methods for Parametrized Partial Differential Equations. Springer Briefs in Mathematics. Springer, Cham (2016)
45. Hesthaven, J.S., Stamm, B., Zhang, S.: Certified reduced basis method for the electric field integral equation. *SIAM J. Sci. Comput.* **34**(3), A1777–A1799 (2012)
46. Hoang, V.H., Schwab, Ch.: n -term Wiener chaos approximation rates for elliptic PDEs with lognormal Gaussian random inputs. *Math. Mod. Methods Appl. Sci.* **24**(4), 797–826 (2014)
47. Hoang, V.H., Schwab, Ch., Stuart, A.: Complexity analysis of accelerated MCMC methods for Bayesian inversion. *Inverse Probl.* **29**(8), 085010 (2013)
48. Huynh, D.B.P., Knezevic, D.J., Chen, Y., Hesthaven, J.S., Patera, A.T.: A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Methods Appl. Mech. Eng.* **199**(29), 1963–1975 (2010)
49. Huynh, D.B.P., Rozza, G., Sen, S., Patera, A.T.: A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathématique, Analyse Numérique* **345**(8), 473–478 (2007)
50. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: Generalized reduced basis methods and n-width estimates for the approximation of the solution manifold of parametric PDEs. In: Brezzi, F., Colli Franzone, P., Gianazza, U., Gilardi, G. (eds.) *Analysis and Numerics of Partial Differential Equations*. Springer INdAM Series, vol. 4, pp. 307–329. Springer, Milan (2013)
51. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: A reduced computational and geometrical framework for inverse problems in hemodynamics. *Int. J. Numer. Methods Biomed. Eng.* **29**(7), 741–776 (2013)
52. Lassila, T., Rozza, G.: Parametric free-form shape design with PDE models and reduced basis method. *Comput. Methods Appl. Mech. Eng.* **199**(23), 1583–1592 (2010)
53. Logg, A., Mardal, K.A., Wells, G.: *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, vol. 84. Springer, Berlin/New York (2012)
54. Ma, X., Zabaras, N.: An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* **228**(8), 3084–3113 (2009)
55. Maday, Y., Mula, O., Patera, A.T., Yano, M.: The generalized empirical interpolation method: stability theory on Hilbert spaces with an application to the Stokes equation. *Comput. Methods Appl. Mech. Eng.* **287**, 310–334 (2015)
56. Maday, Y., Mula, O., Turinici, G.: A priori convergence of the generalized empirical interpolation method. In: 10th International Conference on Sampling Theory and Applications (SampTA 2013), Bremen, pp. 168–171 (2013)
57. Maday, Y., Nguyen, N.C., Patera, A.T., Pau, G.S.H.: A general, multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.* **8**(1), 383–404 (2009)
58. Maday, Y., Patera, A.T., Turinici, G.: A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *J. Sci. Comput.* **17**(1), 437–446 (2002)
59. Patera, A.T., Rozza, G.: Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Copyright MIT, <http://augustine.mit.edu> (2007)
60. Pousin, J., Rappaz, J.: Consistency, stability, a priori and a posteriori errors for Petrov-Galerkin methods applied to nonlinear problems. *Numerische Mathematik* **69**(2), 213–231 (1994)
61. Prudhomme, C., Maday, Y., Patera, A.T., Turinici, G., Rovas, D.V., Veroy, K., Machiels, L.: Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods. *J. Fluids Eng.* **124**(1), 70–80 (2002)
62. Quarteroni, A.: Numerical Models for Differential Problems, 2nd edn. Springer, Milano (2013)
63. Rozza, G., Huynh, D.B.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **15**(3), 229–275 (2008)

-
- 64. Rozza, G., Veroy, K.: On the stability of the reduced basis method for stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Eng.* **196**(7), 1244–1260 (2007)
 - 65. Sapsis, T.P., Lermusiaux, P.F.J.: Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Phys. D: Nonlinear Phenom.* **238**(23), 2347–2360 (2009)
 - 66. Schillings, C., Schwab, Ch.: Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. *Inverse Probl.* **29**(6), 065011 (2013)
 - 67. Schillings, C., Schwab, Ch.: Scaling limits in computational Bayesian inversion. In: *ESAIM: M2AN* (2014, to appear). <http://dx.doi.org/10.1051/m2an/2016005>
 - 68. Schillings, C., Schwab, Ch.: Sparsity in Bayesian inversion of parametric operator equations. *Inverse Probl.* **30**(6), 065007, 30 (2014)
 - 69. Schwab, Ch., Gittelson, C.: Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. *Acta Numerica* **20**, 291–467 (2011)
 - 70. Schwab, Ch., Stevenson, R.: Space-time adaptive wavelet methods for parabolic evolution problems. *Math. Comput.* **78**(267), 1293–1318 (2009)
 - 71. Schwab, Ch., Stuart, A.M.: Sparse deterministic approximation of Bayesian inverse problems. *Inverse Probl.* **28**(4), 045003, 32 (2012)
 - 72. Schwab, Ch., Todor, R.A.: Karhunen–Loève approximation of random fields by generalized fast multipole methods. *J. Comput. Phys.* **217**(1), 100–122 (2006)
 - 73. Stuart, A.M.: Inverse problems: a Bayesian perspective. *Acta Numerica* **19**(1), 451–559 (2010)
 - 74. Taddei, T., Perotto, S., Quarteroni, A.: Reduced basis techniques for nonlinear conservation laws. *ESAIM: Math. Modell. Numer. Anal.* **49**(3), 787–814 (2015)
 - 75. Willcox, K., Peraire, J.: Balanced model reduction via the proper orthogonal decomposition. *AIAA J.* **40**(11), 2323–2330 (2002)

Multifidelity Uncertainty Quantification Using Spectral Stochastic Discrepancy Models

28

Michael S. Eldred, Leo W. T. Ng, Matthew F. Barone, and
Stefan P. Domino

Abstract

When faced with a restrictive evaluation budget that is typical of today's high-fidelity simulation models, the effective exploitation of lower-fidelity alternatives within the uncertainty quantification (UQ) process becomes critically important. Herein, we explore the use of multifidelity modeling within UQ, for which we rigorously combine information from multiple simulation-based models within a hierarchy of fidelity, in seeking accurate high-fidelity statistics at lower computational cost. Motivated by correction functions that enable the provable convergence of a multifidelity optimization approach to an optimal high-fidelity point solution, we extend these ideas to discrepancy modeling within a stochastic domain and seek convergence of a multifidelity uncertainty quantification process to globally integrated high-fidelity statistics. For constructing stochastic models of both the low-fidelity model and the model discrepancy, we employ stochastic expansion methods (non-intrusive polynomial chaos and stochastic collocation) computed by integration/interpolation on structured sparse grids or regularized regression on unstructured grids. We seek to employ a coarsely resolved grid for the discrepancy in combination with a more finely resolved

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

M.S. Eldred (✉)

Optimization and Uncertainty Quantification Department, Sandia National Laboratories,
Albuquerque, NM, USA
e-mail: mseldre@sandia.gov

L.W.T. Ng

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge,
MA, USA
e-mail: leo_ng@alum.mit.edu

M.F. Barone • S.P. Domino

Sandia National Laboratories, Albuquerque, NM, USA
e-mail: mbarone@sandia.gov; spdomin@sandia.gov

grid for the low-fidelity model. The resolutions of these grids may be defined statically or determined through uniform and adaptive refinement processes. Adaptive refinement is particularly attractive, as it has the ability to preferentially target stochastic regions where the model discrepancy becomes more complex, i.e., where the predictive capabilities of the low-fidelity model start to break down and greater reliance on the high-fidelity model (via the discrepancy) is necessary. These adaptive refinement processes can either be performed separately for the different grids or within a coordinated multifidelity algorithm. In particular, we present an adaptive greedy multifidelity approach in which we extend the generalized sparse grid concept to consider candidate index set refinements drawn from multiple sparse grids, as governed by induced changes in the statistical quantities of interest and normalized by relative computational cost. Through a series of numerical experiments using statically defined sparse grids, adaptive multifidelity sparse grids, and multifidelity compressed sensing, we demonstrate that the multifidelity UQ process converges more rapidly than a single-fidelity UQ in cases where the variance of the discrepancy is reduced relative to the variance of the high-fidelity model (resulting in reductions in initial stochastic error), where the spectrum of the expansion coefficients of the model discrepancy decays more rapidly than that of the high-fidelity model (resulting in accelerated convergence rates), and/or where the discrepancy is more sparse than the high-fidelity model (requiring the recovery of fewer significant terms).

Keywords

Multifidelity • Uncertainty quantification • Discrepancy model • Polynomial chaos • Stochastic collocation • Sparse grid • Compressed sensing • Wind turbine

Contents

1	Introduction	993
2	Stochastic Expansions	996
2.1	Non-intrusive Polynomial Chaos	997
2.2	Stochastic Collocation	999
2.3	Sparse Grid Construction	1001
2.4	Compressed Sensing	1003
3	Multifidelity Extensions	1004
3.1	Corrected Low-Fidelity Model	1005
3.2	Sparse Grids with Predefined Offset	1006
3.3	Adaptive Sparse Grids	1006
3.4	Compressed Sensing	1009
3.5	Analytic Moments	1010
4	Computational Results	1012
4.1	Simple One-Dimensional Example	1012
4.2	Short Column Example	1013
4.3	Elliptic PDE Example	1017
4.4	Horn Acoustics Example	1022
4.5	Production Engineering Example: Vertical-Axis Wind Turbine	1025
5	Conclusions	1031
	References	1034

1 Introduction

The rapid advancement of both computer hardware and physics simulation capabilities has revolutionized science and engineering, placing computational simulation on an equal footing with theoretical analysis and physical experimentation. This rapidly increasing reliance on the predictive capabilities of computational models has created the need for rigorous quantification of the effect that all types of uncertainties have on these predictions, in order to inform decisions and design processes with critical information on quantified variability and risk. A variety of challenges arise when deploying uncertainty quantification (UQ) to ever more complex physical systems. Foremost among these is the intersection of UQ problem scale with simulation resource constraints. When pushing the boundaries of computational science using complex three-dimensional (multi-)physics simulation capabilities, a common result is a highest-fidelity simulation that can be performed only a very limited number of times, even when executing on leadership-class parallel computers. Combining this state with the significant number of uncertainty sources (often in the hundreds, thousands, or beyond) that exist in these complex applications can result in an untenable proposition for UQ or at least a state of significant imbalance in which resolution of spatiotemporal physics has been given much higher priority over resolution of quantities of interest (QoIs) throughout the stochastic domain. This primary challenge can be further compounded by a number of factors, including (*i*) the presence of a mixture of aleatory and epistemic uncertainties, (*ii*) the need to evaluate the probability of rare events, and/or (*iii*) the presence of nonsmoothness in the variation of the response QoIs over the range of the random parameters. In these challenging cases, our most sophisticated UQ algorithms may be inadequate on their own for affordably generating accurate high-fidelity statistics; we need to additionally exploit opportunities that exist within a hierarchy of model fidelity.

The computational simulation of a particular physical phenomenon often has multiple discrete model selection possibilities. Here, we will broadly characterize this situation into two classes: a hierarchy of model fidelity and an ensemble of model peers. In the former case of a hierarchy, a clear preference structure exists among the models such that “high-fidelity” and “low-fidelity” judgments are readily assigned. Here, the goal is to manage the trade-off between accuracy and expense among the different model fidelities in order to achieve high-quality statistical results at lower cost. In the latter case, a clear preference structure is lacking and there is additional uncertainty created by the lack of a best model. In this case, the goal becomes one of management and propagation of this model form uncertainty [15] or, in the presence of experimental data for performing inference, reducing the model form uncertainty through formal model selection processes [10, 37]. Taking computational fluid dynamics (CFD) applied to a canonical cavity flow as an example, Fig. 28.1 depicts a multifidelity hierarchy that may include inviscid and boundary-element-based simulations, Reynolds averaged Navier-Stokes (RANS), unsteady RANS, large-eddy simulation (LES), and direct numerical simulation (DNS). Peers at a given level (e.g., RANS and LES) involve a variety

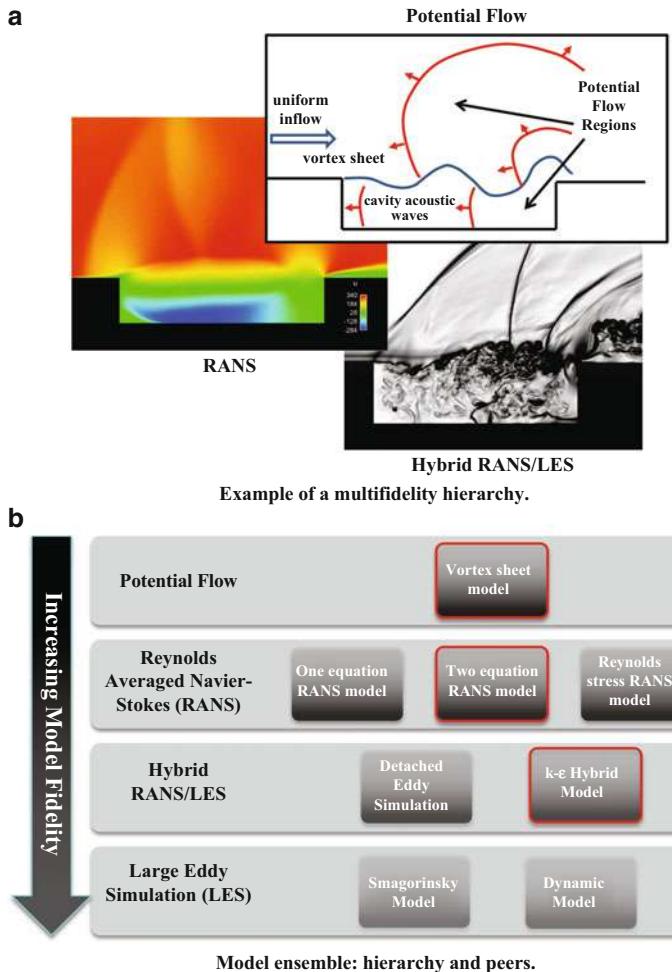


Fig. 28.1 Example of an ensemble of CFD models for a cavity flow problem (a) and hierarchy and ensemble of peer alternatives (b). Observational data can be used to inform model selection among peers (red boxes)

of turbulence (Spalart-Allmaras, $k-\epsilon$, $k-\omega$, etc.) and sub-grid scale (Smagorinsky, dynamic Smagorinsky, etc.) model formulations, as depicted in Fig. 28.1b.

In this work (an extension of [36]) we address the model hierarchy case where the system responses can be obtained accurately by evaluating an expensive high-fidelity model or less accurately by evaluating an inexpensive low-fidelity model. Extension to a deeper hierarchy that includes mid-fidelity options is straightforward (each additional mid-fidelity model introduces an additional level of model discrepancy), but here we focus on two fidelities for simplicity of exposition. The low-fidelity model may be based on simplified physics, coarser discretization of the

high-fidelity model, projection-based reduced-order models (e.g., proper orthogonal decomposition), or other techniques that exchange accuracy for reduced cost. We investigate a multifidelity approach to compute high-fidelity statistics without the expense of relying exclusively on high-fidelity model evaluations. Such multifidelity approaches have previously been developed for the *optimization* of expensive high-fidelity models. In the multifidelity trust-region model-management approach [3], the optimization is performed on a corrected low-fidelity model. The correction function can be additive, multiplicative, or a combination of the two [16] and is updated periodically using high-fidelity model evaluations. First- or second-order polynomials are used to enforce local first- or second-order consistency, respectively, at high-fidelity model evaluation points [3, 16]. Other variations have employed global correction functions (typically enforcing zeroth-order consistency) based on the interpolation of the discrepancy between the high-fidelity model evaluations and the low-fidelity model evaluations [18, 26, 33, 38]. The central concept is that a surrogate based on a physics-based low-fidelity model and a model of the discrepancy may provide a more cost-effective approximation of the high-fidelity model than a surrogate based only on fitting limited sets of high-fidelity data. We carry this idea over to uncertainty propagation and form a stochastic approximation for the discrepancy model as well as a separate stochastic approximation for the low-fidelity model. Both approximations employ expansions of global polynomials defined in terms of the stochastic parameters. After forming the low-fidelity and discrepancy expansions, we combine their polynomial terms to create a multifidelity stochastic expansion that approximates the high-fidelity model, and we use this expansion to generate the desired high-fidelity statistics. Compared to a single-fidelity expansion formed exclusively from high-fidelity evaluations, the multifidelity expansion will typically carry more terms after combination due to the greater resolution used in creating the low-fidelity expansion. If the low-fidelity model is sufficiently predictive, then less computational effort is required to resolve the discrepancy, reducing the number of high-fidelity model evaluations necessary to obtain the high-fidelity response statistics to a desired accuracy. Depending on the specific formulation, we may choose to strictly enforce zeroth- and first-order consistency (values and first derivatives) of our combined approximation with the high-fidelity results at each of the high-fidelity collocation points, mirroring the convergence theory requirements for surrogate-based optimization methods.

Our foundation for multifidelity UQ is provided by stochastic expansion methods, using either multivariate orthogonal polynomials in the case of non-intrusive polynomial chaos or multivariate interpolation polynomials in the case of stochastic collocation. In the former case, polynomial chaos expands the system response as a truncated series of polynomials that are orthogonal with respect to the probability density functions of the stochastic parameters [22, 43]. Exponential convergence in integrated statistical quantities (e.g., mean, variance) can be achieved for smooth functions with finite variance. The chaos coefficients can be obtained by either projecting the system response onto each basis and then computing the coefficients by multidimensional numerical integration or by solving for the coefficients using regression approaches with either least squares for overdetermined or ℓ_1 -regularized

regression [27] for under-determined systems. Stochastic collocation is a related stochastic expansion method which constructs multidimensional interpolation polynomials over the system responses evaluated at a structured set of collocation points [5, 42]. If the collocation points are selected appropriately, then similar performance can be achieved, and in the limiting case of tensor-product Gaussian quadrature rules corresponding to the density functions of the random variables, the two approaches generate identical polynomial forms.

Many other choices for stochastic approximation are possible, and Gaussian process (GP) models have been used extensively for discrepancy modeling in the context of Bayesian inference [30], including multifidelity Bayesian inference approaches [24, 29]. GPs would be a particularly attractive choice in the case of resolution of rare events, as adaptive refinement of GPs to resolve failure domains [8] would provide an effective single-fidelity framework around which to tailor a multifidelity strategy for adaptively refining GP surrogates for the low-fidelity and discrepancy models.

Another related multifidelity capability is multilevel Monte Carlo (MLMC [6, 23]), which relies on correlation among the QoI predictions that are produced for different discretization levels within a model hierarchy. This correlation manifests as a reduction in variance for statistical estimators applied to model discrepancies, which in turn leads to a reduction in the leading constant for the $\frac{1}{\sqrt{N}}$ convergence rates of Monte Carlo methods. Finally, recent work [35, 44] has explored an approach in which lower-fidelity models are used to inform how to most effectively sample the high-fidelity model as well as for reconstructing approximate high-fidelity evaluations. These different approaches place differing requirements on the predictive capability of lower-fidelity models, which have important ramifications on the efficacy of the multifidelity approaches. For example, the requirement of correlation between QoI prediction levels in MLMC and control variate approaches is expected to relax smoothness requirements that are present when explicitly resolving model discrepancy based on polynomial expansion. Conversely, when smoothness is present, its effective exploitation should lead to superior multifidelity convergence rates. Herein, we focus on the spectral stochastic discrepancy modeling approach, with relative comparisons reserved as an important subject for future study.

In the following, we first review our foundation of stochastic expansion methods and their multidimensional construction via sparse grids or compressed sensing. Next, we present the extension to the multifidelity case and provide an algorithmic framework. Finally, we demonstrate our approach with a variety of computational experiments and provide concluding remarks.

2 Stochastic Expansions

In this section, we briefly review the non-intrusive polynomial chaos expansion (PCE) and stochastic collocation (SC) methods. Both methods construct a global polynomial approximation to the system response and have been shown to be

pointwise equivalent if non-nested Gaussian quadrature nodes are used [11]. However, one form of the polynomial expansion may be preferred over the other, depending on the needs of the application (e.g., support for unstructured grids, fault tolerance, and/or local error estimation).

2.1 Non-intrusive Polynomial Chaos

In polynomial chaos, one must estimate the chaos coefficients for a set of basis functions. To reduce the nonlinearity of the expansion and improve convergence, the polynomial bases are chosen such that their orthogonality weighting functions match the probability density functions of the stochastic parameters up to a constant factor [43]. The basis functions are typically obtained from the Askey family of hypergeometric orthogonal polynomials [4], and Table 28.1 lists the appropriate polynomial bases for some commonly used continuous probability distributions. If the stochastic parameters do not follow these standard probability distributions, then the polynomial bases may be generated numerically [19, 25, 41]. Alternatively, or if correlations are present, variable transformations [1, 12, 39] may be used. These transformations provide support for statistical independence (e.g., decorrelation of standard normal distributions as in [12]) and can also enable the use of nested quadrature rules such as Gauss-Patterson and Genz-Keister for uniform and normal distributions, respectively, within the transformed space.

Let $R(\xi)$ be a “black box” that takes d stochastic parameters $\xi = (\xi_1, \dots, \xi_d)$ as inputs and return the system response R as the output.

The system response is approximated by the expansion

$$R(\xi) \approx \sum_{\mathbf{i} \in \mathcal{I}_p} \alpha_{\mathbf{i}} \Psi_{\mathbf{i}}(\xi), \quad (28.1)$$

where the basis functions $\Psi_{\mathbf{i}}(\xi)$ with multi-index $\mathbf{i} = (i_1, \dots, i_d)$, $i_k = 0, 1, 2, \dots$ are the product of the appropriate one-dimensional orthogonal polynomial basis of

Table 28.1 Some standard continuous probability distributions and their corresponding Askey polynomial bases. $B(\alpha, \beta)$ is the Beta function and $\Gamma(\alpha)$ is the Gamma function

Distribution	Density function	Polynomial basis	Orthogonality weight	Support
Normal	$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	Hermite $He_n(x)$	$e^{-\frac{x^2}{2}}$	$[-\infty, \infty]$
Uniform	$\frac{1}{2}$	Legendre $P_n(x)$	1	$[-1, 1]$
Beta	$\frac{(1-x)^\alpha(1+x)^\beta}{2^{\alpha+\beta+1}B(\alpha+1, \beta+1)}$	Jacobi $P_n^{(\alpha, \beta)}(x)$	$(1-x)^\alpha(1+x)^\beta$	$[-1, 1]$
Exponential	e^{-x}	Laguerre $L_n(x)$	e^{-x}	$[0, \infty]$
Gamma	$\frac{x^\alpha e^{-x}}{\Gamma(\alpha+1)}$	Gen. Laguerre $L_n^{(\alpha)}(x)$	$x^\alpha e^{-x}$	$[0, \infty]$

From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.

order i_k in each dimension $k = 1, \dots, d$. The series is typically truncated in one of two ways. For the total-order expansion of order p , the index set is defined as

$$\mathcal{I}_p = \{\mathbf{i} : |\mathbf{i}| \leq p\}, \quad (28.2)$$

where $|\mathbf{i}| = i_1 + \dots + i_d$, while for the tensor-product expansion of order $\mathbf{p} = (p_1, \dots, p_d)$, the index set is defined as

$$\mathcal{I}_{\mathbf{p}} = \{\mathbf{i} : i_k \leq p_k, k = 1, \dots, d\}. \quad (28.3)$$

The number of terms required in each case is

$$M_{TO} = \frac{(d+p)!}{d! p!} = \binom{d+p}{d}, \quad (28.4)$$

and

$$M_{TP} = \prod_{k=1}^d (p_k + 1), \quad (28.5)$$

respectively.

One approach to calculate the chaos coefficients $\alpha_{\mathbf{i}}$ is the spectral projection method that takes advantage of the orthogonality of the bases. This results in

$$\alpha_{\mathbf{i}} = \frac{\langle R(\xi), \Psi_{\mathbf{i}}(\xi) \rangle}{\langle \Psi_{\mathbf{i}}^2(\xi) \rangle} = \frac{1}{\langle \Psi_{\mathbf{i}}^2(\xi) \rangle} \int_{\Omega} R(\xi) \Psi_{\mathbf{i}}(\xi) \rho(\xi) d\xi, \quad (28.6)$$

where $\rho(\xi) = \prod_{k=1}^d \rho_k(\xi_k)$ is the joint probability density of the stochastic parameters over the support $\Omega = \Omega_1 \times \dots \times \Omega_d$ and $\langle \Psi_{\mathbf{i}}^2(\xi) \rangle$ is available in closed form. Thus, the bulk of the work is in evaluating the multidimensional integral in the numerator. Tensor-product quadrature may be employed to evaluate these integrals or, if d is more than a few parameters, sparse grid quadrature may be employed as described in Eqs. 28.19–28.21 to follow.

Once the chaos coefficients are known, the statistics of the system response can be estimated directly and inexpensively from the expansion. For example, the mean and the variance can be obtained analytically as

$$\mu_R = \langle R(\xi) \rangle \approx \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{p}}} \alpha_{\mathbf{i}} \langle \Psi_{\mathbf{i}}(\xi) \rangle = \alpha_0 \quad (28.7)$$

and

$$\sigma_R^2 = \langle R(\xi)^2 \rangle - \mu_R^2 \approx \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{p}}} \sum_{\mathbf{j} \in \mathcal{I}_{\mathbf{p}}} \alpha_{\mathbf{i}} \alpha_{\mathbf{j}} \langle \Psi_{\mathbf{i}}(\xi) \Psi_{\mathbf{j}}(\xi) \rangle - \alpha_0^2 = \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{p}} \setminus \mathbf{0}} \alpha_{\mathbf{i}}^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle \quad (28.8)$$

where $\mathbf{0}$ is a vector of zeros (i.e., the first multi-index). Other statistics such as probabilities and quantiles can be estimated by sampling the polynomial expansion.

2.2 Stochastic Collocation

In stochastic collocation, a multivariate polynomial interpolant is formed over the system responses evaluated at a set of collocation points. In one-dimension, a degree $n - 1$ expansion (identified by the index i) has the form

$$R(\xi) \approx \mathcal{U}_i(R) \stackrel{\text{def}}{=} \sum_{j=1}^n R(\xi^{(j)}) \ell^{(j)}(\xi) \quad (28.9)$$

where $R(\xi^{(j)})$ is the system response evaluated at collocation points $\xi^{(j)}$, $j = 1, \dots, n$. The basis functions $\ell^{(j)}(\xi)$ can be either local or global and either value based or gradient enhanced [1], but the most common option is the global Lagrange polynomial based on interpolation of values:

$$\ell^{(j)}(\xi) = \prod_{k=1, k \neq j}^n \frac{\xi - \xi^{(k)}}{\xi^{(j)} - \xi^{(k)}}. \quad (28.10)$$

The collocation points can be chosen for accuracy in interpolation (as indicated by the Lebesgue measure) or integration (as indicated by polynomial exactness). Here, we emphasize the latter and choose Gaussian quadrature nodes that correspond to the same orthogonal polynomial selections used for the polynomial chaos expansion. In the multivariate case, we start from a tensor-product formulation with the multi-index $\mathbf{i} = (i_1, \dots, i_d)$, $i_k = 0, 1, 2, \dots$:

$$\begin{aligned} R(\xi) &\approx \mathcal{U}_{\mathbf{i}}(R) \stackrel{\text{def}}{=} (\mathcal{U}_{i_1} \otimes \dots \otimes \mathcal{U}_{i_d})(R) \\ &= \sum_{j_1=1}^{n_1} \dots \sum_{j_d=1}^{n_d} R(\xi_1^{(j_1)}, \dots, \xi_d^{(j_d)}) \ell_1^{(j_1)}(\xi_1) \dots \ell_d^{(j_d)}(\xi_d). \end{aligned} \quad (28.11)$$

Statistics of the system response such as the mean and the variance of a tensor-product expansion can be obtained analytically as

$$\begin{aligned} \mu_R &= \langle R(\xi) \rangle \approx \sum_{j_1=1}^{n_1} \dots \sum_{j_d=1}^{n_d} R(\xi_1^{(j_1)}, \dots, \xi_d^{(j_d)}) \langle \ell_1^{(j_1)}(\xi_1) \dots \ell_d^{(j_d)}(\xi_d) \rangle \\ &= \sum_{j_1=1}^{n_1} \dots \sum_{j_d=1}^{n_d} R(\xi_1^{(j_1)}, \dots, \xi_d^{(j_d)}) w_1^{(j_1)} \dots w_d^{(j_d)} \\ &\stackrel{\text{def}}{=} \mathcal{Q}_{\mathbf{i}}(R) \end{aligned} \quad (28.12)$$

and

$$\begin{aligned}
\sigma_R^2 &= \left\langle R(\xi)^2 \right\rangle - \mu_R^2 \\
&\approx \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} \sum_{k_1=1}^{n_1} \cdots \sum_{k_d=1}^{n_d} R\left(\xi_1^{(j_1)}, \dots, \xi_d^{(j_d)}\right) R\left(\xi_1^{(k_1)}, \dots, \xi_d^{(k_d)}\right) \\
&\quad \left\langle \ell_1^{(j_1)}(\xi_1) \dots \ell_d^{(j_d)}(\xi_d) \ell_1^{(k_1)}(\xi_1) \dots \ell_d^{(k_d)}(\xi_d) \right\rangle - \mu_R^2 \\
&= \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} R^2\left(\xi_1^{(j_1)}, \dots, \xi_d^{(j_d)}\right) w_1^{(j_1)} \dots w_d^{(j_d)} - \mu_R^2 \\
&\stackrel{\text{def}}{=} \mathbf{Q}_i(R^2) - \mu_R^2,
\end{aligned} \tag{28.13}$$

where the expectation integrals of the basis polynomials use the property that $\ell^{(s)}(\xi^{(t)}) = \delta_{s,t}$, such that numerical quadrature of these integrals leaves only the quadrature weights w . Higher moments can be obtained analytically in a similar manner, and other statistics such as probabilities and quantiles can be estimated by sampling the expansion.

We can collapse these tensor-product sums by moving to a multivariate basis representation

$$R(\xi) \approx \sum_{j=1}^{N_{TP}} R(\xi^{(j)}) \mathbf{L}^{(j)}(\xi) \tag{28.14}$$

where N_{TP} is the number of tensor-product collocation points in the multidimensional grid and the multivariate interpolation polynomials are defined as

$$\mathbf{L}^{(j)}(\xi) = \prod_{k=1}^n \ell^{(c_k^j)}(\xi_k) \tag{28.15}$$

where c_k^j is a collocation multi-index (similar to the polynomial chaos multi-index in Eqs. 28.1–28.3). The corresponding moment expressions are then

$$\mu_R = \sum_{j=1}^{N_{TP}} R(\xi^{(j)}) \mathbf{w}^{(j)} \tag{28.16}$$

$$\sigma_R^2 = \sum_{j=1}^{N_{TP}} R^2(\xi^{(j)}) \mathbf{w}^{(j)} - \mu_R^2 \tag{28.17}$$

where the multivariate weight is

$$\mathbf{w}^{(j)} = \prod_{k=1}^n w^{(c_k^j)}. \tag{28.18}$$

A sparse interpolant and its moments are then formed from a linear combination of these tensor products, as described in the following section.

2.3 Sparse Grid Construction

If d is moderately large, then a sparse grid construction may be used to alleviate the exponential increase in the number of collocation points with respect to d . As employed in numerical integration [20, 21] and interpolation [7, 9], sparse grids are constructed from a linear combination of tensor-product grids with relatively small numbers of grid points in such a way that preserves a high level of accuracy.

2.3.1 Isotropic Sparse Grids

The isotropic sparse grid at level q where $q = 0, 1, 2, \dots$ is defined as

$$\mathcal{A}_{q,d}(R) = \sum_{q-d+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \mathcal{U}_{\mathbf{i}}(R), \quad (28.19)$$

where the tensor-product interpolation formulas $\mathcal{U}_{\mathbf{i}}(R)$, $\mathbf{i} = (i_1, \dots, i_d)$, $i_k = 0, 1, 2, \dots$ can be replaced by tensor-product quadrature formulas $\mathcal{Q}_{\mathbf{i}}(R)$ for the case of sparse grid integration. Alternatively, Eq. 28.19 may be expressed in terms of the difference formulas:

$$\mathcal{A}_{q,d}(R) = \sum_{|\mathbf{i}| \leq q} \Delta_{\mathbf{i}}(R), \quad (28.20)$$

where $\Delta_{\mathbf{i}}(R) = (\Delta_{i_1} \otimes \dots \otimes \Delta_{i_d})(R)$ and $\Delta_{i_k} = \mathcal{U}_{i_k} - \mathcal{U}_{i_k-1}$ with $\mathcal{U}_{-1} = 0$. Thus, the tensor-product grids used in the isotropic sparse grid construction are those whose multi-indices lie within a simplex defined by the sparse grid level q .

The relationship between the index i_k and the number of collocation points n_k in each dimension $k = 1, \dots, d$ is called the growth rule and is an important detail of the sparse grid construction. If the collocation points are chosen based on a fully nested quadrature rule, then a nonlinear growth rule that approximately doubles n_k with every increment in i_k (e.g., $n_k = 2^{i_k+1} - 1$ for open nested rules such as Gauss-Patterson) is used to augment existing model evaluations. If the collocation points are based on a weakly nested or non-nested quadrature rule, then a linear growth rule (e.g., $n_k = 2i_k + 1$) may alternatively be used to provide finer granularity in the order of the rule and associated degree of the polynomial basis.

2.3.2 Generalized Sparse Grids

The generalized sparse grid construction relaxes the simplex constraint on the multi-indices in Eq. 28.20 to provide flexibility for adaptive refinement. In the relaxed constraint, the set of multi-indices \mathcal{J} is admissible if $\mathbf{i} - \mathbf{e}_k \in \mathcal{J}$ for all $\mathbf{i} \in \mathcal{J}$, $i_k \geq 1$, $k = 1, \dots, d$, where \mathbf{e}_k is the k^{th} unit vector [21]. This admissibility criterion is depicted graphically in Fig. 28.2. In the left graphic, both children of the current

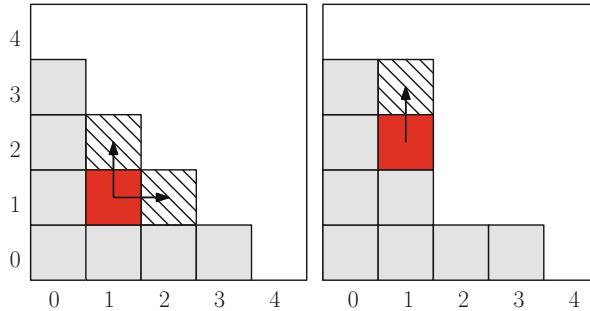


Fig. 28.2 Identification of the admissible forward neighbors for an index set (red). The indices of the reference basis are gray and admissible forward indices are hashed. An index set is admissible only if its backward neighbors exist in every dimension

index are admissible, because their backward neighbors exist in every dimension. In the right graphic only the child in the vertical dimension is admissible, as not all parents of the horizontal child exist.

Thus, admissible multi-indices can be added one by one starting from an initial reference grid, often the level 0 grid ($\mathbf{i} = \mathbf{0}$) corresponding to a single point in parameter space. The refinement process is governed by a greedy algorithm, where each admissible index set is evaluated for promotion into the reference grid based on a refinement metric. The best index set is selected for promotion and used to generate additional admissible index sets, and the process is continued until convergence criteria are satisfied. The resulting generalized sparse grid is then defined from

$$\mathcal{A}_{\mathcal{J},d}(R) = \sum_{\mathbf{i} \in \mathcal{J}} \Delta_{\mathbf{i}}(R). \quad (28.21)$$

2.3.3 Sparse Projection

When forming a polynomial chaos expansion from isotropic, anisotropic, or generalized sparse grids, the expansion terms should be composed from a sum of tensor expansions, where the coefficients for each tensor expansion should be computed from the corresponding tensor-product quadrature rule that is defined from each term in the sparse grid summation (e.g., Eq. 28.19). This integrate-then-combine approach avoids numerical noise in the coefficients of higher-order terms [11].

2.3.4 Sparse Interpolation

For computing moments, the tensor-product mean and variance definitions in Eqs. 28.12–28.13 can be extended to estimate the mean and variance of the sparse grid interpolant using a linear combination of $\mathcal{Q}_{\mathbf{i}}(R)$ or $\mathcal{Q}_{\mathbf{i}}(R^2)$, respectively, at multi-indices \mathbf{i} corresponding to the sparse grid construction. The result can be collapsed to the form of Eqs. 28.16–28.17 (with N_{TP} replaced with the number

of unique points in the sparse grid, N_{SG}), where the weights $w^{(j)}$ are now the sparse weights defined by the linear combination of all tensor-product weights applied to the same collocation point $\xi^{(j)}$. Note that while each tensor-product expansion interpolates the system responses at the collocation points, the sparse grid expansion will not interpolate in general unless a nested set of collocation points is used [7]. Another important detail in the non-nested case is that the summations in Eqs. 28.16–28.17 correspond to sparse numerical integration of R and R^2 , differing slightly from expectations of the sparse interpolant and its square.

Hierarchical interpolation on nested grids [2, 31] is particularly useful in an adaptive refinement setting, where we are interested in measuring small increments in statistical QoI due to small increments (e.g., candidate index sets in generalized sparse grids) in the sparse grid definition. By reformulating using hierarchical interpolants, we mitigate the loss of precision due to subtractive cancelation. In hierarchical sparse grids, the Δ_i in Eqs. 28.20 and 28.21 are tensor products of the one-dimensional difference interpolants defined over the increment in collocation points:

$$\Delta_i(R) = \sum_{j_1=1}^{n_{\Delta_1}} \cdots \sum_{j_d=1}^{n_{\Delta_d}} s \left(\xi_{j_1}^{\Delta_1}, \dots, \xi_{j_d}^{\Delta_d} \right) \left(L_{j_1}^{\Delta_1} \otimes \cdots \otimes L_{j_d}^{\Delta_d} \right) \quad (28.22)$$

where s are the hierarchical surpluses computed from the difference between the newly added response value and the value predicted by the previous interpolant level. Hierarchical increments in expected value are formed simply from the expected value of the new Δ_i contributions induced by a refinement increment (a single Δ_i for each candidate index set in the case of generalized sparse grids). Hierarchical increments in a variety of other statistical QoI may also be derived in a manner that preserves precision [1], starting from increments in response mean μ and covariance Σ :

$$\Delta\mu_i = \Delta\mathbb{E}[R_i] \quad (28.23)$$

$$\Delta\Sigma_{ij} = \Delta\mathbb{E}[R_i R_j] - \mu_i \Delta\mathbb{E}[R_j] - \mu_j \Delta\mathbb{E}[R_i] - \Delta\mathbb{E}[R_i] \Delta\mathbb{E}[R_j] \quad (28.24)$$

where $\Delta\mathbb{E}[\cdot]$ denotes a hierarchical increment in expected value, involving summation of hierarchical surpluses combined with the quadrature weights across the collocation point increments.

2.4 Compressed Sensing

PCE may also be constructed through regression by solving the following $N \times M$ linear system of equations over an *unstructured* set of ξ_j :

$$\begin{bmatrix} \psi_1(\xi_1) & \psi_2(\xi_1) & \dots & \psi_M(\xi_1) \\ \psi_1(\xi_2) & \psi_2(\xi_2) & \dots & \psi_M(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\xi_N) & \psi_2(\xi_N) & \dots & \psi_M(\xi_N) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_M \end{bmatrix} = \Psi\alpha = \mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix}. \quad (28.25)$$

This system may be over-, under-, or uniquely determined, based on the number of candidate basis terms M and the number of response QoI observations N . We are particularly interested in the under-determined case $N \ll M$, for which the problem is ill-posed and a regularization is required to enforce solution uniqueness. One approach is to apply the pseudo-inverse of Ψ , which returns the solution with minimum 2-norm $\|\alpha\|_{\ell_2}$. However, if the solution is sparse (many terms are zero) or compressible (many terms are small due to rapid solution decay), then an effective alternative is to seek the solution with the minimum number of terms. This sparse solution is the minimum zero-norm solution

$$\alpha = \arg \min \|\alpha\|_{\ell_0} \quad \text{such that} \quad \|\Psi\alpha - \mathbf{R}\|_{\ell_2} \leq \varepsilon, \quad (28.26)$$

which can be solved using the orthogonal matching pursuit (OMP), a greedy heuristic algorithm that iteratively constructs the solution vector term by term. Other approaches such as basis pursuit denoising (BPDN) compute a sparse solution using the minimum one norm

$$\alpha = \arg \min \|\alpha\|_{\ell_1} \quad \text{such that} \quad \|\Psi\alpha - \mathbf{R}\|_{\ell_2} \leq \varepsilon, \quad (28.27)$$

which allows for computational efficiency within optimization-based approaches by replacing expensive combinatorial optimization with continuous optimization. In both cases, the noise parameter ε can be tuned through cross-validation to avoid overfitting the observations \mathbf{R} .

3 Multifidelity Extensions

Let $R_{\text{high}}(\xi)$ be the system response obtained by evaluating the expensive high-fidelity model and $R_{\text{low}}(\xi)$ be the system response obtained by evaluating the inexpensive low-fidelity model. In a multifidelity stochastic expansion, the low-fidelity model values are corrected to match the high-fidelity model values (and potentially their derivatives) at the high-fidelity collocation points.

3.1 Corrected Low-Fidelity Model

We investigate additive correction, multiplicative correction, and combined additive and multiplicative correction for the low-fidelity model. Defining the additive correction function and multiplicative correction function as

$$\delta_A(\xi) = R_{\text{high}}(\xi) - R_{\text{low}}(\xi) \quad (28.28)$$

and

$$\delta_M(\xi) = \frac{R_{\text{high}}(\xi)}{R_{\text{low}}(\xi)}, \quad (28.29)$$

respectively, then

$$R_{\text{high}}(\xi) = R_{\text{low}}(\xi) + \delta_A(\xi) \quad (28.30)$$

or

$$R_{\text{high}}(\xi) = R_{\text{low}}(\xi)\delta_M(\xi). \quad (28.31)$$

In the case of a combined correction

$$R_{\text{high}}(\xi) = \gamma(R_{\text{low}}(\xi) + \delta_A(\xi)) + (1 - \gamma)R_{\text{low}}(\xi)\delta_M(\xi), \quad (28.32)$$

the parameter $\gamma \in [0, 1]$ defines a convex combination that determines the proportion of additive correction or multiplicative correction employed in the combined correction. γ provides a tuning parameter that can be optimized to prevent overfitting. One approach is to employ cross-validation to select from among preselected γ values, as described previously for the compressed sensing noise tolerance ε in Eqs. 28.26–28.27. Another option is to compute γ based on a regularization of the combined correction function by minimizing the magnitude of the additive and multiplicative correction in the mean-square sense

$$\min_{\gamma \in [0,1]} \left\langle \gamma^2 \delta_A^2(\xi) + (1 - \gamma)^2 \delta_M^2(\xi) \right\rangle. \quad (28.33)$$

This gives the solution

$$\gamma = \frac{\langle \delta_M^2(\xi) \rangle}{\langle \delta_A^2(\xi) \rangle + \langle \delta_M^2(\xi) \rangle}, \quad (28.34)$$

where the second moments of the $\delta_A(\xi)$ and $\delta_M(\xi)$ can be estimated analytically from their stochastic expansions as described in Eqs. 28.7–28.8 and Eqs. 28.16–

28.17. This choice of γ balances the additive correction and the multiplicative correction such that neither becomes too “large.”

3.2 Sparse Grids with Predefined Offset

Let $S_{q,d}[R]$ be the stochastic expansion (non-intrusive polynomial chaos or stochastic collocation) of $R(\xi)$ at sparse grid level q with dimension d . We add the superscript “pc” or “sc” when we refer specifically to non-intrusive polynomial chaos or stochastic collocation, respectively. Also, let $N_{q,d}$ be the number of model evaluations required to construct $S_{q,d}[R]$. Thus, $R_{\text{high}}(\xi) \approx S_{q,d}[R_{\text{high}}](\xi)$. We further approximate the stochastic expansion of the high-fidelity model with $S_{q,d}[R_{\text{high}}](\xi) \approx \tilde{R}_{\text{high}}(\xi)$, where $\tilde{R}_{\text{high}}(\xi)$ is the multifidelity stochastic expansion based on the additive, multiplicative, or combined correction of the low-fidelity model

$$\tilde{R}_{\text{high}} = S_{q,d}[R_{\text{low}}] + S_{q-r,d}[\delta_A], \quad (28.35)$$

$$\tilde{R}_{\text{high}} = S_{q,d}[R_{\text{low}}]S_{q-r,d}[\delta_M], \quad (28.36)$$

$$\tilde{R}_{\text{high}} = \gamma(S_{q,d}[R_{\text{low}}] + S_{q-r,d}[\delta_A]) + (1 - \gamma)S_{q,d}[R_{\text{low}}]S_{q-r,d}[\delta_M], \quad (28.37)$$

respectively, where r is a sparse grid level offset between the stochastic expansion of the low-fidelity model and the stochastic expansion of the correction function and $r \leq q$. Thus, the multifidelity stochastic expansion at sparse grid level q can be constructed with $N_{q-r,d}$ instead of $N_{q,d}$ high-fidelity model evaluations plus $N_{q,d}$ low-fidelity model evaluations. If the low-fidelity model is significantly less expensive to evaluate than the high-fidelity model, then significant computational savings will be obtained. Furthermore, if the low-fidelity model is sufficiently predictive, then accuracy comparable to the single-fidelity expansion with $N_{q,d}$ high-fidelity model evaluations can be achieved. This notion of a predetermined sparse grid level offset enforces computational savings explicitly, with less regard to managing the accuracy in $\tilde{R}_{\text{high}}(\xi)$. In the case of adaptive refinement, to be discussed in the following section, we instead manage the accuracy explicitly by investing resources where they are most needed for resolving statistics of $\tilde{R}_{\text{high}}(\xi)$ and the computational savings that result are achieved indirectly.

3.3 Adaptive Sparse Grids

The goal of an adaptive refinement procedure applied to multifidelity modeling should be to preferentially refine where the model discrepancy has the greatest complexity. This corresponds to regions of the stochastic domain where the low-fidelity model becomes less predictive. It is often the case in real-world applications that

low-fidelity models may be predictive for significant portions of a parameter space, but, in other portions of the space, the simplifying assumptions break down and a higher-fidelity model must be relied upon. By selectively targeting these regions, we rely more on the low-fidelity model where it is effective and more faithfully resolve the discrepancy where it is not. Thus, adaptive refinement procedures can extend the utility of multifidelity uncertainty quantification approaches in cases where the predictive capability of low-fidelity models is strongly parameter dependent.

For adaptive refinement in a multifidelity context, we will employ a greedy adaptation based on the generalized sparse grid procedure described with Eq. 28.21. One option is to separately adapt the low-fidelity and discrepancy models for accuracy in their individual statistics; however, this performs the refinements in isolation from one another, which may result in a non-optimal allocation of resources. Rather, we prefer to assess the effects of the individual candidate refinements within the aggregated multifidelity context, i.e., their effect on the high-fidelity statistical QoIs such as variance or failure probability. To accomplish this, we present a further generalization to generalized sparse grids in which we consider candidate index sets from multiple sparse grids simultaneously and measure their effects within the aggregated context using appropriate cost normalization. The algorithmic steps can be summarized as:

1. *Initialization:* Starting from an initial reference sparse grid (e.g., level $q = 0$) for the lowest-fidelity model and each level of model discrepancy within a multifidelity hierarchy, aggregate active index sets from each grid using the admissible forward neighbors of all reference index sets.
2. *Trial set evaluation:* For each trial active index set, perform the tensor grid evaluations for the corresponding low-fidelity or discrepancy model, form the tensor polynomial chaos expansion or tensor interpolant corresponding to the grid, combine the trial expansion with the reference expansion for the particular level in the multifidelity hierarchy to which it corresponds (update $S_{\mathcal{J}_{low,d}}[R_{low}]$, $S_{\mathcal{J}_A,d}[\delta_A]$, or $S_{\mathcal{J}_M,d}[\delta_M]$), and then combine each of the levels to generate a trial high-fidelity expansion (\tilde{R}_{high}). Note that index sets associated with discrepancy expansions require evaluation of two levels of fidelity, so caching and reuse of the lowest and all intermediate fidelity evaluations should be performed among the different sparse grids. Bookkeeping should also be performed to allow efficient restoration of previously evaluated tensor expansions, as they will remain active until either selected or processed in the finalization step.
3. *Trial set selection:* From among all of the candidates, select the trial index set that induces the largest change in the high-fidelity statistical QoI, normalized by the cost of evaluating the trial index set (as indicated by the number of new collocation points and the relative model run time(s) per point). Initial estimates of relative simulation cost among the different fidelities (which we will denote as the ratio ρ_{work}) are thus required to appropriately bias the adaptation. To exploit greater coarse-grained parallelism or to achieve load balancing targets, multiple index sets may be selected, resulting in additional trial sets to evaluate on the following cycle. When multiple QoIs exist, trial sets may be rank-ordered using

a norm of QoI increments, or multiple sets could be selected that are each the best for at least one QoI (a non-dominated Pareto set).

4. *Update sets:* If the largest change induced by the active trial sets exceeds a specified convergence tolerance, then promote the selected trial set(s) from the active set to the reference set and update the active set with new admissible forward neighbors; return to step 2 and evaluate all active trial sets with respect to the new reference grid. If the convergence tolerance is satisfied, advance to step 5. An important algorithm detail for this step involves recursive set updating. In one approach, the selection of a discrepancy set could trigger the promotion of that set for both the discrepancy grid and the grid level(s) below it, in support of the notion that the grid refinements should be strictly hierarchical and support a spectrum from less to more resolved. Alternatively, one could let the sparse grid at each level evolve without constraint and ensure only the caching and reuse of evaluations among related levels. In this work, we employ the latter approach, as we wish to preserve the ability to under-resolve levels that are not providing predictive utility.
5. *Finalization:* Promote all remaining active sets to the reference set, update all expansions within the hierarchy, and perform a final combination of the low-fidelity and discrepancy expansions to arrive at the final result for the high-fidelity statistical QoI.

Figure 28.3 depicts a multifidelity sparse grid adaptation in process, for which the low-fidelity grid has three active index sets under evaluation and the discrepancy grid has four. These seven candidates are evaluated for influence on the statistical QoI(s), normalized by relative cost, in step 2 above. It is important to emphasize that all trial set evaluations (step 2) involve sets of actual model evaluations. That is, this algorithm does not identify the best candidates based on any a priori estimation; rather, the algorithm incrementally constructs sparse grids based on greedy selection of the best evaluated candidates in an a posteriori setting, followed by subsequent augmentation with new index sets that are the children of the selected index sets. In the end, the multi-index frontiers for the sparse grids have been advanced only in the regions with the greatest influence on the QoI, and all of the model evaluations (including index sets that were evaluated but never selected) contribute to the final answer, based on step 5.

In the limiting case where the low-fidelity model provides little useful information, this algorithm will prefer refinements to the model discrepancy and will closely mirror the single-fidelity case, with the penalty of carrying along the low-fidelity evaluations needed to resolve the discrepancy. This suggests an additional adaptive control, in which one drops low (and intermediate)-fidelity models from the hierarchy that are adding expense but not adding value (as measured by their frequency of selection in step 3). In addition, this general framework can be extended to include pointwise local refinement [28] (for handling nonsmoothness) as well as adjoint-enhanced approaches [1, 17] (for improving scalability with respect to random input dimension).

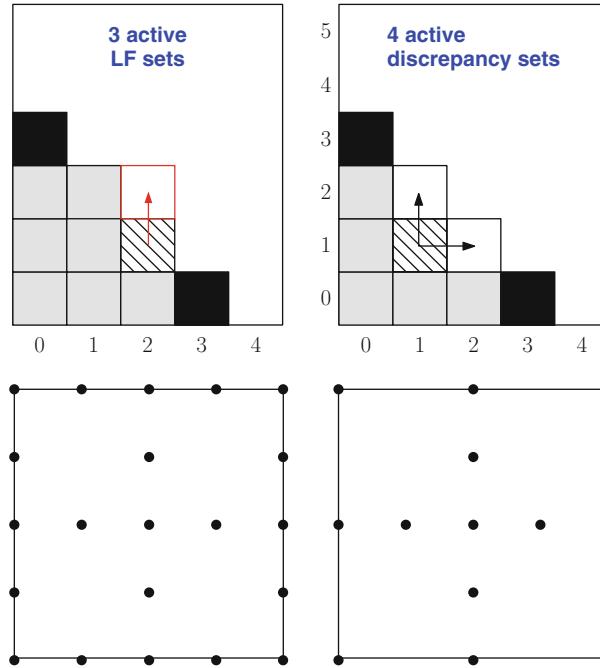


Fig. 28.3 Multifidelity generalized sparse grid with reference index sets in gray (existing) and hashed (newly promoted) and active index sets in black (existing) and white (newly created). Newton-Cotes grid points associated with the reference index sets are shown at bottom

3.4 Compressed Sensing

In the case of compressed sensing, we control the relative refinement of the low-fidelity model and the discrepancy model via the number of points within each sample set. We define a ratio $\rho_{\text{points}} = \frac{m_{\text{low}}}{m_{\text{high}}} \geq 1$, where the high-fidelity points must be contained within the low-fidelity point set to support the definition of the model discrepancy at these points.

We now revisit our surrogate high-fidelity expansions produced from additive, multiplicative, or combined correction to the low-fidelity model. Replacing our previous sparse grid operator $S_{q,d}[\cdot]$ in Eqs. 28.35–28.37 with a compressed sensing operator, we obtain

$$\tilde{R}_{\text{high}} = CS_{m_{\text{low}}, p^*, \varepsilon^*, d}[R_{\text{low}}] + CS_{m_{\text{high}}, p^*, \varepsilon^*, d}[\delta_A], \quad (28.38)$$

$$\tilde{R}_{\text{high}} = CS_{m_{\text{low}}, p^*, \varepsilon^*, d}[R_{\text{low}}]CS_{m_{\text{high}}, p^*, \varepsilon^*, d}[\delta_M], \quad (28.39)$$

$$\tilde{R}_{\text{high}} = \gamma(CS_{m_{\text{low}}, p^*, \varepsilon^*, d}[R_{\text{low}}] + CS_{m_{\text{high}}, p^*, \varepsilon^*, d}[\delta_A]) + (1 - \gamma)$$

$$CS_{m_{\text{low}}, p^*, \varepsilon^*, d}[R_{\text{low}}]CS_{m_{\text{high}}, p^*, \varepsilon^*, d}[\delta_M], \quad (28.40)$$

where $CS[.]$ is dependent on sample size m , total-order p of candidate basis, noise tolerance ε , and dimension d . We employ cross-validation separately for the low-fidelity model and each discrepancy in order to select the best p and ε for each of these sparse recoveries, as denoted by p^* and ε^* above.

3.5 Analytic Moments

In order to compute the moments of the multifidelity stochastic expansion analytically, we collapse the sum or product of the expansion of the low-fidelity model and the expansion of the discrepancy function into a single expansion and then employ the standard moment calculation techniques from Eqs. 28.7–28.8 and Eqs. 28.16–28.17.

3.5.1 Moments of Multifidelity PCE with Additive Discrepancy

This is most straightforward for polynomial chaos expansions with additive discrepancy, and we will start with this case using a predefined sparse grid offset. Let $\mathcal{J}_{q,d}$ be the set of multi-indices of the d -dimensional polynomial chaos expansion bases at sparse grid level q . Then,

$$S_{q,d}^{\text{pc}}[R_{\text{low}}](\xi) = \sum_{\mathbf{i} \in \mathcal{J}_{q,d}} \alpha_{\text{low}\mathbf{i}} \Psi_{\mathbf{i}}(\xi) \quad (28.41)$$

and

$$S_{q-r,d}^{\text{pc}}[\delta_A](\xi) = \sum_{\mathbf{i} \in \mathcal{J}_{q-r,d}} \alpha_{\delta_A \mathbf{i}} \Psi_{\mathbf{i}}(\xi). \quad (28.42)$$

Since the $\mathcal{J}_{q-r,d} \subset \mathcal{J}_{q,d}$, the bases $\Psi_{\mathbf{i}}(\xi)$, $\mathbf{i} \in \mathcal{J}_{q-r,d}$ are common between $S_{q,d}^{\text{pc}}[R_{\text{low}}]$ and $S_{q-r,d}^{\text{pc}}[\delta_A]$. Therefore, the chaos coefficients of those bases can be added to produce a single polynomial chaos expansion

$$S_{q,d}^{\text{pc}}[R_{\text{low}}](\xi) + S_{q-r,d}^{\text{pc}}[\delta_A](\xi) = \sum_{\mathbf{i} \in \mathcal{J}_{q-r,d}} (\alpha_{\text{low}\mathbf{i}} + \alpha_{\delta_A \mathbf{i}}) \Psi_{\mathbf{i}}(\xi) + \sum_{\mathbf{i} \in \mathcal{J}_{q,d} \setminus \mathcal{J}_{q-r,d}} \alpha_{\text{low}\mathbf{i}} \Psi_{\mathbf{i}}(\xi). \quad (28.43)$$

The mean and the variance can be computed directly from this multifidelity expansion by simply collecting terms for each basis and applying Eqs. 28.7 and 28.8:

$$\mu_R \approx \alpha_{\text{low}\mathbf{0}} + \alpha_{\delta_A \mathbf{0}} \quad (28.44)$$

$$\sigma_R^2 \approx \sum_{\mathbf{i} \in \mathcal{J}_{q-r,d} \setminus \mathbf{0}} (\alpha_{\text{low}\mathbf{i}} + \alpha_{\delta_A \mathbf{i}})^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle + \sum_{\mathbf{i} \in \mathcal{J}_{q,d} \setminus \mathcal{J}_{q-r,d}} \alpha_{\text{low}\mathbf{i}}^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle. \quad (28.45)$$

In the adaptive sparse grid and compressed sensing cases, the primary difference is the loss of a strict subset relationship between the discrepancy and low-fidelity multi-indices, which we will represent in generalized form as \mathcal{J}_{δ_A} and \mathcal{J}_{low} , respectively. We introduce a new multi-index for the set intersection $\mathcal{J}_I = \mathcal{J}_{\delta_A} \cap \mathcal{J}_{\text{low}}$. Eq. 28.44 remains the same, as all approaches retain the $\mathbf{0}$ term, but Eq. 28.45 becomes generalized as follows:

$$\sigma_R^2 \approx \sum_{\mathbf{i} \in \mathcal{J}_I \setminus \mathbf{0}} (\alpha_{\text{low}\mathbf{i}} + \alpha_{\delta_A \mathbf{i}})^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle + \sum_{\mathbf{i} \in \mathcal{J}_{\text{low}} \setminus \mathcal{J}_I} \alpha_{\text{low}\mathbf{i}}^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle + \sum_{\mathbf{i} \in \mathcal{J}_{\delta_A} \setminus \mathcal{J}_I} \alpha_{\delta_A \mathbf{i}}^2 \langle \Psi_{\mathbf{i}}^2(\xi) \rangle. \quad (28.46)$$

3.5.2 Moments of Multifidelity PCE with Multiplicative Discrepancy

In the multiplicative discrepancy case for non-intrusive polynomial chaos, we again combine the low-fidelity and discrepancy expansions and then compute the moments from the aggregated expansion. Multiplication of chaos expansions is a kernel operation within stochastic Galerkin methods. The coefficients of a product expansion are computed as follows (shown generically for $z = xy$ where x , y , and z are each expansions of arbitrary form):

$$\sum_{k=0}^{P_z} z_k \Psi_k(\xi) = \sum_{i=0}^{P_x} \sum_{j=0}^{P_y} x_i y_j \Psi_i(\xi) \Psi_j(\xi) \quad (28.47)$$

$$z_k = \frac{\sum_{i=0}^{P_x} \sum_{j=0}^{P_y} x_i y_j \langle \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k^2 \rangle} \quad (28.48)$$

where three-dimensional tensors of one-dimensional basis triple products $\langle \psi_i \psi_j \psi_k \rangle$ are typically sparse and can be efficiently precomputed using one-dimensional quadrature for fast lookup within the multidimensional basis triple products $\langle \Psi_i \Psi_j \Psi_k \rangle$. The form of the high-fidelity expansion (enumerated by P_z in Eq. 28.47) must first be defined to include all polynomial orders indicated by the products of each of the basis polynomials in the low-fidelity and discrepancy expansions, in order to avoid any artificial truncation in the product expansion. These are readily estimated from total-order, tensor, or sum of tensor expansions since they involve simple polynomial order additions for each tensor or total-order expansion product.

3.5.3 Moments of Multifidelity SC

Evaluating the moments for stochastic collocation with either additive or multiplicative discrepancy involves forming a new interpolant on the more refined (low-fidelity) grid. Therefore, we perform an additional step to create a single stochastic expansion

$$S_{q,d}^{\text{sc}} \left\{ S_{q,d}^{\text{sc}} [R_{\text{low}}] + S_{q-r,d}^{\text{sc}} [\delta_A] \right\} \quad (28.49)$$

$$S_{q,d}^{\text{sc}} \left\{ S_{q,d}^{\text{sc}} [R_{\text{low}}] S_{q-r,d}^{\text{sc}} [\delta_M] \right\} \quad (28.50)$$

from which the variance and higher moments can be obtained analytically. This requires evaluating $S_{q,d}^{\text{sc}}[R_{\text{low}}]$ and either $S_{q-r,d}^{\text{sc}}[\delta_A]$ or $S_{q-r,d}^{\text{sc}}[\delta_M]$ at the collocation points associated with the multi-indices in $\mathcal{J}_{q,d}$. For the former, the low-fidelity model values at all of the collocation points are readily available and can be used directly. For the latter, the discrepancy expansion $S_{q-r,d}^{\text{sc}}[\delta_A]$ or $S_{q-r,d}^{\text{sc}}[\delta_M]$ must be evaluated. Since the discrepancy function values are available at collocation points associated with the multi-indices of $\mathcal{J}_{q-r,d}$, it may be tempting to only evaluate $S_{q-r,d}^{\text{sc}}[\delta_A]$ or $S_{q-r,d}^{\text{sc}}[\delta_M]$ at collocation points associated with the multi-indices in $\mathcal{J}_{q,d} \setminus \mathcal{J}_{q-r,d}$. However, because sparse grid stochastic collocation does not interpolate unless the set of collocation points are nested [7], the function values from δ_A and δ_M and from $S_{q-r,d}^{\text{sc}}[\delta_A]$ and $S_{q-r,d}^{\text{sc}}[\delta_M]$ should not be mixed together when they are not consistent.

To generalize to adaptive sparse grid cases that could lack the strict subset relationship $\mathcal{J}_{q-r,d} \subset \mathcal{J}_{q,d}$ provided by a predefined level offset, we introduce a multi-index union \mathcal{J}_U and form a new interpolant on the union grid:

$$S_U^{\text{sc}} \{ S_{\text{low}}^{\text{sc}}[R_{\text{low}}] + S_{\delta_A}^{\text{sc}}[\delta_A] \} \quad \text{for } \mathcal{J}_U = \mathcal{J}_{\text{low}} \cup \mathcal{J}_{\delta_A} \quad (28.51)$$

$$S_U^{\text{sc}} \{ S_{\text{low}}^{\text{sc}}[R_{\text{low}}] S_{\delta_M}^{\text{sc}}[\delta_M] \} \quad \text{for } \mathcal{J}_U = \mathcal{J}_{\text{low}} \cup \mathcal{J}_{\delta_M}. \quad (28.52)$$

Similar to the predefined offset case, the new interpolant is formed by evaluating $S_{\text{low}}^{\text{sc}}[R_{\text{low}}]$, $S_{\delta_A}^{\text{sc}}[\delta_A]$, and/or $S_{\delta_M}^{\text{sc}}[\delta_M]$ at the collocations points associated with the multi-indices in \mathcal{J}_U and computing the necessary sum or product.

4 Computational Results

We compare the performance of multifidelity stochastic expansion and single-fidelity stochastic expansion for several algebraic and PDE models of increasing complexity. We demonstrate cases for which the multifidelity stochastic expansion converges more quickly than the single-fidelity stochastic expansion as well as cases for which the multifidelity stochastic expansion offers little to no efficiency gain.

4.1 Simple One-Dimensional Example

First, we present a simple example to motivate the approach and demonstrate the efficiency improvements that are possible when an accurate low-fidelity model is available. The system responses of the high-fidelity model and the low-fidelity model are, respectively,

$$R_{\text{high}}(\xi) = e^{-0.05\xi^2} \cos 0.5\xi - 0.5e^{-0.02(\xi-5)^2}$$

$$R_{\text{low}}(\xi) = e^{-0.05\xi^2} \cos 0.5\xi,$$

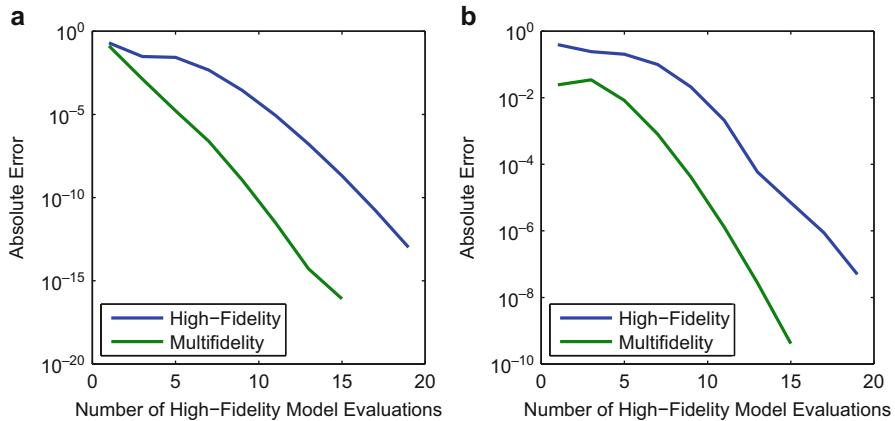


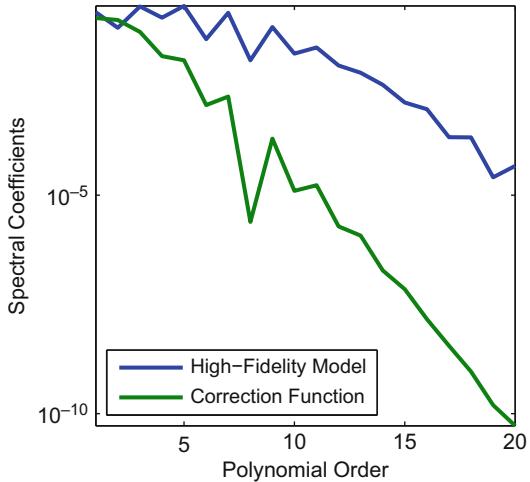
Fig. 28.4 Convergence of single-fidelity PCE and multifidelity PCE with additive discrepancy for the one-dimensional example. (a) Error in mean. (b) Error in standard deviation (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

where $\xi \sim \text{Uniform}[-8, 12]$. An additive discrepancy $\delta_\alpha(\xi) = R_{\text{high}}(\xi) - R_{\text{low}}(\xi)$ is used, which is just the second term of $R_{\text{high}}(\xi)$. In Fig. 28.4, we compare the convergence in mean and standard deviation of the single (high)-fidelity PCE with the convergence of the multifidelity PCE. The multifidelity PCE is constructed from a PCE of the discrepancy function at order 1 to 20 combined with a PCE of the low-fidelity model at order 60 (for which the low-fidelity statistics are converged to machine precision). This corresponds to a case where low-fidelity expense can be considered to be negligible, and by eliminating any issues related to low-fidelity accuracy, we can focus more directly on comparing the convergence of the discrepancy function with convergence of the high-fidelity model (in the next example, we will advance to grid level offsets that accommodate nontrivial low-fidelity expense). The error is plotted against the number of high-fidelity model evaluations and is measured with respect to an overkill single-fidelity PCE solution at order 60. It is evident that the multifidelity stochastic expansion converges much more rapidly because the additive discrepancy function in the example has lower complexity than the high-fidelity model. This can be seen from comparison of the decay of the normalized spectral coefficients as plotted in Fig. 28.5, which shows that the discrepancy expansion decays more rapidly allowing the statistics of the discrepancy expansion to achieve a given accuracy using fewer PCE terms than that required by the high-fidelity model.

4.2 Short Column Example

The short column example [32] demonstrates a higher-dimensional algebraic problem involving multifidelity models. For this example, we will employ predefined

Fig. 28.5 Normalized spectral coefficients of the high-fidelity model and the additive discrepancy function for the one-dimensional example (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)



sparse grid level offsets between low fidelity and high fidelity, which are more appropriate for cases where the low-fidelity model expense is non-negligible. Let the system response of the high-fidelity model be

$$R_{\text{high}}(\xi) = 1 - \frac{4M}{bh^2Y} - \left(\frac{P}{bhY} \right)^2, \quad (28.53)$$

where $\xi = (b, h, P, M, Y)$, $b \sim \text{Uniform}(5, 15)$, $h \sim \text{Uniform}(15, 25)$, $P \sim \text{Normal}(500, 100)$, $M \sim \text{Normal}(2000, 400)$, and $Y \sim \text{Lognormal}(5, 0.5)$ and we neglect the traditional correlation between P and M for simplicity. We consider three artificially constructed low-fidelity models of varying predictive quality, which yield additive discrepancy forms of varying complexity:

$$R_{\text{low1}}(\xi) = 1 - \frac{4P}{bh^2Y} - \left(\frac{P}{bhY} \right)^2, \quad \delta_{A1}(\xi) = \frac{4(P - M)}{bh^2Y}, \quad (28.54)$$

$$R_{\text{low2}}(\xi) = 1 - \frac{4M}{bh^2Y} - \left(\frac{M}{bhY} \right)^2, \quad \delta_{A2}(\xi) = \frac{M^2 - P^2}{(bhY)^2}, \quad (28.55)$$

$$R_{\text{low3}}(\xi) = 1 - \frac{4M}{bh^2Y} - \left(\frac{P}{bhY} \right)^2 - \frac{4(P - M)}{bhY}, \quad \delta_{A3}(\xi) = \frac{4(P - M)}{bhY}. \quad (28.56)$$

4.2.1 Isotropic Sparse Grids

In Fig. 28.6, PCE with isotropic sparse grids is used, and the offset r in the sparse grid level between the low-fidelity model and the discrepancy function is fixed at one. It can be seen that the multifidelity case using $R_{\text{low}1}(\xi)$ results in a reduction in the number of high-fidelity model evaluations required for a given error compared to the single-fidelity case using $R_{\text{high}}(\xi)$. For example, at 10^{-5} error, the number of high-fidelity model evaluations is reduced from about 500 to about 100. While still a rational function with broad spectral content, the discrepancy function, $\delta_{A1}(\xi)$, is similar to the middle term in Eq. 28.53 and has eliminated the final term possessing the greatest nonlinearity. Conversely, the discrepancy function for the second low-fidelity model, $\delta_{A2}(\xi)$, is similar to the final term in Eq. 28.53 and has expanded it to include an additional dimension, resulting in a larger number of high-fidelity model evaluations for a given error compared to the single-fidelity case. For the third low-fidelity model, the convergence rate is faster than the single-fidelity case but the starting error is also larger, resulting in a break-even point at about 11 high-fidelity model evaluations. This suggests that while a less complex discrepancy results in more rapid convergence, it is also important to consider the magnitude and resulting variance of the discrepancy function.

We modify $R_{\text{low}3}(\xi)$ by changing the scalar in the last term from 4 to 0.4 and label it $R_{\text{low}4}(\xi)$. Similarly, we also change the scalar in the last term from 4 to 40 and label it $R_{\text{low}5}(\xi)$. Thus, the discrepancy functions $\delta_{A3}(\xi)$, $\delta_{A4}(\xi)$, and $\delta_{A5}(\xi)$ have the same smoothness and spectral content, but the magnitude of the discrepancy is an order of magnitude smaller for $\delta_{A4}(\xi)$ and an order of magnitude larger for $\delta_{A5}(\xi)$. As plotted in Fig. 28.7, the means have similar convergence rates, but a smaller discrepancy results in lower error.

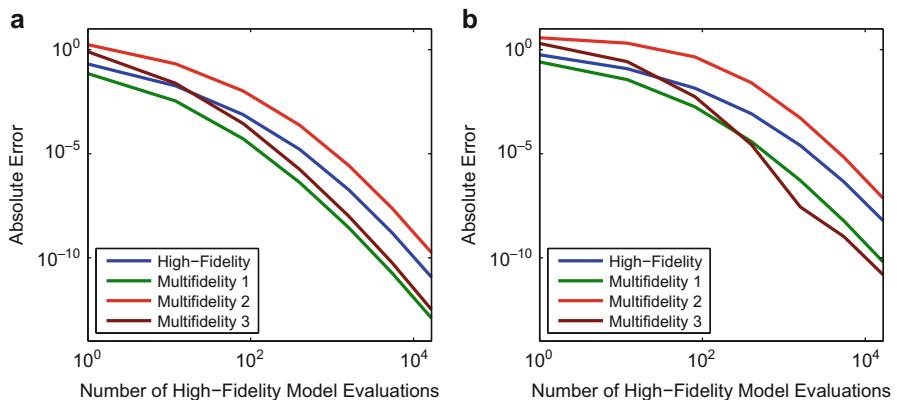


Fig. 28.6 Convergence of single-fidelity and multifidelity PCE with isotropic sparse grids and additive discrepancy for the short column example. The multifidelity sparse grid level offset $r = 1$. **(a)** Error in mean. **(b)** Error in standard deviation

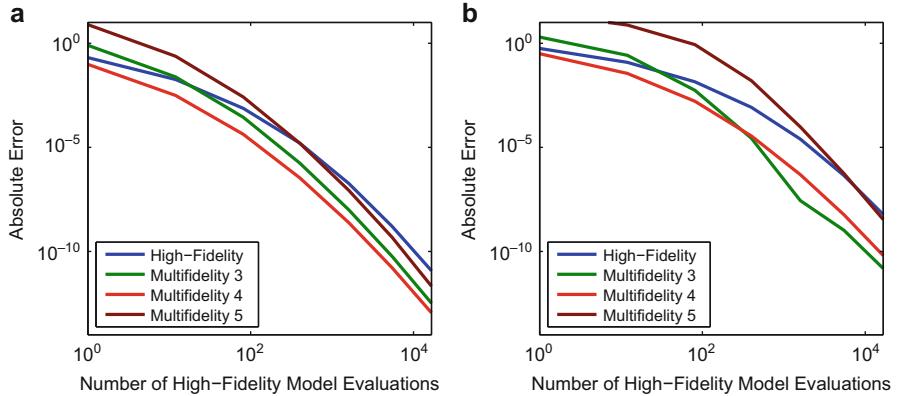


Fig. 28.7 Convergence of single-fidelity PCE and multifidelity PCE with isotropic sparse grids and additive discrepancy for the short column example. The multifidelity sparse grid level offset $r = 1$. (a) Error in mean. (b) Error in standard deviation (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

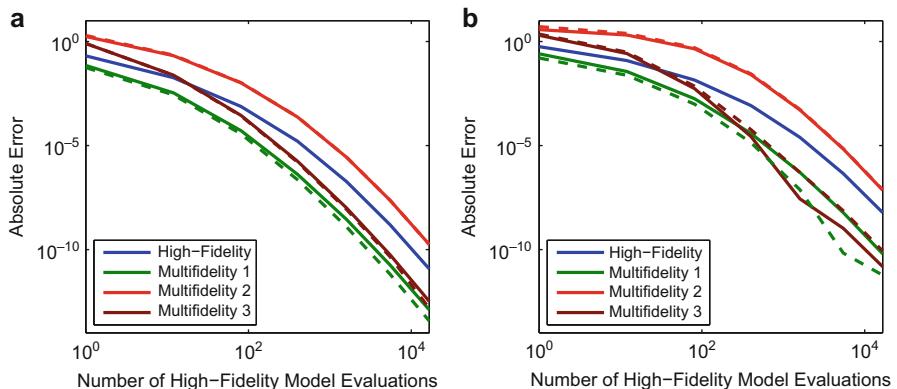


Fig. 28.8 Convergence of single-fidelity PCE and multifidelity PCE with isotropic sparse grids and additive discrepancy for the short column example. The multifidelity sparse grid level offset is compared using $r = 1$ from Fig. 28.6 (solid lines) and $r = 2$ (dashed lines). (a) Error in mean. (b) Error in standard deviation (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

Finally, we investigate the effect of the sparse grid level offset r . Figure 28.8 is the same as Fig. 28.6 but with r increased from one to two, resulting in greater resolution in the low-fidelity expansion for a particular discrepancy expansion resolution. The results from Fig. 28.6 are included as solid lines, and the new results with $r = 2$ are shown as dashed lines. A small improvement can be seen

in the mean convergence for multifidelity using $R_{\text{low}1}(\xi)$ and $R_{\text{low}3}(\xi)$ and for standard deviation convergence using $R_{\text{low}1}(\xi)$, but results are mixed and it is unclear whether the benefit of increasing the offset is worth the additional low-fidelity evaluations, especially in the case where their expense is nontrivial. Thus, it appears that an automated procedure will be needed to optimize these offsets accounting for relative cost. This motivates the multifidelity adaptive sparse grid algorithm described previously (Fig. 28.3).

4.2.2 Compressed Sensing

In Fig. 28.9, we compare the results of applying compressed sensing (CS) for resolving the low-fidelity and discrepancy expansions, using the OMP algorithm for solving Eq. 28.26 in combination with cross-validation for candidate basis order p and noise tolerance ε . The reference sparse grid results are for a level offset $r = 1$ and are consistent with those from Fig. 28.6. For CS, we employ a sampling ratio $\rho_{\text{points}} = 10$ between the low- and high-fidelity sample sets, and since these samples are randomly generated, we average the results of 10 runs for the reported errors. Monte Carlo results are provided as another reference point and are also averaged over 10 runs. The relative low-fidelity expense will differ when using a predefined sparse grid level offset r and a fixed CS sample ratio ρ_{points} due to the nonlinear growth in sparse grid size. Therefore, we introduce the cost of the low-fidelity model by selecting a representative cost ratio $\rho_{\text{work}} = 10$ and plot convergence against the number of equivalent high-fidelity evaluations, which is defined as $m_{\text{eqv}} = m_{\text{high}} + m_{\text{low}}/\rho_{\text{work}}$. We observe similar trends with CS as for sparse grids: multifidelity CS is an improvement over single-fidelity CS for $R_{\text{low}1}$ but not for $R_{\text{low}2}$, and multifidelity CS becomes more competitive for higher resolution levels in $R_{\text{low}3}$. It is also evident that neither the single-fidelity nor the multifidelity CS approaches outperform their sparse grid counterparts for this problem. This implies that the spectra of coefficients in these artificially constructed discrepancy models (Eqs. 28.54–28.56) are dense and relatively low order.

4.3 Elliptic PDE Example

Next, we consider the stochastic PDE in one spatial dimension

$$-\frac{d}{dx} \left[\kappa(x, \omega) \frac{du(x, \omega)}{dx} \right] = 1, \quad x \in (0, 1), \quad u(0, \omega) = u(1, \omega) = 0.$$

with coefficient κ described by the following 10-dimensional Karhunen-Loëve expansion

$$\kappa(x, \omega) = 0.1 + 0.03 \sum_{k=1}^{10} \sqrt{\lambda_k} \phi_k(x) Y_k(\omega), \quad Y_k \sim \text{Uniform}(-1, 1)$$

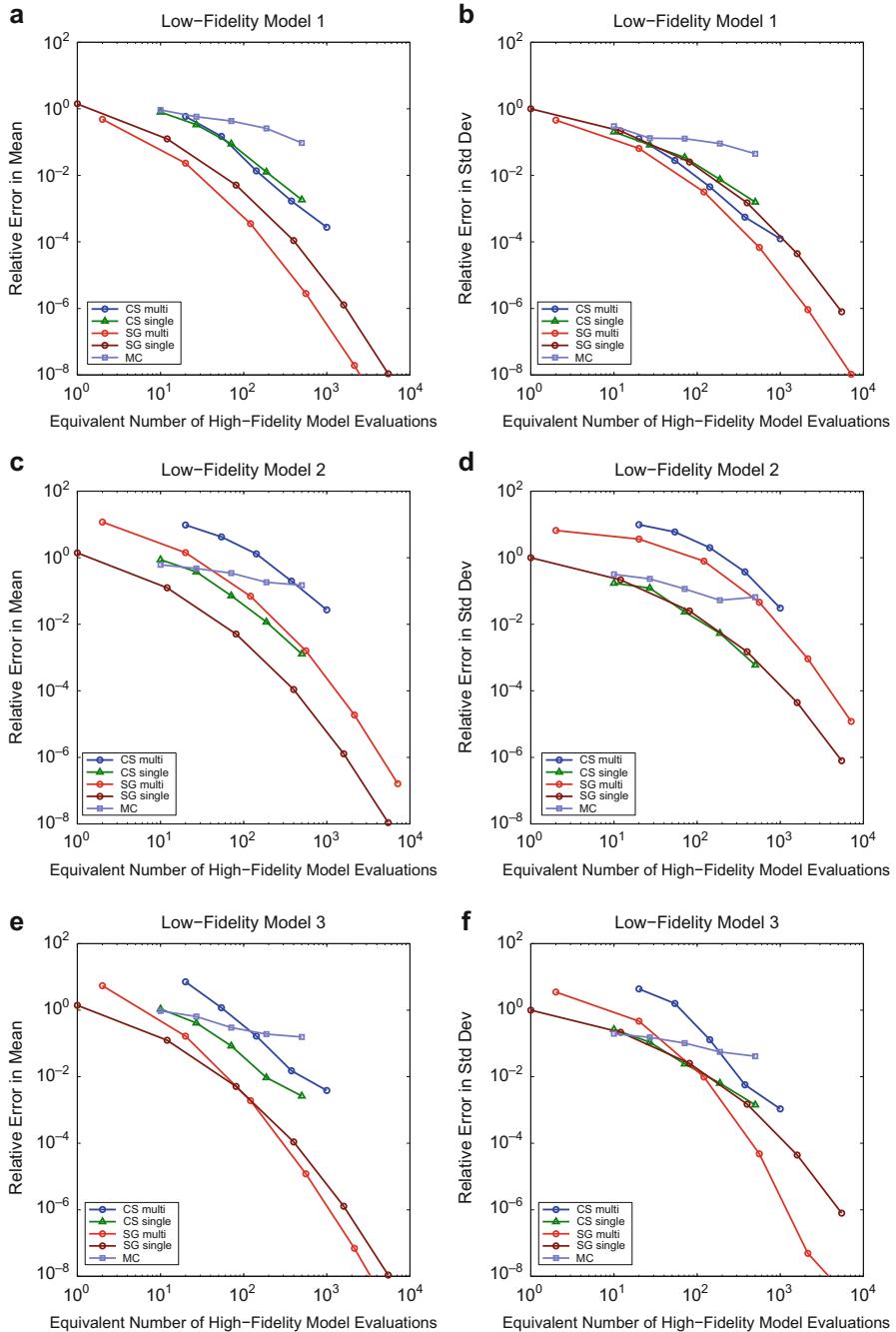
**Fig. 28.9** (continued)

Table 28.2 Comparison of the relative error and the number of model evaluations for the elliptic PDE example

	Relative error in mean	Relative error in std deviation	High fidelity evaluations	Low fidelity evaluations
Single fidelity ($q = 3$)	5.3×10^{-6}	2.7×10^{-4}	1981	—
Single fidelity ($q = 4$)	4.1×10^{-7}	2.3×10^{-5}	12,981	—
Multifidelity ($q = 4, r = 1$)	4.7×10^{-7}	2.6×10^{-5}	1981	12,981

From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.

for the Gaussian covariance kernel

$$C_{kk}(x, x') = \exp \left[-\left(\frac{x - x'}{0.2} \right)^2 \right].$$

The PDE is solved by finite elements and the output of interest is $u(0.5, \omega)$. We use a fine spatial grid with 500 states for the high-fidelity model and a coarse spatial grid with 50 states for the low-fidelity model. The ratio of average run time between the high-fidelity model and the low-fidelity model is $\rho_{\text{work}} = 40$.

4.3.1 Sparse Grids

We compute the mean and standard deviation using multifidelity PCE with sparse grid level 4 applied to the low-fidelity model and sparse grid level 3 applied to the additive correction function (i.e., sparse grid level offset $r = 1$). Table 28.2 compares the relative error of this multifidelity approach with single-fidelity PCE of the high-fidelity model at sparse grid levels 3 and 4. It can be seen that the multifidelity PCE is able to achieve an order of magnitude lower error than the single-fidelity PCE at sparse grid level 3 while using the same number of high-fidelity evaluations. The cost of low-fidelity evaluations is equivalent to about 325 additional high-fidelity evaluations, resulting in greater than an 80% reduction in total cost for comparable accuracy to the single-fidelity PCE result at sparse grid level 4.

Fig. 28.9 Convergence of single-fidelity and multifidelity PCE comparing compressed sensing with isotropic sparse grids for the short column example for R_{low1} , R_{low2} , and R_{low3} . Discrepancy is additive, multifidelity sparse grid level offset is $r = 1$, $\rho_{\text{work}} = 10$, and $\rho_{\text{points}} = 10$. (a) Error in mean for R_{low1} . (b) Error in standard deviation for R_{low1} . (c) Error in mean for R_{low2} . (d) Error in standard deviation for R_{low2} . (e) Error in mean for R_{low3} . (f) Error in standard deviation for R_{low3}

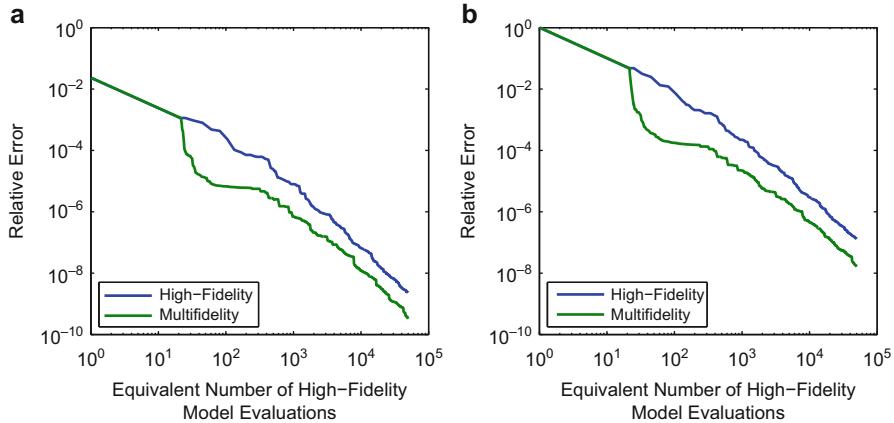


Fig. 28.10 Convergence of single-fidelity and multifidelity PCE with additive discrepancy using adaptive sparse grids for the elliptic PDE example. **(a)** Error in mean. **(b)** Error in standard deviation (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

Figure 28.10 shows the convergence for the single-fidelity and multifidelity PCE with additive discrepancy based on adaptive refinement using generalized sparse grids. The single-fidelity case uses the standard generalized sparse grid procedure [21], whereas the multifidelity case uses the multifidelity adaptive sparse grid algorithm depicted in Fig. 28.3. The initial grid for both the low-fidelity model and the correction function is a level one sparse grid (requiring 11 model evaluations). We use the equivalent number of high-fidelity model evaluations (m_{eqv}) to include the additional cost of low-fidelity model evaluations in the comparison with the single-fidelity case. By considering the potential error reduction per unit cost of refining the sparse grid of the discrepancy versus that of refining the sparse grid of the low-fidelity model, the multifidelity adaptive algorithm is able to achieve a faster convergence than the single-fidelity adaptive generalized sparse grid. For achieving error levels consistent with the non-adapted multifidelity approach in Table 28.2, the adaptive multifidelity algorithm reduces the equivalent number of high-fidelity evaluations by 33% for the mean and by 62% for the standard deviation.

4.3.2 Compressed Sensing

Figure 28.11 compares the convergence in standard deviation for multifidelity compressed sensing with different point ratios ρ_{points} against single-fidelity compressed sensing, non-adaptive single-fidelity sparse grids, non-adaptive multifidelity sparse grids using level offset $r = 1$, and Monte Carlo sampling. As for the short column example, convergence plots for Monte Carlo sampling and CS are averaged over 10

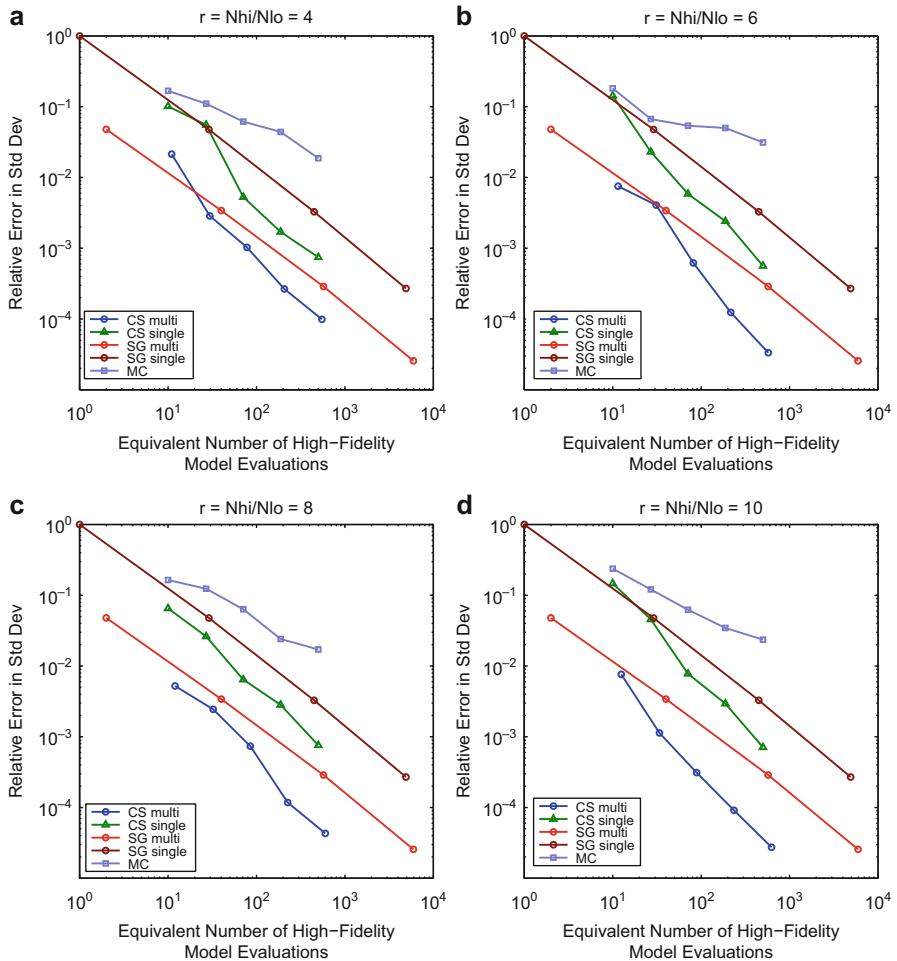


Fig. 28.11 Convergence of standard deviation in elliptic PDE problem using multifidelity compressed sensing with different ρ_{points} ratios. (a) $\rho_{\text{points}} = 4$. (b) $\rho_{\text{points}} = 6$. (c) $\rho_{\text{points}} = 8$. (d) $\rho_{\text{points}} = 10$

runs, and CS employs the OMP solver in combination with cross-validation to select the total-order polynomial degree p of the candidate basis and the noise tolerance ε . It is evident that CS-based approaches perform better for this problem than the sparse grid approaches, and the benefit of multifidelity CS over single-fidelity CS is comparable to that of multifidelity sparse grids over single-fidelity sparse grids. Finally, increasing ρ_{points} is advantageous for this problem, implying strong predictivity in the low-fidelity model and allowing for lower relative investment in resolving the discrepancy.

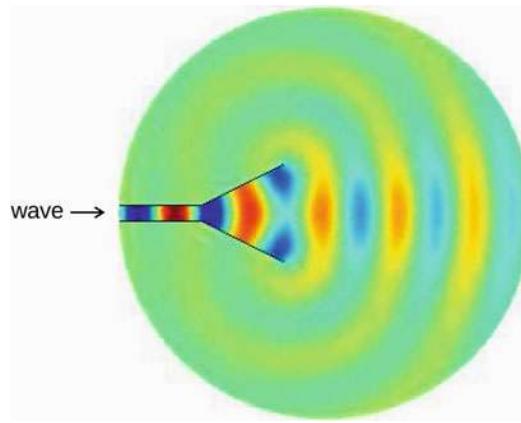


Fig. 28.12 2D horn geometry and the propagation of acoustic waves (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

4.4 Horn Acoustics Example

We model the propagation of acoustic waves through a two-dimensional horn with the non-dimensional Helmholtz equation $\nabla^2 u + k^2 u = 0$ for wave number k . The incoming wave enters the waveguide and exits the flare of the horn into the exterior domain with a truncated absorbing boundary [14]. The horn geometry is illustrated in Fig. 28.12. The stochastic parameters are the wave number $k \sim \text{Uniform}(1.3, 1.5)$, upper horn wall impedance $z_u \sim \text{Normal}(50, 9)$, and lower horn wall impedance $z_l \sim \text{Normal}(50, 9)$, where the latter two represent imperfections in the horn wall. We compute the mean and standard deviation of the reflection coefficient, where a low reflection coefficient is desired for an efficient horn. The high-fidelity model solves the Helmholtz equation by finite elements using 35895 states, and the low-fidelity model is a reduced- basis model constructed from the finite-element discretization [40] using 50 bases. The ratio of average run time between the high-fidelity model and the low-fidelity model is $\rho_{\text{work}} = 40$.

4.4.1 Adaptive Sparse Grids

Figure 28.13 compares the convergence between single-fidelity and multifidelity PCE with an additive discrepancy based on adaptive refinement with generalized sparse grids. For this problem, the multifidelity adaptive sparse grid approach offers little discernable improvement over a single-fidelity adaptive sparse grid, with at best a slight reduction in standard deviation error at low resolution levels. The reduced-basis model (i.e., the low-fidelity model) interpolates the finite-element model at the 50 snapshots used to generated the bases, but despite its accuracy (the maximum discrepancy between the reduced-basis model and the finite-element

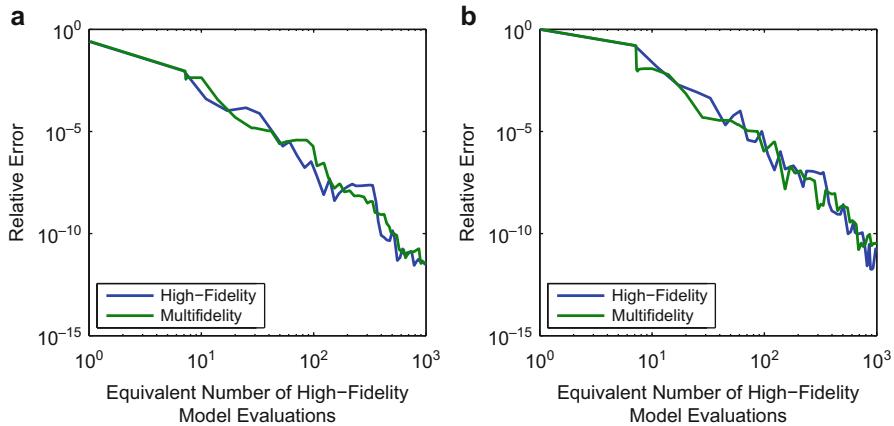


Fig. 28.13 Convergence of single-fidelity PCE and multifidelity PCE with additive correction using adaptive generalized sparse grids for the acoustic horn example. (a) Error in mean. (b) Error in standard deviation (From *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*, by Ng and Eldred, 2012, AIAA-2012-1852, published in the Proceedings of the 53rd SDM conference; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

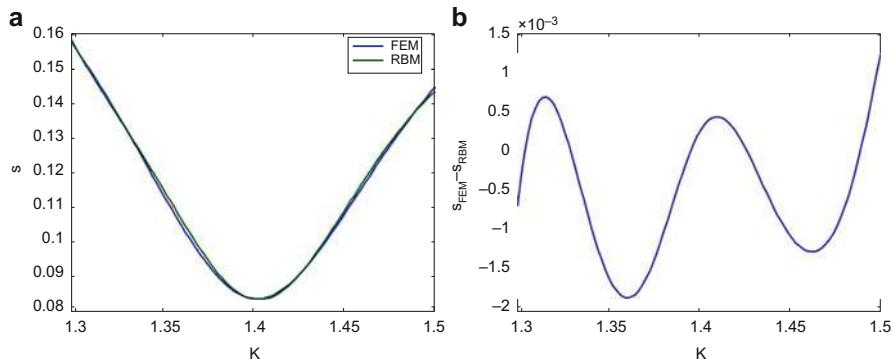


Fig. 28.14 Comparison of finite-element model (FEM) and reduced-basis model (RBM) results for the acoustic horn example. (a) Comparison of low- and high-fidelity QoI. (b) Model discrepancy

model is about 2%), its interpolatory nature results in oscillations that require a higher-order PCE expansion to resolve (Fig. 28.14). In addition, reduced-order modeling approaches based on projection of a truncated basis will in general tend to retain dominant lower-order effects and omit higher-order behavior. This highlights the primary weakness of a multifidelity sparse grid approach; it must step through lower grid levels to reach higher grid levels, such that a multifidelity sparse grid approach has difficulty benefiting from a low-fidelity model that only predicts low-order effects. The presence of similar high-order content within the discrepancy

and high-fidelity models then results in similar high-fidelity resolution requirements for the single-fidelity and multifidelity approaches. This is precisely the case that motivates methods that can utilize sparse recovery to more efficiently resolve high-order discrepancy terms.

4.4.2 Compressed Sensing

Figure 28.15 compares the convergence in standard deviation for multifidelity CS with different point ratios ρ_{points} against single-fidelity CS, non-adaptive single-fidelity sparse grids, non-adaptive multifidelity sparse grids using level offset $r = 1$,

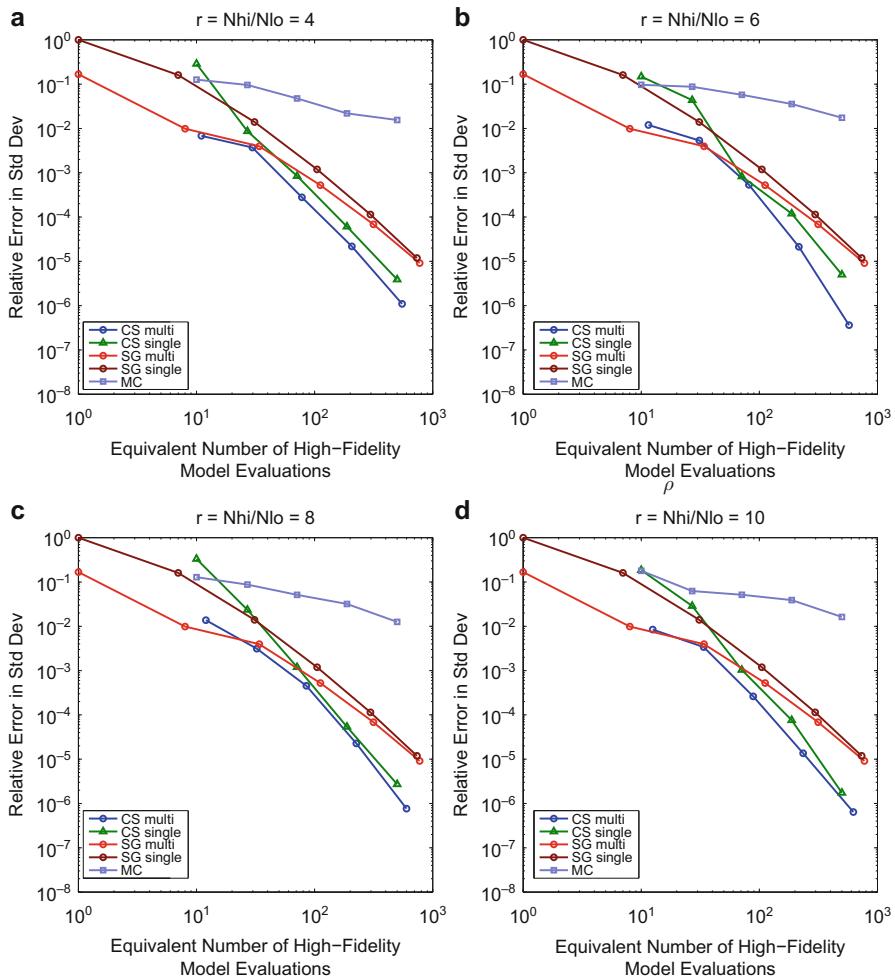


Fig. 28.15 Convergence of standard deviation in horn problem using multifidelity compressed sensing with different ρ_{points} values. (a) $\rho_{\text{points}} = 4$ (b) $\rho_{\text{points}} = 6$. (c) $\rho_{\text{points}} = 8$. (d) $\rho_{\text{points}} = 10$

and Monte Carlo sampling. As for previous examples, convergence plots for Monte Carlo sampling and CS are averaged over 10 runs, and CS employs the OMP solver in combination with cross-validation to select the total-order polynomial degree p of the candidate basis and the noise tolerance ε . For the non-adapted sparse grid approaches, the multifidelity PCE shows a small improvement relative to the single-fidelity approach, although the amount of improvement decreases with resolution level (similar to the adapted sparse grid result in Fig. 28.13b). For the CS approaches, more rapid convergence overall is evident than for the sparse grid approaches, indicating sparsity or compressibility in the coefficient spectrum. The multifidelity CS approaches show modest improvements relative to the single-fidelity approaches, with the greatest separation corresponding to $\rho_{\text{points}} = 6$. Comparing this observation to the elliptic PDE problem where the highest point ratios performed best, one could infer that solutions for the horn problem require greater reliance on the high-fidelity model for resolving the high-order discrepancy effects.

Figure 28.16 plots the spectrum of expansion coefficients α_i comparing single-fidelity with multifidelity CS for different point ratios ρ_{points} . The single-fidelity PCE coefficients recovered from $m = 200$ samples are compared to the multifidelity PCE coefficients using the same 200 high-fidelity samples in combination with 800, 1200, 1600, and 2000 low-fidelity samples for $\rho_{\text{points}} = 4, 6, 8$, and 10, respectively. These cases are plotted against reference single-fidelity cases for which 800, 1200, 1600, and 2000 samples are performed solely on the high-fidelity model. All CS solutions employ cross-validation to select the most accurate candidate basis order p and noise tolerance ε . It is evident that all cases are in agreement with respect to capturing the dominant terms with the largest coefficient magnitude (terms greater than approximately 10^{-5}). Differences manifest for the smaller coefficients, with the more resolved single-fidelity reference solutions (blue) recovering many additional terms relative to the 200-sample single-fidelity solutions (green), with some relative inaccuracy apparent in the smallest terms for the latter. Augmenting the 200 high-fidelity samples with low-fidelity simulations allows the multifidelity approach (red) to more effectively approximate these less-dominant terms, effectively extending the recovered spectrum to a level similar to that of the high-fidelity reference solution.

4.5 Production Engineering Example: Vertical-Axis Wind Turbine

Wind turbine reliability plays a critical role in the long-term prospects for cost-effective wind-based energy generation. The computational assessment of failure probability or life expectancy of turbine components is fundamentally hindered by the presence of large uncertainties in the environmental conditions, the blade structure, and in the form of turbulence closure models that are used to simulate complex flow. Rigorous quantification of the impact of such uncertainties can fundamentally improve the state of the art in computational predictions and, as a result, aid in the design of more cost-effective devices.

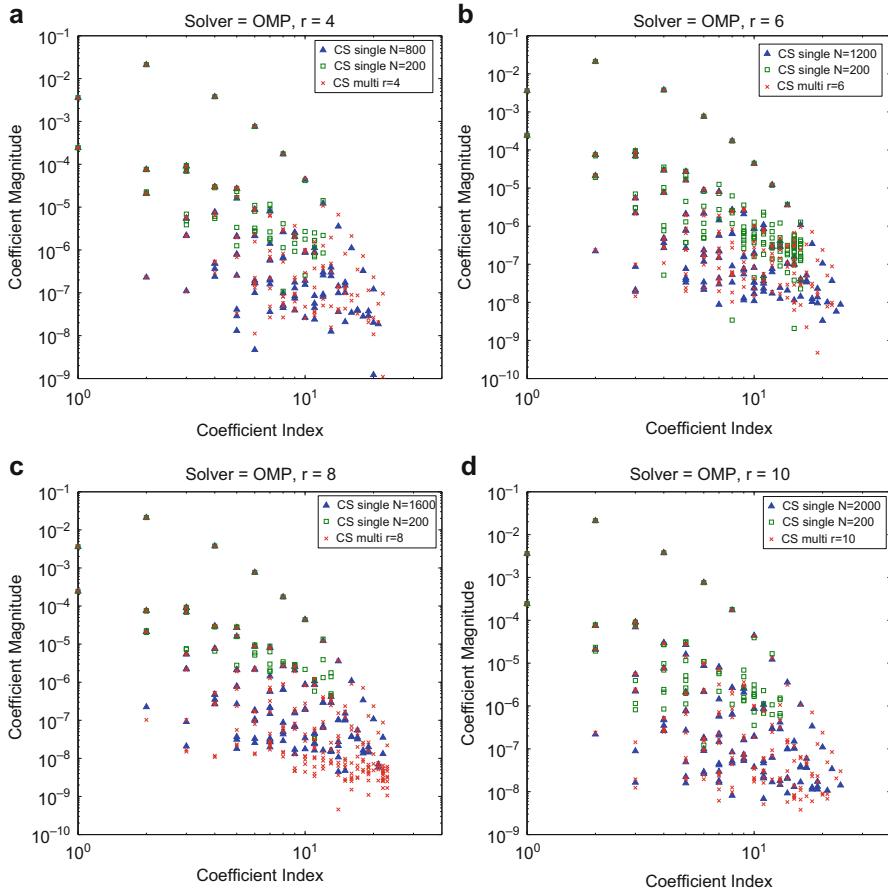


Fig. 28.16 Coefficient spectrum for horn problem using multifidelity compressed sensing with different ρ_{points} values. (a) $\rho_{\text{points}} = 4$. (b) $\rho_{\text{points}} = 6$. (c) $\rho_{\text{points}} = 8$. (d) $\rho_{\text{points}} = 10$

4.5.1 Simulation Tools

An aerodynamic model for a horizontal-axis wind turbine is necessarily three-dimensional, since it is comprised of two or three blades rotating about an axis parallel to the oncoming wind. A vertical-axis wind turbine (VAWT), where the axis of rotation is normal to the wind vector (Fig. 28.17a), allows for a meaningful two-dimensional analysis of one cross section of the rotor. This makes the VAWT a useful bridging problem for investigation of UQ methods employing high-fidelity simulation, since methods can be developed and verified using 2D problems before extension to 3D. In the current context, the 2D analysis of a VAWT subject to uncertain gust phenomena provides our final production-level demonstration of multifidelity UQ methods.

Our low-fidelity model is CACTUS (Code for Axial and Crossflow TURbine Simulation) [34]. CACTUS is a three-dimensional potential flow code developed

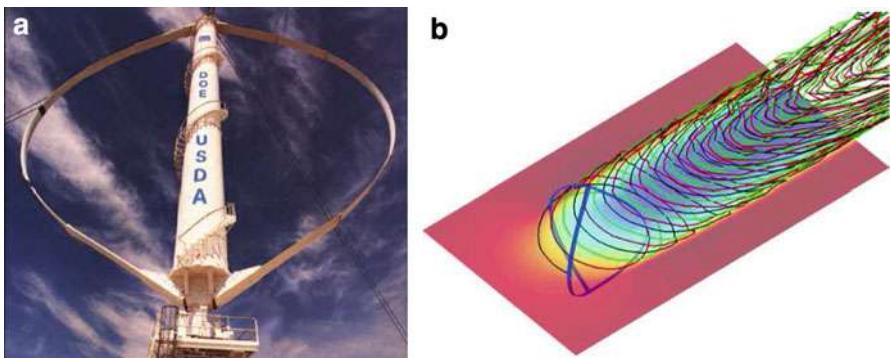


Fig. 28.17 Vertical-axis wind turbine test bed and two-dimensional CACTUS simulation. (a) VAWT test bed. (b) CACTUS simulation

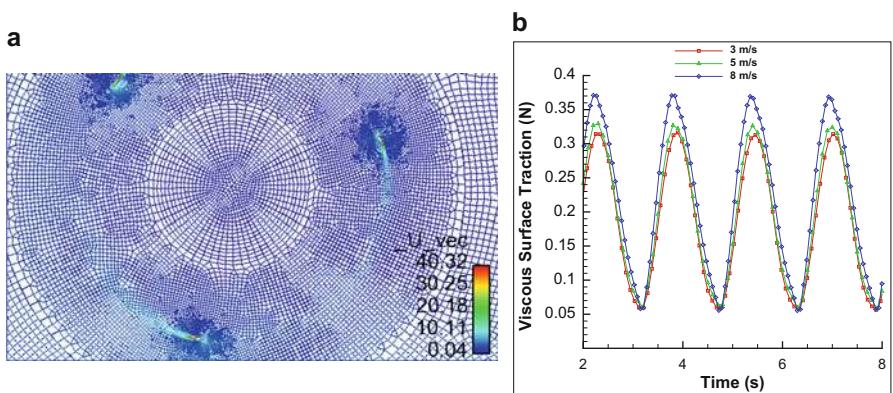


Fig. 28.18 VAWT geometry and leading edge viscous surface traction for three crosswind velocities. (a) Mesh geometry. (b) Integrated viscous surface traction

at Sandia that uses a lifting line/free-vortex formulation to generate predictions of rotor performance and unsteady blade loads (Fig. 28.17b).

Our high-fidelity model is Conchas, which is the module for low-Mach aerodynamics within the SIERRA thermal-fluids code suite. High-fidelity simulations for wind energy applications inherently involve the requirement to solve the turbulent form of the low-Mach Navier-Stokes equation set. The underlying mesh should be adequate to resolve the boundary layers on the blades within the context of a rotating blade scenario. The core methodology involves the use of sliding mesh boundaries between the inner VAWT mesh and the outer free stream mesh. The sliding mesh algorithm combines the control-volume finite-element method at interior domains with a discontinuous Galerkin (DG) implementation at the nonconformal mesh interface [13]. The low-Mach numerical scheme uses equal-order interpolation and a monolithic flow solver using explicit pressure stabilization. Figure 28.18

shows a sample simulation for a VAWT geometry of interest where the cross wind magnitude was varied in a fixed tip speed configuration of 40 m/s. Three different crosswinds were used, 3, 5, and 8 m/s, from which the integrated surface traction was computed (Fig. 28.18b). The mesh outlining the three blade configuration is shown in Fig. 28.18a.

This two-level hierarchy has extreme separation in cost. CACTUS executions for two-dimensional VAWT simulations are fast, typically requiring a few minutes of execution time on a single core of a workstation. Conchas simulations, on the other hand, require approximately 72 h on 48 cores for simulation with 2M finite elements, such that these simulations are strongly resource constrained. For this case, ρ_{work} is approximately 10^5 .

4.5.2 Quantities of Interest

In this final multifidelity UQ example, we focus on the prediction of various statistics for VAWT loads due to an uncertain gust. This has proven to be a challenging problem for PCE approximation using global basis polynomials due to the presence of nonsmooth QoI variations. Figure 28.19 shows an example of this behavior with CACTUS simulations where we overlay a set of centered one-dimensional parameter studies to provide a partial view of a three-dimensional parameter space. It is evident that the variations of these two response QoI are strongly multimodal with multiple slope discontinuities, due to the maximum response changing in space and/or time. In order to partially mitigate this nonsmoothness, we migrated to integrated metrics in subsequent studies.

To facilitate two-dimensional LES simulation of incompressible disturbances with Conchas, the uncertain gust for the multifidelity problem is modeled using

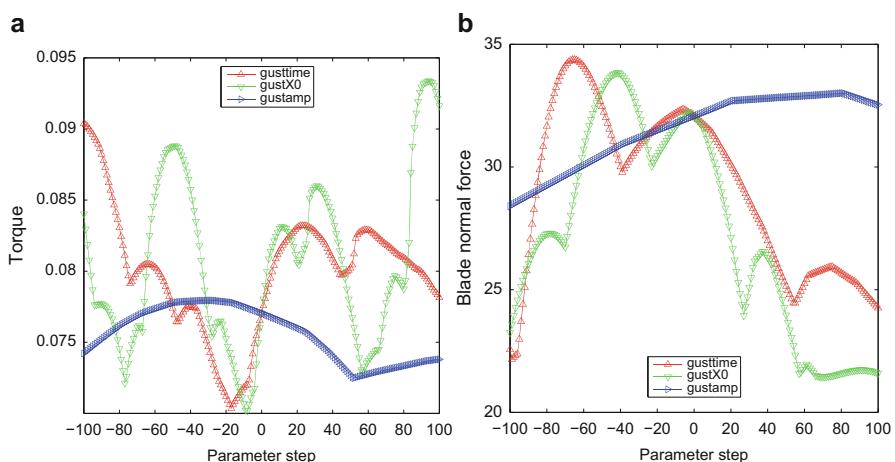


Fig. 28.19 Centered 1D parameter studies for initial formulation using the low-fidelity model. Maximum torque and maximum blade normal force are computed as functions of gust phasing, location, and amplitude. **(a)** Maximum torque. **(b)** Maximum blade normal force

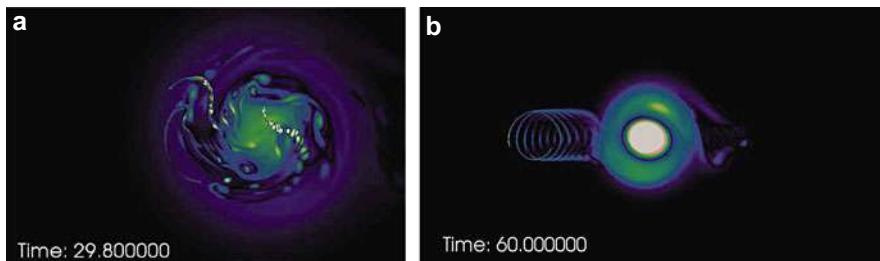


Fig. 28.20 Conchas simulation of a synthetic gust (inviscid Taylor Vortex) passing through a VAWT. (a) Closeup of vortex/rotor interaction. (b) Vortex passing downstream

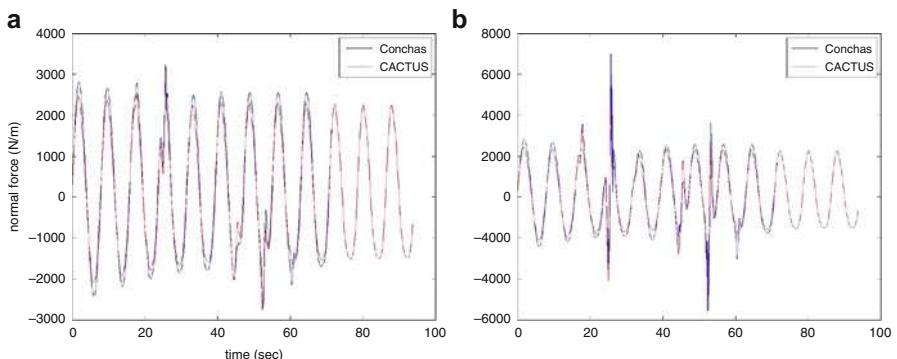


Fig. 28.21 Comparison of CACTUS and Conchas time histories for small and large gust amplitudes. (a) Small gust (amplitude = 5). (b) Large gust (amplitude = 20)

an inviscid Taylor vortex, as shown in Fig. 28.20. The random variables have been slightly modified from those in Fig. 28.19 to describe the vortex radius measured in rotor radii, the lateral position of the vortex in rotor radii, and the amplitude of the vortex. These three random variables are modeled using bounded normal ($\mu = 0.25, \sigma = 0.05, l = 0.0, u = 1.5$), uniform ($l = -1.5, u = 1.5$), and Gumbel ($\alpha = 7.106, \beta = 1.532$) probability distributions, respectively. Figure 28.21 overlays CACTUS and Conchas results for the time histories of blade normal force for two different values of vortex amplitude, where it is evident that there is qualitative agreement between the time histories. The details of flow separation for the large gust have important differences, as expected for the different physics models. Figure 28.22 displays a centered parameter study for the integrated impulse for blade normal force plotted against variations in the vortex radius, lateral position, and amplitude parameters. While the obvious discontinuities have been tamed by the migration to integrated metrics, the response quantity remains a complex function of its parameters.

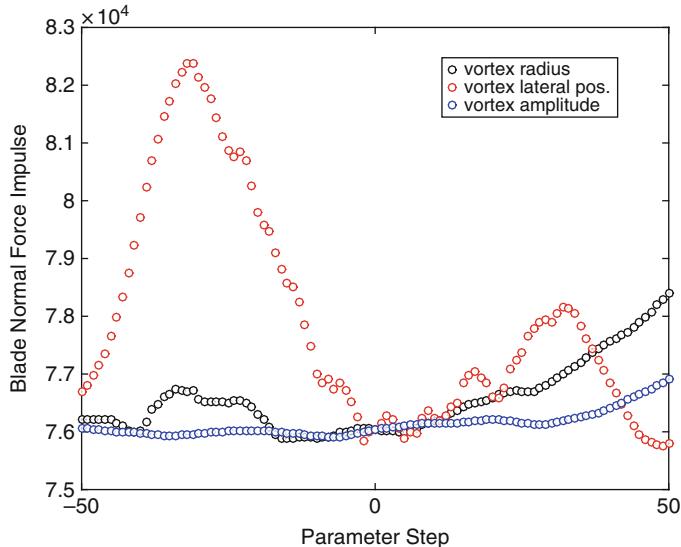


Fig. 28.22 Centered 1D parameter studies for refined formulation using the low-fidelity model. Integrated impulse for blade normal force is computed as a function of Taylor vortex radius, lateral position, and amplitude

4.5.3 Multifidelity Sparse Grid Results

For our numerical experiments, we evaluate the low-fidelity model with an isotropic sparse grid fixed at level 5 (1099 CACTUS simulations executing on a single core), and we evaluate the high-fidelity model at resolution up to level 2 (up to 44 Conchas simulations executing on 48 cores for 72 h each) for forming a low-order model of the discrepancy. Figure 28.23 shows the PCE coefficient magnitudes for the multifidelity spectrum (black) compared to the low-fidelity (green), discrepancy (blue), and high-fidelity (red) spectra. Compared to Fig. 28.16, the spectral decay rate is much slower, and important effects are occurring well beyond the resolution level of the discrepancy sparse grid. Therefore, the multifidelity expansion has to rely on the low-fidelity model for high-order terms which carry significant coefficient magnitudes. This implies that the study would strongly benefit from using less offset in the predefined sparse grid levels (i.e., reducing r in Eq. 28.35 by increasing the discrepancy sparse grid level), requiring many additional high-fidelity runs. Unfortunately, due to the expense of these large-scale LES simulations, this was impractical.

Figure 28.24 shows the evolution of the statistical results as the number of high-fidelity simulations is increased. Since there is no reference solution for this case, errors are not plotted and convergence can only be weakly inferred. While inconclusive at this level of resolution, it appears that the mean and standard deviation statistics are converging from below and that the multifidelity results are closer to their asymptotes. Moreover, the rate of change in the mean statistic is much

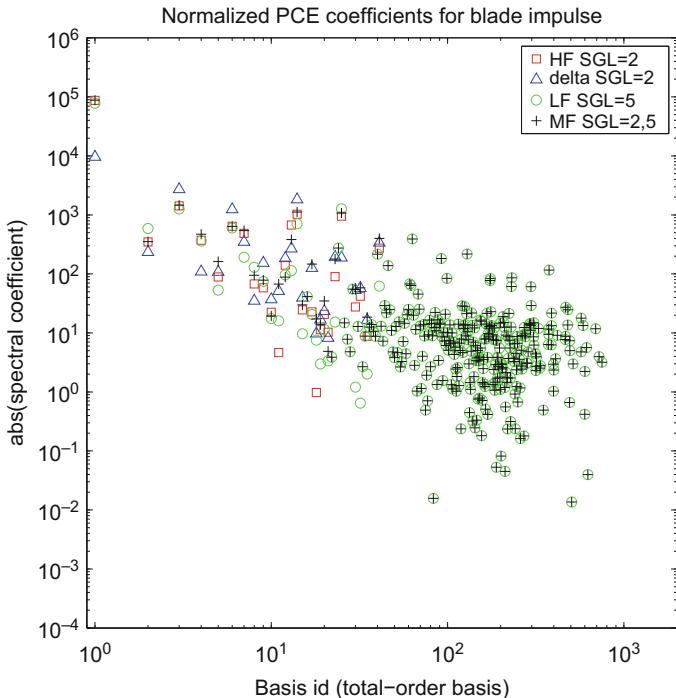


Fig. 28.23 Multifidelity coefficient spectrum: multifidelity coefficients (black) are composed from low fidelity (green) and discrepancy (blue) to extend spectrum from a single-fidelity approach (red)

lower for the multifidelity approaches, indicating more rapid relative convergence. Convergence in PDF and CCDF is less clear, with the most relevant observation being that, while the less resolved single-fidelity ($q = 1$ in red) and corresponding multifidelity results ($q_\delta, q_{\text{low}} = 1, 5$ in blue) are quite different, the more resolved single-fidelity ($q = 2$ in green) and corresponding multifidelity ($q_\delta, q_{\text{low}} = 2, 5$ in black) results have begun to coalesce. Thus, the augmentation with low-fidelity information appears to be providing utility, and the process overall appears to support the trends observed with previous examples, although it is clear that additional resolution is needed to achieve more conclusive statistical convergence.

5 Conclusions

We have presented a general framework for constructing stochastic expansions (non-intrusive polynomial chaos and stochastic collocation) in a multifidelity setting using a hierarchical approximation approach in which we resolve expansions for the low fidelity model and one or more levels of model discrepancy. Compared

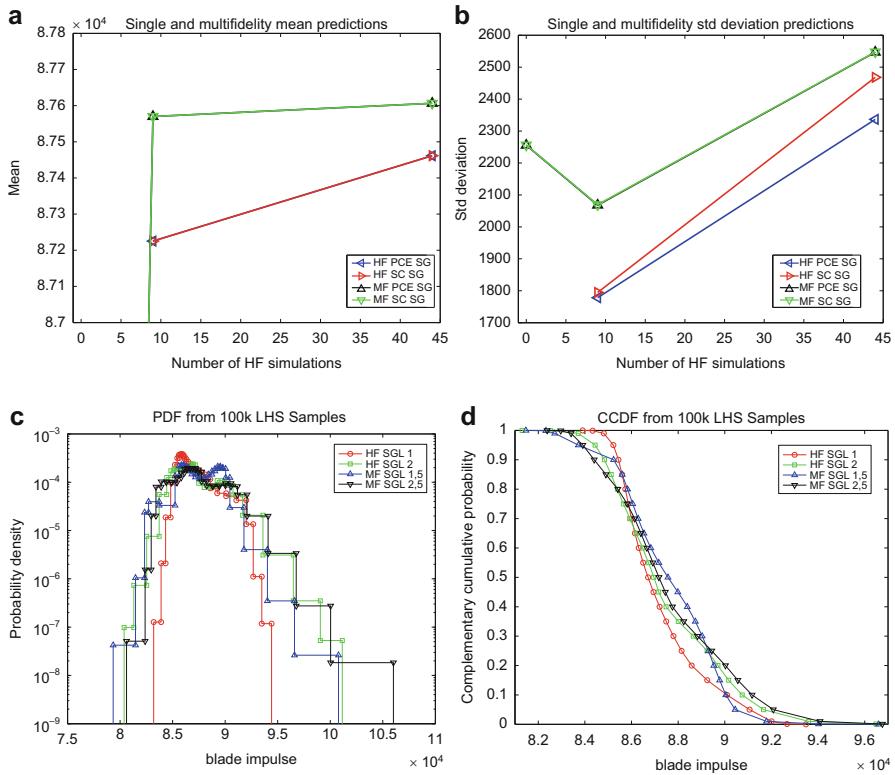


Fig. 28.24 Refinement of blade impulse statistics for multifidelity PCE/SC compared to single-fidelity PCE/SC. (a) Refinement of mean. (b) Refinement of standard deviation. (c) Refinement of PDF. (d) Refinement of CCDF

to the approach of directly estimating the statistics of the system response from a limited number of expensive high-fidelity model evaluations, greater accuracy can be obtained by incorporating information about the system response from a less expensive but still informative low-fidelity model.

Additive, multiplicative, and combined discrepancy formulations have been described within the context of polynomial chaos and stochastic collocation based on sparse grids and within the context of sparse polynomial chaos based on compressed sensing. Multifidelity sparse grid approaches can include simple pre-defined offsets in resolution level that enforce computational savings but do not explicitly manage accuracy, or adaptive approaches that seek the most cost-effective incremental refinements for accurately resolving statistical quantities of interest. Multifidelity compressed sensing approaches employ fixed sample sets but adapt the candidate basis and noise tolerance through the use of cross-validation to achieve an accurate recovery without overfitting.

In the area of adaptive multifidelity algorithms, we present an approach that extends the generalized sparse grid algorithm to consider candidate index sets from multiple sparse grids. Using normalization by the relative cost of the different model fidelities, this adaptive procedure can select the refinements that provide the greatest benefit per unit cost in resolving the high-fidelity statistics. This provides the capability to preferentially refine in dimensions or regions where the discrepancy is more complex, thereby extending the utility of multifidelity UQ to cases where the low-fidelity model is not uniformly predictive.

For the multifidelity UQ approach to be effective, we seek a low-fidelity model that is at least qualitatively predictive in terms of capturing important high-fidelity trends. Examples with good low-fidelity models have demonstrated significant reductions (e.g., 80% in the elliptic PDE example) in the computational expense required to achieve a particular accuracy, and the savings tend to grow as the relative resolution of the low-fidelity model (i.e., r for predefined sparse grid level offsets and ρ_{points} for compressed sensing) is increased. Even without exploitation of special structure (e.g., a priori models of estimator variance and discretization bias in traditional multilevel Monte Carlo), the close relationship between models with differing discretization levels appears to be fertile ground for effective use within multifidelity UQ approaches. On the other hand, low-fidelity models based on reduced-order modeling via projection of low-order singular modes may be a poor choice for multifidelity UQ approaches based on stochastic discrepancy models. The truncation process for defining the basis used in the projection may tend to resolve dominant low-order effects and leave behind a sparse high-order discrepancy function. In the horn problem, the adaptive multifidelity sparse grid approach showed negligible improvement relative to its single-fidelity peer, while the isotropic sparse grid and compressed sensing multifidelity approaches showed only modest gains relative to their benchmarks. Compressed sensing approaches were the best of the three options and are a logical choice for targeting the efficient resolution of sparse high-order discrepancy without requiring one to first resolve all supporting lower-order terms. And in cases where the low-fidelity model introduces additional discontinuities or generates spurious complexity in the model discrepancy that exceeds the original high-fidelity complexity (e.g., short column $R_{\text{low}2}$), it is evident that the multifidelity approaches can converge more slowly than their single-fidelity counterparts, unless this situation can be detected and mitigated by discarding non-informative models from the hierarchy. Finally, the multifidelity UQ approach was demonstrated for an industrial strength application in the statistical assessment of vertical-axis wind turbines subject to uncertain gust loading. While clear convergence evidence is much more challenging to obtain at this scale, affordable resolution levels nevertheless support the basic findings for previous examples in terms of observing accelerated relative convergence in the high-fidelity statistics.

In general, we expect that multifidelity UQ approaches based on spectral stochastic representations of model discrepancy can converge more rapidly than single-fidelity UQ in cases where the variance of the discrepancy is reduced relative to the variance of the high-fidelity model (resulting in reductions in initial stochastic error), where the spectrum of the expansion coefficients of the model discrepancy

decays more rapidly than that of the high-fidelity model (resulting in accelerated convergence rates), and/or where the discrepancy is sparse relative to the high-fidelity model (requiring the recovery of fewer significant terms).

References

1. Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Stephens, J.A., Vigil, D.M., Wildey, T.M.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 6.2 theory manual. Tech. Rep. SAND2014-4253, Sandia National Laboratories, Albuquerque (Updated May 2015). Available online from <http://dakota.sandia.gov/documentation.html>
2. Agarwal, N., Aluru, N.: A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties. *J. Comput. Phys.* **228**, 7662–7688 (2009)
3. Alexandrov, N.M., Lewis, R.M., Gumbert, C.R., Green, L.L., Newman, P.A.: Approximation and model management in aerodynamic optimization with variable fidelity models. *AIAA J. Aircr.* **38**(6), 1093–1101 (2001)
4. Askey, R., Wilson, J.: Some Basic Hypergeometric Orthogonal Polynomials that Generalize Jacobi Polynomials. No. 319 in Memoirs of the American Mathematical Society. AMS, Providence (1985)
5. Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **45**(3), 1005–1034 (2007)
6. Barth, A., Schwab, C., Zollinger, N.: Multi-level monte carlo finite element method for elliptic PDEs with stochastic coefficients. *Numer. Math.* **119**, 123–161 (2011)
7. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.* **12**(4), 273–288 (2000)
8. Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M.: Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA J.* **46**(10), 2459–2468 (2008)
9. Bungartz, H.J., Griebel, M.: Sparse grids. *Acta Numer.* **13**, 147–269 (2004)
10. Cheung, S.H., Oliver, T.A., Prudencio, E.E., Prudhomme, S., Moser, R.D.: Bayesian uncertainty analysis with applications to turbulence modeling. *Reliab. Eng. Syst. Saf.* **96**, 1137–1149 (2011)
11. Constantine, P.G., Eldred, M.S., Phipps, E.T.: Sparse pseudospectral approximation method. *Comput. Methods Appl. Mech. Eng. Volumes* 229–232, pp. 1–12 (2004)
12. Der Kiureghian, A., Liu, P.L.: Structural reliability under incomplete information. *J. Eng. Mech. ASCE* **112**(EM-1), 85–104 (1986)
13. Domino, S.P.: Towards verification of sliding mesh algorithms for complex applications using MMS. In: Proceedings of 2010 Center for Turbulence Research Summer Program, Stanford University (2010)
14. Eftang, J.L., Huynh, D.B.P., Knezevic, D.J., Patera, A.T.: A two-step certified reduced basis method. *J. Sci. Comput.* **51**(1), pp 28–58 (2012)
15. Eldred, M., Wildey, T.: Propagation of model form uncertainty for thermal hydraulics using rans turbulence models in drekar. Tech. Rep. SAND2012-5845, Sandia National Laboratories, Albuquerque (2012)
16. Eldred, M.S., Giunta, A.A., Collis, S.S.: Second-order corrections for surrogate-based optimization with model hierarchies. In: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, AIAA 2004-4457 (2004)
17. Eldred, M.S., Phipps, E.T., Dalbey, K.R.: Adjoint enhancement within global stochastic methods. In: Proceedings of the SIAM Conference on Uncertainty Quantification, Raleigh (2012)

18. Gano, S.E., Renaud, J.E., Sanders, B.: Hybrid variable fidelity optimization by using a Kriging-based scaling function. *AIAA J.* **43**(11), 2422–2430 (2005)
19. Gautschi, W.: *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, New York (2004)
20. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms* **18**(3), 209–232 (1998)
21. Gerstner, T., Griebel, M.: Dimension-adaptive tensor-product quadrature. *Computing* **71**(1), 65–87 (2003)
22. Ghanem, R.G., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
23. Giles, M.B.: Multilevel Monte Carlo path simulation. *Oper. Res.* **56**(3), 607–617 (2008)
24. Goh, J., Bingham, D., Holloway, J.P., Grosskopf, M.J., Kuranz, C.C., Rutter, E.: Prediction and computer model calibration using outputs from multi-fidelity simulators. *Technometrics* in review (2012)
25. Golub, G.H., Welsch, J.H.: Calculation of gauss quadrature rules. *Math. Comput.* **23**(106), 221–230 (1969)
26. Huang, D., Allen, T.T., Notz, W.I., Miller, R.A.: Sequential Kriging optimization using multiple-fidelity evaluations. *Struct. Multidisciplinary Optim.* **32**(5), 369–382 (2006)
27. Jakeman, J., Eldred, M.S., Sargsyan, K.: Enhancing ℓ_1 -minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.* **289**, 18–34 (2015)
28. Jakeman, J.D., Roberts, S.G.: Local and dimension adaptive stochastic collocation for uncertainty quantification. In: Proceedings of the Workshop on Sparse Grids and Applications, Bonn (2011)
29. Kennedy, M.C., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **87**(1), 1–13 (2000)
30. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63**, 425–464 (2001)
31. Klimke, A., Wohlmuth, B.: Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in matlab. *ACM Trans. Math. Softw.* **31**(4), 561–579 (2005)
32. Kuschel, N., Rackwitz, R.: Two basic problems in reliability-based structural optimization. *Math. Method Oper. Res.* **46**, 309–333 (1997)
33. March, A., Willcox, K.: Convergent multifidelity optimization using Bayesian model calibration. In: 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, AIAA 2010-9198 (2010)
34. Murray, J., Barone, M.: The development of CACTUS, a wind and marine turbine performance simulation code. In: 49th AIAA Aerospace Sciences Meeting, Orlando, AIAA 2011-147 (2011)
35. Narayan, A., Gittelson, C., Xiu, D.: A stochastic collocation algorithm with multifidelity models. *SIAM J. Sci. Comput.* **36**(2), A495–A521 (2014)
36. Ng, L.W.T., Eldred, M.S.: Multifidelity uncertainty quantification using nonintrusive polynomial chaos and stochastic collocation. In: Proceedings of the 53rd SDM Conference, Honolulu, Hawaii, AIAA-2012-1852 (2012)
37. Picard, R.R., Williams, B.J., Swiler, L.P., Urbina, A., Warr, R.L.: Multiple model inference with application to uncertainty quantification for complex codes. Tech. Rep. LA-UR-10-06382, Los Alamos National Laboratory, Los Alamos (2010)
38. Rajnarayan, D., Haas, A., Kroo, I.: A multifidelity gradient-free optimization method and application to aerodynamic design. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, AIAA 2008-6020 (2008)
39. Rosenblatt, M.: Remarks on a multivariate transformation. *Ann. Math. Stat.* **23**(3), 470–472 (1952)
40. Rozza, G., Huynh, D.B.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **15**(3), 229–275 (2008)

-
41. Witteveen, J.A.S., Bijl, H.: Modeling arbitrary uncertainties using gram-schmidt polynomial chaos. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, AIAA 2006-896 (2006)
 42. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
 43. Xiu, D., Karniadakis, G.E.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
 44. Zhu, X., Narayan, A., Xiu, D.: Computational aspects of stochastic collocation with multi-fidelity models. *SIAM/ASA J. Uncertain. Quantif.* **2**, 444–463 (2014)

Daniele Venturi, Heyrim Cho, and George Em Karniadakis

Abstract

Determining the statistical properties of nonlinear random systems is a problem of major interest in many areas of physics and engineering. Even with recent theoretical and computational advancements, no broadly applicable technique has yet been developed for dealing with the challenging problems of high dimensionality, low regularity and random frequencies often exhibited by the system. The Mori-Zwanzig and the effective propagator approaches discussed in this chapter have the potential of overcoming some of these limitations, in particular the curse of dimensionality and the lack of regularity. The key idea stems from techniques of irreversible statistical mechanics, and it relies on developing exact evolution equations and corresponding numerical methods for quantities of interest, e.g., functionals of the solution to stochastic ordinary and partial differential equations. Such quantities of interest could be low-dimensional objects in infinite-dimensional phase spaces, e.g., the lift of an airfoil in a turbulent flow, the local displacement of a structure subject to random loads (e.g., ocean waves loading on an offshore platform), or the macroscopic properties of materials with random microstructure (e.g., modeled atomistically in terms of particles). We develop the goal-oriented framework in two different, although related, mathematical settings: the first one is based on the Mori-Zwanzig projection operator method, and it yields exact reduced-order equations for the quantity of interest. The second approach relies on effective propagators,

D. Venturi (✉)

Department of Applied Mathematics and Statistics, University of California Santa Cruz, Santa Cruz, CA, USA

e-mail: venturi@ucsc.edu

H. Cho

Department of Mathematics, University of Maryland, College Park, MD, USA

G.E. Karniadakis (✉)

Division of Applied Mathematics, Brown University, Providence, RI, USA

e-mail: gk@dam.brown.edu, george_karniadakis@brown.edu

i.e., integrals of exponential operators with respect to suitable distributions. Both methods can be applied to nonlinear systems of stochastic ordinary and partial differential equations subject to random forcing terms, random boundary conditions, or random initial conditions.

Keywords

High-dimensional stochastic dynamical systems • Probability density function equations • Projection operator methods • Dimension reduction

Contents

1	Introduction	1038
1.1	Overcoming High Dimensions	1041
1.2	Overcoming Low Regularity	1041
2	Formulation	1042
2.1	Some Properties of the Solution to the Joint PDF Equation	1043
3	Dimension Reduction: BBGKY Hierarchies	1044
4	The Mori-Zwanzig Projection Operator Framework	1045
4.1	Coarse-Grained Dynamics in the Phase Space	1046
4.2	Projection Operators	1047
4.3	Time-Convolutionless Form of the Mori-Zwanzig Equation	1048
4.4	Multilevel Coarse-Graining in Probability and Phase Spaces	1049
5	The Closure Problem	1050
5.1	Beyond Perturbation	1051
5.2	Effective Propagators	1052
5.3	Algorithms and Solvers	1055
6	Applications	1057
6.1	Stochastic Resonance Driven by Colored Noise	1057
6.2	Mori-Zwanzig Equation	1058
6.3	Fractional Brownian Motion, Levy, and Other Noises	1060
6.4	Stochastic Advection-Reaction	1061
6.5	Stochastic Burgers Equation	1063
6.6	Coarse-Grained Models of Particle Systems	1065
7	Conclusions	1066
8	Cross-References	1066
	References	1067

1 Introduction

Experiments on high-dimensional random systems provide observations of macroscopic phase variables such as the mass density in Newtonian fluids, the stress-strain relation in heterogeneous random materials (e.g., carbon fiber), or the velocity distribution in granular flows. These quantities can be related to a thorough microscopic description of the system by taking averages of real-valued (measurable) functions defined on a very high-dimensional phase space. To understand the dynamics of such phase-space functions, one often wishes to obtain closed equations of motion by eliminating the rest of the degrees of freedom. One of the most typical examples

for such contraction of state variables is the derivation of the Boltzmann equation from the Newton's law or from the Liouville equation [16, 118, 142]. Another example of a different type is the Brownian motion of a particle in a liquid, where the master equation governing the position and momentum of the particle is derived from first principles (Hamilton equations of motion of the full system), by eliminating the degrees of freedom of the liquid [17, 67]. In stochastic systems far from equilibrium, one often has to deal with the problem of eliminating macroscopic phase variables, i.e., phase variables with the same order of magnitude and dynamical properties as the ones of interest. For example, to define the turbulent viscosity in the inertial range of fully developed turbulence, one has to eliminate short wavelength components of the fluid velocity which are far from equilibrium. This problem arises more often than one would expect, and it is more challenging than the problem of contracting microscopic phase variables. For example, it arises when deriving the master equation for the series expansion of the solution to a nonlinear stochastic partial differential equation (SPDE), given any discretized form.

In this chapter we illustrate how to perform the contraction of state variables in nonequilibrium stochastic dynamical systems by using the Mori-Zwanzig projection operator method and the effective propagator approach. In particular, we will show how to develop computable evolution equations for quantities of interest in high-dimensional stochastic systems and how to determine their statistical properties. This problem received considerable attention in recent years. Well-known approaches to compute such properties are generalized polynomial chaos (gPC) [50, 154, 155], multilevel generalized polynomial chaos (ME-gPC) [138, 146], multilevel and sparse adaptive probabilistic collocation (ME-PCM) [32, 43, 44, 84], high-dimensional model representations [76, 112], stochastic biorthogonal expansions [131, 132, 137], and generalized spectral decompositions [100, 101]. These techniques can provide considerable speedup in computational time when compared to classical approaches such as Monte Carlo (MC) or quasi-Monte Carlo methods. However, there are still several important computational limitations that have not yet been overcome. They are related to:

1. *High Dimensionality*: Many problems of interest to physics and engineering can be modeled mathematically in terms of systems of nonlinear ODEs or nonlinear PDEs subject to random initial conditions, random parameters, random forcing terms, or random boundary conditions. The large number of phase variables involved in these problems and the high dimensionality of the random input vectors pose major computational challenges in representing the stochastic solution, e.g., in terms of polynomial chaos or probabilistic collocation. In fact, the number of terms in polynomial chaos series expansions, or the number of collocation points in probabilistic collocation methods, grows exponentially fast with the number of dimensions (in tensor product discretizations).
2. *Low Stochastic Regularity*: The computational cost of resolving solutions with *low stochastic regularity* is also an issue. Parametric discontinuities can create Gibbs-type phenomena which can completely destroy the convergence numerical methods – just like in spectral methods [57, 102]. Parametric discontinuities are

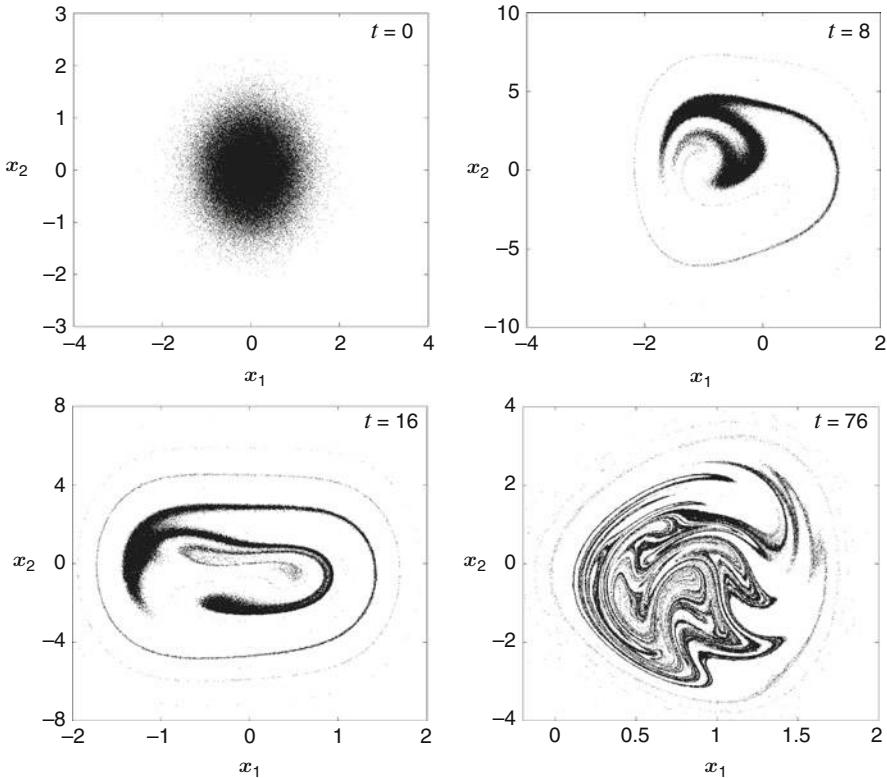


Fig. 29.1 Duffing equation. Poincaré sections of the phase space at different times obtained by evolving a zero-mean jointly Gaussian distribution with covariance $C_{11} = C_{22} = 1/4$, $C_{12} = 0$. Note that simple statistical properties such as the mean and variance are not sufficient to describe the stochastic dynamics of the system (29.5) (Adapted from [136])

unavoidable in nonlinear systems and they are often associated with interesting physics, e.g., around bifurcation points [138, 139]. By using adaptive methods, e.g., ME-gPC or ME-PCM, one can effectively resolve such discontinuities and restore convergence. This is where “ h -refinement” in parameter space is particularly important [43, 144].

3. *Multiple scales:* Stochastic systems can involve multiple scales in space, time, and phase space (see Fig. 29.1) which could be difficult to resolve by conventional numerical methods.
4. *Long-term integration:* The flow map defined by systems of differential equations can yield large deformations, stretching and folding of the phase space. As a consequence, methods that represent the parametric dependence of the solution on random input variables, e.g., in terms of polynomials chaos of fixed order or in terms of a fixed number of collocation points, will lose accuracy as time increases. This phenomenon can be mitigated, although not completely

overcome, by using multielement methods [43, 145], time-evolving bases [117], or a composition of short-term flow maps [80].

The Mori-Zwanzig and the effective propagator approaches have the potential of overcoming some of these limitations, in particular the curse of dimensionality and the lack of regularity.

1.1 Overcoming High Dimensions

The Mori-Zwanzig and the effective propagator approaches allow for a *systematic elimination* of the “irrelevant” degrees of freedom of the system, and they yield formally exact equations for quantities of interest, e.g., functionals of the solution to high-dimensional systems of stochastic ordinary differential equations (SODEs) and stochastic partial differential equations (SPDEs). This allows us to *avoid integrating the full (high-dimensional) stochastic dynamical system* and solve directly for the quantities of interest. In principle, this can break the curse of dimensionality in numerical simulations of SODEs and SPDEs at the price of solving complex integrodifferential PDEs – the Mori-Zwanzig equations. The computability of such PDEs relies on approximations. Over the years many methods have been proposed for this scope, for example, small correlation expansions [30, 46, 121], cumulant resummation methods [13, 17, 45, 78, 136], functional derivative techniques [51–53, 140], path integral methods [86, 108, 130, 153], decoupling approximations [54], and local linearization methods [35]. However, these techniques are not, in general, effective in eliminating degrees of freedom with the *same order of magnitude and dynamical properties* as the quantities of interest. Several attempts have been made to overcome these limitations and establish a computable framework for Mori-Zwanzig equations that goes beyond closures based on perturbation analysis. We will discuss some of these methods later in this chapter.

1.2 Overcoming Low Regularity

The PDF of low-dimensional quantities of interest depending on many phase variables is usually a *regular function*. This is due to a *homogenization effect* induced by multidimensional integration. In other words, the PDF of low-dimensional quantities of interest is often not just low-dimensional but also *smooth*, i.e., amenable to computation. As an example, consider the joint PDF of the Fourier coefficients of a turbulent flow. It is known that such joint PDF lies on an attractor with a possibly fractal structure [41, 42, 49]. However, the linear combination of the Fourier modes, i.e., the Fourier representation of the velocity field at a specific space-time location, turns out to be approximately Gaussian. This behavior is exhibited by other chaotic dynamical systems such as the Lorentz-96 system [79] evolving from a random initial state. In this case, it can be shown that the joint PDF of the phase variables approaches asymptotically in time a fractal attractor whose

dimension depends on the amplitude of the forcing term (see, e.g., [69]). However, the marginal distributions of such a complex joint PDF are approximately Gaussian (see Fig. 29.4).

2 Formulation

Let us consider the nonlinear dynamical system

$$\begin{cases} \frac{dx(t; \omega)}{dt} = f(x(t; \omega), \xi(\omega), t), \\ x(0; \omega) = x_0(\omega) \end{cases}, \quad (29.1)$$

where $x(t; \omega) \in \mathbb{R}^n$ is a multidimensional stochastic process, $f : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^n$ is a deterministic nonlinear map assumed to be Lipschitz continuous in x , $\xi(\omega) \in \mathbb{R}^m$ is a random vector modeling input uncertainty, and $x_0(\omega) \in \mathbb{R}^n$ is a random initial state. The system (29.1) can be very large as it can arise, e.g., from a discretization of a nonlinear SPDE. We assume that the solution to (29.1) exists and is unique for each realization of $\xi(\omega)$ and $x_0(\omega)$. This allows us to consider $x(t; \omega)$ as a deterministic function of $\xi(\omega)$ and $x_0(\omega)$, i.e., we can define the parametrized *flow map* $\hat{x}(t; \xi(\omega), x_0(\omega))$. The joint PDF of $x(t; \omega)$ and $\xi(\omega)$ can be represented as

$$p(a, b, t) = \langle \delta(a - \hat{x}(t; \xi; x_0)) \delta(b - \xi) \rangle, \quad a \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \quad (29.2)$$

where $\langle \cdot \rangle$ denotes an integral with respect to the joint probability distribution of $\xi(\omega)$ and $x_0(\omega)$, while δ are multidimensional Dirac delta functions [68, 71]. Also, the vectors a and b represent the phase-space coordinates corresponding to $x_i(t; \omega)$ and $\xi(\omega)$, respectively. By differentiating (29.2) with respect to time and using well-known identities involving the Dirac delta function, it is straightforward to obtain the following exact hyperbolic conservation law:

$$\frac{\partial p(a, b, t)}{\partial t} = L(a, b, t) p(a, b, t), \quad L(a, b, t) = - \sum_{i=1}^n \left(\frac{\partial f_i(a, b, t)}{\partial a_i} + f_i \frac{\partial}{\partial a_i} \right). \quad (29.3)$$

In the sequel we will often set $p(t) \equiv p(a, b, t)$ and $L(t) \equiv L(a, b, t)$ for notational convenience. Equation (29.3) is equivalent to the Liouville equation of classical statistical mechanics (for non-Hamiltonian systems), with the remarkable difference that the phase variables we consider here can be rather general coordinates – not simply positions and momenta of particles. For instance, they could be the Galerkin or the collocation coefficients arising from a spatial discretization of a SPDE, e.g., if we represent the solution as

$$u(X, t; \omega) = \sum_{j=1}^n x_j(t; \omega) \phi_j(X), \quad (29.4)$$

where $\phi_j(X)$ are spatial basis functions. Early formulations in this direction were proposed by Edwards [33], Herring [56], and Montgomery [90] in the context of fluid turbulence.

2.1 Some Properties of the Solution to the Joint PDF Equation

Nonlinear systems in the form (29.1) can lead to all sorts of dynamics, including bifurcations, fractal attractors, multiple stable steady states, and transition scenarios. Consequently, the solution to the joint PDF equation (29.3) can be very complex as well, since it relates directly to the geometry of the phase space. For example, it is known that the time-asymptotic joint PDF associated with the Lorentz three-mode problem lies on a fractal attractor with Hausdorff dimension of about 2.06 (see [143]). Chaotic states and existence of strange attractors have been well documented for many other systems, such as the Lorenz-84 (see [14]) and the Lorenz-96 [69] models. Even in the much simpler case of the Duffing equation

$$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = -x_1 - 5x_1^3 - \frac{x_2}{50} + 8 \cos\left(\frac{t}{2}\right) \quad (29.5)$$

we can have attractors with fractal structure and chaotic phase similarities [11]. This is clearly illustrated in Fig. 29.1 where we plot the Poincaré sections of the two-dimensional phase space at different times. Such sections are obtained by sampling 10^6 initial states from a zero-mean jointly Gaussian distribution and then evolving them by using (29.5). Since the joint PDF of the phase variables is, in general, a high-dimensional compactly supported distribution with a possibly fractal structure, its numerical approximation is a very challenging task, especially in longtime integration.

However, the statistical description of the system (29.1) in terms of the joint PDF equation (29.3) is often far beyond practical needs. For instance, we may be interested in the PDF of only one component, e.g., $x_1(t; \omega)$, or in the PDF of a phase space function $u = g(x)$ such as (29.4). These PDFs can be obtained either by integrating out several phase variables from the solution to Eq. (29.3), by constructing NARMAX (Nonlinear AutoRegressive Moving Average with exogenous input) models (see [7] §5.7) or by applying the projection operator or the effective propagator approaches discussed in this chapter. This may yield a *low-dimensional* PDF equation whose solution is *more regular* than the one obtained by solving directly Eq. (29.3) and therefore more amenable to computation. The regularization of the reduced-order PDF is due to multidimensional integration (marginalization) of the joint PDF.

3 Dimension Reduction: BBGKY Hierarchies

A family of reduced-order probability density functions can be obtained by integrating the solution to Eq. (29.3) with respect to the phase-space coordinates which are not of interest. This yields, for example,

$$p_i(a_i, t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(a, b, t) da_1 \cdots da_{i-1} da_{i+1} \cdots da_n db, \quad i = 1, \dots, n. \quad (29.6)$$

These reduced-order densities differ from those used in classical BBGKY theory [16], mainly in that they are not, in general, symmetric under interchanges of different phase-space coordinates. For instance, $p_i(a_i, t)$ is not the same function of a_i that $p_j(a_j, t)$ is of a_j , if i and j are different. In the classical BBGKY (Bogoliubov-Born-Green-Kirkwood-Yvon) framework, the phase coordinates of the systems are positions and momenta of identical particles. Therefore, the reduced-order multipoint densities are invariant under interchanges of phase-space coordinates of the same type, e.g., positions or momenta. Most of the added complexity to the classical BBGKY theory stems from this lack of symmetry. A related approach, due to Lundgren [81] and Monin [89], yields a hierarchy of PDF equations involving suitable limits of reduced density functions (see also [48, 59, 134, 135, 152]). The effective computability of both BBGKY-type and Lundgren-Monin hierarchies arising from Eq. (29.3) relies on appropriate closure schemes, e.g., a truncation based on a suitable decoupling approximation of the PDF. In particular, the mean-field approximation

$$p(t, a, b) = p_{\xi}(b) \prod_{i=1}^n p_i(a_i, t), \quad (29.7)$$

where $p_{\xi}(b)$ is the joint PDF of the random vector ξ , yields the system of conservation laws ($i = 1, \dots, n$):

$$\frac{\partial p_i(a_i, t)}{\partial t} = -\frac{\partial}{\partial a_i} \left[p_i(a_i, t) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_i(a, b, t) p_{\xi}(b) \prod_{\substack{j=1 \\ j \neq i}}^n p_j(a_j, t) da_j db \right]. \quad (29.8)$$

These equations are coupled through the integrals appearing within the square bracket. As an example, consider the Lorentz-96 system [69]

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2}) x_{i-1} - x_i + c, \quad i = 1, \dots, 40. \quad (29.9)$$

The first-order truncation of the BBGKY hierarchy is ($i = 1, \dots, 40$)

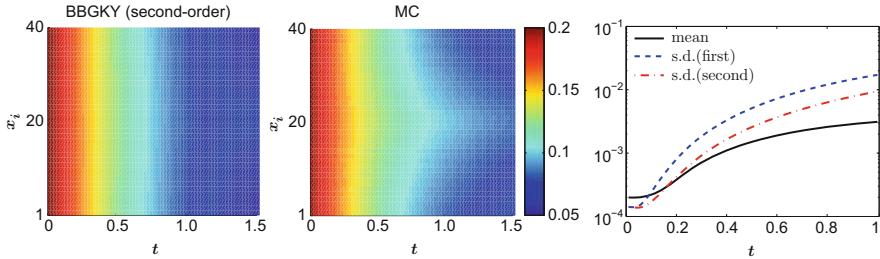


Fig. 29.2 Lorenz-96 system. Standard deviation of the phase variables versus time (left) and absolute errors of first- and second-order truncations of the BBGKY hierarchy relative to MC (Adapted from [22])

$$\frac{\partial p_i(a_i, t)}{\partial t} = -\frac{\partial}{\partial a_i} \left[(\langle x_{i+1} \rangle - \langle x_{i-2} \rangle) \langle x_{i-1} \rangle p_i(a_i, t) - (a_i - c) p_i(a_i, t) \right], \quad (29.10)$$

where $\langle \cdot \rangle$ denotes averaging with respect to the joint PDF of the system, assumed in the form (29.7). Higher-order truncations, i.e., truncations involving multipoint PDFs, can be obtained in a similar way (see [22, 90]). Clearly, higher truncation orders yield better accuracy, but at higher computational cost (see Fig. 29.2).

4 The Mori-Zwanzig Projection Operator Framework

The basic idea of the Mori-Zwanzig formalism is to reduce the dimensionality of the dynamical system (29.1) by splitting the phase variables into two categories: the *relevant* (or resolved) variables and the *irrelevant* (or unresolved) ones. These two sets can be easily classified by means of an orthogonal projection operator P that maps the state vector onto the set of resolved variables. By applying such orthogonal projection to Eq. (29.3), it is straightforward to obtain the following exact equation:

$$\frac{\partial P p(t)}{\partial t} = PL(t) P p(t) + PL(t) G(t, 0) Q p(0) + PL(t) \int_0^t G(t, s) QL(s) P p(s) ds, \quad (29.11)$$

first derived by Nakajima [96], Zwanzig [158, 159], and Mori [91]. Here we have set $p(t) \equiv p(a, b, t)$ and $L(t) \equiv L(a, b, t)$ for notational convenience and denoted by $Q = I - P$ the projection onto the unresolved variables. The operator $G(t, s)$ (forward propagator of the orthogonal dynamics) is formally defined as

$$G(t, s) = \overleftarrow{T} \exp \left[\int_s^t QL(\tau) d\tau \right], \quad (29.12)$$

where \overleftarrow{T} is the chronological time-ordering operator (latest times to the left). For a detailed derivation, see, e.g., [13, 17, 66, 136, 159]. From Eq. (29.11) we see that the

exact dynamics of the PDF of the relevant phase variables (projected PDF $Pp(t)$) depends on three terms: the *Markovian term* $PL(t)Pp(t)$, computable based on the current state $Pp(t)$, the *initial condition (or noise) term* $PL(t)G(t, 0)Qp(0)$, and the *memory term* (time convolution), both depending on the propagator $G(t, 0)$ of the orthogonal dynamics. The critical part of the MZ formulation is to find reliable and accurate approximations of the memory and the initial condition terms.

The nature of the projection operator P will be discussed extensively in subsequent sections. For now, it is sufficient to note that such projection basically extracts from the full joint PDF equation (29.3) only the part that describes (in an exact way) the dynamics of the relevant phase variables. A simple analysis of Eq. (29.11) immediately shows its irreversibility. Roughly speaking, the projected distribution function $Pp(t)$, initially in a certain subspace, leaks out of this subspace so that information is lost, hence the memory (time convolution) and the initial condition terms.

4.1 Coarse-Grained Dynamics in the Phase Space

The Mori-Zwanzig projection operator method just described can be also used to reduce the dimensionality of either deterministic or stochastic *systems of equations in the phase space*, yielding generalized Langevin equations [61, 118] for quantities of interest. One remarkable example of such equations is the one describing the *coarse-grained dynamics* of a particle system. Within this framework the phase variables $x_i(t)$ in (29.1) can represent either the position or the momentum of the particle “ i .” Coarse-graining is achieved by defining a new set of state variables:

$$u(t; \omega) = g(x(t; \omega), t) \quad (\text{quantities of interest}) \quad (29.13)$$

where $g : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^q$ is a phase-space function and q is usually much smaller than n . These variables can represent the position or the momentum of entire *clusters of particles*, e.g., the big green particles shown in Fig. 29.3. The irrelevant phase variables in this case are the components of the full state vector x .

The generalized Langevin equation satisfied by (29.13) can be obtained by using standard methods [29, 61, 94, 118]. For example, if the system is autonomous, i.e., if the right-hand side of Eq. (29.1) reduces to $f(x)$, then we have the formally exact coarse-grained system (see [61, 92, 157]):

$$\frac{du_i(t)}{dt} = e^{tM} PM u_i(0) + \int_0^t e^{(t-s)M} PM R_i(s) ds + R_i(t), \quad i = 1, \dots, q \quad (29.14)$$

where P is an orthogonal projection operator and

$$M = \sum_{j=1}^n f_j(x_0) \frac{\partial}{\partial x_0}, \quad R_i(t) = e^{t(I-P)M}(I - P)Mu_i(t). \quad (29.15)$$

Particle System

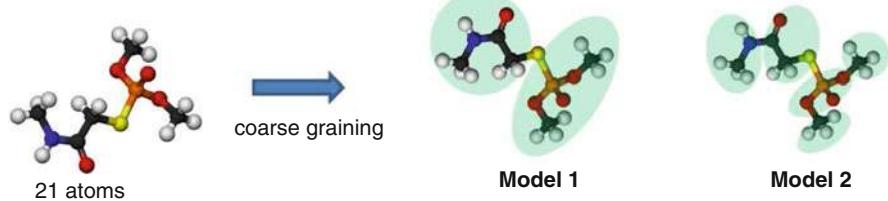


Fig. 29.3 Coarse-graining particle systems using the projection operator method. The microscopic degrees of freedom associated with each atom are condensed into a smaller number of degrees of freedom (those associated with the *big green particles*). Coarse-graining is not unique, and therefore fundamental questions regarding model inadequacy, selection, and validation have to be carefully addressed

The state-space reduced-order equations (29.14) are particularly useful if $u_i(t)$ is a complete set of slowly varying variables relative to the dynamics of the unresolved variables, i.e., the dynamics of $x(t; \omega)$. In this case, the fluctuating forces $R_i(t)$ are rapidly varying in time due to their modified propagator $\exp[t(I - P)M]$, and the memory kernel rapidly decays to zero. Effective approximations are possible in these cases [61, 65, 77, 157]. Based on the phase-space formulation, it is also possible to obtain the Mori-Zwanzig equation for the one-point or the multipoint PDF of the quantities of interest (29.13). To this end, it is sufficient to differentiate the distribution function

$$p_u(a, t) = \langle \delta(a - u(t; \omega)) \rangle, \quad (29.16)$$

with respect to time and substitute (29.14) within the average (see, e.g., [92, 118] for the derivation). If (29.1) represents the semi-discrete form of a SPDE and we are interested in the phase-space function (29.4), then the Mori-Zwanzig formulation yields the *exact PDF equation for the series expansion of the solution to the SPDE*. This overcomes the well-known *closure problem* arising in PDF equations corresponding to SPDEs with diffusion or higher-order terms [105, 110, 141]. On the other hand, if $u_i(t) = x_i(t)$ ($i = 1, \dots, q$) are the first q components of a Galerkin dynamical system, then the Mori-Zwanzig projection operator method allows us to construct a closure approximation in which the *unresolved dynamics* (modes from $q + 1$ to n) are injected in a *formally exact way* into the *resolved dynamics*. This use of projection operators has been investigated, e.g., by Chorin [24, 27], Stinis [119] and Chertok [20]. Early studies in this direction – not involving Mori-Zwanzig – employed *inertial manifolds* [40] and *nonlinear Galerkin projections* [83].

4.2 Projection Operators

The coarse-graining of the microscopic equations of motion can be performed by introducing a projection operator and applying it to the master equation (29.3) (coarse-graining in the PDF space – Eq. (29.11)) or to the dynamical system (29.1)

(coarse-graining in the phase space – Eq. (29.14)). Well-known choices are the Zwanzig projection [159], the Mori projection, projections defined in terms of Boltzmann-Gibbs measures [61, 118, 128], or projections defined by conditional expectations [25, 27, 29, 119]. If the relevant and the irrelevant phase variables (hereafter denoted by a and b , respectively) are statistically independent, i.e., if $p(0) = p_a(0)p_b(0)$, then a convenient projection is

$$Pp(t) = p_b(0) \int p(t)db \quad \Rightarrow \quad p_a(t) = \int Pp(t)db. \quad (29.17)$$

This projection takes the joint PDF $p(t)$ and basically sends it to a separated state. In this case we have that $p(0)$ is in the range of P , i.e., $Pp(0) = p(0)$, and therefore the initial condition term in the MZ-PDF equation drops out since $Qp(0) = (I - P)p(0) = 0$.

4.3 Time-Convolutionless Form of the Mori-Zwanzig Equation

The Mori-Zwanzig PDF (MZ-PDF) equation (29.11) can be transformed into a Markovian (time-convolutionless) form. To this end, we simply consider the formal solution to the orthogonal dynamics equation

$$Qp(t) = G(t, 0)Qp(0) + \int_0^t G(t, s)QL(s)Pp(s)ds, \quad (29.18)$$

and replace $p(s)$ with the solution to Eq. (29.3), propagated backward from time t to time $s < t$, i.e.,

$$p(s) = Z(t, s)p(t), \quad \text{where} \quad Z(t, s) = \overrightarrow{T} \exp \left[- \int_s^t L(\tau)d\tau \right]. \quad (29.19)$$

In the latter definition \overrightarrow{T} is the anti-chronological ordering operator (latest times to the right). Substituting (29.19) into (29.18) yields

$$Qp(t) = [I - \Sigma(t)]^{-1} G(t, 0)Qp(0) + [I - \Sigma(t)]^{-1} \Sigma(t)Pp(t), \quad (29.20)$$

where

$$\Sigma(t) = \int_0^t G(t, s)QL(s)PZ(t, s)ds. \quad (29.21)$$

Equation (29.20) states that the “irrelevant” part of the PDF $Qp(t)$ can, in principle, be determined from the knowledge of the “relevant” part $Pp(t)$ at time t and from the initial condition $Qp(0)$. Thus, the dependence on the history of the relevant part which occurs in the classical Mori-Zwanzig equation has been removed by

the introduction of the backward propagator (29.19). By using the orthogonal dynamics equation (29.20), we obtain the Markovian (time-convolutionless) MZ-PDF equation

$$\frac{\partial Pp(t)}{\partial t} = K(t)Pp(t) + H(t)Qp(0), \quad (29.22)$$

where

$$K(t) = PL(t)[I - \Sigma(t)]^{-1}, \quad H(t) = PL(t)[I - \Sigma(t)]^{-1}G(t, 0). \quad (29.23)$$

Many other equivalent forms of the Mori-Zwanzig equation can be constructed (see the Appendix in [136]), exactly for the same reason as why it is possible to represent an effective propagator of reduced-order dynamics in terms of generalized operator cumulants [55, 73, 74, 97]. So far, everything that has been said is exact, and it led us to the equation of motion (29.22), which is linear and local in time. Unfortunately, such an equation is still of little practical use, because the exact determination of the operators K and H is as complicated as the solution of Eq.(29.3). However, the time-convolutionless form (29.22) is a convenient starting point to construct *systematic approximation schemes*, e.g., by expanding K and H in terms of cumulant operators relative to suitable coupling constants [13, 17, 64, 66, 74, 97, 107, 115].

4.4 Multilevel Coarse-Graining in Probability and Phase Spaces

In [136] we recently proposed a multilevel coarse-graining technique in which the evolution equation for the orthogonal PDF dynamics $Qp(t)$

$$\frac{\partial Qp(t)}{\partial t} = QL(t)[Pp(t) + Qp(t)], \quad (29.24)$$

is decomposed further by introducing a new pair of orthogonal projections P_1 and Q_1 such that $P_1 + Q_1 = I$. This yields the coupled system

$$\frac{\partial P_1 Qp(t)}{\partial t} = P_1 QL(t)[Pp(t) + P_1 Qp(t) + Q_1 Qp(t)], \quad (29.25)$$

$$\frac{\partial Q_1 Qp(t)}{\partial t} = Q_1 QL(t)[Pp(t) + P_1 Qp(t) + Q_1 Qp(t)]. \quad (29.26)$$

Proceeding similarly, we can split the equation for $Q_1 Qp(t)$ by using a new pair of orthogonal projections P_2 and Q_2 satisfying $P_2 + Q_2 = I$. This yields two additional evolution equations for $P_2 Q_1 Qp(t)$ and $Q_2 Q_1 Qp(t)$, respectively. Obviously, one can repeat this process indefinitely to obtain a hierarchy of equations which *generalizes* both the Mori-Zwanzig as well as the BBGKY frameworks. The

advantage of this formulation with respect to the classical approach relies on the fact that the joint PDF $p(t)$ is not simply split into the “relevant” and the “irrelevant” parts by using P and Q . Indeed, the dynamics of the irrelevant part $Qp(t)$ are decomposed further in terms of a new set of projections.

This allows us to coarse-grain relevant features of the orthogonal dynamics further in terms of lower-dimensional quantities. In other words, the *multilevel projection operator method* allows us to *seemingly interface dynamical systems at different scales in a mathematically rigorous way*. This is particularly useful when coarse-graining (in state space) high-dimensional systems in the form (29.1). To this end, we simply have to define a set of quantities of interest $u^{(1)} = g^{(1)}(x, t)$, $u^{(2)} = g^{(2)}(x, t)$, etc. (see (29.13)), e.g., representing clusters of particles of different sizes and corresponding projection operators P_1 , P_2 , etc. This yields a coupled set of equations resembling (29.14) in which relevant features of the microscopic dynamics are interacting at different scales defined by different projection operators.

5 The Closure Problem

Most schemes that attempt to compute the solution of MZ equations or BBGKY-type hierarchies rely on the identification of some small quantity that serves as the basis for a perturbation expansion, e.g., the density for Boltzmann equations [16], the coupling constant or correlation time for Fokker-Planck-type equations [30, 46, 93, 121], or the Kraichnan absolute equilibrium distribution for turbulent inviscid flows [72, 90]. One of the most stubborn impediments for the development of a general theory of reduced-order PDF equations has been the lack of such readily identifiable small parameters. Most of the work that has been done so far refers to the situation in which such small parameters exist, e.g., when the operator L in Eq. (29.3) can be decomposed as

$$L = L_0 + \sigma L_1. \quad (29.27)$$

Here L_0 depends only on the relevant variables of the system, σ is a positive real number (coupling constant in time-dependent quantum perturbation theory), and the norm $\sigma \|L_1\|$ is somehow *small* (see, e.g., [13, 17, 93]). By using the interaction representation of quantum mechanics [9, 149], then it is quite straightforward to obtain from (29.22) and (29.27) an effective approximation (see [136]). One way to do so is to expand the operators (29.23) in a cumulant series, e.g., in terms of Kubo-Van Kampen operator cumulants [55, 64, 74, 97], involving increasing powers of σ (coupling parameter). Any finite-order truncation of such series then represents an *approximation* to the exact MZ-PDF equation. In particular, the approximation obtained by retaining only the first two cumulants is known as *Born approximation* in quantum field theory [17]. We remark that from the point of view of perturbation theory, the convolutionless form (29.22) has distinctive advantages over the usual convolution form (29.11). In particular, in the latter case, a certain amount of

rearrangement is necessary to obtain an expression which is correct up to a certain order in the coupling parameter [126].

5.1 Beyond Perturbation

Several attempts have been made to approximate MZ equations beyond closures based on perturbation analysis. For example, Chorin [24, 25, 27], Stinis [119, 120], and Chertok [20] proposed various models – such as the t -model or the modified t -model – for dimension reduction of autonomous dynamical systems in situations where there is no clear separation of scales between the resolved and the unresolved dynamics.

Another widely used closure approximation is based on the assumption that the distribution of the quantity of interest has a specific form, e.g., approximately Gaussian. This assumption can be justified in some cases on the basis of mixing, high dimensionality, and chaos. For example, the marginal densities of the Lorenz-96 system (29.9) are approximately Gaussian (see Fig. 29.4). In these cases, a Gaussian closure can be used to represent the PDF of the quantity of interest. Alternative methods rely, e.g., on maximum entropy closures [60, 62, 99, 128], functional renormalization [5, 120], and *renormalized perturbation series* ([87], Ch. 5). The key idea is to use methods of many body-theory to generalize traditional perturbation series to the case of strong interactions [85]. These approaches have been used extensively in turbulence theory [49, 87].

A different technique to compute the memory and the initial condition terms appearing in the Mori-Zwanzig equation relies on sampling, e.g., few realizations of the full dynamical system (29.1). In particular, one can leverage implicit sampling

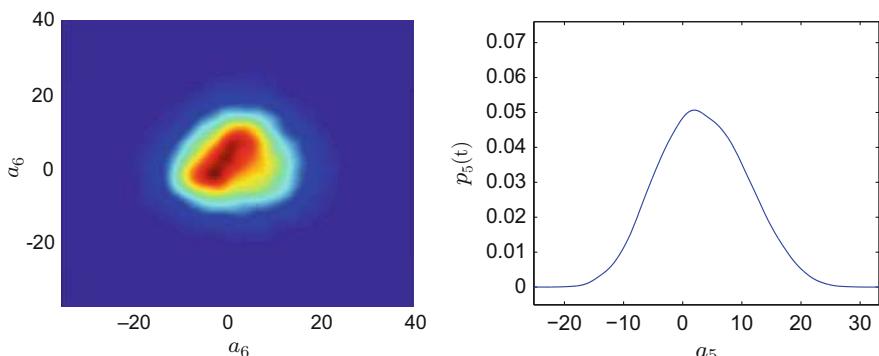


Fig. 29.4 Lorenz-96 system. Joint PDFs of different phase variables at time $t = 100$. Setting $c = 20$ in (29.9) yields a chaotic dynamics. In this case it can be shown that the joint PDF that solves Eq. (29.3) goes to a fractal attractor with Hausdorff dimension 34.5 [69]. However, the reduced-order PDFs are approximately Gaussian. This can be justified on the basis of chaos and multidimensional integration (29.6)

techniques [26] and PDF estimates to construct a hybrid approach in which the memory and the initial condition terms in the MZ equation are computed on the fly based on samples. In this way, one can compensate for the loss of accuracy associated with the approximation of the full MZ equation with few samples of the full dynamical system. A closely related approach is to estimate the expansion coefficients of the effective propagator (subsequent section) by using samples of the full dynamical system (29.1) and retain only the coefficients larger than a certain threshold.

5.2 Effective Propagators

Let us consider a dynamical system in the form (29.1) with time-independent f , i.e., an autonomous system. The formal solution to the joint PDF equation (29.3) in this case can be expressed as

$$p(t) = e^{tL} p(0). \quad (29.28)$$

If the initial state $p(0)$ is separable, i.e., if $p(0) = p_a(0)p_b(0)$ (where a and b are the relevant and irrelevant phase-space coordinates), then the exact evolution of the relevant part of the PDF is given by

$$p_a(t) = \langle e^{tL} \rangle p_a(0), \quad (29.29)$$

where $\langle \cdot \rangle$ is an average with respect to the PDF $p_b(0)$. For example, the exact evolution of the PDF of the first component of the Lorenz-96 system (29.9) is given by

$$p_1(a_1, t) = \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{tL} p_2(a_2, 0) \cdots p_n(a_n, 0) da_2 \cdots da_n \right) p_1(a_1, 0), \quad (29.30)$$

where L , in this case, is

$$L = -nI - \sum_{i=1}^n [(a_{i+1} - a_{i-2}) a_{i-1} + a_i + c] \frac{\partial}{\partial a_i}. \quad (29.31)$$

The linear operator $\langle e^{tL} \rangle$ appearing in (29.29) is known as *relaxation operator* [74] or *effective propagator* [63, 87] of the reduced-order dynamics. Such a propagator is no longer a semigroup as $\langle e^{(t+s)L} \rangle \neq \langle e^{tL} \rangle \langle e^{sL} \rangle$, i.e., the evolution of $p_a(t)$ is non-Markovian. This reflects the memory effect induced in the reduced-order dynamics when we integrate out the phase variables b . To compute the effective propagator, we need to resort to approximations. For example, we could expand it in a power series [34, 70] as

$$\langle e^{tL} \rangle = I + \sum_{k=1}^{\infty} \frac{t^k}{k!} \langle L^k \rangle. \quad (29.32)$$

This expression shows that the dynamics of the PDF $p_a(t)$ is fully determined by the moments of the operator L relative to the joint distribution of the irrelevant phase variables. In particular, the k th-order moment $\langle L^k \rangle$ governing the dynamics of $p_1(a_1, t)$ in the Lorenz-96 system (29.30) is a linear differential operator in a_1 involving derivatives up to order k , i.e.,

$$\langle L^k \rangle = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \underbrace{L \cdots L}_{k \text{ times}} p_2(a_2, 0) \cdots p_n(a_n, 0) da_2 \cdots da_n \quad (29.33)$$

$$= \sum_{j=0}^k \alpha_j^{(k)}(a_1) \frac{\partial^j}{\partial a_1^j}. \quad (29.34)$$

The coefficients $\alpha_j^{(k)}$ can be calculated by substituting (29.31) into (29.33) and performing all integrations. This is a cumbersome calculation, but in principle it can be carried out and yields exact results. The problem is that truncating moment expansions such as (29.32) to any finite order usually yields results of poor accuracy. This is because we may be discarding terms growing like t^k , if the norm of $\langle L^k \rangle$ does not decay rapidly enough. A classical approach to overcome these limitations is to use operator cumulants [55, 64, 66, 73, 74]. For autonomous dynamical systems, we have the exact formula (see, e.g., [4])

$$\langle e^{tL} \rangle = e^{\langle e^{tL} - I \rangle_c}, \quad (29.35)$$

where $\langle \cdot \rangle_c$ here denotes a cumulant average, e.g.,

$$\langle L \rangle_c = \langle L \rangle, \quad \langle L^2 \rangle_c = \langle L^2 \rangle - \langle L \rangle^2, \quad \dots. \quad (29.36)$$

Following Kubo [73, 74], we emphasize that many different types of operator cumulants can be defined. Disregarding, for the moment, the specific *prescription* we use to construct such operator cumulants (see [55, 74]), let us differentiate (29.29) with respect to time and take (29.35) into account. This yields the following exact reduced-order PDF equation

$$\frac{\partial p_a(t)}{\partial t} = \left(\langle L \rangle_c + \sum_{k=2}^{\infty} \frac{t^{k-1}}{(k-1)!} \langle L^k \rangle_c \right) p_a(t), \quad (29.37)$$

which is completely equivalent to the MZ-PDF equation (29.22). Any truncation of the series expansion in (29.37) yields an approximated equation whose specific form depends on the way we define the cumulant average $\langle \cdot \rangle_c$. For example,

we can get expansions in terms of *Kubo-Van Kampen*, *Waldenfels*, or *Speicher* operator cumulants (see the Appendix of [136] or [55, 97]). The choice of the most appropriate operator cumulant expansion is problem dependent.

Other methods to compute approximations to the effective propagator $\langle e^{tL} \rangle$ rely on functional renormalization, in particular on *renormalized perturbation series* ([87], Ch. 5). The key idea of these approaches is to use methods of many body-theory to generalize traditional perturbation series to the case of strong interactions. Formal treatment of this subject, along with the introduction of diagrammatic representations, can be found in [5, 87]. If $p_a(t)$ involves q phase variables (a_1, \dots, a_q) , then each $\langle L^k \rangle_c$ is a linear operator of order k involving a linear combination of generators $\partial^j / \partial a_k^j$ in the form

$$\langle L^k \rangle_c = \sum_{i_1, \dots, i_q=0}^k \beta_{i_1 \dots i_q}^{(k)}(a_1, \dots, a_q) \frac{\partial^{i_1 + \dots + i_q}}{\partial a_1^{i_1} \dots \partial a_q^{i_q}}. \quad (29.38)$$

A substitution of this series expansion into Eq. (29.37) immediately suggests that the exact evolution of the reduced-order PDF $p_a(t)$ is governed, in general, by a linear PDE involving derivatives of *infinite order* in the phase variables (a_1, \dots, a_q) . All coefficients $\beta_{i_1 \dots i_q}^{(k)}(a_1, \dots, a_q)$ appearing in (29.38) can be expressed in terms of integrals of polynomial functions of f_i (see Eq. (29.1)). However, computing such coefficients at all orders is neither trivial nor practical. On the other hand, determining an approximated advection-diffusion form of (29.37) is possible, simply by taking into account those coefficients leading to second-order derivatives in the phase variables. This can be achieved in a systematic way by truncating the series (29.38) to derivatives of second order and computing the corresponding coefficients.

5.2.1 Operator Splitting and Series Expansions

Let us consider an autonomous dynamical system evolving from a random initial state. The propagator $U(t, t_0) = e^{(t-t_0)L}$ forms a semigroup and therefore it can be split as

$$U(t_n, t_0) = U(t_n, t_{n-1}) \cdots U(t_2, t_1) U(t_1, t_0). \quad (29.39)$$

Each operator $U(t_i, t_{i-1})$ (short-time propagator) can be then approximated according to an appropriate decomposition formula [10, 122, 123, 150]. In particular, if L is given by (29.3) and if $\Delta t = |t_i - t_{i-1}|$ is small, then one can use the following first-order approximation

$$\exp \left[-\Delta t \left(\sum_{i=1}^n \frac{\partial f_i}{\partial a_i} + f_i \frac{\partial}{\partial a_i} \right) \right] \simeq \exp \left[-\Delta t \sum_{i=1}^n \frac{\partial f_i}{\partial a_i} \right] \prod_{k=1}^n \exp \left[-\Delta t f_k \frac{\partial}{\partial a_k} \right]. \quad (29.40)$$

This allows us to split the joint PDF equation (29.3) into a system of PDF equations. This approach is quite standard in numerical methods to solve linear PDEs in which the generator of the semigroup can be represented as a superimposition of linear operators. The error estimate for the decomposition formula (29.40) is given in [122, 124]. Higher-order formulas such as Lie-Trotter, Suzuki, and related Backer-Campbell-Hausdorff formulas can be constructed as well. The literature on this subject is very rich e.g., [9, 15, 122, 127, 148, 151].

A somewhat related approach relies on approximating the exponential semigroup e^{tL} in terms of operator polynomials, e.g., the *Faber polynomials* F_k [103]. In this case, the exact evolution of the PDF can be expressed as

$$p_a(t) = \sum_{k=0}^N \psi_k(t) \Phi_k(a), \quad \text{where} \quad \Phi_k(a) = \langle F_k(L) \rangle p_a(0). \quad (29.41)$$

In particular, if F_k are generated by elliptic conformal mappings, then they satisfy a three-term recurrence in the form

$$F_{k+1}(L) = (L - c_0) F_k(L) - c_1 F_{k-1}(L), \quad F_0(L) = I, \quad (29.42)$$

which yields an *unclosed* three-term recurrence for the *modes* Φ_k

$$\Phi_{k+1}(a) = \langle L F_k(L) \rangle p_a(0) - c_0 \Phi_k(a) - c_1 \Phi_{k-1}(a), \quad \Phi_0(a) = 1. \quad (29.43)$$

In some cases, the operator averages $\langle L^n \rangle$ appearing in $\langle L F_k(L) \rangle$ can be reduced to one-dimensional integrals. This happens, in particular, if the initial $p(0)$ is separable and if the functions f_k appearing in (29.1) are separable as well. Although this might seem a severe restriction, it is actually satisfied by many systems including Lorentz-96 [79], Kraichnan-Orszag [106], and the semi-discrete form of SPDEs with polynomial-type nonlinearities (e.g., viscous Burgers and Navier-Stokes equations).

5.3 Algorithms and Solvers

MZ-PDF equations are a particular class of probability density function equations involving memory and initial condition terms. Computing the numerical solution to a probability density function equation is, in general, a very challenging task that involves several problems of different nature. In particular,

High dimensionality: PDF equations describing realistic physical systems usually involve many phase variables. For example, the Fokker-Planck equation of classical statistical mechanics yields a joint probability density function in n phase variables, where n is the dimension of the underlying dynamical system, plus time.

Multiple scales: PDF equations may involve multiple scales in space and time, which could be hardly accessible by conventional numerical methods. For

example, the joint PDF equation (29.3) is a hyperbolic conservation law whose solution is purely advected (with no diffusion) by the compressible flow G . This can easily yield mixing, fractal attractors, and all sorts of complex dynamics (see Fig. 29.1).

Lack of regularity: The solution to a PDF equation is, in general, a distribution [68]. For example, it could be a multivariate Dirac delta function, a function with shock-type discontinuities [23], or even a fractal object. From a numerical viewpoint, resolving such distributions is not trivial although in some cases it can be done by taking integral transformations or projections [156]. An additional numerical difficulty inherent to the simulation of PDF equations arises due to the fact that the solution could be compactly supported over disjoint domains. This obviously requires the development of appropriate numerical techniques such as adaptive discontinuous Galerkin methods [21, 28, 113].

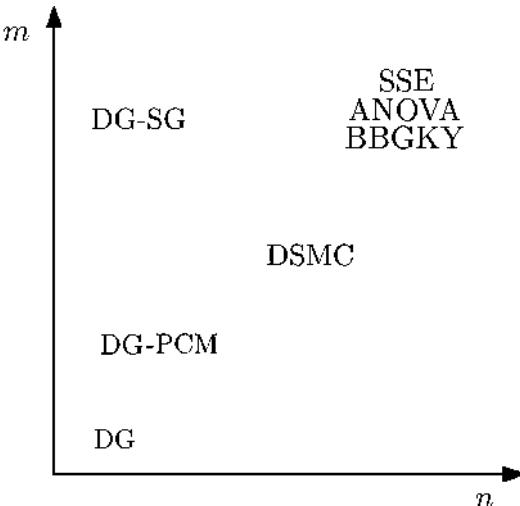
Conservation properties: There are several properties of the solution to a PDF equation that must be preserved in time. The most obvious one is mass, i.e., the solution always integrates to one. Other properties that must be preserved are the positivity of the joint PDF and the fact that a partial marginalization of a joint PDF still yields a PDF.

Long-term integration: The flow map defined by nonlinear dynamical systems can yield large deformations, stretching and folding of the phase space. As a consequence, numerical schemes for kinetic equations associated with such systems will generally loose accuracy in time.

Over the years, many different methods have been proposed to address these issues, with the most efficient ones being problem dependent. For example, a widely used method in statistical fluid mechanics is the particle/mesh method [95, 109–111], which is based directly on stochastic Lagrangian models. Other methods make use of stochastic fields [129] or direct quadrature of moments [47]. In the case of Boltzmann equation, there is a very rich literature. Both probabilistic approaches such as direct simulation Monte Carlo [8, 116] and deterministic methods, e.g., discontinuous Galerkin and spectral methods [18, 19, 39], have been proposed to compute the solution. Probabilistic methods such as direct simulation Monte Carlo are extensively used because of their very low computational cost compared to finite volumes, finite differences, or spectral methods, especially in the multidimensional case. However, Monte Carlo usually yields poorly accurate and fluctuating solutions, which need to be post-processed appropriately, for example, through variance reduction techniques. We refer to Dimarco and Pareschi [31] for a recent review.

In our previous work [21], we addressed the lack of regularity and high dimensionality (in the space of parameters) of kinetic equations by using adaptive discontinuous Galerkin methods [28, 114] combined with sparse probabilistic collocation. Specifically, the phase variables of the system were discretized by using spectral elements on an adaptive nonconforming grid that tracks the support of the PDF in time, while the parametric dependence of the solution was handled by using sparse grids. More recently, we proposed and validated new classes of algorithms

Fig. 29.5 Range of applicability of numerical methods for solving PDF equations as a function of the number of phase variables n and the number parameters m appearing in the equation. Shown are: Separated series expansion methods (SSE), BBGKY closures, high-dimensional model representations (ANOVA), adaptive discontinuous Galerkin methods (DG) combined with sparse grids (SG) or tensor product probabilistic collocation (PCM), direct simulation Monte Carlo (DSMC)



addressing the high-dimensional challenge in PDF equations [22]. These algorithms rely on separated series expansions, high-dimensional model representations, and BBGKY hierarchies. Their range of applicability is sketched in Fig. 29.5 as a function of the number of phase variables n and the number of parameters m appearing in the PDF equation (see also Eq. (29.1)).

The numerical treatment of MZ-PDF equations is even more challenging than classical PDF equations, due to the complexity of the memory and the initial condition terms. Such terms involve the projected part of the full orthogonal dynamics, which is represented by an exponential operator of very high dimension. Computing the solution to MZ-PDF equations, therefore, heavily relies on the approximation of memory and initial condition terms, e.g., in terms of operator cumulants [23, 136], approximate exponential matrices [2, 3, 88], or samples of the full dynamical system [24]. Developing new algorithms to compute the solution to MZ-PDF equations is a matter for future research.

6 Applications

In this section we illustrate the application of the the Mori-Zwanzig formulation to some well-known stochastic systems.

6.1 Stochastic Resonance Driven by Colored Noise

Let us consider a nonlinear dynamical system subject to a weak deterministic periodic signal and additive colored random noise. As is well known, in some cases, e.g., in bistable systems, the cooperation between noise and signal can yield

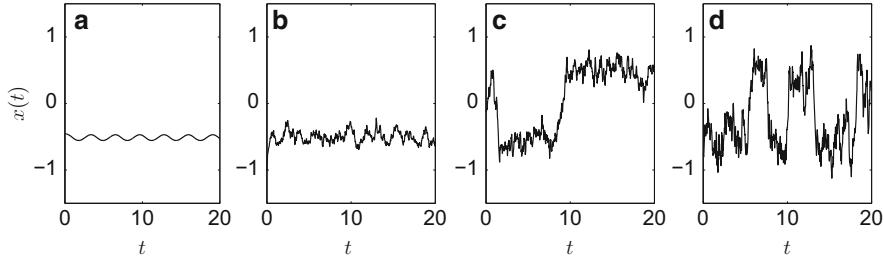


Fig. 29.6 Stochastic resonance. We study the system (29.44) with parameters $\mu = 10$, $\nu = 3$, $\Omega = 2$, $\epsilon = 0.2$ subject to weakly colored random noise ($\tau = 0.01$) of different amplitudes: (a) $\sigma = 0$; (b) $\sigma = 0.2$; (c) $\sigma = 0.4$; (d) $\sigma = 0.8$. Each figure shows only one solution sample. At low noise levels, the average residence time in the two states is much longer than the driving period. However, if we increase the noise level to $\sigma = 0.8$ (d), then we observe almost periodic transitions between the two metastable states. In most cases, we have a jump from one state to the other and back again approximately once per modulation period (Adapted from [136])

a phenomenon known as *stochastic resonance* [6, 82, 104, 147], namely, random noise can enhance significantly the transmission of the weak periodic signal. The mechanism that makes this possible is explained in Fig. 29.6, with reference to the system

$$\begin{cases} \frac{dx(t)}{dt} = \frac{2\mu x - 2\nu x^3 - \nu x^5}{2(1+x^2)^2} + \sigma f(t; \xi) + \epsilon \cos(\Omega t) \\ x(0) = x_0(\omega) \end{cases}. \quad (29.44)$$

Here $\xi \in \mathbb{R}^m$ is a vector of uncorrelated Gaussian random variables, while $x_0 \in \mathbb{R}$ is a Gaussian initial state. Also, the random noise $f(t; \xi)$ is assumed to be a zero-mean Gaussian process with exponential covariance function

$$C(t, s) = \frac{1}{2\tau} e^{-|t-s|/\tau} \quad (29.45)$$

and finite correlation time τ .

6.2 Mori-Zwanzig Equation

The exact evolution equation for the PDF of $x(t)$ can be obtained by applying the convolutionless projection operator method described in previous sections. Such an equation is a linear partial differential equation of *infinite order* in the phase variable a . If we consider a second-order approximation, i.e., if we expand the propagator of p_x in terms of cumulant operators and truncate the expansion at the second order, then we obtain

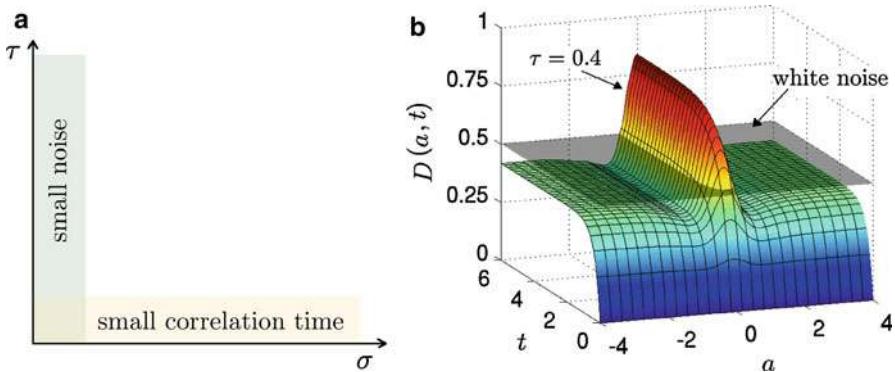


Fig. 29.7 Stochastic resonance. Range of validity of the MZ-PDF equation (29.46) as a function of the noise amplitude σ and correlation time τ (a). Effective diffusion coefficient $D(a, t)$ (see Eq. (29.48)) corresponding to exponentially correlated Gaussian noises (b) (Adapted from [136])

$$\begin{aligned} \frac{\partial p_x}{\partial t} &= L_0 p_x - \epsilon \cos(\Omega t) \frac{\partial p_x}{\partial a} \\ &\quad + \sigma^2 \left[\int_0^t C(t, s) \frac{\partial}{\partial a} e^{(t-s)L_0} \frac{\partial}{\partial a} e^{(s-t)L_0} ds \right] p_x, \end{aligned} \quad (29.46)$$

where

$$L_0 = \frac{\partial}{\partial a} \left(\frac{2\mu a - 2\nu a^3 - \nu a^5}{2(1+a^2)^2} \right) I + \left(\frac{2\mu a - 2\nu a^3 - \nu a^5}{2(1+a^2)^2} \right) \frac{\partial}{\partial a}. \quad (29.47)$$

The rationale behind this approximation is that higher-order cumulants can be neglected [36,37]. This happens, in particular, if both ϵ and σ are small. Faetti et al. [36,37] have shown that for $\epsilon = 0$, the correction due to the fourth-order cumulants is of order $\sigma \tau^2$ for Gaussian noise and order $\sigma \tau$ for other noises. Thus, (29.46) holds true either for small ϵ and σ and arbitrary correlation time τ or for small ϵ and τ and arbitrary noise amplitude σ (see Fig. 29.7). It can be shown (see, e.g., [93,136]) that (29.46) is equivalent to the following *advection-diffusion equation*

$$\frac{\partial p_x}{\partial t} = L_0 p_x - \varepsilon \cos(\Omega t) \frac{\partial p_x}{\partial a} + \sigma^2 \frac{\partial^2}{\partial a^2} (D(a, t) p_x), \quad (29.48)$$

where the effective diffusion coefficient $D(a, t)$ depends on the type of noise. Note that if the correlation time τ goes to zero (white-noise limit), then Eq. (29.46), with $C(t, s)$ defined in (29.45), consistently reduces to the classical Fokker-Planck equation. The proof is simple, and it relies on the limits

$$\lim_{\tau \rightarrow 0} \int_0^t \frac{1}{2\tau} e^{-s/\tau} ds = \frac{1}{2}, \quad \lim_{\tau \rightarrow 0} \int_0^t \frac{1}{2\tau} e^{-s/\tau} s^k ds = 0, \quad k \in \mathbb{N}. \quad (29.49)$$

These equations allow us to conclude that

$$\lim_{\tau \rightarrow 0} \int_0^t \frac{1}{2\tau} e^{-s/\tau} \frac{\partial}{\partial a} e^{sL_0} \frac{\partial}{\partial a} e^{-sL_0} ds = \lim_{\tau \rightarrow 0} \left[\int_0^t \frac{1}{2\tau} e^{-s/\tau} ds \right] \frac{\partial^2}{\partial a^2} = \frac{1}{2} \frac{\partial^2}{\partial a^2}, \quad (29.50)$$

i.e., for $\tau \rightarrow 0$ Eq. (29.46) reduces to the Fokker-Planck equation

$$\frac{\partial p_x}{\partial t} = L_0 p_x - \epsilon \cos(\Omega t) \frac{\partial p_x}{\partial a} + \frac{\sigma^2}{2} \frac{\partial^2 p_x}{\partial a^2}. \quad (29.51)$$

Next, we study the transient dynamics of the one-time PDF of the solution $x(t)$ within the period $T = 3$. To this end, we consider the following set of parameters $\mu = 1$, $\nu = 1$, $\Omega = 10$, $\epsilon = 0.5$, leading to a slow relaxation to statistical equilibrium. This allows us to study the transient of the PDF more carefully and compare the results with Monte Carlo (MC). This is shown in Fig. 29.8, where it is seen that for small σ the random forcing term in (29.44) does not influence significantly the dynamics and therefore the PDF of $x(t)$ is mainly advected by the operator L_0 . Note that the PDF tends to accumulate around the metastable equilibrium states $\pm \sqrt{\sqrt{3} - 1}$. For larger σ the probability of switching between the metastable states increases and therefore the strong bimodality observed in Fig. 29.8 (left) is attenuated.

6.3 Fractional Brownian Motion, Levy, and Other Noises

There exists a close connection between the statistical properties of the random noise and the structure of the MZ-PDF equation for the response variables of the system. In particular, it has been recently shown, e.g., in [75], that the PDF of the solution to the Langevin equation driven by Levy flights satisfies a *fractional* Fokker-Plank equation. Such an equation can be easily derived by using the Mori-Zwanzig projection operator framework, which represents therefore a very general tool to exploit the relation between noise and reduced-order PDF equations. For example, let us consider the non-stationary covariance function of *fractional Brownian motion*

$$C(t, s) = \frac{1}{2} (|t|^{2h} + |s|^{2h} - |t - s|^{2h}), \quad 0 < h < 1, \quad t, s > 0 \quad (29.52)$$

which reduces to the covariance function of standard Levy noise for $h = 1/2$, i.e., $C_{Levy}(t, s) = \min\{t, s\}$. A substitution of (29.52) into (29.46) yields an equation for the PDF of the solution to the system (29.1) driven by fractional Brownian motion of *small amplitude*. As is well known, such noise can trigger either sub-diffusion or super-diffusion in the PDF dynamics.

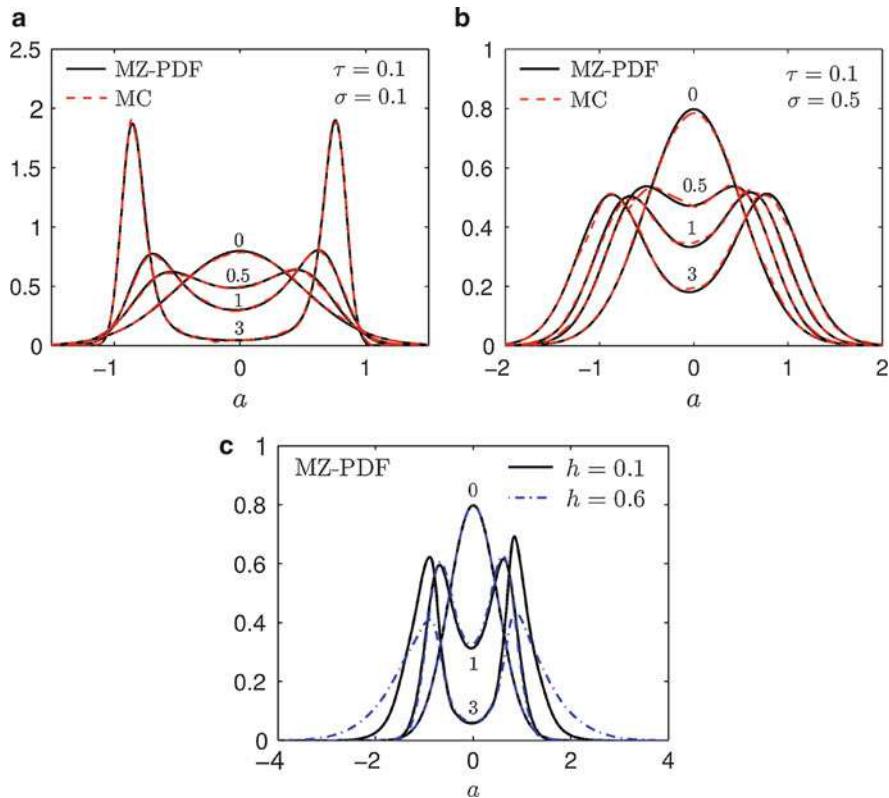


Fig. 29.8 Stochastic resonance. Time snapshots of the PDF of $x(t)$ as predicted by Eq. (29.46) (continuous lines) and MC simulation (10^5 samples) (dashed lines). The Gaussian process $f(t; \xi)$ in (29.44) is exponentially correlated with small correlation time τ ((a) and (b)). Note that the Karhunen-Lo  e expansion of such noise requires 280 Gaussian random variables to achieve 99% of the correlation energy in the time interval $[0, 3]$. We also show the PDF dynamics corresponding to fractional Brownian motion of small amplitude and different Hurst indices (c) (Adapted from [136])

6.4 Stochastic Advection-Reaction

Let us consider the advection-reaction equation for a scalar concentration field

$$\frac{\partial u}{\partial t} + V(x) \cdot \nabla u = [\kappa_0(x) + \sigma \kappa_1(x; \xi)] R(u), \quad \xi \in \mathbb{R}^m, \quad (29.53)$$

where $V(x)$ is a divergence-free (deterministic) advection velocity field, $R(u)$ is a nonlinear reaction term, and $\kappa_1(x; \xi)$ is a zero-mean random perturbation in the reaction rate $\kappa_0(x)$. In Fig. 29.9 we plot a few samples of the concentration field solving (29.53) in one spatial dimension, for different realizations of the random reaction rate and the random initial condition.

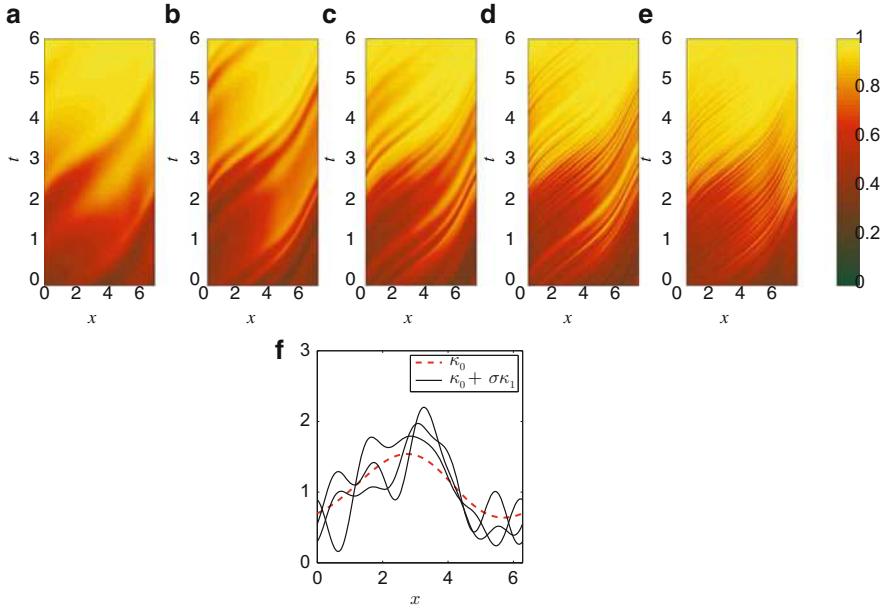


Fig. 29.9 Stochastic advection-reaction. Samples of the concentration field solving (29.53) in one spatial dimension for periodic boundary conditions and different realizations of the random initial condition and the random reaction rate. The correlation length of the random initial condition decreases from (a) to (e). In (f) we plot a few realizations of the random reaction rate ($\sigma = 0.3$) (Adapted from [136])

In [135, 141] we have studied Eq. (29.53) by using the response-excitation PDF method as well as the large-eddy-diffusivity (LED) closure [125]. Here we consider a different approach based on the MZ-PDF equation [136]. To this end, we assume that σ is reasonably small and that the concentration field u is independent of ξ at initial time, i.e., that the initial joint PDF of the system has the form $p(0) = p_u(0)p_\xi$. In these hypotheses, we obtain the following second-order approximation to the MZ-PDF equation

$$\frac{\partial p_u(t)}{\partial t} = L_0 p_u(t) + \sigma^2 \left[\int_0^t \langle \kappa_1 e^{sL_0} \kappa_1 \rangle e^{-sL_0} ds \right] F^2 p_u(t), \quad (29.54)$$

where the average $\langle \cdot \rangle$ is relative to the joint PDF of ξ and

$$L_0 = -\kappa_0(x)F - V(x) \cdot \nabla, \quad F = \frac{\partial R(a)}{\partial a} + R(a) \frac{\partial}{\partial a}. \quad (29.55)$$

Equation (29.54) is linear, but it involves derivatives of infinite order in both variables x and a . Such derivatives come from the exponential operators e^{sL_0} within the time convolution term. Note that such convolution can be also expressed as

a functional derivative [133] of the exponential operator along $\kappa_1(x)$, by using an identity of Feynman (see [38], Eq. (6) or [151]). In a finite-dimensional setting, these quantities can be computed by using efficient numerical algorithms, e.g., based on scaling-squaring methods and Padé approximants [2,3,88]. In Fig. 29.10 we plot the time snapshots of the PDF of the concentration field as predicted by the MZ-PDF equation (29.54). The comparison between such PDF and a Monte Carlo solution is done in Fig. 29.11. It is seen that in this case the second-order operator cumulant approximation provides accurate results for a quite large degree of perturbation (see Fig. 29.9f).

6.5 Stochastic Burgers Equation

The Mori-Zwanzig formulation can be applied also to the Burgers equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \sigma f(x, t; \omega) \quad (29.56)$$

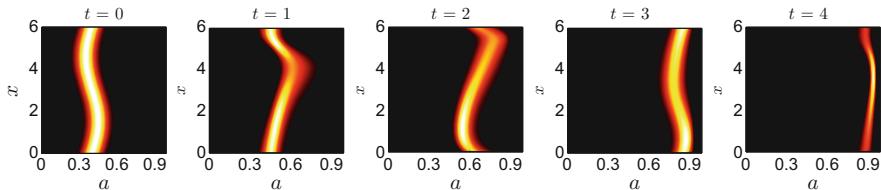


Fig. 29.10 Stochastic advection-reaction. Time snapshots of the concentration PDF predicted by the MZ-PDF equation (29.54) (Adapted from [136])

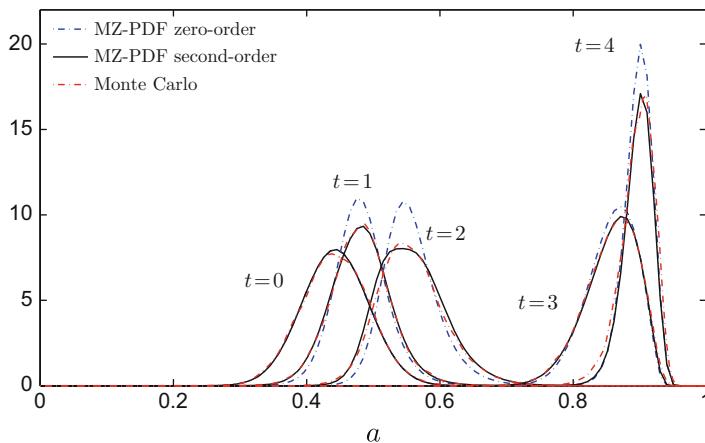


Fig. 29.11 Stochastic advection-reaction. Comparison between the MZ-PDF solution at $x = 1$ and a nonparametric kernel density estimation [12] of the PDF based on 10000 MC solution samples. The zero-order approximation is obtained by neglecting the second-order term in σ in Eq. (29.54) (Adapted from [136])

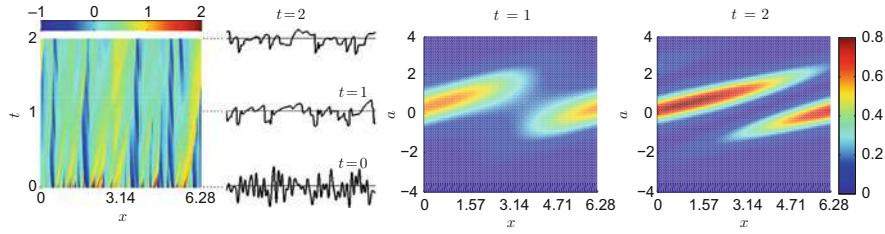


Fig. 29.12 Stochastic Burgers equation. One realization of the velocity field computed by using adaptive discontinuous Galerkin methods (left). Time snapshots of the one-point PDF obtained by solving the MZ-PDF equation (29.57) (Adapted from [23])

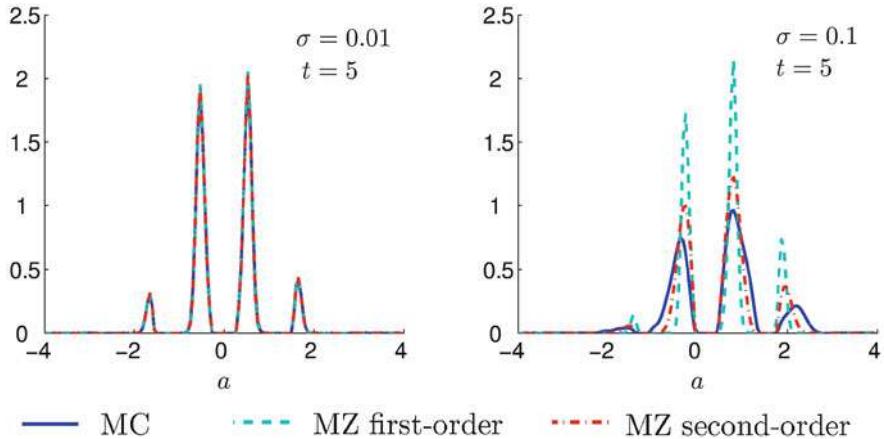


Fig. 29.13 Stochastic Burgers equation. One-point PDF of the velocity field at $x = \pi$ for exponentially correlated, homogeneous (in space) random forcing processes with correlation time 0.01 and amplitude $\sigma = 0.01$ and $\sigma = 0.1$ (second row). Shown are results obtained from MC and from two different truncations of the MZ-PDF equation (29.57) (Adapted from [23])

to formally integrate out the random forcing term. This yields the following equation (second-order approximation) for the one-point one-time PDF of the velocity field (see Fig. 29.12 and [23])

$$\frac{\partial p_u(t)}{\partial t} = L_0 p_u(t) + \sigma \langle f(x, t) \rangle \frac{\partial p_u(t)}{\partial a} + \sigma^2 \left[\int_0^t \left\langle f(x, t) \frac{\partial}{\partial a} e^{(t-s)L_0} f(x, s) \right\rangle \frac{\partial}{\partial a} e^{-(t-s)L_0} ds \right] p_u(t), \quad (29.57)$$

where L_0 is given by

$$L_0 = - \int_{-\infty}^a da \frac{\partial}{\partial x} - a \frac{\partial}{\partial x}. \quad (29.58)$$

In Fig. 29.13 we compare the PDF dynamics obtained by solving Eq. (29.57) with Monte Carlo simulation. It is seen that, as we increase the amplitude σ of the forcing, the second-order approximation (29.57) loses accuracy and higher-order corrections have to be included.

6.6 Coarse-Grained Models of Particle Systems

Particle systems are often used in models of system biology and soft matter physics to simulate and understand large-scale effects based on microscopic first principles. The computability of such systems depends critically on the number of particles and the interaction potentials. Full molecular dynamics (MD) simulations can be performed for particle systems with $\mathcal{O}(10^{13})$ particles. However, such “hero” simulations require hundred of thousands of computer cores and significant time and data processing to be successfully completed. This motivates the use of coarse-graining techniques, such as dissipative particle dynamics (DPD) [98], for particle systems to compute macroscopic/mesoscopic observables at a reasonable computational cost. Among different approaches, the Mori-Zwanzig formulation [61, 118] has proved to be effective in achieving this goal [1, 58, 77]. The key idea is shown in Fig. 29.14, where a star polymer described atomistically is coarse-grained to bigger particles – the MZ-DPD particles – by following the procedure sketched in Fig. 29.3 for a star polymer. The calculation of the solution to the MZ-DPD system, e.g., Eq. (29.14), relies on approximations. In particular, the memory term plays an important role in the dynamics of the coarse-grained system, and this role becomes more relevant as we increase the coarse-graining level [77, 157]. In Fig. 29.14 we compare the velocity autocorrelation function obtained from molecular dynamics simulations (MD) and the coarse-grained MZ-DPD system.

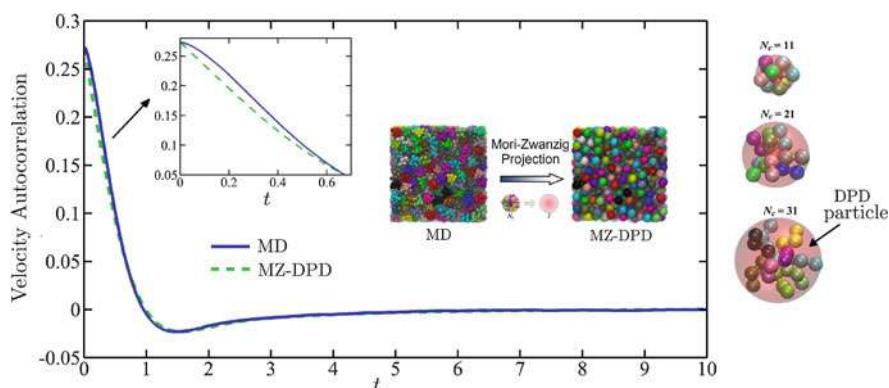


Fig. 29.14 Coarse-grained model of a particle system. Comparison between the velocity autocorrelation function obtained from molecular dynamics simulation (MD) and Mori-Zwanzig dissipative particle-dynamics (MZ-DPD) (Courtesy of Dr. Zhen Li, Brown University (unpublished))

7 Conclusions

In this chapter we discussed how to perform the contraction of state variables in nonequilibrium stochastic dynamical systems by using the Mori-Zwanzig projection operator method and the effective propagator approaches. Both techniques yield exact equations of motion for quantities of interest in high-dimensional systems, e.g., functionals of the solution to systems of stochastic ordinary and partial differential equations. Examples of such functionals are the position and momentum of clusters of particles (MZ-DPD methods), the series expansion of the solution to a SPDE, or the turbulent viscosity in the inertial range of fully developed turbulence. One of the main advantages of developing such exact equations is that they allow us to avoid integrating the full (high-dimensional) stochastic dynamical system and solve directly for the quantities of interest, thus reducing the computational cost significantly. In principle, this can break the curse of dimensionality in numerical simulations of SODEs and SPDEs at the price of solving complex integrodifferential equations. Computing the solution to the Mori-Zwanzig equations relies on approximations and appropriate numerical schemes. Over the years many different techniques have been proposed for this scope, with the most efficient ones being problem dependent. We discussed classical perturbation methods such as truncated operator cumulant expansions, as well as more recent approaches, e.g., based on orthogonal expansions of memory kernels, renormalized perturbation theory, sampling techniques, and maximum entropy principles. There is no general recipe to effectively approximate the Mori-Zwanzig equations for systems in which the relevant and the irrelevant phase variables have similar dynamical properties and order of magnitude. This situation arises very often when dealing with the problem of eliminating macroscopic phase variables, and it should be approached on a case-by-case basis.

8 Cross-References

- ▶ [Hierarchical Models for Uncertainty Quantification: An Overview](#)
- ▶ [Multiresolution Analysis for Uncertainty Quantification](#)
- ▶ [Polynomial Chaos: Modeling, Estimation, and Approximation](#)
- ▶ [Random Vectors and Random Fields in High Dimension: Parametric Model-based Representation, Identification from Data, and Inverse Problems](#)
- ▶ [Sparse Collocation Methods for Stochastic Interpolation and Quadrature](#)
- ▶ [Stochastic Collocation Methods: A Survey](#)

References

1. Akkermans, R.L.C., Briels, W.J.: Coarse-grained dynamics of one chain in a polymer melt. *J. Chem. Phys.* **113**(15), 620–630 (2000)
2. Al-Mohy, A.H., Higham, N.J.: Computing the Fréchet derivative of the matrix exponential with an application to condition number estimation. *SIAM J. Matrix Anal. Appl.* **30**(4), 1639–1657 (2009)
3. Al-Mohy, A.H., Higham, N.J.: Computing the action of the matrix exponential with an application to exponential integrators. *SIAM J. Sci. Comput.* **33**(2), 488–511 (2011)
4. Arai, T., Goodman, B.: Cumulant expansion and Wick theorem for spins. Application to the antiferromagnetic ground state. *Phys. Rev.* **155**(2), 514–527 (1967)
5. Balescu, R.: Equilibrium and Non-equilibrium Statistical Mechanics. Wiley, New York (1975)
6. Benzi, R., Sutera, A., Vulpiani, A.: The mechanism of stochastic resonance. *J. Phys. A: Math. Gen.* **14**:L453–L457 (1981)
7. Billings, S.A.: Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains. Wiley, Chichester (2013)
8. Bird, G.A.: Molecular Gas Dynamics and Direct Numerical Simulation of Gas Flows. Clarendon Press, Oxford (1994)
9. Blanes, S., Casas, F., Oteo, J.A., Ros, J.: The Magnus expansion and some of its applications. *Phys. Rep.* **470**, 151–238 (2009)
10. Blanes, S., Casas, F., Murua, A.: Splitting methods in the numerical integration of non-autonomous dynamical systems. *RACSAM* **106**, 49–66 (2012)
11. Bonatto, C., Gallas, J.A.C., Ueda, Y.: Chaotic phase similarities and recurrences in a damped-driven Duffing oscillator. *Phys. Rev. E* **77**, 026217(1–5) (2008)
12. Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. *Ann. Stat.* **38**(5), 2916–2957 (2010)
13. Breuer, H.P., Kappler, B., Petruccione, F.: The time-convolutionless projection operator technique in the quantum theory of dissipation and decoherence. *Ann. Phys.* **291**, 36–70 (2001)
14. Broer, H., Simó, C., Vitolo, R.: Bifurcations and strange attractors in the Lorenz-84 climate model with seasonal forcing. *Nonlinearity* **15**, 1205–1267 (2002)
15. Casas, F.: Solutions of linear partial differential equations by Lie algebraic methods. *J. Comput. Appl. Math.* **76**, 159–170 (1996)
16. Cercignani, C., Gerasimenko, U.I., Petrina, D.Y. (eds.): Many Particle Dynamics and Kinetic Equations, 1st edn. Kluwer Academic, Dordrecht/Boston (1997)
17. Chaturvedi, S., Shibata, F.: Time-convolutionless projection operator formalism for elimination of fast variables. Applications to Brownian motion. *Z. Phys. B* **35**, 297–308 (1979)
18. Cheng, Y., Gamba, I.M., Majorana, A., Shu, C.W.: A discontinuous Galerkin solver for Boltzmann-Poisson systems in nano devices. *Comput. Methods Appl. Mech. Eng.* **198**, 3130–3150 (2009)
19. Cheng, Y., Gamba, I.M., Majorana, A., Shu, C.W.: A brief survey of the discontinuous Galerkin method for the Boltzmann-Poisson equations. *SEMA J.* **54**, 47–64 (2011)
20. Chertock, A., Gottlieb, D., Solomonoff, A.: Modified optimal prediction and its application to a particle method problem. *J. Sci. Comput.* **37**(2), 189–201 (2008)
21. Cho, H., Venturi, D., Karniadakis, G.E.: Adaptive discontinuous Galerkin method for response-excitation PDF equations. *SIAM J. Sci. Comput.* **5**(4), B890–B911 (2013)
22. Cho, H., Venturi, D., Karniadakis, G.E.: Numerical methods for high-dimensional probability density function equations. *J. Comput. Phys. Under Rev.* (2014)
23. Cho, H., Venturi, D., Karniadakis, G.E.: Statistical analysis and simulation of random shocks in Burgers equation. *Proc. R. Soc. A* **2171**(470), 1–21 (2014)

24. Chorin, A., Lu, F.: A discrete approach to stochastic parametrization and dimensional reduction in nonlinear dynamics, pp. 1–12. arXiv:submit/1219662 (2015)
25. Chorin, A.J., Stinis, P.: Problem reduction, renormalization and memory. *Commun. Appl. Math. Comput. Sci.* **1**(1), 1–27 (2006)
26. Chorin, A.J., Tu, X.: Implicit sampling for particle filters. *PNAS* **106**(41), 17249–17254 (2009)
27. Chorin, A.J., Hald, O.H., Kupferman, R.: Optimal prediction and the Mori-Zwanzig representation of irreversible processes. *Proc. Natl. Acad. Sci. U. S. A.* **97**(7), 2968–2973 (2000)
28. Cockburn, B., Karniadakis, G.E., Shu, C.W.: Discontinuous Galerkin Methods, Vol. 11 of Lecture Notes in Computational Science and Engineering. Springer, New York (2000)
29. Darve, E., Solomon, J., Kia, A.: Computing generalized Langevin equations and generalized Fokker-Planck equations. *Proc. Natl. Acad. Sci. U. S. A.* **106**(27), 10884–10889 (2009)
30. Dekker, H.: Correlation time expansion for multidimensional weakly non-Markovian Gaussian processes. *Phys. Lett. A* **90**(1–2), 26–30 (1982)
31. Dimarco, G., Pareschi, L.: Numerical methods for kinetic equations. *Acta Numer.* **23**(4), 369–520 (2014)
32. Doostan, A., Owhadi, H.: A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.* **230**(8), 3015–3034 (2011)
33. Edwards, S.F.: The statistical dynamics of homogeneous turbulence. *J. Fluid Mech.* **18**, 239–273 (1964)
34. Engel, K.J., Nagel, R.: One-Parameter Semigroups for Linear Evolution Equations. Springer, New York (2000)
35. Faetti, S., Grigolini, P.: Unitary point of view on the puzzling problem of nonlinear systems driven by colored noise. *Phys. Rev. A* **36**(1), 441–444 (1987)
36. Faetti, S., Fronzoni, L., Grigolini, P., Mannella, R.: The projection operator approach to the Fokker-Planck equation. I. Colored Gaussian noise. *J. Stat. Phys.* **52**(3/4), 951–978 (1988)
37. Faetti, S., Fronzoni, L., Grigolini, P., Palleschi, V., Tropiano, G.: The projection operator approach to the Fokker-Planck equation. II. Dichotomic and nonlinear Gaussian noise. *J. Stat. Phys.* **52**(3/4), 979–1003 (1988)
38. Feynman, R.P.: An operator calculus having applications in quantum electrodynamics. *Phys. Rev.* **84**, 108–128 (1951)
39. Filbet, F., Russo, G.: High-order numerical methods for the space non-homogeneous Boltzmann equations. *J. Comput. Phys.* **186**, 457–480 (2003)
40. Foias, C., Sell, G.R., Temam, R.: Inertial manifolds for nonlinear evolutionary equations. *Proc. Natl. Acad. Sci. U.S.A.* **73**(2), 309–353 (1988)
41. Foias, C., Manley, O.P., Rosa, R., Temam, R.: Navier-Stokes equations and turbulence, 1st edn. Cambridge University Press (2001)
42. Foias, C., Jolly, M.S., Manley, O.P., Rosa, R.: Statistical estimates for the Navier-Stokes equations and Kraichnan theory of 2-D fully developed turbulence. *J. Stat. Phys.* **108**(3/4), 591–646 (2002)
43. Foo, J., Karniadakis, G.E.: The multi-element probabilistic collocation method (ME-PCM): error analysis and applications. *J. Comput. Phys.* **227**, 9572–9595 (2008)
44. Foo, J., Karniadakis, G.E.: Multi-element probabilistic collocation method in high dimensions. *J. Comput. Phys.* **229**, 1536–1557 (2010)
45. Fox, R.F.: A generalized theory of multiplicative stochastic processes using Cumulant techniques. *J. Math. Phys.* **16**(2), 289–297 (1975)
46. Fox, R.F.: Functional-calculus approach to stochastic differential equations. *Phys. Rev. A* **33**(1), 467–476 (1986)
47. Fox, R.O.: Computational Models for Turbulent Reactive Flows. Cambridge University Press, Cambridge (2003)
48. Friedrich, R., Daitche, A., Kamps, O., Lülf, J., Voßkuhle, M., Wilczek, M.: The Lundgren-Monin-Novikov hierarchy: kinetic equations for turbulence. *Comp. Rend. Phys.* **13**(9–10), 929–953 (2012)
49. Frisch, U.: Turbulence: the legacy of A. N. Kolmogorov. Cambridge University Press, Cambridge (1995)

50. Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach. Springer, New York (1998)
51. Hänggi, P.: Correlation functions and master equations of generalized (non-Markovian) Langevin equations. *Z. Phys. B* **31**, 407–416 (1978)
52. Hänggi, P.: On derivations and solutions of master equations and asymptotic representations. *Z. Phys. B* **30**, 85–95 (1978)
53. Hänggi, P.: The functional derivative and its use in the description of noisy dynamical systems. In: Pesquera, L., Rodriguez, M. (eds.) *Stochastic Processes Applied to Physics*, pp. 69–95. World Scientific, Singapore (1985)
54. Hänggi, P., Jung, P.: Colored noise in dynamical systems. In: Prigogine, I., Rice, S.A. (eds.) *Advances in Chemical Physics*, vol. 89, pp. 239–326. Wiley-Interscience, New York (1995)
55. Hegerfeldt, G.C., Schulze, H.: Noncommutative cumulants for stochastic differential equations and for generalized Dyson series. *J. Stat. Phys.* **51**(3/4), 691–710 (1988)
56. Herring, J.R.: Self-consistent-field approach to nonstationary turbulence. *Phys. Fluids* **9**(11), 2106–2110 (1966)
57. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, Cambridge (2007)
58. Hijón, C., nol, P.E., Vanden-Eijnden, E., Delgado-Buscalioni, R.: Mori-Zwanzig formalism as a practical computational tool. *Faraday Discuss.* **144**, 301–322 (2010)
59. Hosokawa, I.: Monin-Lundgren hierarchy versus the Hopf equation in the statistical theory of turbulence. *Phys. Rev. E* **73**, 067301(1–4) (2006)
60. Hughes, K.H., Burghardt, I.: Maximum-entropy closure of hydrodynamic moment hierarchies including correlations. *J. Chem. Phys.* **136**, 214109(1–18) (2012)
61. Izvekov, S.: Microscopic derivation of particle-based coarse-grained dynamics. *J. Chem. Phys.* **138**, 134106(1–16) (2013)
62. Jaynes, E.T.: *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, (2003)
63. Jensen, R.V.: Functional integral approach to classical statistical dynamics. *J. Stat. Phys.* **25**(2), 183–210 (1981)
64. Kampen, N.G.V.: A cumulant expansion for stochastic linear differential equations. II. *Physica* **74**, 239–247 (1974)
65. Kampen, N.G.V.: Elimination of fast variables. *Phys. Rep.* **124**(2), 69–160 (1985)
66. Kampen, N.G.V.: *Stochastic Processes in Physics and Chemistry*, 3rd edn. North Holland, Amsterdam (2007)
67. Kampen, N.G.V., Oppenheim, I.: Brownian motion as a problem of eliminating fast variables. *Physica A* **138**, 231–248 (1986)
68. Kanwal, R.P.: *Generalized Functions: Theory and Technique*, 2nd edn. Birkhäuser, Boston (1998)
69. Karimi, A., Paul, M.R.: Extensive chaos in the Lorenz-96 model. *Chaos* **20**(4), 043105(1–11) (2010)
70. Kato, T.: *Perturbation Theory for Linear Operators*, 4th edn. Springer, New York (1995)
71. Khuri, A.I.: Applications of Dirac's delta function in statistics. *Int. J. Math. Educ. Sci. Technol.* **35**(2), 185–195 (2004)
72. Kraichnan, R.H.: Statistical dynamics of two-dimensional flow. *J. Fluid Mech.* **67**, 155–175 (1975)
73. Kubo, R.: Generalized cumulant expansion method. *J. Phys. Soc. Jpn.* **17**(7), 1100–1120 (1962)
74. Kubo, R.: Stochastic Liouville equations. *J. Math. Phys.* **4**(2), 174–183 (1963)
75. Kullberg, A., del Castillo-Negrete, D.: Transport in the spatially tempered, fractional Fokker-Planck equation. *J. Phys. A: Math. Theor.* **45**(25), 255101(1–21) (2012)
76. Li, G., Wang, S.W., Rabitz, H., Wang, S., Jaffé, P.: Global uncertainty assessments by high dimensional model representations (HDMR). *Chem. Eng. Sci.* **57**(21), 4445–4460 (2002)
77. Li, Z., Bian, X., Caswell, B., Karniadakis, G.E.: Construction of dissipative particle dynamics models for complex fluids via the Mori-Zwanzig formulation. *Soft. Matter.* **10**, 8659–8672 (2014)

78. Lindenberg, K., West, B.J., Masoliver, J.: First passage time problems for non-Markovian processes. In: Moss, F., McClintock, P.V.E. (eds.) *Noise in Nonlinear Dynamical Systems*, vol. 1, pp. 110–158. Cambridge University Press, Cambridge (1989)
79. Lorenz, E.N.: Predictability – a problem partly solved. In: ECMWF Seminar on Predictability, Reading, vol. 1, pp. 1–18 (1996)
80. Luchtenburg, D.M., Brunton, S.L., Rowley, C.W.: Long-time uncertainty propagation using generalized polynomial chaos and flow map composition. *J. Comput. Phys.* **274**, 783–802 (2014)
81. Lundgren, T.S.: Distribution functions in the statistical theory of turbulence. *Phys. Fluids* **10**(5), 969–975 (1967)
82. Luo, X., Zhu, S.: Stochastic resonance driven by two different kinds of colored noise in a bistable system. *Phys. Rev. E* **67**(3/4), 021104(1–13) (2003)
83. Ma, X., Karniadakis, G.E.: A low-dimensional model for simulating three-dimensional cylinder flow. *J. Fluid Mech.* **458**, 181–190 (2002)
84. Ma, X., Zabaras, N.: An adaptive hierarchical sparse grid collocation method for the solution of stochastic differential equations. *J. Comput. Phys.* **228**, 3084–3113 (2009)
85. Mattuck, R.D.: *A Guide to Feynman Diagrams in the Many-Body Problem*. Dover, New York (1992)
86. McCane, A.J., Luckock, H.C., Bray, A.J.: Path integrals and non-Markov processes. 1. General formalism. *Phys. Rev. A* **41**(2), 644–656 (1990)
87. McComb, W.D.: *The Physics of Fluid Turbulence*. Oxford University Press, Oxford (1990)
88. Moler, C., Loan, C.V.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **45**(1), 3–49 (2003)
89. Monin, A.S.: Equations for turbulent motion. *Prikl. Mat. Mekh.* **31**(6), 1057–1068 (1967)
90. Montgomery, D.: A BBGKY framework for fluid turbulence. *Phys. Fluids* **19**(6), 802–810 (1976)
91. Mori, H.: Transport, collective motion, and Brownian motion. *Prog. Theor. Phys.* **33**(3), 423–455 (1965)
92. Mori, H., Morita, T., Mashiyama, K.T.: Contraction of state variables in non-equilibrium open systems. I. *Prog. Theor. Phys.* **63**(6), 1865–1883 (1980)
93. Moss, F., McClintock, P.V.E. (eds.): *Noise in Nonlinear Dynamical Systems. Volume 1: Theory of Continuous Fokker-Planck Systems*. Cambridge University Press, Cambridge (1995)
94. Mukamel, S., Oppenheim, I., Ross, J.: Statistical reduction for strongly driven simple quantum systems. *Phys. Rev. A* **17**(6), 1988–1998 (1978)
95. Muradoglu, M., Jenny, P., Pope, S.B., Caughey, D.A.: A consistent hybrid finite-volume/particle method for the PDF equations of turbulent reactive flows. *J. Comput. Phys.* **154**, 342–371 (1999)
96. Nakajima, S.: On quantum theory of transport phenomena – steady diffusion. *Prog. Theor. Phys.* **20**(6), 948–959 (1958)
97. Neu, P., Speicher, R.: A self-consistent master equation and a new kind of cumulants. *Z. Phys. B* **92**, 399–407 (1993)
98. Español, P., Warren, P.: Statistical mechanics of dissipative particle dynamics. *EuroPhys. Lett.* **30**(4), 191–196 (1995)
99. Noack, B.R., Niven, R.K.: A hierarchy of maximum entropy closures for Galerkin systems of incompressible flows. *Comput. Math. Appl.* **65**(10), 1558–1574 (2012)
100. Nouy, A.: Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Appl. Mech. Eng.* **17**, 403–434 (2010)
101. Nouy, A., Maître, O.P.L.: Generalized spectral decomposition for stochastic nonlinear problems. *J. Comput. Phys.* **228**, 202–235 (2009)
102. Novak, E., Ritter, K.: High dimensional integration of smooth functions over cubes. *Numer. Math.* **75**, 79–97 (1996)
103. Novati, P.: Solving linear initial value problems by Faber polynomials. *Numer. Linear Algebra Appl.* **10**, 247–270 (2003)

104. Nozaki, D., Mar, D.J., Grigg, P., Collins, J.J.: Effects of colored noise on stochastic resonance in sensory neurons. *Phys. Rev. Lett.* **82**(11), 2402–2405 (1999)
105. O'Brien, E.E.: The probability density function (pdf) approach to reacting turbulent flows. In: *Topics in Applied Physics. Turbulent Reacting Flows*, vol. 44, pp. 185–218. Springer, Berlin/New York (1980)
106. Orszag, S.A., Bissonnette, L.R.: Dynamical properties of truncated Wiener-Hermite expansions. *Phys. Fluids* **10**(12), 2603–2613 (1967)
107. Pereverzev, A., Bittner, E.R.: Time-convolutionless master equation for mesoscopic electron-phonon systems. *J. Chem. Phys.* **125**, 144107(1–7) (2006)
108. Pesquera, L., Rodriguez, M.A., Santos, E.: Path integrals for non-Markovian processes. *Phys. Lett.* **94**(6–7), 287–289 (1983)
109. Pope, S.B.: A Monte Carlo method for the PDF equations of turbulent reactive flow. *Combust. Sci. Technol.* **25**, 159–174 (1981)
110. Pope, S.B.: Lagrangian PDF methods for turbulent flows. *Ann. Rev. Fluid Mech.* **26**, 23–63 (1994)
111. Pope, S.B.: Simple models of turbulent flows. *Phys. Fluids* **23**(1), 011301(1–20) (2011)
112. Rabitz, H., Aliş ÖF, Shorter, J., Shim, K.: Efficient input–output model representations. *Comput. Phys. Commun.* **117**(1–2), 11–20 (1999)
113. Remacle, J.F., Flaherty, J.E., Shephard, M.S.: An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**(1), 53–72 (2003)
114. Remacle, J.F., Flaherty, J.E., Shephard, M.S.: An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**(1), 53–72 (2003)
115. Richter, M., Knorr, A.: A time convolution less density matrix approach to the nonlinear optical response of a coupled system-bath complex. *Ann. Phys.* **325**, 711–747 (2010)
116. Rjasanow, S., Wagner, W.: *Stochastic Numerics for the Boltzmann Equation*. Springer, Berlin/New York (2004)
117. Sapsis, T.P., Lermusiaux, P.F.J.: Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D* **238**(23–24), 2347–2360 (2009)
118. Snook, I.: *The Langevin and Generalised Langevin Approach to the Dynamics of Atomic, Polymeric and Colloidal Systems*, 1st edn. Elsevier, Amsterdam/Boston (2007)
119. Stinis, P.: A comparative study of two stochastic mode reduction methods. *Physica D* **213**, 197–213 (2006)
120. Stinis, P.: Mori-Zwanzig-reduced models for systems without scale separation. *Proc. R. Soc. A* **471**, 20140446(1–13) (2015)
121. Stratonovich, R.L.: *Topics in the Theory of Random Noise*, vols. 1 and 2. Gordon and Breach, New York (1967)
122. Suzuki, M.: Decomposition formulas of exponential operators and Lie exponentials with applications to quantum mechanics and statistical physics. *J. Math. Phys.* **26**(4), 601–612 (1985)
123. Suzuki, M.: General decomposition theory of ordered exponentials. *Proc. Jpn. Acad. B* **69**(7), 161–166 (1993)
124. Suzuki, M.: Convergence of general decompositions of exponential operators. *Commun. Math. Phys.* **163**, 491–508 (1994)
125. Tartakovsky, D.M., Broyda, S.: PDF equations for advective-reactive transport in heterogeneous porous media with uncertain properties. *J. Contam. Hydrol.* **120–121**, 129–140 (2011)
126. Terwiel, R.H.: Projection operator method applied to stochastic linear differential equations. *Physica* **74**, 248–265 (1974)
127. Thalhammer, M.: High-order exponential operator splitting methods for time-dependent Schrödinger equations. *SIAM J. Numer. Anal.* **46**(4), 2022–2038 (2008)
128. Turkington, B.: An optimization principle for deriving nonequilibrium statistical models of Hamiltonian dynamics. *J. Stat. Phys.* **152**, 569–597 (2013)

129. Valino, L.: A field Monte Carlo formulation for calculating the probability density function of a single scalar in a turbulent flow. *Flow Turbul. Combust.* **60**(2), 157–172 (1998)
130. Venkatesh, T.G., Patnaik, L.M.: Effective Fokker-Planck equation: Path-integral formalism. *Phys. Rev. E* **48**(4), 2402–2412 (1993)
131. Venturi, D.: On proper orthogonal decomposition of randomly perturbed fields with applications to flow past a cylinder and natural convection over a horizontal plate. *J. Fluid Mech.* **559**, 215–254 (2006)
132. Venturi, D.: A fully symmetric nonlinear biorthogonal decomposition theory for random fields. *Physica D* **240**(4–5), 415–425 (2011)
133. Venturi, D.: Conjugate flow action functionals. *J. Math. Phys.* **54**, 113502(1–19) (2013)
134. Venturi, D., Karniadakis, G.E.: Differential constraints for the probability density function of stochastic solutions to the wave equation. *Int. J. Uncertain. Quantif.* **2**(3), 131–150 (2012)
135. Venturi, D., Karniadakis, G.E.: New evolution equations for the joint response-excitation probability density function of stochastic solutions to first-order nonlinear PDEs. *J. Comput. Phys.* **231**, 7450–7474 (2012)
136. Venturi, D., Karniadakis, G.E.: Convolutionless Nakajima-Zwanzig equations for stochastic analysis in nonlinear dynamical systems. *Proc. R. Soc. A* **470**(2166), 1–20 (2014)
137. Venturi, D., Wan, X., Karniadakis, G.E.: Stochastic low-dimensional modelling of a random laminar wake past a circular cylinder. *J. Fluid Mech.* **606**, 339–367 (2008)
138. Venturi, D., Wan, X., Karniadakis, G.E.: Stochastic bifurcation analysis of Rayleigh-Bénard convection. *J. Fluid Mech.* **650**, 391–413 (2010)
139. Venturi, D., Choi, M., Karniadakis, G.E.: Supercritical quasi-conduction states in stochastic Rayleigh-Bénard convection. *Int. J. Heat Mass Transf.* **55**(13–14), 3732–3743 (2012)
140. Venturi, D., Sapsis, T.P., Cho, H., Karniadakis, G.E.: A computable evolution equation for the joint response-excitation probability density function of stochastic dynamical systems. *Proc. R. Soc. A* **468**(2139), 759–783 (2012)
141. Venturi, D., Tartakovsky, D.M., Tartakovsky, A.M., Karniadakis, G.E.: Exact PDF equations and closure approximations for advective-reactive transport. *J. Comput. Phys.* **243**, 323–343 (2013)
142. Villani, C.: A review of mathematical topics in collisional kinetic theory. In: Friedlander, S., Serre, D. (eds.) *Handbook of mathematical fluid dynamics*, Vol I, North-Holland, Amsterdam, pp 73–258 (2002)
143. Viswanath, D.: The fractal property of the lorentz attractor. *Physica D* **190**, 115–128 (2004)
144. Wan, X., Karniadakis, G.E.: An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.* **209**(2), 617–642 (2005)
145. Wan, X., Karniadakis, G.E.: Long-term behavior of polynomial chaos in stochastic flow simulations. *Comput. Methods Appl. Mech. Eng.* **195**, 5582–5596 (2006)
146. Wan, X., Karniadakis, G.E.: Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.* **28**(3), 901–928 (2006)
147. Wang, C.J.: Effects of colored noise on stochastic resonance in a tumor cell growth system. *Phys. Scr.* **80**, 065004 (5pp) (2009)
148. Wei, J., Norman, E.: Lie algebraic solutions of linear differential equations. *J. Math. Phys.* **4**(4), 575–581 (1963)
149. Weinberg, S.: *The Quantum Theory of Fields*, vol. I. Cambridge University Press, Cambridge (2002)
150. Wiebe, N., Berry, D., Høyer, P., Sanders, B.C.: Higher-order decompositions of ordered operator exponentials. *J. Phys. A: Math. Theor.* **43**, 065203(1–20) (2010)
151. Wilcox, R.M.: Exponential operators and parameter differentiation in quantum physics. *J. Math. Phys.* **8**, 399–407 (1967)
152. Wilczek, M., Daitche, A., Friedrich, R.: On the velocity distribution in homogeneous isotropic turbulence: correlations and deviations from Gaussianity. *J. Fluid Mech.* **676**, 191–217 (2011)
153. Wio, H.S., Colet, P., San Miguel M, Pesquera, L., Rodríguez, M.A.: Path-integral formulation for stochastic processes driven by colored noise. *Phys. Rev. A* **40**(12), 7312–7324 (1989)

154. Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
155. Xiu, D., Karniadakis, G.E.: Modeling uncertainty in flow simulations via generalized polynomial chaos. *J. Comput. Phys.* **187**, 137–167 (2003)
156. Yang, Y., Shu, C.W.: Discontinuous Galerkin method for hyperbolic equations involving δ -singularities: negative-order norm error estimate and applications. *Numer. Math.* **124**, 753–781 (2013)
157. Yoshimoto, Y., Kinefuchi, I., Mima, T., Fukushima, A., Tokumasu, T., Takagi, S.: Bottom-up construction of interaction models of non-Markovian dissipative particle dynamics. *Phys. Rev. E* **88**, 043305(1–12) (2013)
158. Zwanzig, R.: Ensemble methods in the theory of irreversibility. *J. Chem. Phys.* **33**(5), 1338–1341 (1960)
159. Zwanzig, R.: Memory effects in irreversible thermodynamics. *Phys. Rev.* **124**, 983–992 (1961)

James L. Beck and Konstantin M. Zuev

Abstract

Rare events are events that are expected to occur infrequently or, more technically, those that have low probabilities (say, order of 10^{-3} or less) of occurring according to a probability model. In the context of uncertainty quantification, the rare events often correspond to failure of systems designed for high reliability, meaning that the system performance fails to meet some design or operation specifications. As reviewed in this section, computation of such rare-event probabilities is challenging. Analytical solutions are usually not available for nontrivial problems, and standard Monte Carlo simulation is computationally inefficient. Therefore, much research effort has focused on developing advanced stochastic simulation methods that are more efficient. In this section, we address the problem of estimating rare-event probabilities by Monte Carlo simulation, importance sampling, and subset simulation for highly reliable dynamic systems.

Keywords

Rare-event simulation • Dynamic system reliability • Monte carlo simulation • Subset simulation • Importance sampling • Splitting

Contents

1	Introduction	1076
1.1	Mathematical Formulation of Problem	1076
2	Standard Monte Carlo Simulation	1078
3	Importance Sampling	1081
4	Subset Simulation	1084
5	Splitting	1089
6	Illustrative Example	1091
7	Conclusion	1096
	References	1098

J.L. Beck (✉) • K.M. Zuev

Department of Computing and Mathematical Sciences, California Institute of Technology,
Pasadena, CA, USA

e-mail: jimbeck@caltech.edu; kostia@caltech.edu

1 Introduction

We focus on rare-event simulation for addressing reliability problems corresponding to dynamic systems. To compute the rare-event (failure) probability for a dynamic system, both input (excitation) and modeling uncertainties should be quantified and propagated. Therefore, a probability model must be chosen to describe the uncertainty in the future input for the system, and then a chosen deterministic or stochastic system model is used, preferably in conjunction with a probability model describing the associated modeling uncertainties, to propagate these uncertainties. These input and system models define a probabilistic description of the system output (response). For example, the problem of interest might be to compute the small failure probability for a highly reliable dynamic system such as a bridge or building under uncertain future earthquake excitation, or for an aircraft under uncertain excitation by turbulence, using a finite-element structural model to approximate the dynamics of the system. This model will usually be subject to both parametric uncertainty (what values of the model parameters best represent the behavior of the system?) and nonparametric modeling uncertainty (what are the effects of the aspects of the system behavior not captured by the dynamic model?). The treatment of input uncertainty has a long history in dynamic reliability theory and random vibrations, now more commonly called stochastic dynamics, but the treatment of modeling uncertainty is more recent.

Usually the dynamic model of the system is represented by a time-dependent BVP (boundary-value problem) involving PDEs (partial differential equations) or by a set of coupled ODEs (ordinary differential equations). Typically the failure event is defined as any one of a set of performance quantities of interest exceeding its specified threshold over some time interval. This is the so-called first-passage problem. This challenging problem is characterized by a lack of analytical solutions, even for the simplest case of a single-degree-of-freedom linear oscillator subject to excitation that is modeled as a Gaussian process. Approximate analytical methods exist that are usually limited in scope, and their accuracy is difficult to assess in a given application [43, 51]. Semi-analytical methods from structural reliability theory such as FORM and SORM (first- and second-order reliability methods) [20, 43] cannot be applied directly to the first-passage problem and are inapplicable, anyway, because of the high-dimensional nature of the discrete-time input history [32, 53]. Standard Monte Carlo simulation has general applicability, but it is computationally very inefficient because of the low failure probabilities. As a consequence, advanced stochastic simulation schemes are needed.

1.1 Mathematical Formulation of Problem

We assume that initially there is a continuous-time deterministic model of the real dynamic system that consists of a state-space model with a finite-dimensional state $X(t) \in \mathbb{R}^n$ at time t , and this is converted to a discrete-time state-space model using a numerical time-stepping method to give

$$X(t+1) = f(X(t), U(t), t), \quad X(t) \in \mathbb{R}^n, \quad U(t) \in \mathbb{R}^m, \quad t = 0, \dots, T \quad (30.1)$$

where $U(t) \in \mathbb{R}^m$ is the input at discrete time t .

If the original model consists of a BVP with PDEs describing a response $u(x, t)$ where $x \in \mathbb{R}^d$, then we assume that a finite set of basis functions $\{\phi_1(x), \dots, \phi_n(x)\}$ is chosen (e.g., global bases such as Fourier and Hermite polynomials or localized ones such as finite-element interpolation functions) so that the solution is well approximated by

$$u(x, t) \approx \sum_{i=1}^n X_i(t) \phi_i(x) \quad (30.2)$$

Then a numerical method is applied to the BVP PDEs to establish time-dependent equations for the vector of coefficients $X(t) = [X_1(t), \dots, X_n(t)]$ so that the standard state-space equation in (30.1) still applies. For example, for a finite-element model of a structural system, $\{\phi_1(x), \dots, \phi_n(x)\}$ would be local interpolation functions over the elements. Then, expressing the BVP in weak form, a weighted residual or Galerkin method could be applied to give a state-space equation for the vector of coefficients $X(t)$ [27].

Suppose that a positive scalar performance function $g(X(t))$ is a quantity of interest and that the rare-event \mathcal{E} of concern is that $g(X(t))$ exceeds a threshold b over some discrete-time interval $t = 0, \dots, T$:

$$\mathcal{E} = \left\{ U = (U(0), \dots, U(T)) : \max_{t=0, \dots, T} g(X(t)) > b \right\} \subset \mathbb{R}^{m \times (T+1)} \quad (30.3)$$

where $X(t)$ satisfies (30.1). The performance function $g(X(t))$ may involve exceedance of multiple performance quantities of interest $\{g_k(X(t)) : k = 1, \dots, K\}$ above their corresponding thresholds $\{a_k\}$. This can be accomplished by aggregating them using the *max* and *min* operators in an appropriate combination on the set of g_k 's; for example, for a *pure series* failure criterion, where the threshold exceedance of *any* a_k represents failure, one takes the aggregate performance failure criterion as $g(X(t)) = \max\{g_k(X(t))/a_k : k = 1, \dots, K\} > 1$, while for a *pure parallel* failure criterion, where *all* of the g_k must exceed their thresholds before failure is considered to have occurred, one takes the aggregate performance failure criterion as $g(X(t)) = \min\{g_k(X(t))/a_k : k = 1, \dots, K\} > 1$.

If the uncertainty in the input time history vector $U = [U(0), \dots, U(T)] \in \mathbb{R}^D$ ($D = m \times (T + 1)$) is quantified by a probability distribution for U that has a PDF (probability density function) $p(u)$ with respect to Lebesgue integration over \mathbb{R}^D , then the rare-event probability is given by

$$p_{\mathcal{E}} = \mathbb{P}(U \in \mathcal{E}) = \int_{\mathcal{E}} p(u) du \quad (30.4)$$

The PDF $p(u)$ is assumed to be readily sampled. Although *direct sampling* from a high-dimensional PDF is not possible in most cases, multidimensional Gaussians

are an exception because the Gaussian vector can be readily transformed so that the components are independent and the PDF is a product of one-dimensional Gaussian PDFs. In many applications, the discrete-time stochastic input history is modeled by running discrete-time Gaussian white noise through a digital filter to shape its spectrum in the frequency domain and then multiplying the filtered sequence by an envelope function to shape it in the time domain, if it is nonstationary.

The model in (30.1) may also depend on uncertain parameters $\theta \in \Theta \subset \mathbb{R}^p$ which includes the initial values $X(0)$ if they are uncertain. Then a prior PDF $p(\theta)$ may be chosen to quantify the uncertainty in the value of vector θ . Some of the parameters may characterize the PDF for input U which can then be denoted $p(u|\theta)$. It is convenient to redefine vector U to also include θ ; then the new PDF $p(u)$ is $p(u|\theta)p(\theta)$ in terms of the previous PDFs. We assume that model parameter uncertainty is incorporated in this way, so the basic equations remain the same as (30.1), (30.3), and (30.4). When model uncertainty is incorporated, the calculated $p_{\mathcal{E}}$ has been referred to as the robust rare-event probability [10, 40], meaning robust to model uncertainty, as in robust control theory.

2 Standard Monte Carlo Simulation

The standard *Monte Carlo simulation* method (MCS) is one of the most robust and straightforward ways to simulate rare events and estimate their probabilities. The method was originally developed in [37] for solving problems in mathematical physics. Since then MCS has been used in many applications in physics, statistics, computer science, and engineering, and currently it lays at the heart of all random sampling-based techniques [35, 44].

The basic idea behind MCS is to observe that the probability in (30.4) can be written as an expectation:

$$p_{\mathcal{E}} = \int_{\mathbb{R}^D} I_{\mathcal{E}}(u) p(u) du = \mathbb{E}_p[I_{\mathcal{E}}] \quad (30.5)$$

where $I_{\mathcal{E}}$ is the indicator function of \mathcal{E} , that is, $I_{\mathcal{E}}(u) = 1$ if $u \in \mathcal{E}$ and $I_{\mathcal{E}}(u) = 0$ otherwise, and $D = m \times (T + 1)$ is the dimension of the integral. Recall that the strong law of large numbers [45] states that if U_1, \dots, U_N are independent and identically distributed (i.i.d.) samples of vector U drawn from the distribution $p(u)$, then for any function $h(u)$ with finite mean $\mathbb{E}_p[h(u)]$, the sample average $\frac{1}{N} \sum_{i=1}^N h(U_i)$ converges to the true value $\mathbb{E}_p[h(u)]$ as $N \rightarrow \infty$ almost surely (i.e., with probability 1). Therefore, setting $h(u) = I_{\mathcal{E}}(u)$, the probability in (30.5) can be estimated as follows:

$$p_{\mathcal{E}} \approx p_{\mathcal{E}}^{MCS} = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(U_i) \quad (30.6)$$

It is straightforward to show that $p_{\mathcal{E}}^{MCS}$ is an *unbiased* estimator of $p_{\mathcal{E}}$ with mean and variance:

$$\begin{aligned}\mathbb{E}_p[p_{\mathcal{E}}^{MCS}] &= \mathbb{E}_p \left[\frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(U_i) \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_p[I_{\mathcal{E}}] = p_{\mathcal{E}} \\ \text{Var}_p[p_{\mathcal{E}}^{MCS}] &= \text{Var}_p \left[\frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(U_i) \right] = \frac{1}{N^2} \sum_{i=1}^N \text{Var}_p[I_{\mathcal{E}}] = \frac{p_{\mathcal{E}}(1-p_{\mathcal{E}})}{N}\end{aligned}\tag{30.7}$$

Furthermore, by the central limit theorem [45], as $N \rightarrow \infty$, $p_{\mathcal{E}}^{MCS}$ is distributed asymptotically as Gaussian with this mean and variance.

Frequentist interpretation of MCS: The frequentist interpretation of MCS focuses on the forward problem, arguing that if N is large so that the variance of $p_{\mathcal{E}}^{MCS}$ is relatively small, then the value $\hat{p}_{\mathcal{E}}^{MCS}$ based on (30.6) for a specific set of N samples $\{\hat{U}_1, \dots, \hat{U}_N\}$ drawn from $p(u)$ should be close to the mean $p_{\mathcal{E}}$ of $p_{\mathcal{E}}^{MCS}$. The sample mean estimate $\hat{p}_{\mathcal{E}}^{MCS}$ is very intuitive and, in fact, simply reflects the frequentist definition of probability: $\hat{p}_{\mathcal{E}}^{MCS}$ is the ratio between the number of trials where the event \mathcal{E} occurred, $\hat{N}_{\mathcal{E}} = \sum_{i=1}^N I_{\mathcal{E}}(\hat{U}_i)$, and the total number of trials N .

Bayesian interpretation of MCS: The same MCS estimate $\hat{p}_{\mathcal{E}}^{MCS}$ has a simple Bayesian interpretation (e.g., [56]), which focuses on the inverse problem for the specific set of N samples $\{\hat{U}_1, \dots, \hat{U}_N\}$ drawn from $p(u)$. Following the Bayesian approach [26], the unknown probability $p_{\mathcal{E}}$ is considered as a stochastic variable whose value in $[0, 1]$ is uncertain. The *Principle of Maximum Entropy* [25] leads to the uniform prior distribution for $p_{\mathcal{E}}$, $p(p_{\mathcal{E}}) = 1$, $0 \leq p_{\mathcal{E}} \leq 1$, which implies that all values are taken as equally plausible a priori. Since samples U_1, \dots, U_N are i.i.d., the binary sequence $I_{\mathcal{E}}(U_1), \dots, I_{\mathcal{E}}(U_N)$ is a sequence of Bernoulli trials, and so for the forward problem, $N_{\mathcal{E}}$ is distributed according to the binomial distribution with parameters N and $p_{\mathcal{E}}$, $N_{\mathcal{E}} \sim \text{Bin}(N, p_{\mathcal{E}})$. Therefore, for the set of N samples, the likelihood function is $p(\hat{N}_{\mathcal{E}}|p_{\mathcal{E}}, N) = \binom{N}{\hat{N}_{\mathcal{E}}} p_{\mathcal{E}}^{\hat{N}_{\mathcal{E}}} (1-p_{\mathcal{E}})^{N-\hat{N}_{\mathcal{E}}}$. Using *Bayes' theorem*, the posterior distribution for $p_{\mathcal{E}}$, $p(p_{\mathcal{E}}|\hat{N}_{\mathcal{E}}, N) \propto p(p_{\mathcal{E}})p(\hat{N}_{\mathcal{E}}|p_{\mathcal{E}}, N)$, is therefore the beta distribution $\text{Beta}(\hat{N}_{\mathcal{E}} + 1, N - \hat{N}_{\mathcal{E}} + 1)$, i.e.,

$$p(p_{\mathcal{E}}|\hat{N}_{\mathcal{E}}, N) = \frac{p_{\mathcal{E}}^{\hat{N}_{\mathcal{E}}} (1-p_{\mathcal{E}})^{N-\hat{N}_{\mathcal{E}}}}{B(\hat{N}_{\mathcal{E}} + 1, N - \hat{N}_{\mathcal{E}} + 1)}\tag{30.8}$$

where the beta function B is the normalizing constant that equals $(N + 1)! / (\hat{N}_{\mathcal{E}}!(N - \hat{N}_{\mathcal{E}})!)$ here. The MCS estimate is the *maximum a posteriori (MAP) estimate*, which is the mode of the posterior distribution (30.8) and therefore the most probable value of $p_{\mathcal{E}}$ a posteriori:

$$\hat{p}_{\mathcal{E}}^{MCS} = \frac{\hat{N}_{\mathcal{E}}}{N}\tag{30.9}$$

Notice that the posterior PDF in (30.8) gives a complete description of the uncertainty in the value of $p_{\mathcal{E}}$ based on the specific set of N samples of U drawn from $p(u)$. The posterior distribution in (30.8) is in fact the original Bayes' result [9], although Bayes' theorem was developed in full generality by Laplace [34].

The standard MCS method for estimating the probability in (30.4) is summarized in the following pseudo-code.

Monte Carlo Simulation

Input:

▷ N , total number of samples.

Algorithm:

Set $N_{\mathcal{E}} = 0$, number of trials where the event \mathcal{E} occurred.

for $i = 1, \dots, N$ **do**

 Sample the input excitation $U_i = (U_i(0), \dots, U_i(T)) \sim p(u)$.

 Compute the system trajectory $X_i = (X_i(0), \dots, X_i(T))$
 using the system model (30.1) with $U(t) = U_i(t)$.

if $\max_{t=0, \dots, T} g(X_i(t)) > b$

$N_{\mathcal{E}} \leftarrow N_{\mathcal{E}} + 1$

end if

end for

Output:

► $\hat{p}_{\mathcal{E}}^{MCS} = \frac{N_{\mathcal{E}}}{N}$, MCS estimate of $p_{\mathcal{E}}$

► $p(p_{\mathcal{E}}|N_{\mathcal{E}}, N) = \frac{p_{\mathcal{E}}^{N_{\mathcal{E}}} (1-p_{\mathcal{E}})^{N-N_{\mathcal{E}}}}{B(N_{\mathcal{E}}+1, N-N_{\mathcal{E}}+1)}$, posterior PDF of $p_{\mathcal{E}}$

Assessment of accuracy of MCS estimate: For the frequentist interpretation, the coefficient of variation (c.o.v.) for the estimator $\hat{p}_{\mathcal{E}}^{MCS}$ given by (30.6), conditional on $p_{\mathcal{E}}$ and N , is given by (30.7):

$$\delta(\hat{p}_{\mathcal{E}}^{MCS}|p_{\mathcal{E}}, N) = \sqrt{\frac{\text{Var}_p[\hat{p}_{\mathcal{E}}^{MCS}]}{\mathbb{E}_p[\hat{p}_{\mathcal{E}}^{MCS}]}} = \sqrt{\frac{1-p_{\mathcal{E}}}{Np_{\mathcal{E}}}} \quad (30.10)$$

This can be approximated by replacing $p_{\mathcal{E}}$ by the estimate $\hat{p}_{\mathcal{E}}^{MCS} = \hat{N}_{\mathcal{E}}/N$ for a given set of N samples $\{\hat{U}_1, \dots, \hat{U}_N\}$:

$$\delta(\hat{p}_{\mathcal{E}}^{MCS}|p_{\mathcal{E}}, N) \approx \sqrt{\frac{1-\hat{p}_{\mathcal{E}}^{MCS}}{N\hat{p}_{\mathcal{E}}^{MCS}}} \triangleq \hat{\delta}_N^{MCS} \quad (30.11)$$

For the Bayesian interpretation, the posterior c.o.v. for the stochastic variable $p_{\mathcal{E}}$, conditional on the set of N samples, follows from (30.8):

$$\delta(p_{\mathcal{E}}|\widehat{N}_{\mathcal{E}}, N) = \frac{\sqrt{\text{Var}[p_{\mathcal{E}}|\widehat{N}_{\mathcal{E}}, N]}}{\mathbb{E}[p_{\mathcal{E}}|\widehat{N}_{\mathcal{E}}, N]} = \frac{\sqrt{1 - \frac{\widehat{N}_{\mathcal{E}}+1}{N+2}}}{\sqrt{(N+3)\left(\frac{\widehat{N}_{\mathcal{E}}+1}{N+2}\right)}} \xrightarrow{\text{as } N \rightarrow \infty} \sqrt{\frac{1 - \hat{p}_{\mathcal{E}}^{MCS}}{N \hat{p}_{\mathcal{E}}^{MCS}}} = \hat{\delta}_N^{MCS}$$
(30.12)

as $N \rightarrow \infty$. Therefore, the same expression $\hat{\delta}_N^{MCS}$ can be used to assess the accuracy of the MCS estimate, even though the two c.o.v.s have distinct interpretations.

The approximation $\hat{\delta}_N^{MCS}$ for the two c.o.v.s reveals both the main advantage of the standard MCS method and its main drawback. The main strength of MCS, which makes it very robust, is that its accuracy does not depend on the geometry of the domain $\mathcal{E} \subset \mathbb{R}^D$ and its dimension D . As long as an algorithm for generating i.i.d. samples from $p(u)$ is available, MCS, unlike many other methods (e.g., numerical integration), does not suffer from the “curse of dimensionality.” Moreover, an irregular, or even fractal-like, shape of \mathcal{E} will not affect the accuracy of MCS.

On the other hand, the serious drawback of MCS is that this method is not computationally efficient in estimating the *small probabilities* $p_{\mathcal{E}}$ corresponding to *rare events*, where from (30.10),

$$\delta(p_{\mathcal{E}}^{MCS}|p_{\mathcal{E}}, N) \approx \frac{1}{\sqrt{N p_{\mathcal{E}}}} \quad (30.13)$$

Therefore, to achieve a prescribed level of accuracy $\delta < 1$, the required total number of samples is $N = (p_{\mathcal{E}}\delta^2)^{-1} \gg 1$. For each sampled excitation U_i , a system analysis – usually computationally very intensive – is required to compute the corresponding system trajectory X_i and to check whether U_i belongs to \mathcal{E} . This makes MCS excessively costly and inapplicable for generating rare events and estimating their small probabilities. Nevertheless, essentially all sampling-based methods for estimation of rare-event probability are either based on MCS (e.g., importance sampling) or have it as a part of the algorithm (e.g., subset simulation).

3 Importance Sampling

The *importance sampling* (IS) method belongs to the class of *variance reduction techniques* that aim to increase the accuracy of the estimates by constructing (sometimes biased) estimators with a smaller variance [1, 22]. It seems it was first proposed in [29], soon after the standard MCS method appeared.

The inefficiency of MCS for rare-event estimation stems from the fact that most of the generated samples $U_i \sim p(u)$ do not belong to \mathcal{E} so that the vast majority of the terms in the sum (30.6) are zero and only very few (if any) are equal to one. The basic idea of IS is to make use of the information available about the rare-event \mathcal{E} to generate samples that lie more frequently in \mathcal{E} or in the *important region* $\tilde{\mathcal{E}} \subset \mathcal{E}$ that accounts for most of the probability content in (30.4). Rather than estimating $p_{\mathcal{E}}$ as an average of many 0's and very few 1's like in $\hat{p}_{\mathcal{E}}^{MCS}$, IS seeks to reduce the

variance by constructing an estimator of the form $p_{\mathcal{E}}^{IS} = \frac{1}{N} \sum_{i=1}^{N'} w_i$, where N' is an appreciable fraction of N and the w_i are small but not zero, ideally of the same order as the target probability, $w_i \approx p_{\mathcal{E}}$.

Specifically, for an appropriate PDF $q(u)$ on the excitation space \mathbb{R}^D , the integral in (30.5) can be rewritten as follows:

$$p_{\mathcal{E}} = \int_{\mathbb{R}^D} I_{\mathcal{E}}(u) p(u) du = \int_{\mathbb{R}^D} \frac{I_{\mathcal{E}}(u) p(u)}{q(u)} q(u) du = \mathbb{E}_q \left[\frac{I_{\mathcal{E}} p}{q} \right] \quad (30.14)$$

The IS estimator is now constructed similarly to (30.6) by utilizing the law of large numbers:

$$p_{\mathcal{E}} \approx p_{\mathcal{E}}^{IS} = \frac{1}{N} \sum_{i=1}^N \frac{I_{\mathcal{E}}(U_i) p(U_i)}{q(U_i)} = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(U_i) w(U_i) \quad (30.15)$$

where U_1, \dots, U_N are i.i.d. samples from $q(u)$, called the *importance sampling density* (ISD), and $w(U_i) = \frac{p(U_i)}{q(U_i)}$ is the *importance weight* of sample U_i .

The IS estimator $p_{\mathcal{E}}^{IS}$ converges almost surely as $N \rightarrow \infty$ to $p_{\mathcal{E}}$ by the strong law of large numbers, provided that the support of $q(u)$, i.e., the domain in \mathbb{R}^D where $q(u) > 0$, contains the support of $I_{\mathcal{E}}(u)p(u)$. Intuitively, the latter condition guarantees that all points of \mathcal{E} that can be generated by sampling from the original PDF $p(u)$ can also be generated by sampling from the ISD $q(u)$. Note that if $q(u) = p(u)$, then $w(U_i) = 1$ and IS simply reduces to MCS, $p_{\mathcal{E}}^{MCS} = p_{\mathcal{E}}^{IS}$. By choosing the ISD $q(u)$ appropriately, IS aims to obtain an estimator with a smaller variance.

The IS estimator $p_{\mathcal{E}}^{IS}$ is also *unbiased* with mean and variance:

$$\begin{aligned} \mathbb{E}_q[p_{\mathcal{E}}^{IS}] &= \mathbb{E}_q \left[\frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(U_i) w(U_i) \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_q \left[\frac{I_{\mathcal{E}} p}{q} \right] = p_{\mathcal{E}} \\ \text{Var}_q[p_{\mathcal{E}}^{IS}] &= \frac{1}{N^2} \sum_{i=1}^N \text{Var}_q \left[\frac{I_{\mathcal{E}} p}{q} \right] = \frac{1}{N} \left(\mathbb{E}_q \left[\frac{I_{\mathcal{E}} p^2}{q^2} \right] - p_{\mathcal{E}}^2 \right) \end{aligned} \quad (30.16)$$

The IS method is summarized in the following pseudo-code.

The most important task in applying IS for estimating small probabilities of rare events is the construction of the ISD, since the accuracy of $\hat{p}_{\mathcal{E}}^{IS}$ depends critically on $q(u)$. If the ISD is “good,” then one can get great improvement in efficiency over standard MCS. If, however, the ISD is chosen inappropriately so that, for instance, $N_{\mathcal{E}} = 0$ or the importance weights have a large variation, then IS will yield a very poor estimate. Both scenarios are demonstrated below in Sect. 6.

It is straightforward to show that the *optimal* ISD, which minimizes the variance in (30.16), is simply the original PDF $p(u)$ conditional on the domain \mathcal{E} :

Importance Sampling

Input:

- ▷ N , total number of samples.
- ▷ $q(u)$, importance sampling density.

Algorithm:

```

Set  $j = 0$ , counter for the number of samples in  $\mathcal{E}$ .
for  $i = 1, \dots, N$  do
    Sample the input excitation  $U_i = (U_i(0), \dots, U_i(T)) \sim q(u)$ .
    Compute the system trajectory  $X_i = (X_i(0), \dots, X_i(T))$ 
    using the system model (30.1) with  $U(t) = U_i(t)$ .
    if  $\max_{t=0,\dots,T} g(X_i(t)) > b$ 
         $j \leftarrow j + 1$ 
        Compute the importance weight of the  $j^{th}$  sample in  $\mathcal{E}$ ,  $w_j = \frac{p(U_i)}{q(U_i)}$ .
    end if
end for

```

$N_{\mathcal{E}} = j$, the total number of trials where the event \mathcal{E} occurred.

Output:

► $\hat{p}_{\mathcal{E}}^{IS} = \frac{\sum_{j=1}^{N_{\mathcal{E}}} w_j}{N}$, IS estimate of $p_{\mathcal{E}}$

$$q_0(u) = p(u|\mathcal{E}) = \frac{I_{\mathcal{E}}(u)p(u)}{p_{\mathcal{E}}} \quad (30.17)$$

Indeed, in this case, all generated sample excitations satisfy $U_i \in \mathcal{E}$, so their importance weights $w(U_i) = p_{\mathcal{E}}$, and the IS estimate $\hat{p}_{\mathcal{E}}^{IS} = p_{\mathcal{E}}$. Moreover, just one sample ($N = 1$) generated from $q_0(u)$ is enough to find the probability $p_{\mathcal{E}}$ exactly. Note, however, that this is a purely theoretical result since in practice sampling from the conditional distribution $p(u|\mathcal{E})$ is challenging, and, most importantly, it is impossible to compute $q_0(u)$: this would require the knowledge of $p_{\mathcal{E}}$, which is unknown. Nevertheless, this result indicates that the ISD $q(u)$ should be chosen as close to $q_0(u)$ as possible. In particular, most of the probability mass of $q(u)$ should be concentrated on \mathcal{E} . Based on these considerations, several ad hoc techniques for constructing ISDs have been developed, e.g., variance scaling and mean shifting [15].

In the special case of linear dynamics and Gaussian excitation, an extremely efficient algorithm for estimating the rare-event probability $p_{\mathcal{E}}$ in (30.4), referred to as ISEE (*importance sampling using elementary events*), has been presented [3]. The choice of the ISD exploits known information about each elementary event, defined as an outcrossing of the performance threshold b in (30.3) at a specific time $t \in \{0, \dots, T\}$. The c.o.v. of the ISEE estimator for N samples of U from $p(u)$ is given by

$$\delta_N^{ISEE} = \frac{\alpha}{\sqrt{N}} \quad (30.18)$$

where the proportionality constant α is close to 1, regardless of how small the value of $p_{\mathcal{E}}$. In fact, α decreases slightly as $p_{\mathcal{E}}$ decreases, exhibiting the opposite behavior to MCS.

In general, it is known that in many practical cases of rare-event estimation, it is difficult to construct a good ISD that leads to a low-variance IS estimator, especially if the dimension of the uncertain excitation space \mathbb{R}^D is large, as it is in dynamic reliability problems [5]. A geometric explanation as to why IS is often inefficient in high dimensions is given in [32]. Au [2] has presented an efficient IS method for estimating $p_{\mathcal{E}}$ in (30.4) for elastoplastic systems subject to Gaussian excitation. In recent years, substantial progress has been made by tailoring the *sequential importance sampling* (SIS) methods [35], where the ISD is iteratively refined, to rare-event problems. SIS and its modifications have been successfully used for estimating rare events in dynamic portfolio credit risk [19], structural reliability [33], and other areas.

4 Subset Simulation

The *subset simulation* (SS) method [4] is an advanced stochastic simulation method for estimating rare events which is based on *Markov chain Monte Carlo* (MCMC) [35, 44]. The basic idea behind SS is to represent a very small probability $p_{\mathcal{E}}$ of the rare-event \mathcal{E} as a product of larger probabilities of “more-frequent” events and then estimate these larger probabilities separately. To implement this idea, let

$$\mathbb{R}^D \equiv \mathcal{E}_0 \supset \mathcal{E}_1 \dots \supset \mathcal{E}_L \equiv \mathcal{E} \quad (30.19)$$

be a sequence of nested subsets of the uncertain excitation space starting from the entire space $\mathcal{E}_0 = \mathbb{R}^D$ and shrinking to the target rare-event $\mathcal{E}_L = \mathcal{E}$. By analogy with (30.3), subsets \mathcal{E}_i can be defined by relaxing the value of the critical threshold b :

$$\mathcal{E}_i = \left\{ U \in \mathbb{R}^D : \max_{t=0, \dots, T} g(X(t)) > b_i \right\} \quad (30.20)$$

where $b_1 < \dots < b_L = b$. In the actual implementation of SS, the number of subsets L and the values of intermediate thresholds $\{b_i\}$ are chosen adaptively.

Using the notion of conditional probability and exploiting the nesting of the subsets, the target probability $p_{\mathcal{E}}$ can be factorized as follows:

$$p_{\mathcal{E}} = \prod_{i=1}^L \mathbb{P}(\mathcal{E}_i | \mathcal{E}_{i-1}) \quad (30.21)$$

An important observation is that by choosing the intermediate thresholds $\{b_i\}$ appropriately, the conditional events $\{\mathcal{E}_i | \mathcal{E}_{i-1}\}$ can be made more frequent, and their

probabilities can be made large enough to be amenable to efficient estimation by MCS-like methods.

The first probability $\mathbb{P}(\mathcal{E}_1|\mathcal{E}_0) = \mathbb{P}(\mathcal{E}_1)$ can be readily estimated by standard MCS:

$$\mathbb{P}(\mathcal{E}_1) \approx \frac{1}{n} \sum_{j=1}^n I_{\mathcal{E}_1}(U_j), \quad (30.22)$$

where U_1, \dots, U_n are i.i.d. samples from $p(u)$. Estimating the remaining probabilities $\mathbb{P}(\mathcal{E}_i|\mathcal{E}_{i-1})$, $i \geq 2$, is more challenging since one needs to generate samples from the conditional distribution $p(u|\mathcal{E}_{i-1}) = \frac{I_{\mathcal{E}_{i-1}}(u)p(u)}{\mathbb{P}(\mathcal{E}_{i-1})}$, which, in general, is not a trivial task. Notice that a sample U from $p(u|\mathcal{E}_{i-1})$ is one drawn from $p(u)$ that lies in \mathcal{E}_{i-1} . However, it is not efficient to use MCS for generating samples from $p(u|\mathcal{E}_{i-1})$: sampling from $p(u)$ and accepting only those samples that belong to \mathcal{E}_{i-1} is computationally very expensive, especially at higher levels i .

In standard SS, samples from the conditional distribution $p(u|\mathcal{E}_{i-1})$ are generated by the *modified Metropolis algorithm* (MMA) [4] which belongs to the family of MCMC methods for sampling from complex probability distributions that are difficult to sample directly from [35, 44]. An alternative strategy – *splitting* – is described in the next section.

The MMA algorithm is a component-wise version of the original Metropolis algorithm [38]. It is specifically tailored for sampling from high-dimensional conditional distributions and works as follows. First, without loss of generality, assume that $p(u) = \prod_{k=1}^D p_k(u_k)$, i.e., components of U are independent. This assumption is indeed not a limitation, since in simulation one always starts from independent variables to generate correlated excitation histories U . Suppose further that some vector $U_1 \in \mathbb{R}^D$ is already distributed according to the target conditional distribution, $U_1 \sim p(u|\mathcal{E}_{i-1})$. MMA prescribes how to generate another vector $U_2 \sim p(u|\mathcal{E}_{i-1})$, and it consists of two steps:

1. Generate a “candidate” state V as follows: first, for each component $k = 1, \dots, D$ of V , sample $v(k)$ from the symmetric univariate *proposal distribution* $q_{k,i}(v|U_1(k))$ centered on the k^{th} component of U_1 , where symmetry means that $q_{k,i}(v|u) = q_{k,i}(u|v)$; then, compute the *acceptance ratio* $r_k = \frac{p_k(v(k))}{p_k(U_1(k))}$; and finally, set

$$V(k) = \begin{cases} v(k), & \text{with probability } \min\{1, r_k\} \\ U_1(k), & \text{with probability } 1 - \min\{1, r_k\} \end{cases} \quad (30.23)$$

2. Accept or reject the candidate state V :

$$U_2 = \begin{cases} V, & \text{if } V \in \mathcal{E}_{i-1} \\ U_1, & \text{if } V \notin \mathcal{E}_{i-1} \end{cases} \quad (30.24)$$

It can be shown that U_2 generated by MMA is indeed distributed according to the target conditional distribution $p(u|\mathcal{E}_{i-1})$ when U_1 is [4]. For a detailed discussion of MMA, the reader is referred to [56].

The procedure for generating conditional samples at level i is as follows. Starting from a “seed” $U_1 \sim p(u|\mathcal{E}_{i-1})$, one can now use MMA to generate a sequence of random vectors U_1, \dots, U_n , called a *Markov chain*, distributed according to $p(u|\mathcal{E}_{i-1})$. At each step, U_j is used to generate the next state U_{j+1} . Note that although these MCMC samples are identically distributed, they are clearly not independent: the correlation between successive samples is due to the proposal PDFs $\{q_{k,i}\}$ at level i that govern the generation of U_{j+1} from U_j . Nevertheless, U_1, \dots, U_n can still be used for statistical averaging as if they were i.i.d., although with certain reduction in efficiency [4]. In particular, similar to (30.22), the conditional probability $\mathbb{P}(\mathcal{E}_i|\mathcal{E}_{i-1})$ can be estimated as follows:

$$\mathbb{P}(\mathcal{E}_i|\mathcal{E}_{i-1}) \approx \frac{1}{n} \sum_{j=1}^n I_{\mathcal{E}_i}(U_j) \quad (30.25)$$

To obtain an estimator for the target probability $p_{\mathcal{E}}$, it remains to multiply the MCS (30.22) and MCMC (30.25) estimators of all factors in (30.21). In real applications, however, it is often difficult to rationally define the subsets $\{\mathcal{E}_i\}$ in advance, since it is not clear how to specify the values of the intermediate thresholds $\{b_i\}$. In SS, this is done adaptively. Specifically, let $U_1^{(0)}, \dots, U_n^{(0)}$ be the MCS samples from $p(u)$, $X_1^{(0)}, \dots, X_n^{(0)}$ be the corresponding trajectories from (30.1), and $G_j^{(0)} = \max_{t=0, \dots, T} g(X_j^{(0)}(t))$ be the resulting performance values. Assume that the sequence $\{G_j^{(0)}\}$ is ordered in a nonincreasing order, i.e., $G_1^{(0)} \geq \dots \geq G_n^{(0)}$, renumbering the samples where necessary. Define the first intermediate threshold b_1 as follows:

$$b_1 = \frac{G_{np_0}^{(0)} + G_{np_0+1}^{(0)}}{2} \quad (30.26)$$

where p_0 is a chosen probability satisfying $0 < p_0 < 1$. This choice of b_1 has two immediate consequences: first, the MCS estimate of $\mathbb{P}(\mathcal{E}_1)$ in (30.22) is exactly p_0 , and, second, $U_1^{(0)}, \dots, U_{np_0}^{(0)}$ not only belong to \mathcal{E}_1 but also are distributed according to the conditional distribution $p(u|\mathcal{E}_1)$. Each of these np_0 samples can now be used as mother seeds in MMA to generate $(\frac{1}{p_0} - 1)$ offspring, giving a total of n samples $U_1^{(1)}, \dots, U_n^{(1)} \sim p(u|\mathcal{E}_1)$. Since these seeds start in the stationary state $p(u|\mathcal{E}_1)$ of the Markov chain, this MCMC method gives *perfect sampling*, i.e., no wasteful burn-in period is needed. Similarly, b_2 is defined as

$$b_2 = \frac{G_{np_0}^{(1)} + G_{np_0+1}^{(1)}}{2} \quad (30.27)$$

where $\{G_j^{(1)}\}$ are the (ordered) performance values corresponding to excitations $\{U_j^{(1)}\}$. Again by construction, the estimate (30.25) gives $\mathbb{P}(\mathcal{E}_2|\mathcal{E}_1) \approx p_0$, and $U_1^{(1)}, \dots, U_{n^{(1)}}^{(1)} \sim p(u|\mathcal{E}_2)$. The SS method proceeds in this manner until the target rare-event \mathcal{E} is reached and is sufficiently sampled. All but the last factor in (30.21) are approximated by p_0 , and the last factor $\mathbb{P}(\mathcal{E}|\mathcal{E}_{L-1}) \approx \frac{n_{\mathcal{E}}}{n} \geq p_0$, where $n_{\mathcal{E}}$ is the number of samples in \mathcal{E} among $U_1^{(L-1)}, \dots, U_n^{(L-1)} \sim p(u|\mathcal{E}_{L-1})$. The method is more formally summarized in the following pseudo-code.

Subset Simulation

Input :

- ▷ n , number of samples per conditional level.
- ▷ p_0 , level probability; e.g. $p_0 = 0.1$
- ▷ $\{q_{k,i}\}$, proposal distributions; e.g. $q_{k,i}(v|u) = \mathcal{N}(v|u, \sigma_{k,i}^2)$

Algorithm :

```

Set  $i = 0$ , number of conditional level.
Set  $n_{\mathcal{E}}^{(0)} = 0$ , number of the MCS samples in  $\mathcal{E}$ .
Sample the input excitations  $U_1^{(0)}, \dots, U_n^{(0)} \sim p(u)$ .
Compute the corresponding trajectories  $X_1^{(0)}, \dots, X_n^{(0)}$ .
for  $j = 1, \dots, n$  do
    if  $G_j^{(0)} = \max_{t=0, \dots, T} g(X_j^{(0)}(t)) > b$  do
         $n_{\mathcal{E}}^{(0)} \leftarrow n_{\mathcal{E}}^{(0)} + 1$ 
    end if
end for
while  $n_{\mathcal{E}}^{(i)}/n < p_0$  do
     $i \leftarrow i + 1$ , a new subset  $\mathcal{E}_i$  is needed.
    Sort  $\{U_j^{(i-1)}\}$  so that  $G_1^{(i-1)} \geq G_2^{(i-1)} \geq \dots \geq G_n^{(i-1)}$ 
    Define the  $i^{\text{th}}$  intermediate threshold:  $b_i = (G_{n^{(i-1)}}^{(i-1)} + G_{n^{(i-1)}+1}^{(i-1)})/2$ 
    for  $j = 1, \dots, n^{(i)}$  do
        Using  $W_{j,1} = U_j^{(i-1)} \sim p(u|\mathcal{E}_i)$  as a seed, use MMA to generate
         $(\frac{1}{p_0} - 1)$  additional states of a Markov chain
         $W_{j,1}, \dots, W_{j,1/p_0} \sim p(u|\mathcal{E}_i)$ .
    end for
    Renumber:  $\{W_{j,s}\}_{j=1,s=1}^{n^{(i-1)}/p_0} \mapsto U_1^{(i)}, \dots, U_n^{(i)} \sim p(u|\mathcal{E}_i)$ 
    Compute the corresponding trajectories  $X_1^{(i)}, \dots, X_n^{(i)}$ 
    for  $j = 1, \dots, n$  do
        if  $G_j^{(i)} = \max_{t=0, \dots, T} g(X_j^{(i)}(t)) > b$  do
             $n_{\mathcal{E}}^{(i)} \leftarrow n_{\mathcal{E}}^{(i)} + 1$ 
        end if
    end for
end while
 $L = i + 1$ , number of levels, i.e. subsets  $\mathcal{E}_i$  in (30.19) and (30.20)
 $N = n + n(1 - p_0)(L - 1)$ , total number of samples.

```

Output :

► $\hat{p}_{\mathcal{E}}^{SS} = p_0^{L-1} \frac{n_{\mathcal{E}}^{(L-1)}}{n}$, SS estimate of p_F :

Implementation details of SS, in particular the choice of level probability p_0 and proposal distributions $\{q_k\}$, are thoroughly discussed in [56]. It has been confirmed that $p_0 = 0.1$ proposed in the original paper [4] is a nearly optimal value. The choice of $\{q_{k,i}\}$ is more delicate, since the efficiency of MMA strongly depends on the proposal PDF variances in a nontrivial way: proposal PDFs with both small and large variance tend to increase the correlation between successive samples, making statistical averaging in (30.25) less efficient. In general, finding the optimal variance of proposal distributions is a challenging task not only for MMA but also for almost all MCMC algorithms. Nevertheless, it has been found in many applications that using $q_{k,i}(v|u) = \mathcal{N}(v|u, \sigma_{k,i}^2)$, the Gaussian distribution with mean u and variance $\sigma_{k,i}^2$ yields good efficiency if $\sigma_{k,i}^2 = \sigma_0^2$ and $p(u)$ is a multi-dimensional Gaussian with all variances equal to σ_0^2 . For an adaptive strategy for choosing $\{q_{k,i}\}$, the reader is referred to [56]; for example, $\sigma_{k,i}^2 = \sigma_i^2$ can be chosen so that the observed average acceptance rate in MMA, based on a subset of samples at level i , lies in the interval [0.3, 0.5].

It can be shown [4, 7] that, given p_ε , p_0 , and the total number of samples N , the c.o.v. of the SS estimator p_ε^{SS} is given by

$$\delta^2(p_\varepsilon^{SS}|p_\varepsilon, p_0, N) = \frac{(1 + \gamma)(1 - p_0)}{Np_0(\ln p_0^{-1})^r} (\ln p_\varepsilon^{-1})^r \quad (30.28)$$

where $2 \leq r \leq 3$ and γ is approximately a constant that depends on the state correlation of the Markov chain at each level. Numerical experiments show that $r = 2$ gives a good approximation to the c.o.v. and that $\gamma \approx 3$ if the proposal variance σ_i^2 for each level is appropriately chosen [4, 7, 56]. It follows from (30.13) that $\delta_{MCS}^2 \propto p_\varepsilon^{-1}$ for MCS, while for SS, $\delta_{SS}^2 \propto (\ln p_\varepsilon^{-1})^r$. This drastically different scaling behavior of the c.o.v.'s with small p_ε directly exhibits the improvement in efficiency.

To compare an advanced stochastic simulation algorithm directly with MCS, which is always applicable (but not efficient) for rare-event estimation, [11] introduced the relative computation efficiency of an algorithm, η_A , which is defined as the ratio of the number of samples N_{MCS} required by MCS to the number of samples N_A required by the algorithm for the same c.o.v. δ . The *relative efficiency* of SS is then

$$\eta_{SS} = \frac{N_{MCS}}{N_{SS}} = \frac{p_0(\ln p_0^{-1})^r}{(1 + \gamma)(1 - p_0)p_\varepsilon(\ln p_\varepsilon^{-1})^r} \approx \frac{0.03 p_\varepsilon^{-1}}{(\log_{10} p_\varepsilon^{-1})^2} \quad (30.29)$$

for $r = 2$, $\gamma = 3$, and $p_0 = 0.1$. For rare events, p_ε^{-1} is very large, and, as expected, SS outperforms MCS; for example, if $p_\varepsilon = 10^{-6}$, then $\eta_{SS} \approx 800$.

In recent years, a number of modifications of SS have been proposed, including SS with splitting [17] (described in the next section), hybrid SS [18], two-stage SS [30], spherical SS [31], and SS with delayed rejection [57]. A Bayesian post-processor for SS, which generalizes the Bayesian interpretation of MCS described

above, was developed in [56]. In the original paper [4], SS was developed for estimating reliability of complex civil engineering structures such as tall buildings and bridges at risk from earthquakes. It was applied for this purpose in [5] and [24]. SS and its modifications have also been successfully applied to rare-event simulation in fire risk analysis [8], aerospace [39, 52], nuclear [16], wind [49] and geotechnical engineering [46], and other fields. A detailed exposition of SS at an introductory level and a MATLAB code implementing the above pseudo-code are given in [55]. For more advanced and complete reading, the fundamental monograph on SS [7] is strongly recommended.

5 Splitting

In the previously presented stochastic simulation methods, samples of the input and output discrete-time histories, $\{U(t) : t = 0, \dots, T\} \subset \mathbb{R}^m$ and $\{X(t) : t = 0, \dots, T\} \subset \mathbb{R}^n$, are viewed geometrically as vectors U and X that define points in the vector spaces $\mathbb{R}^{(T+1)m}$ and $\mathbb{R}^{(T+1)n}$, respectively. In the splitting method, however, samples of the input and output histories are viewed as trajectories defining paths of length $(T + 1)$ in \mathbb{R}^m and \mathbb{R}^n , respectively. Samples that reach a certain designated subset in the input or output spaces at some time are treated as “mothers” and are then split into multiple offspring trajectories by separate sampling of the input histories subsequent to the splitting time. These multiple trajectories can themselves subsequently be treated as mothers if they reach another designated subset nested inside the first subset at some later time and so be split into multiple offspring trajectories. This is continued until a certain number of the trajectories reach the smallest nested subset corresponding to the rare event of interest.

Splitting methods were originally introduced by Kahn and Harris [28], and they have been extensively studied (e.g., [12, 17, 42, 54]). We describe splitting here by using the framework of subset simulation where the only change is that the conditional sampling in the nested subsets is done by splitting the trajectories that reach each subset, rather than using them as seeds to generate more samples from Markov chains in their stationary state. As a result, only standard Monte Carlo simulation is needed, instead of MCMC simulation.

The procedure in [17] is followed here to generate offspring trajectories at the i^{th} level ($i = 1, \dots, L$) of subset simulation from each of the mother trajectories in \mathcal{E}_i constructed from samples from the previous level, except that we present it from the viewpoint of trajectories in the input space, rather than the output space. Therefore, at the i^{th} level, each of the np_0 sampled input histories U_j , $j = 1, \dots, np_0$, from the previous level that satisfy $U_j \in \mathcal{E}_i$, as defined in (30.20) (so the corresponding output history X_j satisfies $\max_{t=0, \dots, T} g(X_j(t)) > b_i$), is split at their first-passage time

$$t_j = \min\{t = 0, \dots, T : g(X_j(t)) > b_i\} \quad (30.30)$$

This means that the mother trajectory U_j is partitioned as $[U_j^-, U_j^+]$ where $U_j^- = [U_j(0), \dots, U_j(t_j)]$ and $U_j^+ = [U_j(t_j + 1), \dots, U_j(T)]$; then a subtrajectory sample $\tilde{U}_j^+ = [\tilde{U}_j(t_j + 1), \dots, \tilde{U}_j(T)]$ is drawn from

$$p(u_j^+ | U_j^-, \mathcal{E}_i) = \frac{\mathbb{P}(\mathcal{E}_i | u_j^+, U_j^-)}{\mathbb{P}(\mathcal{E}_i | U_j^-)} p(u_j^+ | U_j^-) = p(u_j^+ | U_j^-) = p(u_j^+) \quad (30.31)$$

where the last equation follows if one assumes independence of the $U_j(t), t = 0, \dots, T$ (although it is not necessary). Also, $\mathbb{P}(\mathcal{E}_i | u_j^+, U_j^-) = 1 = \mathbb{P}(\mathcal{E}_i | U_j^-)$. Note that the new input sample $\tilde{U}_j = [U_j^-, \tilde{U}_j^+]$ also lies in \mathcal{E}_i since it has the subtrajectory U_j^- in common with U_j , which implies that the corresponding outputs at the first-passage time t_j are equal: $\tilde{X}_j(t_j) = X_j(t_j) > b_i$. The offspring trajectory \tilde{U}_j is a sample from $p(u)$ lying in \mathcal{E}_i , and so, like its mother U_j , it is a sample from $p(u|\mathcal{E}_i)$. This process is repeated to generate $(\frac{1}{p_0} - 1)$ such offspring trajectories from each mother trajectory, giving a total of $np_0(\frac{1}{p_0} - 1) + np_0 = n$ input histories that are samples from $p(u|\mathcal{E}_i)$ at the i^{th} level.

The pseudo-code for the splitting version of subset simulation is the same as the previously presented pseudo-code for the MCMC version except that the part describing the generation of conditional samples at level i using the MMA algorithm is replaced by

Generation of conditional samples at level i with Splitting

```

for  $j = 1, \dots, np_0$  do
    Using  $U_j^{(i-1)} \sim p(u|\mathcal{E}_i)$  as a mother trajectory, generate  $(\frac{1}{p_0} - 1)$  offspring
    trajectories by splitting of this input trajectory.
end for
```

To generate the same number of samples n at a level, the splitting version of subset simulation is slightly more efficient than the MCMC version using MMA because when generating the conditional samples, the input offspring trajectories $\tilde{U} = [\tilde{U}^-, \tilde{U}^+]$ already have made available the first part \tilde{X}^- of the corresponding output trajectory $\tilde{X} = [\tilde{X}^-, \tilde{X}^+]$. Thus, (30.1) need only be solved for \tilde{X}^+ starting from the final value of \tilde{X}^- (which corresponds to the first-passage time of the trajectory). A disadvantage of the splitting version is that it cannot handle parameter uncertainty in the model in (30.1) since the offspring trajectories must use (30.1) with the same parameter values as their mothers. Furthermore, the splitting version applies only to dynamic problems, as considered here. The MCMC version of subset simulation can handle parameter uncertainty and is applicable to both static and dynamic uncertainty quantification problems.

Ching, Au, and Beck [17] discuss the statistical properties of the estimators corresponding to (30.22) and (30.25) when the sampling at each level is done by the

trajectory splitting method. They show that as long as the conditional probability in subset simulation satisfies $p_0 \geq 0.1$, the coefficient of variation for $p_{\mathcal{E}}$ when estimating it by (30.21) and (30.25) is insensitive to p_0 .

Ching, Beck, and Au [18] also introduce a hybrid version of subset simulation that combines some advantages of the splitting and MCMC versions when generating the conditional samples $U_j, j = 1, \dots, n$ at each level. It is limited to dynamic problems because of the splitting, but it can handle parameter uncertainty through using MCMC. All three variants of subset simulation are applied to a series of benchmark reliability problems in [6]; their results imply that for the same computational effort in the dynamic benchmark problems, the hybrid version gives slightly better accuracy for the rare-event probability than the MCMC version. For a comparison between these results and those of other stochastic simulation methods that are applied to some of the same benchmark problems (e.g., spherical subset simulation, auxiliary domain method, and line sampling), the reader may wish to check [47].

6 Illustrative Example

To illustrate MCS, IS, and SS with MCMC and splitting for rare-event estimation, consider the following forced Lorenz system of ordinary differential equations:

$$\dot{X}_1 = \sigma(X_2 - X_1) + U(t) \quad (30.32)$$

$$\dot{X}_2 = rX_1 - X_2 - X_1X_3 \quad (30.33)$$

$$\dot{X}_3 = X_1X_2 - bX_3 \quad (30.34)$$

where $X(t) = (X_1(t), X_2(t), X_3(t))$ defines the system state at time t and $U(t)$ is the external excitation to the system. If $U(t) \equiv 0$, these are the original equations due to E. N. Lorenz that he derived from a model of fluid convection [36]. In this example, the three parameters σ , r , and b are set to $\sigma = 3$, $b = 1$, and $r = 26$. It is well known (e.g., [50]) that in this case, the Lorenz system has three unstable equilibrium points, one of which is

$$X^* = \left(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1 \right) = (5, 5, 25) \quad (30.35)$$

that lies on one “wing” of the “butterfly” attractor. Let

$$X(0) = X^* + (1/2, 1/2, 1/2) = (5.5, 5.5, 25.5) \quad (30.36)$$

be the initial condition and $X(t)$ be the corresponding solution. Lorenz showed [36] that the solution of (30.32), (30.33), and (30.34) with $U(t) \equiv 0$ always (for any t) stays inside the bounding ellipsoid \mathbb{E} :

$$\frac{X_1(t)^2}{R^2 \frac{b}{\sigma}} + \frac{X_2(t)^2}{bR^2} + \frac{(X_3(t) - R)^2}{R^2} \leq 1, \quad R = r + \sigma \quad (30.37)$$

Suppose that the system is now excited by $U(t) = \alpha B(t)$, where $B(t)$ is the standard Brownian process (Gaussian white noise) and α is some scaling constant. The uncertain stochastic excitation $U(t)$ makes the corresponding system trajectory $X(t)$ also stochastic. Let us say that the event \mathcal{E} occurs if $X(t)$ leaves the bounding ellipsoid \mathbb{E} during the time interval of interest $[0, T]$.

The discretization of the excitation U is obtained by the standard discretization of the Brownian process:

$$U(0) = 0, \quad U(k) = \alpha B(k \Delta t) = U(k-1) + \alpha \sqrt{\Delta t} Z_k = \alpha \sqrt{\Delta t} \sum_{i=1}^k Z_i, \quad (30.38)$$

where $\Delta t = 0.1$ s is the sampling interval and $k = 1, \dots, D = T/\Delta t$, and Z_1, \dots, Z_D are i.i.d. standard Gaussian random variables. The target domain $\mathcal{E} \subset \mathbb{R}^D$ is then

$$\mathcal{E} = \{(Z_1, \dots, Z_D) : \max_{0 \leq k \leq D} g(k) > 1\}, \quad (30.39)$$

where the system response $g(k)$ at time $t = k \Delta t$ is

$$g(k) = \frac{X_1(k \Delta t)^2}{R^2 \frac{b}{\sigma}} + \frac{X_2(k \Delta t)^2}{bR^2} + \frac{(X_3(k \Delta t) - R)^2}{R^2} \quad (30.40)$$

Figure 30.1 shows the solution of the unforced Lorenz system (with $\alpha = 0$ so $U(t) = 0$), and an example of the solution of the forced system (with $\alpha = 3$) that corresponds to excitation $U \in \mathcal{E}$ (slightly abusing notation, $U = U(Z_1, \dots, Z_D) \in \mathcal{E}$ means that the corresponding Gaussian vector $(Z_1, \dots, Z_D) \in \mathcal{E}$).

Monte Carlo Simulation: For $\alpha = 3$, Fig. 30.2 shows the probability $p_{\mathcal{E}}$ of event \mathcal{E} as a function of T estimated using standard MCS:

$$\hat{p}_{\mathcal{E}}^{MCS} = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{E}}(Z^{(i)}) \quad (30.41)$$

where $Z^{(i)} = (Z_1^{(i)}, \dots, Z_D^{(i)}) \sim \phi(z)$ are i.i.d. samples from the standard D -dimensional Gaussian PDF $\phi(z)$. For each value of T , $N = 10^4$ samples were used. When $T < 25$, the accuracy of the MCS estimate (30.41) begins to degenerate since the total number of samples N becomes too small for the corresponding target probability. Moreover, for $T < 15$, none of the N -generated MCS samples belong to the target domain \mathcal{E} , making the MCS estimate zero. Figure 30.2 shows, as expected, that $p_{\mathcal{E}}$ is an increasing function of T , since the more time the system has, the more likely its trajectory eventually penetrates the boundary of ellipsoid \mathbb{E} .

Importance Sampling: IS is a variance reduction technique, and, as it was discussed in previous sections, its efficiency critically depends on the choice of the

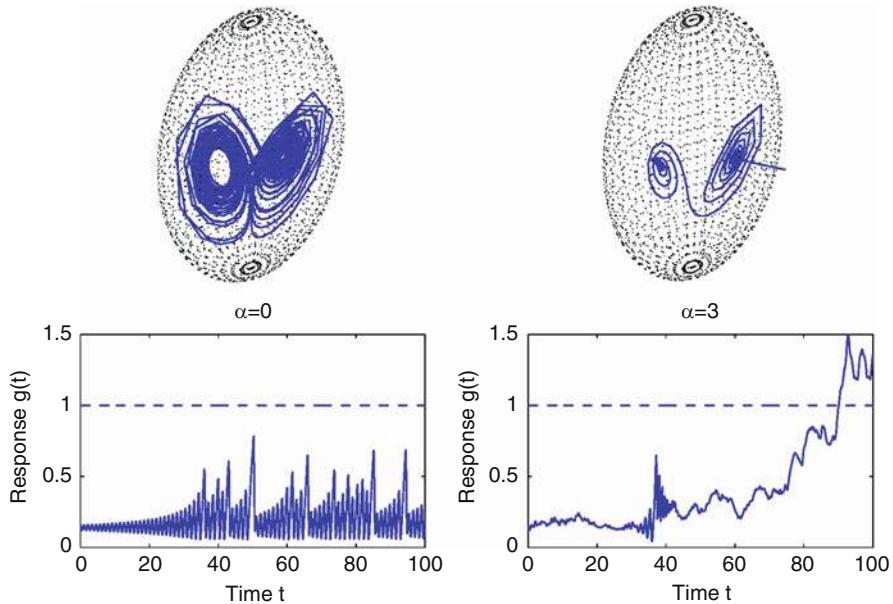


Fig. 30.1 The left column shows the solution of the unexcited Lorenz system ($\alpha = 0$) enclosed in the bounding ellipsoid \mathbb{E} (top) and the corresponding response function $g(t)$ (bottom), where $t \in [0, T]$, $T = 100$. The right top panel shows the solution of the forced Lorenz system ($\alpha = 3$) that corresponds to an excitation $U \in \mathcal{E}$. As it is clearly seen, this solution leaves the ellipsoid \mathbb{E} . According to the response function $g(t)$ shown in the right bottom panel, this first-passage event happens around $t = 90$

ISD q . Usually some geometric information about the target domain \mathcal{E} is needed for constructing a good ISD. To get some intuition, Fig. 30.3 shows the domain \mathcal{E} for two lower-dimensional cases: $T = 1$, $\Delta t = 0.5$ ($D = 2$) and $T = 1.5$, $\Delta t = 0.5$ ($D = 3$). Notice that in both cases, \mathcal{E} consists of two well-separated subsets, $\mathcal{E} = \mathcal{E}_- \cup \mathcal{E}_+$, which are approximately symmetric about the origin. This suggests that a good ISD must be a mixture of two distributions q_- and q_+ that effectively sample \mathcal{E}_- and \mathcal{E}_+ ,

$$q(z) = \frac{q_-(z) + q_+(z)}{2} \quad (30.42)$$

In this example, three different ISDs, denoted q_1 , q_2 , and q_3 , are considered:

Case 1: $q_{\pm}(z) = \phi(z| \pm z_{\mathcal{E}})$, where $z_{\mathcal{E}} \sim \phi(z|\mathcal{E})$. That is, we first generate a sample $z_{\mathcal{E}} \in \mathcal{E}$ and then take ISD q_1 as the mixture of Gaussian PDFs centered at $z_{\mathcal{E}}$ and $-z_{\mathcal{E}}$.

Case 2: $q_{\pm}(z) = \phi(z| \pm z_{\mathcal{E}}^*)$, where $z_{\mathcal{E}}^*$ is obtained as follows. First we generate $n = 1000$ samples from $\phi(z)$ and define $z_{\mathcal{E}}^*$ to be the sample in \mathcal{E} with the smallest norm. Sample $z_{\mathcal{E}}^*$ can be interpreted as the “best representative” of \mathcal{E}_- (or \mathcal{E}_+),

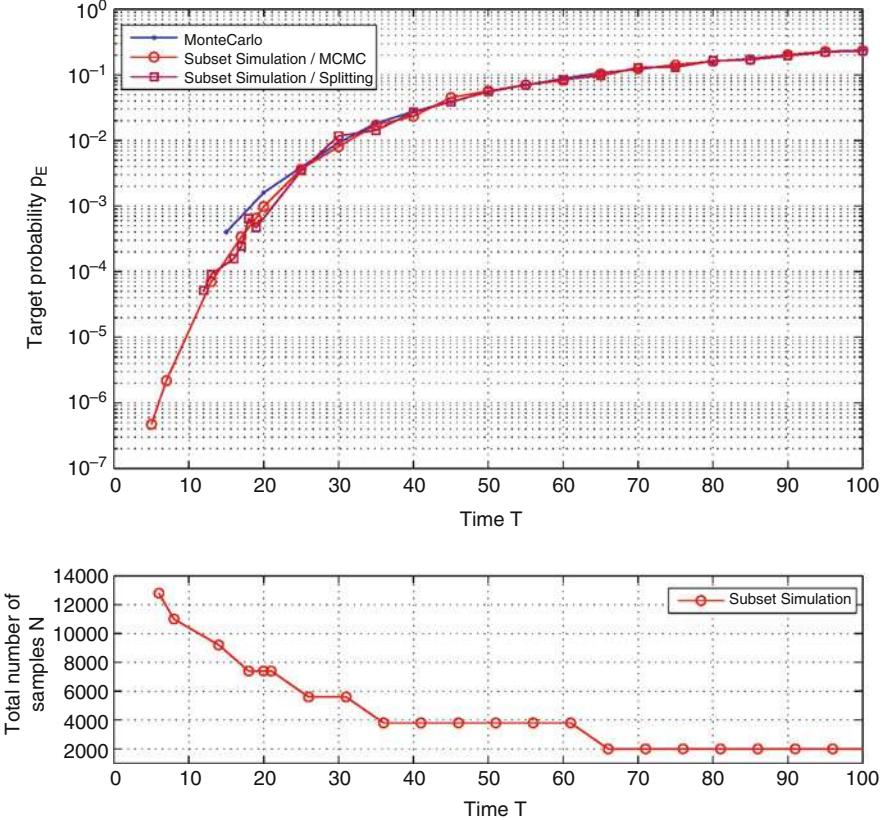


Fig. 30.2 Top panel shows the estimate of the probability $p_{\mathcal{E}}$ of event \mathcal{E} where $\alpha = 3$ as a function of duration time T . For each value of $T \in [5, 100]$, $N = 10^4$ samples were used in MCS, and $n = 2 \times 10^3$ samples per conditional level were used in the two versions of SS. The MCS and SS/splitting estimates for $p_{\mathcal{E}}$ are zero for $T < 15$ and $T < 12$, respectively. The bottom panel shows the total computational effort automatically chosen by both SS algorithms

since $\phi(z_{\mathcal{E}}^*)$ has the largest (among generated samples) value. We then take ISD q_2 as the mixture of Gaussian PDFs centered at $z_{\mathcal{E}}^*$ and $-z_{\mathcal{E}}^*$.

Case 3: To illustrate what happens if one ignores the geometric information about two components of \mathcal{E} , we choose $q_3(z) = \phi(z|z_{\mathcal{E}}^*)$, as given in Case 2.

Let $T = 1$ and $\alpha = 20$. The dimension of the uncertain excitation space is then $D = 10$. Table 30.1 shows the simulation results for the above three cases as well as for standard MCS. The IS method with q_1 , on average, correctly estimates $p_{\mathcal{E}}$. However, the c.o.v. of the estimate is very large, which results in large fluctuations of the estimate in independent runs. IS with q_2 works very well and outperforms MCS: the c.o.v. is reduced by half. Finally, IS with q_3 completely misses one component

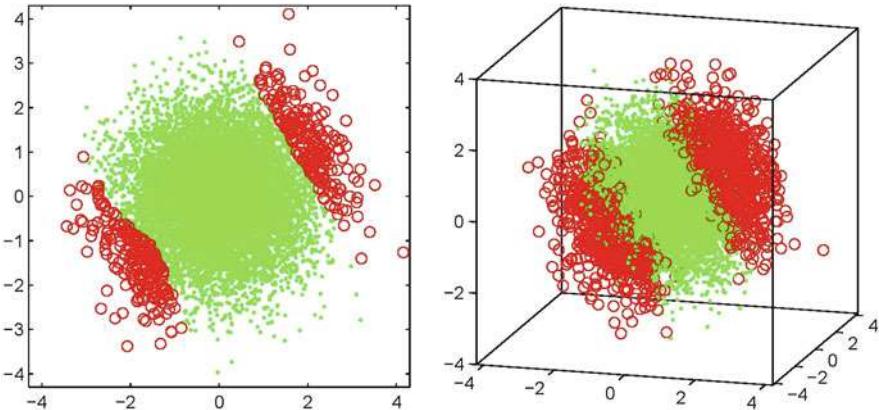


Fig. 30.3 Left panel: visualization of the domain \mathcal{E} in two-dimensional case $D = 2$, where $T = 1$, $\Delta t = 0.5$, and $\alpha = 20$. $N = 10^4$ samples were generated and marked by red circles (respectively, green dots) if they do (respectively, do not) belong to \mathcal{E} . Right panel: the same as on the left panel but with $D = 3$ and $T = 1.5$

Table 30.1 Simulation results for IS and MCS. For each method, mean values $\langle \hat{p}_{\mathcal{E}} \rangle$ of the estimates and their coefficient of variations $\delta(\hat{p}_{\mathcal{E}})$ are based on 100 independent runs

	$\langle \hat{p}_{\mathcal{E}} \rangle$	$\delta(\hat{p}_{\mathcal{E}})$
MCS	3.4×10^{-3}	17%
IS q_1	3.2×10^{-3}	132.4%
IS q_2	3.4×10^{-3}	8.3%
IS q_3	1.8×10^{-3}	5.5%

part of the target domain \mathcal{E} , and the resulting estimate is about half of the correct value. Note that the c.o.v. in this case is very small, which is very misleading.

It was mentioned in previous sections that IS is often not efficient in high dimensions because it becomes more difficult to construct a good ISD [5, 32]. To illustrate this effect, IS with q_2 was used to estimate $p_{\mathcal{E}}$ for a sequence of problems where the total duration time gradually grows from $T = 1$ to $T = 10$. This results in an increase of the dimension D of the underlying uncertain excitation space from 10 to 100. Figure 30.4 shows how the IS estimate degenerates as the dimension D of the problem increases. While IS is accurate when $D = 10$ ($T = 1$), it strongly underestimates the true value of $p_{\mathcal{E}}$ as D approaches 100 ($T = 10$).

Subset Simulation: SS is a more advanced simulation method, and, unlike IS, it does not suffer from the curse of dimensionality. For $\alpha = 3$, Fig. 30.2 shows the estimate of the target probability $p_{\mathcal{E}}$ as a function of T using SS with MCMC and splitting. For each value of T , $n = 2 \times 10^3$ samples were used in each conditional level in SS. Unlike MCS, SS is capable of efficiently simulating very rare events and estimating their small probabilities. The total computational effort, i.e., the total number N of samples automatically chosen by SS, is shown in the bottom panel of Fig. 30.2. Note that the larger the value of $p_{\mathcal{E}}$, the smaller the number of conditional levels in SS, and, therefore, the smaller the total number of samples N . The total

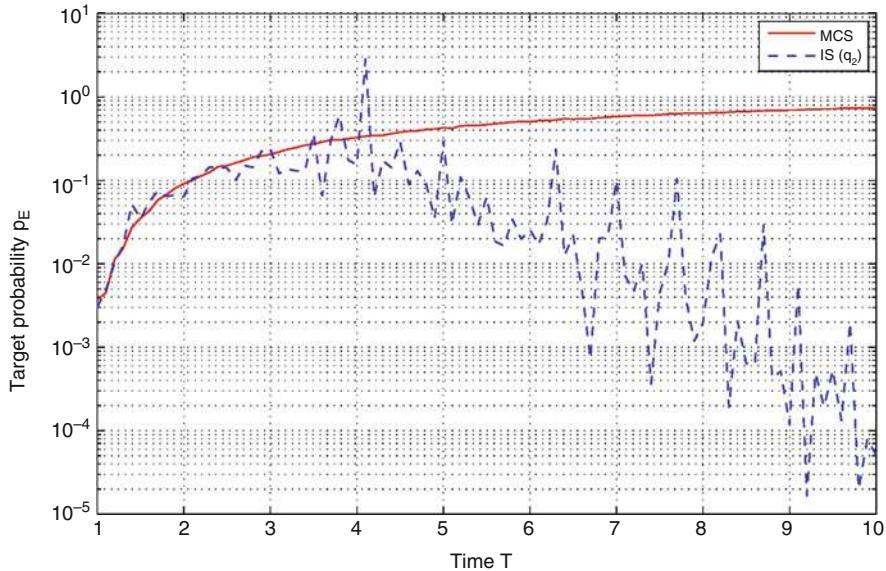


Fig. 30.4 Estimation of the target probability p_E as a function of duration time T . Solid red and dashed blue curves correspond to MCS and IS with q_2 , respectively. In this example, $\alpha = 20$ and $N = 10^4$ samples for each value of T are used. It is clearly visible how the IS estimate degenerates as the dimension D goes from 10 ($T = 1$) to 100 ($T = 10$)

computational effort in SS is thus a decreasing function of T . In this example, the original MCMC strategy [4] for generating conditional samples outperforms the splitting strategy [17] that exploits the causality of the system: while the SS/MCMC method works even in the most extreme case ($T = 5$), the SS/splitting estimate for p_E becomes zero for $T < 12$.

7 Conclusion

This chapter examines computational methods for rare-event simulation in the context of uncertainty quantification for dynamic systems that are subject to future uncertain excitation modeled as a stochastic process. The rare events are assumed to correspond to some time-varying performance quantity exceeding a specified threshold over a specified time duration, which usually means that the system performance fails to meet some design or operation specifications.

To analyze the reliability of the system against this performance failure, a computational model for the input-output behavior of the system is used to predict the performance of interest as a function of the input stochastic process discretized in time. This dynamic model may involve explicit treatment of parametric and non-parametric uncertainties that arise because the model only approximately describes

the real system behavior, implying that there are usually no true values of the model parameters and the accuracy of its predictions is uncertain. In the engineering literature, the mathematical problem to be solved numerically for the probability of performance failure, commonly called the failure probability, is referred to as the first-passage reliability problem. It does not have an analytical solution, and numerical solutions must face two challenging aspects:

1. The vector representing the time-discretized stochastic process that models the future system excitation lies in an input space of high dimension;
2. The dynamic systems of interest are assumed to be highly reliable so that their performance failure is a rare event, that is, the probability of its occurrence, $p_{\mathcal{E}}$, is very small.

As a result, standard Monte Carlo simulation and importance sampling methods are not computationally efficient for first-passage reliability problems. On the other hand, subset simulation has proved to be a general and powerful method for numerical solution of these problems. Like MCS, it is not affected by the dimension of the input space, and for a single run, it produces a plot of $p_{\mathcal{E}}$ vs. threshold b covering $p_{\mathcal{E}} \in [p_0^{-L}, 1]$, where L is the number of levels used. For a critical appraisal of methods for first-passage reliability problems in high dimensions, the reader may wish to check Schuëller et al. [48].

Several variants of subset simulation have been developed motivated by the goal of further improving the computational efficiency of the original version, although the efficiency gains, if any, are modest. All of them have an accuracy described by a coefficient of variation for the estimate of the rare-event probability that depends on $\ln(1/p_{\mathcal{E}})$ rather than $\sqrt{1/p_{\mathcal{E}}}$ as in standard Monte Carlo simulation. For all methods covered in this section, the dependence of this coefficient of variation on the number of samples N is proportional to $N^{-1/2}$. Therefore, in the case of very low probabilities, $p_{\mathcal{E}}$, it still requires thousands of simulations (large N) of the response time history based on a dynamic model as in (30.1) in order to get acceptable accuracy. For complex models, this computational effort may be prohibitive.

One approach to reduce the computational effort when estimating very low rare-event probabilities is to utilize additional information about the nature of the problem for specific classes of reliability problems (e.g., [2, 3]). Another more general approach is to construct surrogate models (meta-models) based on using a relatively small number of complex-model simulations as training data. The idea is to use a trained surrogate model to rapidly calculate an approximation of the response of the complex computational model as a substitute when drawing new samples. Various methods for constructing surrogate models have been applied in reliability engineering, including response surfaces [14], support vector machines [13, 23], neural networks [41], and Gaussian process modeling (Kriging) [21]. The latter method is a particularly powerful one because it also provides a probabilistic assessment of the approximation error. It deserves further exploration, especially with regard to the optimal balance between the accuracy of the surrogate model

as a function of the number of training samples from the complex model and the accuracy of the estimate of the rare-event probability as a function of the total number of samples from both the complex model and the surrogate model.

References

1. Asmussen, S., Glynn, P.W.: *Stochastic Simulation: Algorithms and Analysis*. Springer, New York (2010)
2. Au, S.K.: Importance sampling for elasto-plastic systems using adapted process with deterministic control. *Int. J. Nonlinear Mech.* **44**, 189–198 (2009)
3. Au, S.K., Beck, J.L.: First excursion probabilities for linear systems by very efficient importance sampling. *Prob. Eng. Mech.* **16**, 193–207 (2001)
4. Au, S.K., Beck, J.L.: Estimation of small failure probabilities in high dimensions by subset simulation. *Prob. Eng. Mech.* **16**(4), 263–277 (2001)
5. Au, S.K., Beck, J.L.: Importance sampling in high dimensions. *Struct. Saf.* **25**(2), 139–163 (2003)
6. Au, S.K., Ching, J., Beck, J.L.: Application of subset simulation methods to reliability benchmark problems. *Struct. Saf.* **29**(3), 183–193 (2007)
7. Au, S.K., Wang, Y.: *Engineering Risk Assessment and Design with Subset Simulation*. Wiley, Singapore (2014)
8. Au, S.K., Wang, Z.H., Loa, S.M.: Compartment fire risk analysis by advanced Monte Carlo method. *Eng. Struct.* **29**, 2381–2390 (2007)
9. Bayes, T.: An essay towards solving a problem in the doctrine of chances. *Philos. Trans. R. Soc. Lond.* **53**, 370–418 (1763). Reprinted in *Biometrika* **45**, 296–315 (1989)
10. Beck, J.L.: Bayesian system identification based on probability logic. *Struct. Control Health Monit.* **17**, 825–847 (2010)
11. Beck, J.L., Au, S.K.: Reliability of Dynamic Systems using Stochastic Simulation. In: *Proceedings of the 6th European Conference on Structural Dynamics*, Paris (2005)
12. Botev, Z.I., Kroese, D.P.: Efficient Monte Carlo simulation via the generalized splitting method. *Stat. Comput.* **22**(1), 1–16 (2012)
13. Bourinet, J.M., Deheeger, F., Lemaire, M.: Assessing small failure probabilities by combined subset simulation and support vector machines. *Struct. Saf.* **33**(6), 343–353 (2011)
14. Bucher, C., Bourgund, U.: A fast and efficient response surface approach for structural reliability problems. *Struct. Saf.* **7**, 57–66 (1990)
15. Bucklew, J.A.: *Introduction to Rare Event Simulation*. Springer Series in Statistics. Springer, New York (2004)
16. Cadini, F., Avram, D., Pedroni, N., Zio, E.: Subset simulation of a reliability model for radioactive waste repository performance assessment. *Reliab. Eng. Syst. Saf.* **100**, 75–83 (2012)
17. Ching, J., Au, S.K., Beck, J.L.: Reliability estimation for dynamical systems subject to stochastic excitation using subset simulation with splitting. *Comput. Methods Appl. Mech. Eng.* **194**, 1557–1579 (2005)
18. Ching, J., Beck, J.L., Au, S.K.: Hybrid subset simulation method for reliability estimation of dynamical systems subject to stochastic excitation. *Prob. Eng. Mech.* **20**, 199–214 (2005)
19. Deng, S., Giesecke, K., Lai, T.L.: Sequential importance sampling and resampling for dynamic portfolio credit risk. *Oper. Res.* **60**(1), 78–91 (2012)
20. Ditlevsen, O., Madsen, H.O.: *Structural Reliability Methods*. Wiley, Chichester/New York (1996)
21. Dubourg, V., Sudret, B., Deheeger, F.: Meta-model based importance sampling for structural reliability analysis. *Prob. Eng. Mech.* **33**, 47–57 (2013)

22. Dunn, W.L., Shultis, J.K.: Exploring Monte Carlo Methods. Elsevier, Amsterdam/Boston (2012)
23. Hurtado, J.: Structural Reliability: Statistical Learning Perspectives. Springer, Berlin/New York (2004)
24. Jalayer, F., Beck, J.L.: Effects of two alternative representations of ground-motion uncertainty on probabilistic seismic demand assessment of structures. *Earthq. Eng. Struct. Dyn.* **37**, 61–79 (2008)
25. Jaynes, E.T.: Information theory and statistical mechanics. *Phys. Rev.* **106**(4), 620–630 (1957)
26. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge (2003)
27. Johnson, C.: Numerical Solution of Partial Differential Equations by the Finite Element Method. Dover, Mineola (2009)
28. Kahn, H., Harris, T.E.: Estimation of particle transmission by random sampling. *Natl. Bur. Stand. Appl. Math. Ser.* **12**, 27–30 (1951)
29. Kahn, H., Marshall, A. W.: Methods of reducing sample size in Monte Carlo computations. *J. Oper. Res. Soc. Am.* **1**(5), 263–278 (1953)
30. Katafygiotis, L.S., Cheung, S.H.: A two-stage subset simulation-based approach for calculating the reliability of inelastic structural systems subjected to Gaussian random excitations. *Comput. Methods Appl. Mech. Eng.* **194**, 1581–1595 (2005)
31. Katafygiotis, L.S., Cheung, S.H.: Application of spherical subset simulation method and auxiliary domain method on a benchmark reliability study. *Struct. Saf.* **29**(3), 194–207 (2007)
32. Katafygiotis, L.S., Zuev, K.M.: Geometric insight into the challenges of solving high-dimensional reliability problems. *Prob. Eng. Mech.* **23**, 208–218 (2008)
33. Katafygiotis, L.S., Zuev, K.M.: Estimation of small failure probabilities in high dimensions by adaptive linked importance sampling. In: Proceedings of the COMPDYN-2007, Rethymno (2007)
34. Laplace, P.S.: Théorie Analytique des Probabilités. Courcier, Paris (1812)
35. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2001)
36. Lorenz, E.N.: Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**(2), 130–141 (1963)
37. Metropolis, N., Ulam, S.: The Monte Carlo method. *J. Am. Stat. Assoc.* **44**, 335–341 (1949)
38. Metropolis, N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953)
39. Pellissetti, M.F., Schuëller, G.I., Pradlwarter, H.J., Calvi, A., Fransen, S., Klein, M.: Reliability analysis of spacecraft structures under static and dynamic loading. *Comput. Struct.* **84**, 1313–1325 (2006)
40. Papadimitriou, C., Beck, J.L., Katafygiotis, L.S.: Updating robust reliability using structural test data. *Prob. Eng. Mech.* **16**, 103–113 (2001)
41. Papadopoulos, V., Giovanis, D.G., Lagaros, N.D., Papadrakakis, M.: Accelerated subset simulation with neural networks for reliability analysis. *Comput. Methods Appl. Mech. Eng.* **223**, 70–80 (2012)
42. Pradlwarter, H.J., Schuëller, G.I., Melnik-Melnikov, P.G.: Reliability of MDOF-systems. *Prob. Eng. Mech.* **9**, 235–43 (1994)
43. Rackwitz, R.: Reliability analysis – a review and some perspectives. *Struct. Saf.* **32**, 365–395 (2001)
44. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2004)
45. Ross, S.M.: A First Course in Probability, 8th edn. Prentice Hall, Upper Saddle River (2009)
46. Santoso, A.M., Phoon, K.K., Quek, S.T.: Modified Metropolis-Hastings algorithm with reduced chain correlation for efficient subset simulation. *Prob. Eng. Mech.* **26**, 331–341 (2011)
47. Schuëller, G.I., Pradlwarter, H.J.: Benchmark study on reliability estimation in higher dimensions of structural systems – an overview. *Struct. Saf.* **29**(3), 167–182 (2007)
48. Schuëller, G.I., Pradlwarter, H.J., Koutsourelakis, P.S.: A critical appraisal of reliability estimation procedures for high dimensions. *Prob. Eng. Mech.* **19**, 463–474 (2004)
49. Sichani, M.T., Nielsen, S.R.K.: First passage probability estimation of wind turbines by Markov chain Monte Carlo. *Struct. Infrastruct. Eng.* **9**, 1067–1079 (2013)

-
50. Sparrow, C.: *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*. Springer, New York (1982)
 51. Taflanidis, A.A., Beck, J.L.: Analytical approximation for stationary reliability of certain and uncertain linear dynamic systems with higher dimensional output. *Earthq. Eng. Struct. Dyn.* **35**, 1247–1267 (2006)
 52. Thunnissen, D.P., Au, S.K., Tsuyuki, G.T.: Uncertainty quantification in estimating critical spacecraft component temperatures. *AIAA J. Thermophys. Heat Transf.* **21**(2), 422–430 (2007)
 53. Valdebenito, M.A., Pradlwarter, H.J., Schüller, G.I.: The role of the design point for calculating failure probabilities in view of dimensionality and structural nonlinearities. *Struct. Saf.* **32**, 101–111 (2010)
 54. Villén-Altamirano, M., Villén-Altamirano, J.: Analysis of RESTART simulation: theoretical basis and sensitivity study. *Eur. Trans. Telecommun.* **13**(4), 373–386 (2002)
 55. Zuev, K.: Subset simulation method for rare event estimation: an introduction. In: M. Beer et al. (Eds.) *Encyclopedia of Earthquake Engineering*. Springer, Berlin/Heidelberg (2015). Available on-line at <http://www.springerreference.com/docs/html/chapterdbid/369348.html>
 56. Zuev, K.M., Beck, J.L., Au, S.K., Katafygiotis, L.S.: Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Comput. Struct.* **92–93**, 283–296 (2012)
 57. Zuev, K.M., Katafygiotis, L.S.: Modified Metropolis-Hastings algorithm with delayed rejection. *Prob. Eng. Mech.* **26**, 405–412 (2011)

Part IV

Introduction to Sensitivity Analysis

Bertrand Iooss and Andrea Saltelli

Abstract

Sensitivity analysis provides users of mathematical and simulation models with tools to appreciate the dependency of the model output from model input and to investigate how important is each model input in determining its output. All application areas are concerned, from theoretical physics to engineering and socio-economics. This introductory paper provides the sensitivity analysis aims and objectives in order to explain the composition of the overall “Sensitivity Analysis” chapter of the Springer Handbook. It also describes the basic principles of sensitivity analysis, some classification grids to understand the application ranges of each method, a useful software package, and the notations used in the chapter papers. This section also offers a succinct description of sensitivity auditing, a new discipline that tests the entire inferential chain including model development, implicit assumptions, and normative issues and which is recommended when the inference provided by the model needs to feed into a regulatory or policy process. For the “Sensitivity Analysis” chapter, in addition to this introduction, eight papers have been written by around twenty practitioners from different fields of application. They cover the most widely used methods for this subject: the deterministic methods as the local sensitivity analysis, the experimental design strategies, the sampling-based and variance-based methods developed from the 1980s, and the new importance measures and metamodel-

B. Iooss (✉)

Industrial Risk Management Department, EDF R&D, Chatou, France

Institut de Mathématiques de Toulouse, Université Paul Sabatier, Toulouse, France

e-mail: bertrand.iooss@edf.fr; biooss@yahoo.fr

A. Saltelli

Centre for the Study of the Sciences and the Humanities (SVT), University of Bergen (UIB),
Bergen, Norway

Institut de Ciència i Tecnologia Ambientals (ICTA), Universitat Autònoma de Barcelona (UAB),
Barcelona, Spain

e-mail: andrea.saltelli@svt.uib.no; andrea.saltelli@jrc.ec.europa.eu

based techniques established and studied since the 2000s. In each paper, toy examples or industrial applications illustrate their relevance and usefulness.

Keywords

Computer experiments • Uncertainty analysis • Sensitivity analysis • Sensitivity auditing • Risk assessment • Impact assessment

Contents

1	Introduction	1104
2	Basic Principles of Sensitivity Analysis	1105
3	Methods Contained in the Chapter	1108
4	Specialized R Software Packages	1112
5	Sensitivity Auditing	1114
6	Conclusion	1118
	References	1119

1 Introduction

In many fields such as environmental risk assessment, behavior of agronomic systems, and structural reliability or operational safety, mathematical models are used for simulation, when experiments are too expensive or impracticable, and for prediction. Models are also used for uncertainty quantification and sensitivity analysis studies. Complex computer models calculate several output values (scalars or functions) that can depend on a high number of input parameters and physical variables. Some of these input parameters and variables may be unknown, unspecified, or defined with a large imprecision range. Inputs include engineering or operating variables, variables that describe field conditions, and variables that include unknown or partially known model parameters. In this context, the investigation of computer code experiments remains an important challenge.

This computer code exploration process is the main purpose of the sensitivity analysis (SA) process. SA allows the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input [51]. It may be used to determine the input variables that contribute the most to an output behavior, and the non-influential inputs, or to ascertain some interaction effects within the model. The SA process entails the computation and analysis of the so-called sensitivity or importance indices of the input variables with respect to a given quantity of interest in the model output. Importance measures of each uncertain input variable on the response variability provide a deeper understanding of the modeling in order to reduce the response uncertainties in the most effective way [23, 30, 57]. For instance, putting more efforts on knowledge of influential inputs will reduce their uncertainties. The underlying goals for SA are model calibration, model validation, and assisting with the decision-making process. This chapter is for engineers, researchers, and students who wish to apply SA techniques in any scientific field (physics, engineering, socio-economics, environmental studies, astronomy, etc.).

Several textbooks and specialist works [2, 3, 8, 10–12, 21, 56, 59] have covered most of the classic SA methods and objectives. In parallel, a scientific conference called SAMO (“Sensitivity Analysis on Model Output”) has been organized every 3 years since 1995 and extensively covers SA related subjects. Works presented at the different SAMO conferences can be found in their proceedings and several special issues published in international journals (mainly in “Reliability Engineering and System Safety”).

The main goal of this chapter is to provide an overview of classic and advanced SA methods, as none of the referenced works have reported all the concepts and methods in one single document. Researchers and engineers will find this document to be an up-to-date report on SA as it currently stands, although this scientific field remains very active in terms of new developments. The present chapter is only a snapshot in time and only covers well-established methods.

The next section of this paper provides the SA basic principles, including elementary graphic methods. In the third section, the SA methods contained in the chapter are described using a classification grid, together with the main mathematical notations of the chapter papers. Then, the SA-specialized packages developed in the R software environment are discussed. To finish this introductory paper, a process for the sensitivity auditing of models in a policy context is discussed, by providing seven rules that extend the use of SA. As discussed in Saltelli et al. [61], SA, mandated by existing guidelines as a good practice to use in conjunction with mathematical modeling, is insufficient to ensure quality in the treatment of scientific uncertainty for policy purposes. Finally, the concluding section lists some important and recent research works that could not be covered in the present chapter.

2 Basic Principles of Sensitivity Analysis

The first historical approach to SA is known as the local approach. The impact of small input perturbations on the model output is studied. These small perturbations occur around nominal values (the mean of a random variable, for instance). This deterministic approach consists of calculating or estimating the partial derivatives of the model at a specific point of the input variable space [68]. The use of adjoint-based methods allows models with a large number of input variables to be processed. Such approaches are particularly well-suited to tackling uncertainty analysis and SA and data assimilation problems in environmental systems such as those in climatology, oceanography, hydrogeology, etc. [3, 4, 48].

To overcome the limitations of local methods (linearity and normality assumptions, local variations), another class of methods has been developed in a statistical framework. In contrast to local SA, which studies how small variations in inputs around a given value change the value of the output, global sensitivity analysis (“global” in opposition to the local analysis) does not distinguish any initial set of model input values, but considers the numerical model in the entire domain of possible input parameter variations [57]. Thus, the global SA is an instrument used

to study a mathematical model as a whole rather than one of its solution around parameters specific values.

Numerical model users and modelers have shown high interest in these global tools that take full advantage of the development of computing equipment and numerical methods (see Helton [20], de Rocquigny et al. [10], and [11] for industrial and environmental applications). Saltelli et al. [58] emphasized the need to specify clearly the objectives of a study before performing an SA. These objectives may include:

- The factor prioritization setting, which aims at identifying the most important factors. The most important factor is the one that, if fixed, would lead to the greatest reduction in the uncertainty of the output;
- The factor fixing setting, which aims at reducing the number of uncertain inputs by fixing unimportant factors. Unimportant factors are the ones that, if fixed to any value, would not lead to a significant reduction of the output uncertainty. This is often a preliminary step before the calibration of model inputs using some available information (real output observations, constraints, etc.);
- The variance cutting setting, which can be a part of a risk assessment study. Its aim is to reduce the output uncertainty from its initial value to a lower pre-established threshold value;
- The factor mapping setting, which aims at identifying the important inputs in a specific domain of the output values, for example, which combination of factors produce output values above or below a given threshold.

In a deterministic framework, the model is analyzed at specific values for inputs, and the space of uncertain inputs may be explored in statistical approaches. In a probabilistic framework instead, the inputs are considered as random variables $\mathbf{X} = (X_1, \dots, X_d) \in \mathbb{R}^d$. The random vector \mathbf{X} has a known joint distribution, which reflects the uncertainty of the inputs. The computer code (also called “model”) is denoted $G(\cdot)$, and for a scalar output $Y \in \mathbb{R}$, the model formula writes

$$Y = G(\mathbf{X}). \quad (31.1)$$

Model function G can represent a system of differential equations, a program code, or any other correspondence between \mathbf{X} and Y values that can be calculated for a finite period of time. Therefore, the model output Y is also a random variable whose distribution is unknown (increasing the knowledge of it is the goal of the uncertainty propagation process). SA statistical methods consist of techniques stemming from the design of experiments theory (in the case of a large number of inputs), the Monte Carlo techniques (to obtain precise sensitivity results), and modern statistical learning methods (for complex CPU time-consuming numerical models).

For example, to begin with the most basic (and essential) methods, simple graphical tools can be applied on an initial sample of inputs/output $(x_1^{(i)}, \dots, x_d^{(i)}, y^{(i)})_{i=1 \dots n}$ (sampling strategies are numerous and are described in

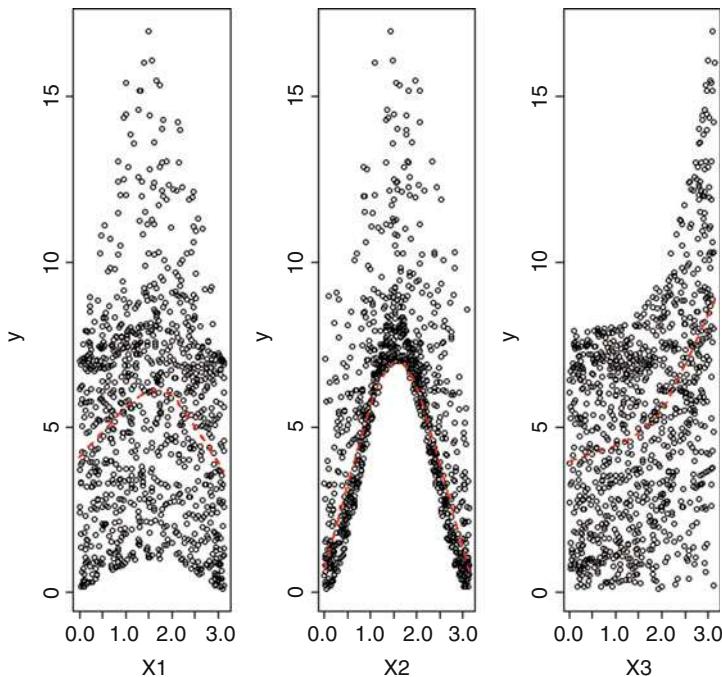


Fig. 31.1 Scatterplots of 200 simulations on a numerical model with three inputs (in abscissa of each plot) and one output (in ordinate). *Dotted curves* are local-polynomial-based smoothers

many other chapters of this handbook). To begin with, simple scatterplots between each input variable and the model output can allow the detection of linear or nonlinear input/output relation. Figure 31.1 gives an example of scatterplots on a simple function with three input variables and one output variable. As the two-dimensional scatterplots do not capture the possible interaction effects between the inputs, the cobweb plots [32] can be used. Also known as parallel coordinate plots, cobweb plots allow to visualize the simulations as a set of trajectories by joining the value (or the corresponding quantile) of each variables' combination of the simulation sample by a line (see Fig. 31.2): Each vertical line represents one variable, the last vertical line representing the output variable. In Fig. 31.2, the simulations leading to the smallest values of the model output have been highlighted in red. This allows to immediately understand that these simulations correspond to combinations of small and large values of the first and second inputs, respectively.

Moreover, from the same sample $\left(x_1^{(i)}, \dots, x_d^{(i)}, y^{(i)}\right)_{i=1 \dots n}$, quantitative global sensitivity measures can be easily estimated, as the linear (or Pearson) correlation coefficient and the rank (or Spearman) correlation coefficient [56]. It is also possible to fit a linear model explaining the behavior of Y given the values of \mathbf{X} , provided that the sample size n is sufficiently large (at least $n > d$). The main indices are then

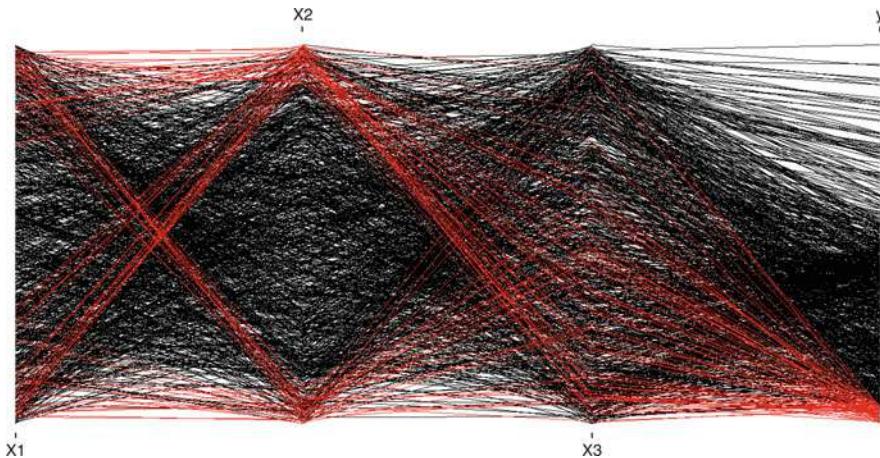


Fig. 31.2 Cobweb plot of 200 simulations of a numerical model with three inputs (first three columns) and one output (last column)

the Standard Regression Coefficients $\text{SRC}_j = \beta_j \sqrt{\frac{\text{Var}(X_j)}{\text{Var}(Y)}}$, where β_j is the linear regression coefficient associated to X_j . SRC_j^2 represents a share of variance if the linearity hypothesis is confirmed. Among many simple sensitivity indices, all these indices are included in the so-called sampling-based sensitivity analysis methods (see the description of the content of this chapter in the next section and Helton et al. [23]).

3 Methods Contained in the Chapter

Three families of methods are described in this chapter, based on which is the objective of the analysis:

1. First, screening techniques aim to a qualitative ranking of input factors at minimal cost in the number of model evaluations. Paper 2 (see ▶ Chap. 32, “Variational Methods”, written by Maëlle Nodet and Arthur Vidard) introduces the local SA based on variational methods, while Paper 3 (see ▶ Chap. 33, “Design of Experiments for Screening”, written by Sue Lewis and David Woods) makes an extensive review on design of experiments techniques, including some screening designs and numerical exploration designs specifically developed in the context of computer experiments;
2. Second, sampling-based methods are described. Paper 4 (see ▶ Chap. 34, “Weights and Importance in Composite Indicators: Mind the Gap”, written by William Becker, Paolo Paruolo, Michaela Saisana, and Andrea Saltelli)

shows how, from an initial sample of input and output values, quantitative sensitivity indices can be obtained by various methods (correlation, multiple linear regression, nonparametric regression) and applied in analyzing composite indicators. In Paper 5 (see ► Chap. 35, “[Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms](#)”, written by Clémentine Prieur and Stefano Tarantola), the definitions of the variance-based importance measures (the so-called Sobol indices) and the algorithms to calculate them will be detailed. In Paper 6 (see ► Chap. 36, “[Derivative-Based Global Sensitivity Measures](#)”, written by Sergeï Kucherenko and Bertrand Iooss), the global SA based on derivatives sample (the DGSM indices) are explained, while in Paper 7 (see ► Chap. 37, “[Moment-Independent and Reliability-Based Importance Measures](#)”, written by Emanuele Borgonovo and Bertrand Iooss), the moment-independent and reliability importance measures are described.

3. Third, in-depth exploration of model behavior with respect to input variation can be carried out. Paper 8 (see ► Chap. 38, “[Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes](#)”, written by Loïc Le Gratiet, Stefano Marelli, and Bruno Sudret) includes recent advances made in the modeling of computer experiments. A metamodel is used as a surrogate model of the computer model with any SA techniques, when the computer model is too CPU time-consuming to allow a sufficient number of model calculations. Special attention is paid to two of the most popular metamodels: the polynomial chaos expansion and the Gaussian process model, for which the Sobol indices can be efficiently obtained. Finally, Paper 9 (see ► Chap. 39, “[Sensitivity Analysis of Spatial and/or Temporal Phenomena](#)”, written by Amandine Marrel, Nathalie Saint Geours, and Matthias De Lozzo) extends the SA tools in the context of temporal and/or spatial phenomena.

All the SA techniques explained in this chapter present a trade-off between the number of model computations required and the assumed model complexity. Figure 31.3 proposes a coarse classification of the global method families described before. This figure shows how to place a method depending on its required number of computations and its underlying hypothesis on the complexity of the G model. For example, “non-monotonic” means that the method can be applied to non-monotonic models, as of course to monotonic and linear ones. A distinction is made between screening techniques (identification of non-influential variables among a large number of variables) and more precise variance-based quantitative methods. As most of the methods have a dimension-dependent cost in terms of required model evaluations, another distinction is made with the few methods whose costs are dimension independent.

With the same axes than the previous figure, Fig. 31.4 proposes a more accurate classification of the classic global SA methods described in the present SA chapter. Note that this classification is not exhaustive and does not take full account of ongoing attempts to improve the existing methods. Overall, this classification tool has several levels of reading:

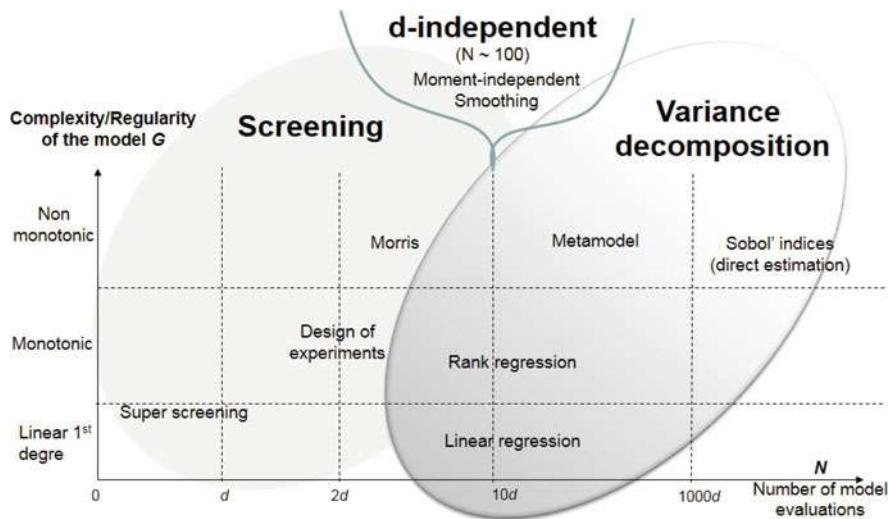


Fig. 31.3 Coarse classification of main global SA methods in terms of required number of model evaluations and model complexity

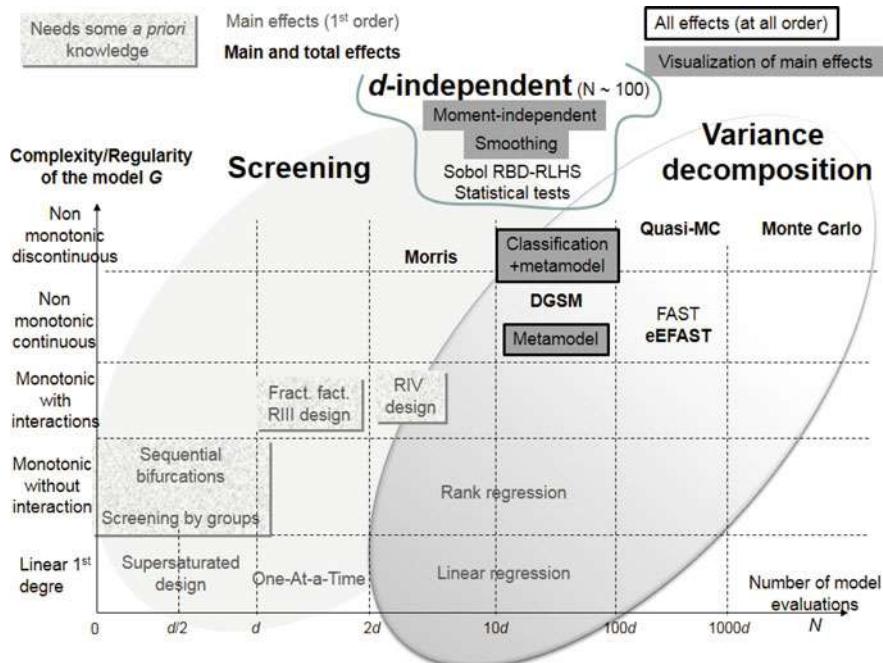


Fig. 31.4 Classification of global SA methods in terms of the required number of model evaluations and model complexity

- positioning methods based on their cost in terms of the number of model calls. Most of these methods linearly depend on the dimension (number of inputs), except for the moment-independent measures (estimated with given-data approaches), smoothing methods, Sobol-RBD (random balance design), Sobol-RLHS (replicated Latin hypercube sampling) and statistical tests;
- positioning methods based on assumptions about model complexity and regularity;
- distinguishing the type of information provided by each method;
- identifying methods which require some prior knowledge about the model behavior.

Each of these techniques corresponds to different categories of problems met in practice. One should use the simplest method that is adapted to the study's objectives, the number of numerical model evaluations that can be performed, and the prior knowledge on the model's regularity. Each sensitivity analysis should include a validation step, which helps to understand if another method should be applied, if the number of model evaluations should be increased, and so on. Based on the characteristics of the different methods, some authors [10, 26] have proposed decision trees to help the practitioner to choose the most appropriate method for their problem and model.

Finally, the different papers of this chapter use the same mathematical notations, which are summarized below:

$G(\cdot)$	Numerical model
N, n	Sample sizes
d	Dimension of an input vector
p	Dimension of an output vector
$\mathbf{x} = (x_1, \dots, x_d)$	Deterministic input vector
$\mathbf{X} = (X_1, \dots, X_d)$	Random input vector
x_j, X_j	Deterministic and random input variable
\mathbf{x}^T	Transpose of \mathbf{x}
$\mathbf{x}_{\sim i} = \mathbf{x}_{-i} = \mathbf{x}_{\bar{i}}$	$= (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_d)$
$\mathbf{x}^{(i)}$	Sample vector of \mathbf{x}
$y, Y \in \mathbb{R}$	Deterministic and random output variable when $p = 1$
$y^{(i)}$	Sample of y
$f_X(\cdot)$	Density function of a real random variable X
$F_X(\cdot)$	Distribution function of X
μ_X, σ_X	Mean and standard deviation of X
$x_\alpha = q_\alpha(X)$	α -quantile of X
$V = \text{Var}(Y)$	Total variance of the model output Y
A	Subset of indices in the set $\{1, \dots, d\}$
V_A	Partial variance
S_A, S_A^{tot}	First order and total Sobol indices of A

4 Specialized R Software Packages

From a practical point of view, the application of SA methods by researchers, engineers, end users, and students is conditioned by the availability of an easy-to-use software. Several software include some SA methods (see the Software chapter of the Springer Handbook), but only a few are specialized on the SA issues. In this section, the **sensitivity** package of the R environment is presented [49]. Its development has started since 2006 and the several contributions that this package has received have made it particularly complete. It includes most of the methods presented in the papers of this chapter.

The R software is a powerful tool for knowledge diffusion in the statistical community. The open source and availability have made R the software of choice for many statisticians in education and industry. The characteristics of R are the following:

- R is easy to run and install on main operating systems. It can be efficiently used with an old computer, with a single workstation and with one of the most recent supercomputer;
- R is a programming language which is interpreted and object oriented and which contains vector operators and matrix computation;
- The main drawbacks of R are its virtual memory limits and non-optimized computation times. To overcome these problems, compiled languages as Fortran or C are much more efficient and can be introduced as compiled codes inside R algorithms requiring huge computation time.
- R contains a lot of built-in statistical functions;
- R is extremely well documented with built-in help system;
- R encourages the collaboration, the discussion forums, and the creation of new packages by researchers and students. Thousands of packages are made available on the CRAN website (<http://cran.r-project.org/>).

All these benefits highlight the interest to develop specific softwares in R and many packages have been developed on SA. For instance, the **FME** package contains basic SA and local SA methods (see ▶ Chap. 32, “Variational Methods”), while the **spartan** package contains basic methods for exploring stochastic numerical models.

For global SA, the **sensitivity** package includes a collection of functions for factor screening, sensitivity index estimation, and reliability sensitivity analysis of model outputs. It implements:

- A few screening techniques as the sequential bifurcations and the Morris method (see ▶ Chap. 33, “Design of Experiments for Screening”). Note that the R package **planor** allows to build fractional factorial design (see ▶ Chap. 33, “Design of Experiments for Screening”);
- The main sampling-based procedures as linear regression coefficients, partial correlations, and rank transformation (see ▶ Chap. 34, “Weights and Importance in Composite Indicators: Mind the Gap”). Note that the **ipcp** function of the R package **ipplots** provides an interactive cobweb graphical tool (see Fig. 31.2),

while the R package **CompModSA** implements various nonparametric regression procedures for SA (see ► Chap. 34, “Weights and Importance in Composite Indicators: Mind the Gap”);

- The variance-based sensitivity indices (Sobol indices), by various schemes of the so-called pick-freeze method, the Extended-FAST method, and the replicated orthogonal array-based Latin hypercube sample (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”). The R package **fast** is fully devoted to the FAST method;
- The Poincaré constants for the derivative-based global sensitivity measures (DGSM) (see ► Chap. 36, “Derivative-Based Global Sensitivity Measures”);
- The sensitivity indices based on Csiszar f-divergence and Hilbert-Schmidt Independence Criterion of [6] (see ► Chap. 37, “Moment-Independent and Reliability-Based Importance Measures”);
- The reliability sensitivity analysis by the perturbation law-based indices (PLI) (see ► Chap. 37, “Moment-Independent and Reliability-Based Importance Measures”);
- The estimation of the Sobol indices with a Gaussian process metamodel (see ► Chap. 38, “Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes”) with a Gaussian process metamodel coming from the R package **DiceKriging**. Note that the R package **tgp** performs the same job using treed Gaussian process and that the R package **GPC** allows to estimate the Sobol indices by building a polynomial chaos metamodel (see ► Chap. 38, “Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes”);
- Sobol indices for multidimensional outputs: Aggregated Sobol indices and functional (1D) Sobol indices (see ► Chap. 39, “Sensitivity Analysis of Spatial and/or Temporal Phenomena”). Note that the R package **multisensi** is fully devoted to this subject, while the R package **safi** implements new SA methods of models with functional inputs;
- The Distributed Evaluation of Local Sensitivity Analysis (DELSA) described in Rakovec et al. [50].

The **sensitivity** package has been designed to work either models written in R than external models such as heavy computational codes. This is achieved with the input argument `model` present in all functions of this package. The argument `model` is expected to be either a function or a predictor (i.e., an object with a `predict` function such as `lm`). The model is invoked once for the whole design of experiment. The argument `model` can be left to `NULL`. This is referred to as the decoupled approach and used with external computational codes that rarely run on the statistician’s computer. Examples of use of all the **sensitivity** functions can be found using the R built-in help system.

As a global and generic platform allowing to include all the methods of these different R packages, the **mtk** package [70] has recently been proposed. It is an object-oriented framework which aims at dealing with external simulation platforms and managing all the different tasks of uncertainty and sensitivity analyses. Finally, the **ATmet** and **pse** packages interface several **sensitivity** package functions for, respectively, metrology applications and parameter space exploration.

5 Sensitivity Auditing

It may happen that a sensitivity analysis of a model-based study is meant to underpin an inference and to certify its robustness, in a context where the inference feeds into a policy or decision-making process. In these cases the framing of the analysis itself, its institutional context, and the motivations of its author may become a matter of great importance, and a pure SA – with its emphasis on parametric uncertainty – may be seen as insufficient. The emphasis on the framing may derive inter alia from the relevance of the policy study to different constituencies that are characterized by different norms and values and hence by a different story about “what the problem is” and foremost about “who is telling the story.” Most often the framing includes more or less implicit assumptions, which could be political (e.g., which group needs to be protected) all the way to technical (e.g., which variable can be treated as a constant).

These concerns about how the story is told and who tells it are all the more urgent in a climate as today’s where science’s own quality assurance criteria are under scrutiny due to a systemic crisis in reproducibility [25], and the explosion of the blogosphere invites more open debates on the scientific basis of policy decisions [42]. The Economist, a weekly magazine, has entered the fray by commenting on the poor state of current scientific practices and devoting its cover to “How Science goes wrong.” It adds that, *The false trails laid down by shoddy research are an unforgivable barrier to understanding* (The Economist [66], p. 11). Among the possible causes of such a predicament is a process of hybridization [33] of fact and values and of public and private institutions and actors. Thus, the classical division of roles among science, providing tested fact, and policy, providing legitimized norms, becomes arduous to maintain.

As an additional difficulty, according to Grundmann [19], *One might suspect that the more knowledge is produced in hybrid arrangements, the more the protagonists will insist on the integrity, even veracity of their findings.*

In order to take these concerns into due consideration, the instruments of SA have been extended to provide an assessment of the entire knowledge and model-generating process. This approach has been called sensitivity auditing. It takes inspiration from NUSAP, a method used to qualify the worth of quantitative information with the generation of “pedigrees” of numbers [17, 69]. Likewise, sensitivity auditing has been developed to provide pedigrees of models and model-based inferences [53, 54, 61].

Sensitivity auditing has been especially designed for an adversarial context, where not only the nature of the evidence but also the degree of certainty and uncertainty associated to the evidence will be the subject of partisan interests. Sensitivity auditing is structured along a set of seven rules/imperatives:

1. Check against the rhetorical use of mathematical modeling. Question addressed: is the model being used to elucidate or to obfuscate?;

2. Adopt an “assumption hunting” attitude. Question addressed: what was “assumed out”? What are the tacit, pre-analytical, possibly normative assumptions underlying the analysis?;
3. Detect garbage in, garbage out (GIGO). Issue addressed: artificial deflation of uncertainty operated in order to achieve a desired inference at a desired level of confidence. It also works on the reverse practice, the artificial inflation of uncertainties, e.g., to deter regulation;
4. Find sensitive assumptions before they find you. Issue addressed: anticipate criticism by doing careful homework via sensitivity and uncertainty analyses before publishing results.
5. Aim for transparency. Issue addressed: stakeholders should be able to make sense of, and possibly replicate, the results of the analysis;
6. Do the right sums, which is more important than “Do the sums right.” Issue addressed: is the viewpoint of a relevant stakeholder being neglected? Who decided that there was a problem and what the problem was?
7. Focus the analysis on the key question answered by the model, exploring the entire space of the assumptions holistically. Issue addressed: don’t perform perfunctory analyses that just “scratch the surface” of the system’s potential uncertainties.

The first rule looks at the instrumental use of mathematical modeling to advance one’s agenda. This use is called rhetorical, or strategic, like the use of Latin by the elites and clergy before the Reformation. At times the use of models is driven a simple pursuit of profit; according to Stiglitz [64], this was the case for the modelers “pricing” the derivatives at the root of the sub-prime mortgages crisis:

[...] Part of the agenda of computer models was to maximize the fraction of, say, a lousy sub-prime mortgage that could get an AAA rating, then an AA rating, and so forth, [...] This was called rating at the margin, and the solution was still more complexity, p. 161.

At times this use of models can be called “ritual,” in the sense that it offers a false sense of reassurance. An example is Fisher [13] (quoting Szenberg [65]):

Kenneth Arrow, one of the most notable Nobel Laureates in economics, has his own perspective on forecasting. During World War II, he served as a weather officer in the U.S. Army Air Corps and worked with a team charged with the particularly difficult task of producing month-ahead weather forecasts. As Arrow and his team reviewed these predictions, they confirmed statistically what you and I might just as easily have guessed: The Corps’ weather forecasts were no more useful than random rolls of a die. Understandably, the forecasters asked to be relieved of this seemingly futile duty. Arrow’s recollection of his superiors’ response was priceless: “The commanding general is well aware that the forecasts are no good. However, he needs them for planning purposes” Szenberg [65].

The second rule about “assumption hunting” is a reminder to look for what was assumed when the model was originally framed. Modes are full of *caeteris paribus* assumptions, meaning that, e.g., in economics the model can predict the result of a

shock to a given set of equations assuming that all the rest – all other input variables and inputs – remains equal, but in real life *caeteris* are never *paribus*, the meaning by this that variables tend to be linked with one another, so that they can hardly change in isolation.

Furthermore, at times the assumption made by modelers do not to withstand scrutiny. A good example of assumption hunting is from John Kay [28], where the author takes issue with modeling used in transport policy. This author discovered that among the input values assumed (and hence fixed) in the model was *average car occupancy rates, differentiated by time of day, in 2035*. The point is that such assumptions are very difficult to justify. This comment was published in the Financial Times (where John Kay is a columnist) showing that at present times controversies that could be called epistemological evade the confines of academia and populate the media.

Rule three is about artificially exaggerating or playing down uncertainties wherever convenient. The tobacco lobbies exaggerated the uncertainties about the health effects of smoking according to Oreskes and Conway [44], while advocates of the death penalty played down the uncertainties in the negative relations between capital punishment and crime rate [34]. Clearly the latter wanted the policy, in this case the death penalty, and were interested in showing that the supporting evidence was robust. In the former case the lobbies did not want regulation (e.g., bans on tobacco smoking in public places) and were hence interested in amplifying the uncertainty in the smoking-health effect causality relationship.

Rule four is about “confessing” uncertainties before going public with the analysis. This rule is also one of the commandments of applied econometrics according to Kennedy [29]: *Thou shall confess in the presence of sensitivity. Corollary: Thou shall anticipate criticism.* According to this rule a sensitivity analysis should be performed before the results of a modeling study are published. There are many good reasons for doing this, one being that a carefully performed sensitivity analysis often uncovers plain coding mistakes or model inadequacies. The other is that most often than not the analysis reveals uncertainties that are larger than those anticipated by the model developers. Econometrician Edward Leamer in discussing this [34] argues that *One reason these methods are rarely used is their honesty seems destructive.* In Saltelli and d’Hombres [52], the negative consequences of doing a sensitivity analysis *a posteriori* are discussed. The case is the first review of the cost of offsetting climate change done by Nicholas Stern of the London School of Economics (the so-called Stern Review) which was criticized by William Nordhaus, of the University of Yale, on the basis of large sensitivity of the estimates upon the discount factors employed by Stern. Stern’s own sensitivity analysis, published as an annex to the review, revealed according to the authors in Saltelli and d’Hombres [52] that while the discount factors were not the only important factors determining the cost estimate, the estimates were indeed very uncertain. For the large uncertainties of integrated assessment models of climate’s impact, see also Saltelli et al. [62].

Rule five is about presenting the results of the modeling study in a transparent fashion. Both rules originate from the practice of impact assessment, where a

modeling study presented without a proper SA, or as originating from a model which is in fact a black box, may end up being rejected by stakeholders [52]. Both rules four and five suggest that reproducibility may be a condition for transparency and that this latter may be a condition for legitimacy. This debate on science's transparency is very much alive in the USA, in the dialectic relationship between the US Environmental Protection Agency (EPA) and the US Congress (especially the Republican Party) which objects to EPA's regulations on the basis that these are based on "secret science."

Rule six, about doing the right sum, is not far from the "assumption hunting" rule; it is just more general. It deals with the fact that often an analyst is set to work on an analysis arbitrarily framed to the advantage of a party. Sometime this comes via the choice of the discipline selected to do the analysis. Thus, an environmental impact problem may be framed through the lenses of economics and presented as a cost benefit or risk analysis, while the issue has little to do with costs or benefits or risks and a lot to do with profits, controls, and norms. An example is in Marris et al. [41] on the issue of GMOs, mostly presented in the public discourse as a food safety issue, while the spectrum of concerns of GMO opponents – including lay citizens – appears broader. According to Winner [71] (pp. 138–163), ecologists should not be led into the trap of arguing about the "safety" of a technology after the technology has been introduced. They should instead question the broader power, policy, and profit implications of that introduction and its desirability.

Rule seven is about avoiding perfunctory sensitivity analyses. As discussed in Saltelli et al. [60], an SA where each uncertain input is moved at a time while leaving all other inputs fixed is perfunctory. A true SA should make an honest effort at activating all uncertainties simultaneously, leaving the model free to display its full nonlinear and possibly nonadditive behavior. A similar point is made in Sam L. Savage's book, *The Flaw of Averages* [63].

In conclusion, these rules are meant to help an analyst to anticipate criticism. In drafting these rules, the authors in [53, 54, 61] have tried to put themselves in the shoes of a modeler also based on their own experience and tried to imagine a model-based inference feeding into an impact assessment. What questions and objections may be received by the modeler? Here is a possible list:

- "You treated X as a constant when we know it is uncertain by at least 30%"
- "It would be sufficient for a 5% error in X to make your statement about Z fragile"
- "Your model is but one of the plausible models – you neglected model uncertainty"
- "You have instrumentally maximized your level of confidence in the results"
- "Your model is a black box – why should I trust your results?"
- "You have artificially inflated the uncertainty"
- "Your framing is not socially robust"
- "You are answering the wrong question"

The reader may easily check that carefully going through the sensitivity auditing checklist should provide ammunition to anticipate objections of this nature.

Sensitivity auditing can then be seen as a user guide to criticize the model-based studies, and SA is a part of this guide. In the following section, we go back to sensitivity analysis in order to conclude and give some perspectives.

6 Conclusion

This introductory paper has presented the SA aims and objectives, the SA and sensitivity auditing basic principles, the different methods explained in the chapter papers by positioning them in a classification grid, and the useful R software packages and the notations used in the chapter papers. The chapter's Editor, Bertrand Iooss, would like to sincerely thank all authors and co-authors of the "Sensitivity Analysis" chapter for their efforts and the quality of their contributions. Dr. Jean-Philippe Argaud (EDF R&D), Dr. Géraud Blatman (EDF R&D), Dr. Nicolas Bousquet (EDF R&D), Dr. Sébastien da Veiga (Safran), Dr. Hervé Monod (INRA), Dr. Matieyendou Lamboni (INRA), and Dr. Loïc Le Gratiet (EDF R&D) are also greatly thanked for their advices on the different chapter papers.

The papers in this chapter include the most widely used methods for sensitivity analysis: the deterministic methods as the local sensitivity analysis, the experimental design strategies, the sampling-based and variance-based methods developed from the 1980s, and the new importance measures and metamodel-based techniques established and studied since the 2000s. However, with such a rich subject, choices had to be made for the different chapter papers, and some important omissions are present. For instance, the robust Bayesian analysis [1, 24] is not discussed, while it is nevertheless a great ingredient in the study of the sensitivity of Bayesian answers to assumptions and uncertain calculation inputs. Moreover, in the context of non-probabilistic representation of uncertainty (as in the interval analysis, the evidence theory and the possibility theory), a small amount of SA methods has been developed [22]. This subject is deferred to a future review work.

Due to their very new or incomplete nature, several other SA issues are not discussed in this chapter. For instance, estimating total Sobol indices at low cost remains a problem of primary importance in many applications (see Saltelli et al. [60] for a recent review on the subject). Second-order Sobol index estimations have recently been considered by Fruth et al. [16] (by way of total interaction indices) and Tissot and Prieur [67] (by way of replicated orthogonal-array LHS). The latter work offers a powerful estimation method because the number of model calls is independent of the number of inputs (as in the spirit of permutation-based technique [37, 38]). However, for high-dimensional models (several hundreds inputs), estimation biases and computational costs remain considerable; De Castro and Janon [9] have proposed to introduce modern statistical techniques based on variable selection in regression models. Owen [46] has introduced generalized Sobol indices allowing to compare and search efficient estimators (as the new one found in Owen [45]).

Another mathematical difficulty is the consideration of the dependence between the inputs X_i ($i = 1 \dots d$). Nonindependence between inputs in SA has been

discussed by many authors such as Saltelli and Tarantola [55], Jacques et al. [27], Xu and Gertner [72], Da Veiga et al. [7], Li et al. [36], Kucherenko et al. [31], Mara and Tarantola [39], and Chastaing et al. [5]. Despite all these works, much confusion still exists in practical applications. In practice, it would be useful to be able to measure the influence of the potential dependence between some inputs on the output quantity of interest.

Note that most of the works in this chapter focus on SA relative to the overall variability of model output (second-order statistics). In practice, one can be interested in other quantities of interest, such as the output entropy, the probability that the output exceeds a threshold and a quantile estimation [15,35,56]. This is an active area of research as shown in this chapter (see ► Chap. 37, “[Moment-Independent and Reliability-Based Importance Measures](#)”), but innovative and powerful ideas have recently been developed by Owen et al. [47] and Geraci et al. [18] using higher-order statistics, Fort et al. [14] using contrast functions, and Da Veiga [6] using a kernel point of view.

Finally, in some situations, the computer code is not a deterministic simulator but a stochastic one. This means that two model calls with the same set of input variables lead to different output values. Typical stochastic computer codes are queuing models, agent-based models, and models involving partial differential equations applied to heterogeneous or Monte Carlo-based numerical models. For this type of code, Marrel et al. [40] have proposed a first solution for dealing with Sobol indices. Moutoussamy et al. [43] have also tackled the issue in the context of the metamodel building of the output probability density function. Developing relevant SA methods in this context will certainly be subject of future works.

References

1. Berger, J.: An overview of robust Bayesian analysis (with discussion). *Test* **3**, 5–124 (1994)
2. Borgonovo, E., Plischke, E.: Sensitivity analysis: a review of recent advances. *Eur. J. Oper. Res.* **248**, 869–887 (2016)
3. Cacuci, D.: *Sensitivity and Uncertainty Analysis – Theory*. Chapman & Hall/CRC, Boca Raton (2003)
4. Castaings, W., Dartus, D., Le Dimet, F.X., Saulnier, G.M.: Sensitivity analysis and parameter estimation for distributed hydrological modeling: potential of variational methods. *Hydrol. Earth Syst. Sci. Discuss.* **13**, 503–517 (2009)
5. Chastaing, G., Gamboa, F., Prieur, C.: Generalized Hoeffding-Sobol decomposition for dependent variables – application to sensitivity analysis. *Electron. J. Stat.* **6**, 2420–2448 (2012)
6. Da Veiga, S.: Global sensitivity analysis with dependence measures. *J. Stat. Comput. Simul.* **85**, 1283–1305 (2015)
7. Da Veiga, S., Wahl, F., Gamboa, F.: Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics* **51**(4), 452–463 (2009)
8. Dean, A., Lewis, S. (eds.): *Screening – Methods for Experimentation in Industry, Drug Discovery and Genetics*. Springer, New York (2006)
9. De Castro, Y., Janon, A.: Randomized pick-freeze for sparse Sobol indices estimation in high dimension. *ESAIM Probab. Stat.* **19**, 725–745 (2015)
10. de Rocquigny, E., Devictor, N., Tarantola, S. (eds.): *Uncertainty in Industrial Practice*. Wiley, Chichester/Hoboken (2008)

11. Faivre, R., Iooss, B., Mahévas, S., Makowski, D., Monod, H. (eds.): Analyse de sensibilité et exploration de modèles. Éditions Quaé (2013)
12. Fang, K.T., Li, R., Sudjianto, A.: Design and Modeling for Computer Experiments. Chapman & Hall/CRC, Boca Raton (2006)
13. Fisher, R.W.: Remembering Carol Reed, Aesop's Fable, Kenneth Arrow and Thomas Dewey. In: Speech: An Economic Overview: What's Next, Federal Reserve Bank of Dallas. <http://www.dallasfed.org/news/speeches/fisher/2011/fs110713.cfm> (2011)
14. Fort, J., Klein, T., Rachdi, N.: New sensitivity analysis subordinated to a contrast. Commun. Stat. Theory Methods (2014, in press). <http://www.tandfonline.com/doi/full/10.1080/03610926.2014.901369#abstract>
15. Frey, H., Patil, S.: Identification and review of sensitivity analysis methods. Risk Anal. **22**, 553–578 (2002)
16. Fruth, J., Roustant, O., Kuhnt, S.: Total interaction index: a variance-based sensitivity index for second-order interaction screening. J. Stat. Plan. Inference **147**, 212–223 (2014)
17. Funtowicz, S., Ravetz, J.: Uncertainty and Quality in Science for Policy. Kluwer Academic, Dordrecht (1990)
18. Geraci, G., Congedo, P., Iaccarino, G.: Decomposing high-order statistics for sensitivity analysis. In: Thermal & Fluid Sciences Industrial Affiliates and Sponsors Conference, Stanford University, Stanford (2015)
19. Grundmann, R.: The role of expertise in governance processes. For. Policy Econ. **11**, 398–403 (2009)
20. Helton, J.: Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. Reliab. Eng. Syst. Saf. **42**, 327–367 (1993)
21. Helton, J.: Uncertainty and sensitivity analysis for models of complex systems. In: Graziani, F. (ed.) Computational Methods in Transport: Verification and Validation, pp. 207–228. Springer, New-York (2008)
22. Helton, J., Johnson, J., Obekampf, W., Salaberry, C.: Sensitivity analysis in conjunction with evidence theory representations of epistemic uncertainty. Reliab. Eng. Syst. Saf. **91**, 1414–1434 (2006a)
23. Helton, J., Johnson, J., Salaberry, C., Storlie, C.: Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliab. Eng. Syst. Saf. **91**, 1175–1209 (2006b)
24. Insua, D., Ruggeri, F. (eds.): Robust Bayesian Analysis. Springer, New York (2000)
25. Ioannidis, J.P.A.: Why most published research findings are false. PLoS Med. **2**(8), 696–701 (2005)
26. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Meloni, C., Dellino, G. (eds.) Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications. Springer, New York (2015)
27. Jacques, J., Lavergne, C., Devictor, N.: Sensitivity analysis in presence of model uncertainty and correlated inputs. Reliab. Eng. Syst. Saf. **91**, 1126–1134 (2006)
28. Kay, J.: A wise man knows one thing – the limits of his knowledge. Financial Times 29 Nov 2011
29. Kennedy, P.: A Guide to Econometrics, 5th edn. Blackwell Publishing, Oxford (2007)
30. Kleijnen, J.: Sensitivity analysis and related analyses: a review of some statistical techniques. J. Stat. Comput. Simul. **57**, 111–142 (1997)
31. Kucherenko, S., Tarantola, S., Annoni, P.: Estimation of global sensitivity indices for models with dependent variables. Comput. Phys. Commun. **183**, 937–946 (2012)
32. Kurowicka, D., Cooke, R.: Uncertainty Analysis with High Dimensional Dependence Modelling. Wiley, Chichester/Hoboken (2006)
33. Latour, B.: We Have Never Been Modern. Harvard University Press, Cambridge (1993)
34. Leamer, E.E.: Tantalus on the road to asymptopia. J. Econ. Perspect. **4**(2), 31–46 (2010)
35. Lemaître, P., Sergienko, E., Arnaud, A., Bousquet, N., Gamboa, F., Iooss, B.: Density modification based reliability sensitivity analysis. J. Stat. Comput. Simul. **85**, 1200–1223 (2015)

36. Li, G., Rabitz, H., Yelvington, P., Oluwole, O., Bacon, F., Kolb, C., Schoendorf, J.: Global sensitivity analysis for systems with independent and/or correlated inputs. *J. Phys. Chem.* **114**, 6022–6032 (2010)
37. Mara, T.: Extension of the RBD-FAST method to the computation of global sensitivity indices. *Reliab. Eng. Syst. Saf.* **94**, 1274–1281 (2009)
38. Mara, T., Joseph, O.: Comparison of some efficient methods to evaluate the main effect of computer model factors. *J. Stat. Comput. Simul.* **78**, 167–178 (2008)
39. Mara, T., Tarantola, S.: Variance-based sensitivity indices for models with dependent inputs. *Reliability Engineering and System Safety* **107**, 115–121 (2012)
40. Marrel, A., Iooss, B., Da Veiga, S., Ribatet, M.: Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput.* **22**, 833–847 (2012)
41. Marris, C., Wynne, B., Simmons, P., Weldon, S.: Final report of the PABE research project funded by the Commission of European Communities. Technical report contract number: FAIR CT98-3844 (DG12 – SSMI), Commission of European Communities (2001)
42. Monbiot, G.: Beware the rise of the government scientists turned lobbyists. *The Guardian* 29 Apr 2013
43. Moutoussamy, V., Nanty, S., Pauwels, B.: Emulators for stochastic simulation codes. *ESAIM: Proc. Surv.* **48**, 116–155 (2015)
44. Oreskes, N., Conway, E.M.: *Merchants of Doubt: How a Handful of Scientists Obscured the Truth on Issues from Tobacco Smoke to Global Warming*. Bloomsbury Press, New York (2010)
45. Owen, A.: Better estimation of small Sobol' sensitivity indices. *ACM Trans. Model. Comput. Simul.* **23**, 11 (2013a)
46. Owen, A.: Variance components and generalized Sobol' indices. *J. Uncert. Quantif.* **1**, 19–41 (2013b)
47. Owen, A., Dick, J., Chen, S.: Higher order Sobol' indices. *Inf. Inference: J. IMA* **3**, 59–81 (2014)
48. Park, K., Xu, L.: *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*. Springer, Dordrecht (2008)
49. Pujol, G., Iooss, B., Janon, A.: Sensitivity Package, Version 1.11. The Comprehensive R Archive Network. <http://www.cran.r-project.org/web/packages/sensitivity/> (2015)
50. Rakovec, O., Hill, M.C., Clark, M.P., Weerts, A.H., Teuling, A.J., Uijlenhoet, R.: Distributed evaluation of local sensitivity analysis (DELSA), with application to hydrologic models. *Water Resour. Res.* **50**, 1–18 (2014)
51. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002)
52. Saltelli, A., d'Hombres, B.: Sensitivity analysis didn't help. A practitioners critique of the Stern review. *Glob. Environ. Change* **20**(2), 298–302 (2010)
53. Saltelli, A., Funtowicz, S.: When all models are wrong: more stringent quality criteria are needed for models used at the science-policy interface. *Issues Sci. Technol.* **XXX**(2), 79–85 (2014, Winter)
54. Saltelli, A., Funtowicz, S.: Evidence-based policy at the end of the Cartesian dream: the case of mathematical modelling. In: Pereira, G., Funtowicz, S. (eds.) *The End of the Cartesian Dream. Beyond the Techno–Scientific Worldview*. Routledge's Series: Explorations in Sustainability and Governance, pp. 147–162. Routledge, London (2015)
55. Saltelli, A., Tarantola, S.: On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. *J. Am. Stat. Assoc.* **97**, 702–709 (2002)
56. Saltelli, A., Chan, K., Scott, E. (eds.): *Sensitivity Analysis*. Wiley Series in Probability and Statistics. Wiley, Chichester/New York (2000a)
57. Saltelli, A., Tarantola, S., Campolongo, F.: Sensitivity analysis as an ingredient of modelling. *Stat. Sci.* **15**, 377–395 (2000b)
58. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, Chichester/Hoboken (2004)

59. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Salsana, M., Tarantola, S.: *Global Sensitivity Analysis – The Primer*. Wiley, Chichester (2008)
60. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**, 259–270 (2010)
61. Saltelli, A., Pereira, G., Van der Sluijs, J.P., Funtowicz, S.: What do I make of your latinorum? Sensitivity auditing of mathematical modelling. *Int. J. Foresight Innov. Policy* **9**(2/3/4), 213–234 (2013)
62. Saltelli, A., Stark, P., Becker, W., Stano, P.: Climate models as economic guides. Scientific challenge or quixotic quest? *Issues Sci. Technol.* **XXXI**(3), 79–84 (2015)
63. Savage, S.L.: *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*. Wiley, Hoboken (2009)
64. Stiglitz, J.: *Freefall, Free Markets and the Sinking of the Global Economy*. Penguin, London (2010)
65. Szenberg, M.: *Eminent Economists: Their Life Philosophies*. Cambridge University Press, Cambridge (1992)
66. The Economist: How science goes wrong. *The Economist* 19 Oct 2013
67. Tissot, J.Y., Prieur, C.: A randomized orthogonal array-based procedure for the estimation of first- and second-order Sobol' indices. *J. Stat. Comput. Simul.* **85**, 1358–1381 (2015)
68. Turanyi, T.: Sensitivity analysis for complex kinetic system, tools and applications. *J. Math. Chem.* **5**, 203–248 (1990)
69. Van der Sluijs, J.P., Craye, M., Funtowicz, S., Kloprogge, P., Ravetz, J., Risbey, J.: Combining quantitative and qualitative measures of uncertainty in model based environmental assessment: the NUSAP system. *Risk Anal.* **25**(2), 481–492 (2005)
70. Wang, J., Faivre, R., Richard, H., Monod, H.: mtk: a general-purpose and extensible R environment for uncertainty and sensitivity analyses of numerical experiments. *R J.* **7/2**, 206–226 (2016)
71. Winner, L.: *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. The University of Chicago Press, Chicago (1989)
72. Xu, C., Gertner, G.: Extending a global sensitivity analysis technique to models with correlated parameters. *Comput. Stat. Data Anal.* **51**, 5579–5590 (2007)

Maelle Nodet and Arthur Vidard

Abstract

This contribution presents derivative-based methods for local sensitivity analysis, called *Variational Sensitivity Analysis* (VSA). If one defines an output called the *response function*, its sensitivity to input variations around a nominal value can be studied using derivative (gradient) information. The main issue of VSA is then to provide an efficient way of computing gradients.

This contribution first presents the theoretical grounds of VSA: framework and problem statement and tangent and adjoint methods. Then it covers practical means to compute derivatives, from naive to more sophisticated approaches, discussing their various merits. Finally, applications of VSA are reviewed, and some examples are presented, covering various applications fields: oceanography, glaciology, and meteorology.

Keywords

Variational sensitivity analysis • Variational methods • Tangent model • Adjoint model • Gradient • Automatic differentiation • Derivative • Local sensitivity analysis • Stability analysis • Geophysical applications • Meteorology • Glaciology • Oceanography

Contents

1	Introduction	1124
2	Methods	1125
2.1	Problem Statement	1125
2.2	Tangent and Adjoint Models	1126

M. Nodet (✉) • A. Vidard

Laboratoire Jean Kuntzmann (LJK), University Grenoble Alpes, Grenoble, France

INRIA, Rocquencourt, France

e-mail: maelle.nodet@inria.fr; arthur.vidard@inria.fr

2.3	Practical Gradient Computation	1130
2.4	Stability Analysis	1134
3	Applications	1135
3.1	Sensitivity to Initial or Boundary Condition Changes	1135
3.2	Parameter Sensitivity	1138
3.3	Sensitivity of Complex Systems	1139
4	Conclusion	1140
5	Cross-References	1141
	References	1141

1 Introduction

This contribution presents derivative-based methods for local sensitivity analysis, gathered under the name *Variational Sensitivity Analysis* (VSA). The aim of VSA is to provide sensitivity information using derivatives. This is indeed a valuable information, as the derivative of a function at a given point gives the growth rate at that point, in other words the tendency of the function to grow (or not) when the input varies. Approximately one could say *the larger the derivative, the more sensitive the parameter*.

Note that VSA can be extended to global analysis (GSA) and the reader is referred to contribution (see ▶ Chap. 36, “Derivative-Based Global Sensitivity Measures”). Here the focus will be solely on local derivative-based sensitivity analysis. Contrary to GSA, LSA aims to compute sensitivities when the parameters vary *locally* around their nominal values and not *globally* over a potentially large subset.

VSA is closely related to the research domain called *data assimilation*. This one consists in adjusting input parameters of a model so that the system state fits a given set of observations (data). Variational data assimilation translates this into an optimal control problem whose aim is to minimize the model-observation misfit. This minimization is performed using descent methods, and the gradient is computed using the so-called adjoint method, which is also at the core of VSA. Moreover, improving parameters and models through data assimilation assumes that the most sensitive parameters are known, which in turn creates the need to perform VSA beforehand.

This contribution is divided in two parts. First, it will cover the methods of local VSA: the derivative is first defined, then the adjoint method is shown to provide a powerful way to compute it, then a brief overview about practical derivatives computation is given, and finally stability analysis is mentioned, which is closely related to sensitivity analysis.

In a second part, some applications are presented. VSA has been used in a wide range of domains: e.g., meteorology [1, 5, 8, 23, 33, 34], cyclone tracking [14, 22, 32], air quality [25, 26], oceanography [2, 27, 30, 31], surface hydrology [4], groundwater modeling [28], glaciology [13], agronomy [15], chemistry [24], etc. Historically, the adjoint method was first applied to numerical weather prediction, so that meteorology is a primary application domain, with many references on VSA.

As in other geophysical domains, meteorological models are in general of very large dimension, so that GSA is mostly out of reach, which motivates the introduction of the adjoint method and VSA. This contribution chose to focus on a small number of example applications in geophysics.

2 Methods

2.1 Problem Statement

In this section, the sensitivity of the output vector \mathbf{y} with respect to the input vector \mathbf{x} is considered. This output is often called the *response function*, since it represents the response of the system to variation on the input. In the variational framework, practitioners are generally interested in studying sensitivities of given numerical models \mathcal{M} coming from various application domains (physics, geosciences, biology, etc.), so that the output is a function of a state vector $\mathbf{u}(\mathbf{x}; t) \in \mathbb{R}^p$, which depends on the input $\mathbf{x} \in \mathbb{R}^d$ and on time t and represents the state of a given system of interest:

$$\begin{cases} \frac{d\mathbf{u}}{dt} = \mathcal{M}(\mathbf{u}; \mathbf{x}), & t \in [0, T] \\ \mathbf{u}(t = 0) = \mathbf{u}_0(\mathbf{x}) \end{cases} \quad (32.1)$$

As \mathbf{u} lies in \mathbb{R}^p , the model \mathcal{M} is a (possibly nonlinear) operator from $\mathbb{R}^p \times \mathbb{R}^d$ to \mathbb{R}^p .

In that case, the dependency between the input and the output reads

$$\mathbf{y} = G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x})) \quad (32.2)$$

where G is the response function and \mathcal{G} maps the state space into the output space.

Remark 1. In the definition of the response function G lies all the art of sensitivity analysis. It should be designed carefully depending on the aim of the study (model validation, physical phenomenon study, etc.). This is discussed in depth in the contributions ▶ [Chap. 39, “Sensitivity Analysis of Spatial and/or Temporal Phenomena”](#) and [Variables Weights and Importance in Arithmetic Averages: Two Stories to Tell](#).

Remark 2. A more general case would be $\mathbf{y} = \mathcal{G}(\mathbf{u}(\mathbf{x}), \mathbf{x})$. The theory easily extends to that case, but for readability, only this simpler case will be presented here.

In order to simplify the notations and clarify the reading, scalar outputs will be considered in this chapter, i.e., $\mathbf{y} \in \mathbb{R}$, but vector-valued outputs can of course be considered as well.

Variational sensitivity consists in finding the sensitivity of G with respect to the variations of \mathbf{x} , in other words the derivative of G with respect to the vector \mathbf{x} :

$$\frac{dG}{d\mathbf{x}}(\mathbf{x})$$

In this framework, G is differentiable from \mathbb{R}^d to \mathbb{R} , with continuous derivative, i.e., G is of class C^1 . Then the derivative can be identified (using the Euclidean scalar product in \mathbb{R}^d) with the gradient:

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = \left(\frac{\partial G}{\partial x_1}(\mathbf{x}), \frac{\partial G}{\partial x_2}(\mathbf{x}), \dots, \frac{\partial G}{\partial x_d}(\mathbf{x}) \right)^T$$

The partial derivatives of G are particular cases of the directional derivative, also called the Gâteau derivative, which is defined by $G'(\mathbf{x})[\mathbf{h}]$ such that

$$G'(\mathbf{x})[\mathbf{h}] = \lim_{\alpha \rightarrow 0} \frac{G(\mathbf{x} + \alpha \mathbf{h}) - G(\mathbf{x})}{\alpha}$$

where the direction \mathbf{h} is a vector of \mathbb{R}^d . The partial derivative is simply the directional derivative in the direction of a basis vector; it is given by

$$G'(\mathbf{x})[\mathbf{e}_i] = \frac{\partial G}{\partial x_i}(\mathbf{x})$$

As G is a continuously differentiable function, the link between the directional derivative and the gradient is immediate:

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}} G(\mathbf{x}) \cdot \mathbf{h}$$

where “ \cdot ” represents the Euclidean scalar product in \mathbb{R}^d .

2.2 Tangent and Adjoint Models

The most naive approach to track sensitive variables consists in fixing all parameters except one, increasing it by a given percentage (of its standard deviation or its absolute value) and then evaluating its impact on the output. This type of analysis can allow for a quick ranking of the variables if there are not too many of them. The reader can refer to [11] for a brief review on this subject.

A more refined approach would be to obtain a numerical approximation of the gradient using finite differences, i.e., computing the gradient as a limit of a growth rate (see [11] and next paragraph about practical aspects). This method is very simple but has two main drawbacks. First, its computational cost increases rapidly

with the dimension d of \mathbf{x} . Second, the choice of δx_i is critical: if too large, the truncation error becomes large; if too small, rounding error occurs.

To address this last point, one can obtain an exact calculation for the Gâteau derivatives (FSAP, *Forward Sensitivity Analysis Procedure* in [3]’s terminology). Assuming the output is given by Eqs. (32.1) and (32.2),

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{u}} \mathcal{G}(\mathbf{u}(\mathbf{x})).\mathbf{u}'(\mathbf{x})[\mathbf{h}] = \mathcal{G}'(\mathbf{u}(\mathbf{x}))[\mathbf{u}'(\mathbf{x})[\mathbf{h}]]$$

where $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$ is the Gâteau derivative of \mathbf{u} at \mathbf{x} in the direction \mathbf{h} . If \mathbf{v} denotes $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$, then \mathbf{v} is given by the following equations, called the Tangent Linear Model (TLM):

$$\begin{cases} \frac{d\mathbf{v}}{dt} = \frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}).\mathbf{v} + \frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}).\mathbf{h}, & t \in [0, T] \\ \mathbf{v}(t=0) = \mathbf{u}'_0(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}} \mathbf{u}_0. \mathbf{h} \end{cases} \quad (32.3)$$

where $\frac{\partial \mathcal{M}}{\partial \mathbf{u}}$ and $\frac{\partial \mathcal{M}}{\partial \mathbf{x}}$ are the Jacobian matrices of the model with respect to the state \mathbf{u} and the parameters \mathbf{x} . The Tangent Linear Model allows to compute exactly the directional derivative of the response function G , for a given direction \mathbf{h} . To get the entire gradient, all the partial derivatives need to be computed; therefore, d integrations of the TLM are required. The accuracy problem may be solved, but for a large-dimensional set of parameters, the computing cost of the FSAP method remains prohibitive.

In large-dimensional cases, however, an adjoint method can be used to compute the gradient (ASAP, *Adjoint Sensitivity Analysis Procedure* in [3]’s terminology). As the derivation of the adjoint model is tedious in the general abstract case, this contribution will focus on common examples.

One will first consider the case where G and \mathcal{G} are given by

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x})) = \int_{t=0}^{t=T} \mathcal{H}(\mathbf{u}(\mathbf{x}; t)) dt \quad (32.4)$$

where \mathcal{H} is the single time output function, from \mathbb{R}^p to \mathbb{R} .

Then the Gâteau derivative of G with respect to \mathbf{x} in the direction \mathbf{h} is given by

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{h} = \int_{t=0}^{t=T} \nabla_{\mathbf{u}} \mathcal{H}(\mathbf{u}(\mathbf{x}; t)).\mathbf{v} dt \quad (32.5)$$

where \mathbf{v} is as before $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$.

Now the adjoint method consists in introducing a wisely chosen so-called adjoint variable so that the previous gradient can be formulated without the tangent variable \mathbf{v} . To do so, the tangent model (32.3) is multiplied by a variable $\mathbf{p}(t)$, and an integration by parts is performed in order to obtain a formula $\int_{t=0}^{t=T} (\dots).\mathbf{v} dt$, which

will later be identified with (32.5). So first a multiplication by \mathbf{p} is performed and then an integration:

$$0 = - \int_0^T \mathbf{p} \frac{d\mathbf{v}}{dt} dt + \int_0^T \mathbf{p} \left(\frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \cdot \mathbf{v} + \frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \cdot \mathbf{h} \right) dt$$

Then the integration by parts gives

$$\begin{aligned} 0 &= -\mathbf{p}(T)\mathbf{v}(T) + \mathbf{p}(0)\mathbf{v}(0) + \int_0^T \left(\frac{d\mathbf{p}}{dt} + \frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x})^T \mathbf{p} \right) \cdot \mathbf{v} dt \\ &\quad + \int_0^T \left(\left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right)^T \mathbf{p} \right) \cdot \mathbf{h} dt \end{aligned}$$

Using the initial condition (32.3) on \mathbf{v} , it becomes

$$\begin{aligned} \mathbf{p}(T)\mathbf{v}(T) - \int_0^T \left(\frac{d\mathbf{p}}{dt} + \frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x})^T \mathbf{p} \right) \cdot \mathbf{v} dt \\ = \mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0 \cdot \mathbf{h} + \int_0^T \left(\left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right)^T \mathbf{p} \right) \cdot \mathbf{h} dt \\ = \left(\mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0 + \int_0^T \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right)^T \mathbf{p} dt \right) \cdot \mathbf{h} \end{aligned} \quad (32.6)$$

As Eq. (32.5) needs to be rewritten, the following backward equation for \mathbf{p} is set and called the adjoint model:

$$\begin{cases} -\frac{d\mathbf{p}}{dt} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \cdot \mathbf{p} + \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x}); t), & t \in [0, T] \\ \mathbf{p}(t = T) = 0 \end{cases} \quad (32.7)$$

Combining (32.5), (32.6), and (32.7), the following formula for the Gâteau derivative is obtained:

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}}G(\mathbf{x}) \cdot \mathbf{h} = \left(\mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0 + \int_0^T \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right)^T \mathbf{p} dt \right) \cdot \mathbf{h}$$

leading to the gradient:

$$\nabla_{\mathbf{x}}G(\mathbf{x}) = [\nabla_{\mathbf{x}}\mathbf{u}_0]^T \mathbf{p}(t = 0) + \int_{t=0}^{t=T} \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \mathbf{p}(t) dt \quad (32.8)$$

which does not involve variable \mathbf{v} anymore. Thus, the computing cost is independent from the number of parameters, and $\nabla_{\mathbf{x}}G(\mathbf{x})$ can be computed *exactly* using one

integration of the direct model and one backward integration of the adjoint model. Note that if the model is linear, only the adjoint integration is needed.

Similarly, the adjoint model can be computed for the following type of output:

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x}; t_1)) \quad (32.9)$$

for $t_1 \in]0; T[$, by noticing that it can be written as

$$G(\mathbf{x}) = \int_{t=0}^{t=T} \mathcal{H}(\mathbf{u}(\mathbf{x}; t)) \delta_{t=t_1}(t) dt$$

where $\delta_{t=t_1}(t)$ is the Dirac function of t at point t_1 . The adjoint model is then

$$\begin{cases} -\frac{d\mathbf{p}}{dt} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \cdot \mathbf{p} + \nabla_{\mathbf{u}} \mathcal{H}(\mathbf{u}(\mathbf{x}; t)) \delta_{t=t_1}(t), & t \in [0, T] \\ \mathbf{p}(t = T) = 0 \end{cases} \quad (32.10)$$

Notice here that the adjoint variable is equal to zero up to time t_1 . Computationally speaking, it is therefore sufficient to run the adjoint model from t_1 to 0.

Then the gradient is unchanged:

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = [\nabla_{\mathbf{x}} \mathbf{u}_0]^T \mathbf{p}(t = 0) + \int_{t=0}^{t=T} \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \mathbf{p}(t) dt \quad (32.11)$$

One can also be interested in an output at final time:

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x}; T)), \quad G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{u}} \mathcal{H}(\mathbf{u}(\mathbf{x}; T)) \cdot \mathbf{v}(T)$$

Then the adjoint model writes

$$\begin{cases} -\frac{d\mathbf{p}}{dt} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \cdot \mathbf{p}, & t \in [0, T] \\ \mathbf{p}(t = T) = \nabla_{\mathbf{u}} \mathcal{H}(\mathbf{u}(\mathbf{x}; T)) \end{cases} \quad (32.12)$$

and the gradient is unchanged:

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = [\nabla_{\mathbf{x}} \mathbf{u}_0]^T \mathbf{p}(t = 0) + \int_{t=0}^{t=T} \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}) \right]^T \mathbf{p}(t) dt \quad (32.13)$$

Note here that this adjoint method allows to obtain directly every component of the gradient vector with a single run of the adjoint model, thus avoiding the curse of dimensionality on the parameter set dimension.

2.3 Practical Gradient Computation

Practical aspects are an important incentive for the choice of either of gradient computation methods presented above. For a small dimension problem, the finite difference approximation route is pretty straightforward, and unless a high precision is required, it is probably the better choice. For larger dimension problems, however, either choice will require some efforts in terms of computing cost and/or code developments. This section presents practical aspects of such developments, as well as a further approximation of the finite difference method tailored for high dimension problems.

2.3.1 Finite Difference Approximation

By definition, one has

$$\frac{\partial G}{\partial x_i}(\mathbf{x}) = \lim_{\alpha_i \rightarrow 0} \frac{G(\mathbf{x} + (0, \dots, \alpha_i, \dots, 0)^T) - G(\mathbf{x})}{\alpha_i}$$

Thus, the partial derivative (and therefore the gradient) can be numerically approximated by

$$\frac{\partial G}{\partial x_i}(\mathbf{x}) \approx \frac{G(\mathbf{x} + (0, \dots, \delta x_i, \dots, 0)^T) - G(\mathbf{x})}{\delta x_i}$$

where δx_i is “small”; see, e.g., [11]. As it has been mentioned before, this method, although very simple, has two main drawbacks (computational cost, rounding issues), which motivates the need for the adjoint code.

2.3.2 Discrete vs. Continuous Adjoint

There are two approaches to obtain an adjoint code:

- Discretize the continuous direct model, and then write the adjoint of the discrete direct model. This is generally called the *discrete adjoint* approach; see, e.g., [17].
- Write the continuous adjoint from the continuous direct model (as explained before), and then discretize the continuous adjoint equations. This is called the *continuous adjoint approach*.

The two approaches are not equivalent. As an example, a simple ordinary differential equation can be considered:

$$c'(t) = F(t).c(t)$$

where F is a time-dependant linear operator acting on the vector $c(t)$. If this model is discretized using a forward Euler scheme, a typical time step writes as

$$c_{n+1} = (I + \Delta_t F_n)c_n$$

where I is the identity matrix and n the time index. Then the adjoint of this discrete equation would give the following typical time step (*discrete adjoint*):

$$c_n^* = (I + \Delta_t F_n)^T c_{n+1}^*$$

where $*$ denotes the adjoint variable. On the other hand, the continuous adjoint equation is

$$-c^{*'}(t) = F(t)^T \cdot c^*(t)$$

If the same forward Euler explicit scheme is chosen, one obtains for the *continuous adjoint*

$$c_n^* = (I + \Delta_t F_{n+1})^T c_{n+1}^*$$

and the time dependency on F then implies that both approaches are not identical. This illustrates the fact that discretization and transposition (the word “adjointization” does not exist; the process of obtaining an adjoint code is called “transposition” or “derivation”) do not, in general, commute.

The choice of the discrete adjoint approach should be immediate for two main reasons:

1. The response function $G(\mathbf{x})$ is computed through the discrete direct model; its gradient is therefore given by the *discrete adjoint*. The *continuous adjoint* gives only an approximation of this gradient.
2. The discrete adjoint approach allows for the use of automatic adjoint compilers (software that takes in input the direct model code and produces as output the tangent and adjoint codes, e.g., Tapenade [12]).

However, it must be noted here that, for large complex systems, obtaining the discrete adjoint can be a time- and expertise-demanding task. Therefore, if one has limited time and experience in adjoint coding and if one can be satisfied with just an approximation of the gradient, the continuous adjoint approach can be considered. Moreover, for complex nonlinear or non-differentiable equations, the discrete adjoint has been shown in [29] to present problems to compute the sensitivities, so that other approaches may be considered. Similarly, [18] presents an application with adaptive mesh, where it is preferable to go through the continuous adjoint route. In any case, the validation of the gradient (see below) should give a good idea about the quality of the chosen gradient.

2.3.3 Adjoint Code Derivation

As it has been mentioned above, obtaining a *discrete* adjoint code is a complex task. A numerical code is ultimately a sequence of single-line instructions. In other words, a code can be seen as a composition of a (large) number of functions, each line code representing one function in this composition. To obtain the tangent code

(and then the adjoint code, by transposition), it is necessary to apply the chain rule to this function composition. Because of nonlinearities, dependencies, and inputs/outputs, applying the chain rule to a large code is very complex. There exist recipes for adjoint code construction, explaining in details how to get the adjoint for various instruction types (assignment, loop, conditional statements, inputs/outputs, and so on); see, e.g., [9, 10]. Code differentiation can be done by hand following these recipes.

An alternative to adjoint handwriting is to use specially designed softwares that automate the writing of the adjoint code, called automatic differentiation tools, such as the software Tapenade [12]. These tools are powerful, always improving, and can now derive large computer codes in various programming languages. It should be mentioned, however, that these tools cannot be used completely in black box mode and may require some preparatory work on the direct code. Despite these difficulties, some large-scale usages of automatic differentiation have been performed, e.g., for the ocean model of the MIT (MITgcm, see [20]) or a Greenland ice model (see below the Applications section and [13]).

2.3.4 Monte Carlo Approximation

Monte Carlo approximation allows to obtain an approximate gradient using a reasonable number of response function evaluations. Here the approximation proposed in [1] is presented.

Assume that \mathbf{h} is small enough, so that the following approximation holds:

$$\delta G = G(\mathbf{x} + \mathbf{h}) - G(\mathbf{x}) \approx G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{h} \quad (32.14)$$

By right-multiplying by \mathbf{h}^T both sides of Eq. (32.14), one gets

$$\delta G \mathbf{h}^T \approx (\nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{h})\mathbf{h}^T$$

Considering that \mathbf{h} is a stochastic perturbation, one can take the expectation

$$\mathbb{E}(\delta G \mathbf{h}^T) \approx \mathbb{E}([\nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{h}]\mathbf{h}^T) = \nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{A}$$

where $\mathbf{A} = \mathbb{E}(\mathbf{h}\mathbf{h}^T)$ is the covariance matrix of \mathbf{h} . Therefore,

$$\nabla_{\mathbf{x}}G(\mathbf{x}) \approx \mathbb{E}(\delta G \mathbf{h}^T)\mathbf{A}^{-1} \quad (32.15)$$

In practice, \mathbf{A} is a large matrix and may be difficult to inverse; therefore, the authors propose to replace \mathbf{A} by its diagonal. In the end, an approximate gradient is given by the formula:

$$\widetilde{\nabla_{\mathbf{x}}G(\mathbf{x})} = \mathbb{E}(\delta G \mathbf{h}^T) (\text{Diag}(\mathbf{A}))^{-1} \quad (32.16)$$

where the expectation is computed using a Monte Carlo method, requiring a certain number of model runs. This formula is simple enough but should be handled with care; indeed, it holds three successive approximations: finite differences instead of true gradient, Monte Carlo approximation (usually carried over with a small sample size), and A replaced by $\text{Diag}(A)$. However, this kind of approach has been successfully used in data assimilation, to obtain gradient-like information without resorting to adjoint code construction; see, e.g., [7, 19].

2.3.5 Gradient Code Validation

First-Order Test

The first-order test is very simple; the idea is just to check the following approximation at the first order in $\alpha \rightarrow 0$:

$$\frac{G(\mathbf{x} + \alpha \cdot \mathbf{h}) - G(\mathbf{x})}{\alpha} = (\nabla G, \mathbf{h}) + o(1)$$

The principle of the test is then to compute, for various perturbation directions \mathbf{h} and various values of α (with $\alpha \rightarrow 0$, e.g., $\alpha = 10^{-n}, n = 1 \dots 8$), the two following quantities: first one computes

$$\tau(\alpha, \mathbf{h}) = \frac{G(\mathbf{x} + \alpha \cdot \mathbf{h}) - G(\mathbf{x})}{\alpha}$$

with the direct code, and then one computes $\delta(\mathbf{h}) = (\nabla G, \mathbf{h})$, where ∇G is given by the adjoint code. Then one just has to measure the relative error

$$\varepsilon(\alpha, \mathbf{h}) = \frac{|\tau(\alpha, \mathbf{h}) - \delta(\mathbf{h})|}{|\delta(\mathbf{h})|}$$

and check that $\varepsilon(\alpha, \mathbf{h})$ tends to 0 with α for various directions \mathbf{h} .

Second-Order Test

For this test, the Taylor expansion at second order is written:

$$G(\mathbf{x} + \alpha \mathbf{h}) = G(\mathbf{x}) + \alpha (\nabla G, \mathbf{h}) + \frac{\alpha^2}{2} (\mathbf{h}, \nabla^2 G \mathbf{h}) + o(\alpha^2)$$

When \mathbf{h} is given, the last term is therefore a constant: $G(\mathbf{x} + \alpha \mathbf{h}) = G(\mathbf{x}) + \alpha (\nabla G, \mathbf{h}) + \frac{\alpha^2}{2} C(\mathbf{h}) + o(\alpha^2)$. In that case, the second-order test writes as follows. Let $\tau(\alpha, \mathbf{h})$ be defined as previously:

$$\tau(\alpha, \mathbf{h}) = \frac{G(\mathbf{x} + \alpha \cdot \mathbf{h}) - G(\mathbf{x})}{\alpha} \quad \text{and} \quad \delta(\mathbf{h}) = (\nabla G, \mathbf{h})$$

The Taylor expansion gives $\frac{\tau(\alpha, \mathbf{h}) - \delta(\mathbf{h})}{\alpha} = \frac{1}{2}C(\mathbf{h}) + o(1)$. The test consists in computing the quantity $r(\alpha, \mathbf{h})$:

$$r(\alpha, \mathbf{h}) = \frac{\tau(\alpha, \mathbf{h}) - \delta(\mathbf{h})}{\alpha}$$

for various directions \mathbf{h} and various α and checks that it tends to a constant (depending on \mathbf{h}) when $\alpha \rightarrow 0$.

2.4 Stability Analysis

Stability analysis is the study of how perturbations on the system will grow. Such tools, and in particular the so-called singular vectors, can be used for sensitivity analysis. Indeed, looking at extremal perturbations gives an input on sensitivities of the system (see [22, 23, 27] for application examples).

The growth rate of a given perturbation \mathbf{h}_0 of, say, the initial condition \mathbf{u}_0 of the model is classically defined by

$$\rho(\mathbf{h}_0) = \frac{\|\mathcal{M}(\mathbf{u}_0 + \mathbf{h}_0, T) - \mathcal{M}(\mathbf{u}_0, T)\|}{\|\mathbf{h}_0\|} \quad (32.17)$$

where $\|\cdot\|$ is a given norm.

One can then define the optimal perturbation \mathbf{h}_0^1 so that $\rho(\mathbf{h}_0^1) = \max_{\mathbf{h}_0} \rho(\mathbf{h}_0)$ and then deduce a family of maximum growth vectors:

$$\rho(\mathbf{h}_0^i) = \max_{\mathbf{h}_0 \perp \text{Span}(\mathbf{h}_0^1, \dots, \mathbf{h}_0^{i-1})} \rho(\mathbf{h}_0), i \geq 2 \quad (32.18)$$

By restricting the study to the linear part of the perturbation behavior, the growth rate becomes (denoting $\mathbf{L} = \frac{\partial \mathcal{M}}{\partial \mathbf{u}}$ for clarity)

$$\begin{aligned} \rho^2(\mathbf{h}_0) &= \frac{\|\mathbf{L}\mathbf{h}_0\|^2}{\|\mathbf{h}_0\|^2} = \frac{<\mathbf{L}\mathbf{h}_0, \mathbf{L}\mathbf{h}_0>}{<\mathbf{h}_0, \mathbf{h}_0>} \\ &= \frac{<\mathbf{h}_0, \mathbf{L}^*\mathbf{L}\mathbf{h}_0>}{<\mathbf{h}_0, \mathbf{h}_0>} \end{aligned} \quad (32.19)$$

$\mathbf{L}^*\mathbf{L}$ being a symmetric positive definite matrix, its eigenvalues are nonnegative real, and its eigenvectors are (or can be chosen) orthonormal. The strongest growth vectors are the eigenvectors of $\mathbf{L}^*\mathbf{L}$ which correspond to the greatest eigenvalues. They are called forward singular vectors and denoted f_i^+ :

$$\mathbf{L}^*\mathbf{L}f_i^+ = \mu_i f_i^+ \quad (32.20)$$

One can notice then that $\mathbf{L}f_i^+$ is an eigenvector of \mathbf{LL}^* , which allows to define the backward singular vectors, noted f_i^- , as

$$\mathbf{L}f_i^+ = \sqrt{\mu_i} f_i^-$$

The eigenvalue corresponding to f_i^- is μ_i as well. Forward singular vectors represent the directions of perturbation that will grow fastest, while backward singular vectors represent the directions of perturbation that have grown the most.

The computation of the f_i^+ and f_i^- generally requires numerous matrix-vector multiplications, i.e., direct integrations of the model and backward adjoint integrations. The result of these calculations depends on the norm used, the time window, and the initial state if the model is nonlinear. For an infinite time window, singular vectors converge toward Lyapunov vectors. The full nonlinear model can be retained in Eq. (32.17) leading to the computation of so-called nonlinear singular vectors. They are obtained by optimizing directly Eq. (32.17). However, due to nonlinear dissipation, they tend to converge toward infinitesimal perturbations as the time window lengthens; this can be sorted out by adding some constraints on the norm of the perturbation [21].

3 Applications

Applications of VSA can be generally divided into two classes: sensitivity to initial or boundary condition changes and sensitivity to parameter changes. However, one can extend the notion of sensitivity analysis to second-order sensitivity and stability analysis. This section is organized following this classification. Even though VSA has been used extensively in a wide range of problems, examples given here come from geophysical applications. Indeed, in that case, the control vector is generally of very large dimension; therefore, global SA techniques are out of reach. Moreover, quite often tangent and/or adjoint models are already available since they are used for data assimilation.

3.1 Sensitivity to Initial or Boundary Condition Changes

Sensitivity to initial condition changes ($\mathbf{x} = \mathbf{u}_0$) is routinely used in numerical weather prediction systems. In that case, the model (32.1) describes the evolution of the atmosphere, initialized with \mathbf{u}_0 . The parameter vector $\mathbf{x} = \mathbf{u}_0$ represents the full state of the atmosphere. In modern atmospheric models, that amounts to $10^9 - 10^{10}$ unknowns that are correlated.

The response function follows the form of (32.4), with:

$$\mathcal{H}(\mathbf{u}(\mathbf{u}_0)) = \| \mathbf{z}^{\text{obs}}(t) - H[\mathbf{u}(\mathbf{u}_0; t)] \|_{\mathcal{O}}^2$$

where \mathbf{z}^{obs} are observations of the system, H maps the state vector to the observation space, and $\| \cdot \|_{\mathcal{O}}$ is typically a weighted L^2 norm. Following (32.4), the response function is then

$$G(\mathbf{u}_0) = \frac{1}{2} \int_0^T \| \mathbf{z}^{\text{obs}}(t) - H[\mathbf{u}(\mathbf{u}_0; t)] \|_{\mathcal{O}}^2 dt \quad (32.21)$$

Local sensitivities of such functions can be very useful to understand the behavior of a system and have been extensively used in geosciences (see, for instance, [8, 28, 34], or [2]).

One can find an example of application of such methods on the Mercator Ocean's 1/4° global ocean model in [31]. The initial objective was to try to estimate the influences of geographical areas to reduce the forecast error using an adjoint method to compute the sensitivities. A preliminary study has been conducted by considering the misfit to observations as a proxy of the forecast error and sought to determine the sensitivity of this misfit to regional changes in the initial condition and/or to forcing. That should give an indication about the important phenomena to consider to improve this system.

The most easily interpreted case in this study is to consider a sensitivity criterion coming from the difference in sea surface temperature (SST) maps at the final instant of the assimilation cycle, because of its dense coverage in space. The response function (32.21) is a discrete version of the time integral, in which the operator H (mapping the state vector \mathbf{u} to the observation space) simply extracts the SST of the state vector. This can be translated into computing the gradient:

$$G(\mathbf{u}_0, \mathbf{q}) = \frac{1}{2} \sum_{n=1}^{N_{\text{SST}}} \| H_{\text{SST}}(\mathbf{u}_n) - \text{SST}^{\text{obs}} \|_{\mathbf{R}^{-1}}^2 \quad (32.22)$$

with a parameter vector $\mathbf{x} = (\mathbf{u}_0, \mathbf{q})$ made of $\mathbf{u}_0 = (u_0, v_0, T_0, S_0, \eta_0)^T$ the initial state vector (current velocity components, temperature, salinity, and sea level) and of $\mathbf{q} = (q_{sr}, q_{ns}, emp)^T$ (radiative fluxes, total heat fluxes, freshwater fluxes) and SST^{obs} observations of SST.

One can see an example of sensitivity to initial temperature (surface and 100 m) as shown in the two bottom panels of Fig. 32.1. High sensitivity will give a signal similar to the gap in observations (top left), while low sensitivity will show a white area. In this example, it is clear that the SST misfit is highly sensitive to changes in surface temperature where the initial mixed layer depth (top right) is low and insensitive elsewhere. The opposite conclusion can be drawn from the sensitivity to the initial temperature at 100 m. This is obviously not a surprise and corresponds more to the purpose of verification of the model rather than system improvement. However, it highlights the importance of having a good estimate of the vertical mixing. Other components of the gradient show the important role of atmospheric forcing (again this could have been anticipated), and ways to improve the system also appear to point to that direction.

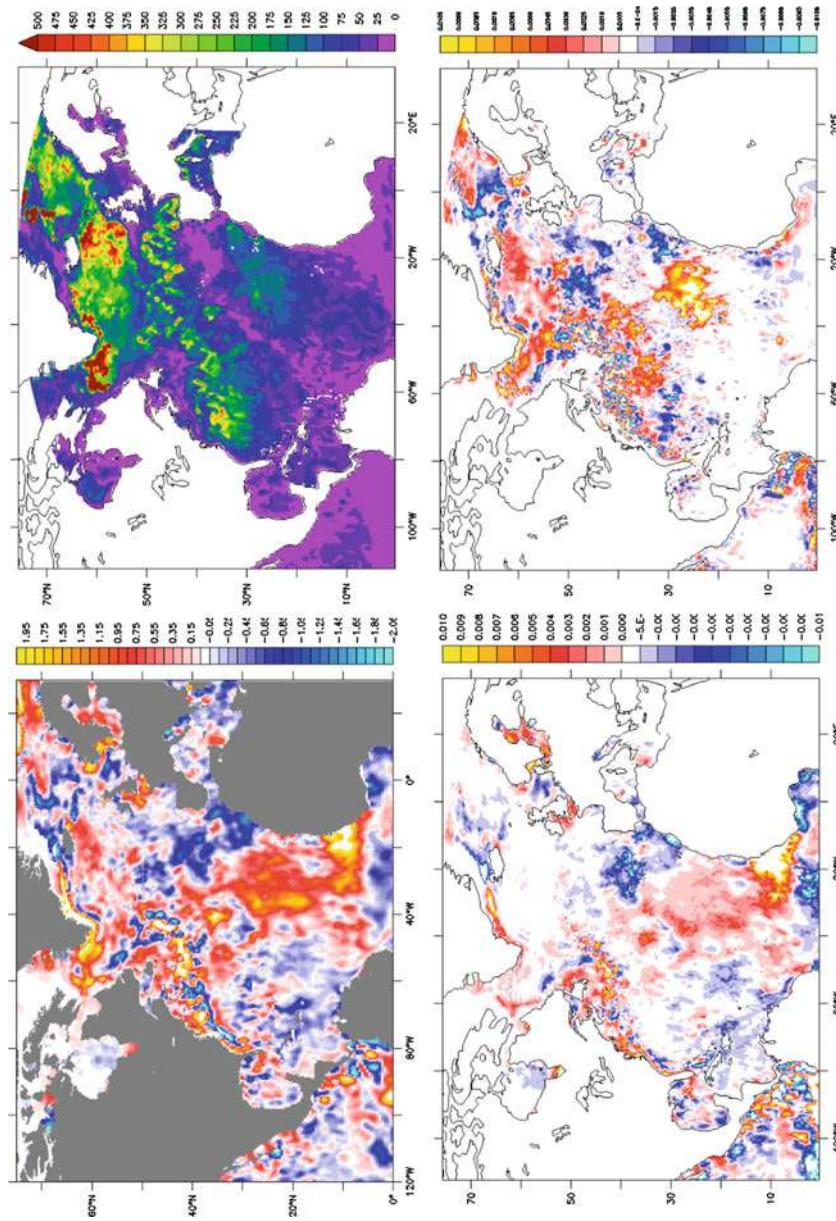


Fig. 32.1 Top: misfit between forecast and observed SST (left) and mixed layer depth (right). Bottom: sensitivity to 1-week lead time SST error with respect to variations in initial surface (left) and 100 m (right) and 1000 m temperature (From [31])

This kind of study is also routinely used to target observations. For example, in order to be able to track tropical cyclones, it is possible to use the so-called adjoint-derived sensitivity steering vectors (ADSSV, [5, 14, 32]). In that case, the model Eq. (32.1) represents the evolution of the atmosphere, starting from an initial state vector \mathbf{u}_0 . Some technicalities allow to choose the parameter vector \mathbf{x} equal to the vorticity at initial time $\mathbf{x} = \xi_0 = \frac{\partial v_0}{\partial x} - \frac{\partial u_0}{\partial y}$. Then the response function follows the form (32.9); it is a two-criteria function at a given verification time t_1 :

$$G(\xi_0) = \left(\frac{1}{|A|} \int_A u(t_1) dx, \frac{1}{|A|} \int_A v(t_1) dx \right)^T \quad (32.23)$$

where $\frac{1}{|A|} \int_A u dx$ and $\frac{1}{|A|} \int_A v dx$ are the zonal and meridional averaged wind velocities over a given area of interest A . By looking at the sensitivities to ξ_0 : $\frac{\partial G_u}{\partial \xi_0}$ and $\frac{\partial G_v}{\partial \xi_0}$, one will get information about the way the tropical cyclone is likely to go. For example, if at a given forecast time at one particular grid point the ADSSV vector points to the east, an increase in the vorticity at this very point at the observing time would be associated with an increase in the eastward steering flow of the storm at the verifying time [32]. This information, in turn, helps to decide where to launch useful observations.

3.2 Parameter Sensitivity

The previous section focuses on variable input quantities (initial/boundary conditions or forcings); however, most of numerical models also rely on a set of physical parameters. They are generally only approximatively known and their settings depend on the studied case. Methods to measure sensitivities to parameter changes are the same as before; differences mostly lie in the parameter set nature: it is generally of small to medium size and elements are mostly uncorrelated. To both respect, they may be better suited for GSA. However, in some cases, these parameters can, for instance, vary spatially and therefore be out of reach of global analysis.

Examples of spatially varying parameters are quite common in geophysics, and an example in glaciology will be given here. In the framework of global change and in particular sea-level change, the volume evolution of the two main ice caps (Antarctica and Greenland) is of crucial interest. In ice cap modeling, experts consider that the basal characteristics of the ice cap are particularly important: basal melt rate and basal sliding coefficient (linked to the sliding velocity). These basal characteristics can be considered as parameters (they are intrinsic to the modeled system) while being spatially varying, and their influence on the ice cap volume must be quantified in order to better understand and predict the future volume evolution.

This has been studied, for example, in [13], where the authors use adjoint methods to compute the sensitivities of the ice volume V over Greenland to

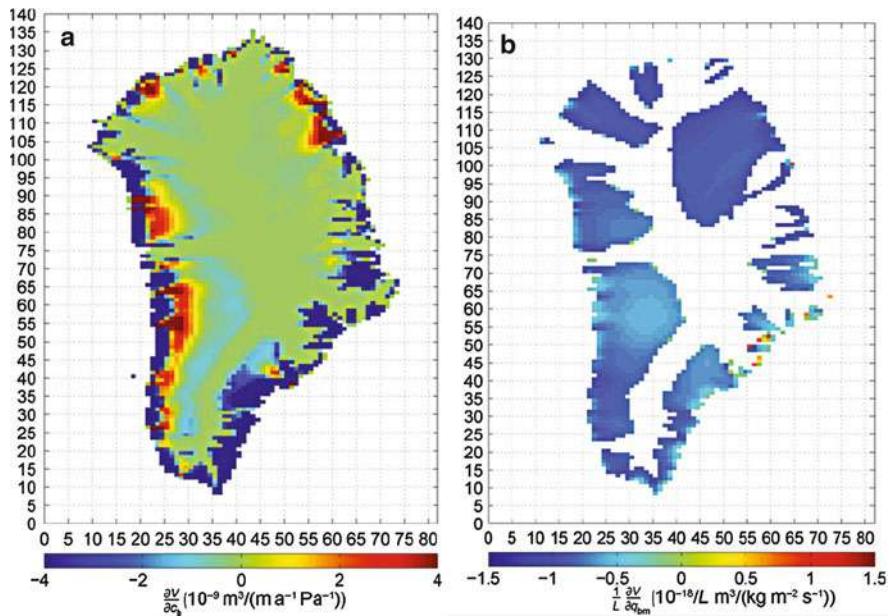


Fig. 32.2 Adjoint sensitivity maps of the total Greenland ice volume V related to (a) basal sliding c_b and (b) basal melt rate q_{bm} (From [13], copyright International Glaciological Society)

perturbations on the basal sliding c_b and the basal melt rate q_{bm} . Let us note here that in this case, the adjoint model has been obtained using automatic differentiation tools. Figure 32.2 shows that sliding sensitivities exhibit significant regional variations and are mostly located in the coastal areas, whereas melt-rate sensitivities are either fairly uniform or they completely vanish. This kind of information is of prime importance when tuning the model parameter and/or designing an observation campaign for measuring said parameters. For instance, in this particular system, there is no gain to be expected by focusing on the interior of the domain.

3.3 Sensitivity of Complex Systems

Previously presented examples focus on looking for sensitivities of a given model to perturbations. However, these approaches can be extended to more complex problems such as coupled models, for instance, or even a forecasting system, i.e., a modeling system that also includes an initialization scheme. Most of the time, this initialization is done through the so-called data assimilation techniques, where observations from the past are used to adjust the present state of the system. There are two kinds of such techniques, either based on filtering approaches or variational methods. Filtering techniques aim at bringing the model trajectory closer

to observations through a sequence of prediction and correction steps, i.e., the model is corrected as follows:

$$\frac{d\mathbf{u}}{dt} = \mathcal{M}(\mathbf{u}; \mathbf{x}) + \mathbf{K} (H(\mathbf{u}(t)) - \mathbf{y}^{\text{obs}}(t)) \quad (32.24)$$

where \mathbf{K} is a gain matrix. For the simplest versions of filtering data assimilation, \mathbf{K} does not depend on $\mathbf{u}(t)$, so computing sensitivities to such system can be done similarly as before (see [33] for an example). However, in more sophisticated approach, like variational data assimilation, it is less straightforward. In that case, the data assimilation problem is solved through the minimization of a cost function that is typically Eq. (32.21). Looking for local sensitivities on the forecasting system would mean to look for sensitivities to the optimal solution of the minimization of (32.21), that is to say to compute the gradient of the whole optimality system (direct model, adjoint model, cost function); doing so by adjoint method may require the second-order adjoint model [6, 16].

Another example of complex system is to perform a sensitivity analysis on a stability analysis (i.e., how given perturbations will affect the stability of a system).

In [27], the authors use stability analysis to study the sensitivity of the thermohaline oceanic circulation (large-scale circulation, mostly dominated by density variations, i.e., temperature and salinity variations). To do so, they look for the optimal initial perturbation of the sea surface salinity that induces the largest variation of the thermohaline circulation.

In [23], the authors are interested in the moist predictability in meteorological models. Since this is a very nonlinear process, they propose to use nonlinear singular vectors as response function G :

$$G(\mathbf{u}_0) = \arg \max_{\|\mathbf{h}_0\|=E_0} \left(\frac{\|\mathcal{M}(\mathbf{u}_0 + \mathbf{h}_0, T) - \mathcal{M}(\mathbf{u}_0, T)\|}{\|\mathbf{h}_0\|} \right) \quad (32.25)$$

This tells which variation in the initial condition will affect the most the optimal perturbations and then the predictability. Note that in that case, computing $\nabla_{\mathbf{u}_0} G(\mathbf{u}_0)$ also requires the second-order adjoint.

Obviously, these are only a handful of possible applications among many; as long as one defines a response function, one could be interested by studying its sensitivities.

4 Conclusion

Variational methods are local sensitivity analysis techniques; they gather a set of methods from very basic to sophisticated as presented above. The main advantage is that they can be used for very large dimension problems if using adjoint methods; the downside is that it may require some heavy developments. This burden can be reduced however by the use of automatic differentiation tools. They have been

used for a very wide range of applications and even on a daily basis in operational numerical weather prediction. Although they are local by essence, adjoint-based variational methods can be extended to global sensitivity analysis as will be presented in ► Chap. 36, “Derivative-Based Global Sensitivity Measures”.

Methods presented here are dedicated to first-order local sensitivity analysis. This can be extended to interaction studies by using the second-order derivatives (Hessian), which can be computed similarly using so-called second-order adjoint models.

Readers interested in going further could start with clearly written and easy to read papers such as [4, 13, 17]. To go further, the book [3] and the application paper [2] are recommended. And, finally, about second-order derivatives, [16] provides a nice introduction, and [23] offers an advanced application.

5 Cross-References

- [Derivative-Based Global Sensitivity Measures](#)
 - [Sensitivity Analysis of Spatial and/or Temporal Phenomena](#)
-

References

1. Ancell, B., Hakim, G.J.: Comparing adjoint- and ensemble-sensitivity analysis with applications to observation targeting. *Mon. Weather Rev.* **135**(12), 4117–4134 (2007)
2. Ayoub, N.: Estimation of boundary values in a North Atlantic circulation model using an adjoint method. *Ocean Model.* **12**(3–4), 319–347 (2006)
3. Cacuci, D.G.: *Sensitivity and Uncertainty Analysis: Theory*. CRC Press, Boca Raton (2005)
4. Castaings, W., Dartus, D., Le Dimet, F.X., Saulnier, G.M.: Sensitivity analysis and parameter estimation for distributed hydrological modeling: potential of variational methods. *Hydrol. Earth Syst. Sci.* **13**(4), 503–517 (2009)
5. Chen, S.G., Wu, C.C., Chen, J.H., Chou, K.H.: Validation and interpretation of adjoint-derived sensitivity steering vector as targeted observation guidance. *Mon. Weather Rev.* **139**, 1608–1625 (2011)
6. Daescu, D.N., Navon, I.M.: Reduced-order observation sensitivity in 4D-var data assimilation. In: American Meteorological Society 88th AMS Annual Meeting, New Orleans (2008)
7. Desroziers, G., Camino, J.T., Berre, L.: 4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations. *Q. J. R. Meteorol. Soc.* **140**, 2097–2110 (2014)
8. Errico, R.M., Vukicevic, T.: Sensitivity analysis using an adjoint of the PSU-NCAR mesoscale model. *Mon. Weather Rev.* **120**(8), 1644–1660 (1992)
9. Giering, R., Kaminski, T.: Recipes for adjoint code construction. *ACM Trans. Math. Softw.* **24**(4), 437–474 (1998)
10. Griewank, A., Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia (2008)
11. Hamby, D.M.: A review of techniques for parameter sensitivity analysis of environmental models. *Environ. Monit. Assess.* **32**(2), 135–154 (1994)
12. Hascoet, L., Pascual, V.: The Tapenade automatic differentiation tool: principles, model, and specification. *ACM Trans. Math. Softw.* **39**(3), 20 (2013)

13. Heimbach, P., Bugnion, V.: Greenland ice-sheet volume sensitivity to basal, surface and initial conditions derived from an adjoint model. *Ann. Glaciol.* **50**, 67–80 (2009)
14. Hoover, B.T., Morgan, M.C.: Dynamical sensitivity analysis of tropical cyclone steering using an adjoint model. *Mon. Weather Rev.* **139**, 2761–2775 (2011)
15. Lauvernet, C., Hascoët, L., Dimet, F.X.L., Baret, F.: Using automatic differentiation to study the sensitivity of a crop model. In: Forth, S., Hovland, P., Phipps, E., Utke, J., Walther, A. (eds.) *Recent Advances in Algorithmic Differentiation*. Lecture Notes in Computational Science and Engineering, vol. 87, pp. 59–69. Springer, Berlin (2012)
16. Le Dimet, F.X., Ngodock, H.E., Luong, B., Verron, J.: Sensitivity analysis in variational data assimilation. *J. Meteorol. Soc. Jpn. Ser. 2*, **75**, 135–145 (1997)
17. Lellouche, J.M., Devenon, J.L., Dekeyser, I.: Boundary control of Burgers' equation—a numerical approach. *Comput. Math. Appl.* **28**(5), 33–34 (1994)
18. Li, S., Petzold, L.: Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement. *J. Comput. Phys.* **198**(1), 310–325 (2004)
19. Liu, C., Xiao, Q., Wang, B.: An ensemble-based four-dimensional variational data assimilation scheme. Part I: technical formulation and preliminary test. *Mon. Weather Rev.* **136**(9), 3363–3373 (2008)
20. Marotzke, J., Wunsch, C., Giering, R., Zhang, K., Stammer, D., Hill, C., Lee, T.: Construction of the adjoint MIT ocean general circulation model and application to atlantic heat transport sensitivity. *J. Geophys. Res.* **104**(29), 529–29 (1999)
21. Mu, M., Duan, W., Wang, B.: Conditional nonlinear optimal perturbation and its applications. *Nonlinear Process. Geophys.* **10**(6), 493–501 (2003)
22. Qin, X., Mu, M.: Influence of conditional nonlinear optimal perturbations sensitivity on typhoon track forecasts. *Q.J. R. Meteorol. Soc.* **138**, 185–197 (2011)
23. Rivièvre, O., Lapeyre, G., Talagrand, O.: A novel technique for nonlinear sensitivity analysis: application to moist predictability. *Q. J. R. Meteorol. Soc.* **135**(643), 1520–1537 (2009)
24. Saltelli, A., Ratto, M., Tarantola, S., Campolongo, F.: Sensitivity analysis for chemical models. *Chem. Rev.* **105**(7), 2811–2828 (2005)
25. Sandu, A., Daescu, D.N., Carmichael, G.R.: Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part I—theory and software tools. *Atmos. Environ.* **37**(36), 5083–5096 (2003)
26. Sandu, A., Daescu, D.N., Carmichael, G.R., Chai, T.: Adjoint sensitivity analysis of regional air quality models. *J. Comput. Phys.* **204**(1), 222–252 (2005)
27. Sévellec, F.: Optimal surface salinity perturbations influencing the thermohaline circulation. *J. Phys. Oceanogr.* **37**(12), 2789–2808 (2007)
28. Sykes, J.F., Wilson, J.L., Andrews, R.W.: Sensitivity analysis for steady state groundwater flow using adjoint operators. *Water Resour. Res.* **21**(3), 359–371 (1985)
29. Thuburn, J., Haine, T.W.N.: Adjoints of nonoscillatory advection schemes. *J. Comput. Phys.* **171**(2), 616–631 (2001)
30. Vidard, A.: Data assimilation and adjoint methods for geophysical applications. PhD thesis, Université de Grenoble, Habilitation thesis (2012)
31. Vidard, A., Rémy, E., Greiner, E.: Sensitivity analysis through adjoint method: application to the GLORYS reanalysis. Contrat n° 08/D43, Mercator Océan (2011)
32. Wu, C.C., Chen, J.H., Lin, P.H., Chou, K.H.: Targeted observations of tropical cyclone movement based on the adjoint-derived sensitivity steering vector. *J. Atmos. Sci.* **64**(7), 2611–2626 (2007)
33. Zhu, Y., Gelaro, R.: Observation sensitivity calculations using the adjoint of the gridpoint statistical interpolation (GSI) analysis system. *Mon. Weather Rev.* **136**(1), 335–351 (2008)
34. Zou, X., Barcilon, A., Navon, I.M., Whitaker, J., Cacuci, D.G.: An adjoint sensitivity study of blocking in a two-layer isentropic model. *Mon. Weather Rev.* **121**, 2833–2857 (1993)

David C. Woods and Susan M. Lewis

Abstract

The aim of this paper is to review methods of designing screening experiments, ranging from designs originally developed for physical experiments to those especially tailored to experiments on numerical models. The strengths and weaknesses of the various designs for screening variables in numerical models are discussed. First, classes of factorial designs for experiments to estimate main effects and interactions through a linear statistical model are described, specifically regular and nonregular fractional factorial designs, supersaturated designs, and systematic fractional replicate designs. Generic issues of aliasing, bias, and cancellation of factorial effects are discussed. Second, group screening experiments are considered including factorial group screening and sequential bifurcation. Third, random sampling plans are addressed including Latin hypercube sampling and sampling plans to estimate elementary effects. Fourth, a variety of modeling methods commonly employed with screening designs are briefly described. Finally, a novel study demonstrates six screening methods on two frequently-used exemplars, and their performances are compared.

Keywords

Computer experiments • fractional factorial designs • Gaussian process models • group screening • space-filling designs • supersaturated designs • variable selection

Contents

1	Introduction	1144
1.1	Linear Regression Models	1146
1.2	Gaussian Process Models	1147
1.3	Screening without a Surrogate Model	1148

D.C. Woods (✉) • S.M. Lewis

Southampton Statistical Sciences Research Institute, University of Southampton,
Southampton, SO17 1BJ, UK

e-mail: D.Woods@southampton.ac.uk; S.M.Lewis@southampton.ac.uk

2	Factorial Screening Designs	1148
2.1	Regular Fractional Factorial Designs	1149
2.2	Nonregular Fractional Factorial Designs	1151
2.3	Supersaturated Designs for Main Effects Screening	1154
2.4	Common Issues with Factorial Screening Designs	1157
2.5	Systematic Fractional Replicate Designs	1157
3	Screening Groups of Variables	1159
3.1	Factorial Group Screening	1159
3.2	Sequential Bifurcation	1161
3.3	Iterated Fractional Factorial Designs	1162
3.4	Two-Stage Group Screening for Gaussian Process Models	1163
4	Random Sampling Plans and Space Filling	1163
4.1	Latin Hypercube Sampling	1163
4.2	Sampling Plans for Estimating Elementary Effects (Morris' Method)	1166
5	Model Selection Methods	1169
5.1	Variable Selection for Nonregular and Supersaturated Designs	1170
5.2	Variable Selection for Gaussian Process Models	1170
6	Examples and Comparisons	1172
7	Conclusions	1179
Cross-References		1180
References		1180

1 Introduction

Screening [32] is the process of discovering, through statistical design of experiments and modeling, those controllable factors or input variables that have a substantive impact on the response or output which is either calculated from a numerical model or observed from a physical process.

Knowledge of these *active* input variables is key to optimization and control of the numerical model or process. In many areas of science and industry, there are often a large number of potentially important variables. Effective screening experiments are then needed to identify the active variables as economically as possible. This may be achieved through careful choice of experiment size and the set of combinations of input variable values (the design) to be run in the experiment. Each run determines an evaluation of the numerical model or an observation to be made on the physical process. The variables found to be active from the experiment are further investigated in one or more follow-up experiments that enable estimation of a detailed predictive statistical model of the output variable.

The need to screen a large number of input variables in a relatively small experiment presents challenges for both design and modeling. Crucial to success is the principle of *factor sparsity* [16] which states that only a small proportion of the input variables have a substantive influence on the output. If this widely observed principle does not hold, then a small screening experiment may fail to reliably detect the active variables, and a much larger investigation will be required.

While most literature has focused on designs for physical experiments, screening is also important in the study of numerical models via computer experiments [97].

Such models often describe complex input-output relationships and have numerous input variables. A primary reason for building a numerical model is to gain better understanding of the nature of these relationships, especially the identification of the active input variables. If a small set of active variables can be identified, then the computational costs of subsequent exploration and exploitation of the numerical model are reduced. Construction of a surrogate model from the active variables requires less experimentation, and smaller Monte Carlo samples may suffice for uncertainty analysis and uncertainty propagation.

The effectiveness of screening can be evaluated in a variety of ways. Suppose there are d input variables held in vector $\mathbf{x} = (x_1, \dots, x_d)^T$ and that $\mathcal{X} \subset \mathbb{R}^d$ contains all possible values of \mathbf{x} , i.e., all possible combinations of input variable values. Let $A_T \subseteq \{1, \dots, d\}$ be the set of indices of the truly active variables and $A_S \subseteq \{1, \dots, d\}$ consist of the indices of those variables selected as active through screening. Then, the following measures may be defined: (i) sensitivity, $\phi_s = |A_S \cap A_T|/|A_T|$, the proportion of active variables that are successfully detected, where ϕ_s is defined as 1 when $A_T = \emptyset$; (ii) false discovery rate [8], $\phi_{\text{fdr}} = |A_S \cap \bar{A}_T|/|A_S|$, where \bar{A}_T is the complement of A_T , the proportion of variables selected as active that are actually inactive, and ϕ_{fdr} is defined as 0 when $A_S = \emptyset$; and (iii) type I error rate, $\phi_I = |A_S \cap \bar{A}_T|/|\bar{A}_T|$, the proportion of inactive variables that are selected as active. In practice, high sensitivity is often considered more important than a low type I error rate or false discovery rate [34] because failure to detect an active input variable results in no further investigation of the variable and no exploitation of its effect on the output for purposes of optimization and control.

The majority of designs for screening experiments are tailored to the identification and estimation of a surrogate model that approximates an output variable $Y(\mathbf{x})$. A class of surrogate models which has been successfully applied in a variety of fields [80] has the form

$$Y(\mathbf{x}) = \mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} + \varepsilon(\mathbf{x}), \quad (33.1)$$

where \mathbf{h} is a $p \times 1$ vector of known functions of \mathbf{x} , $\boldsymbol{\beta} = (\beta_0, \dots, \beta_{p-1})^T$ are unknown parameters, and $\varepsilon(\mathbf{x})$ is a random variable with a $N(0, \sigma^2)$ distribution for constant σ^2 . Note that if multiple responses are obtained from each run of the experiment, then the simplest and most common approach is separate screening of the variables for each response using individual models of the form (33.1).

An important decision in planning a screening experiment is the level of fidelity, or accuracy, required of a surrogate model for effective screening including the choice of the elements of \mathbf{h} in (33.1). Two forms of (33.1) are commonly used for screening variables in numerical models: linear regression models and Gaussian process models.

1.1 Linear Regression Models

Linear regression models assume that $\varepsilon(\mathbf{x}_0)$ and $\varepsilon(\mathbf{x}'_0)$, $\mathbf{x}_0 \neq \mathbf{x}'_0 \in \mathcal{X}$, are independent random variables. Estimation of detailed mean functions $\mathbf{h}^T(\mathbf{x})\boldsymbol{\beta}$ with a large number of terms requires large experiments which can be prohibitively expensive. Hence, many popular screening strategies investigate each input variable x_i at two levels, often coded +1 and -1 and referred to as “high” and “low,” respectively [15, chs. 6 and 7]. Interest is then in identifying those variables that have a large *main effect*, defined for variable x_i as the difference between the average expected responses for the 2^{d-1} combinations of variable values with $x_i = +1$ and the average for the 2^{d-1} combinations with $x_i = -1$. Main effects may be estimated via a *first-order surrogate model*

$$\mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d, \quad (33.2)$$

where $p = d + 1$. Such a “main effects screening” strategy relies on a firm belief in *strong effect heredity* [46], that is, important interactions or other nonlinearities involve only those input variables that have large main effects. Without this property, active variables may be overlooked.

There is evidence, particularly from industrial experiments [18, 99], that strong effect heredity may fail to hold in practice. This has led to the recent development and assessment of design and data analysis methodology that also allows screening of interactions between pairs of variables [34, 57]. For two-level variables, the interaction between x_i and x_j ($i, j = 1, \dots, d; i \neq j$) is defined as one-half of the difference between the conditional main effect for x_i given $x_j = +1$ and the conditional main effect of x_i given $x_j = -1$. Main effects and two-variable interactions can be estimated via a *first-order surrogate model supplemented by two-variable product terms*

$$\mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d + \beta_{12} x_1 x_2 + \dots + \beta_{(d-1)d} x_{d-1} x_d, \quad (33.3)$$

where $p = 1 + d(d + 1)/2$ and $\beta_{d+1}, \dots, \beta_{p-1}$ in (33.1) are relabeled $\beta_{12}, \dots, \beta_{(d-1)d}$ for notational clarity.

The main effects and interactions are collectively known as the factorial effects and can be shown to be the elements of $2\boldsymbol{\beta}$. The screening problem may be cast as variable or model selection, that is, choosing a statistical model composed of a subset of the terms in (33.3).

The parameters in $\boldsymbol{\beta}$ can be estimated by least squares. Let $x_i^{(j)}$ be the value taken by the i th variable in the j th run ($i = 1, \dots, d; j = 1, \dots, n$). Then, the rows of the $n \times d$ *design matrix* $\mathbf{X}^n = (x_1^{(j)}, \dots, x_d^{(j)})_{j=1, \dots, n}$ each hold one run of the design. Let $\mathbf{Y}^n = (Y^{(1)}, \dots, Y^{(n)})$ be the output vector. Then, the least squares estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y}^n, \quad (33.4)$$

where $\mathbf{H} = (\mathbf{h}(\mathbf{x}_1^n), \dots, \mathbf{h}(\mathbf{x}_n^n))^T$ is the *model matrix* and $(\mathbf{x}_j^n)^T$ is the j th row of \mathbf{X}^n . For the least squares estimators to be uniquely defined, \mathbf{H} must be of full column rank.

In physical screening experiments, often no attempt is made to estimate nonlinear effects other than two-variable interactions. This practice is underpinned by the principle of *effect hierarchy* [112] which states that low-order factorial effects, such as main effects and two-variable interactions, are more likely to be important than higher-order effects. This principle is supported by substantial empirical evidence from physical experiments.

However, the exclusion of higher-order terms from surrogate model (33.1) can result in biased estimators (33.4). Understanding, and minimizing, this bias is key to effective linear model screening. Suppose that a more appropriate surrogate model is

$$Y(\mathbf{x}) = \beta_0 + \mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} + \tilde{\mathbf{h}}^T(\mathbf{x})\tilde{\boldsymbol{\beta}} + \varepsilon,$$

where $\tilde{\mathbf{h}}(\mathbf{x})$ is a \tilde{p} -vector of model terms, additional to those held in $\mathbf{h}(\mathbf{x})$, and $\tilde{\boldsymbol{\beta}}$ is a \tilde{p} -vector of constants. Then, the expected value of $\hat{\boldsymbol{\beta}}$ is given by

$$\mathbb{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta} + \mathbf{A}\tilde{\boldsymbol{\beta}}, \quad (33.5)$$

where

$$\mathbf{A} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \tilde{\mathbf{H}}, \quad (33.6)$$

and $\tilde{\mathbf{H}} = (\tilde{\mathbf{h}}(\mathbf{x}_j^n))_j$. The *alias matrix* \mathbf{A} determines the pattern of bias in $\hat{\boldsymbol{\beta}}$ due to omitting the terms $\tilde{\mathbf{h}}^T(\mathbf{x})\tilde{\boldsymbol{\beta}}$ from the surrogate model and can be controlled through the choice of design. The size of the bias is determined by $\tilde{\boldsymbol{\beta}}$ which is outside the experimenter's control.

1.2 Gaussian Process Models

Gaussian process (GP) models are used when it is anticipated that understanding more complex relationships between the input and output variables is necessary for screening. Under a GP model, it is assumed that $\varepsilon(\mathbf{x}_0), \varepsilon(\mathbf{x}'_0)$ follow a bivariate normal distribution with correlation dependent on a distance metric applied to $\mathbf{x}_0, \mathbf{x}'_0$; see [93] and *Metamodel-based sensitivity analysis: polynomial chaos and Gaussian process*.

Screening with a GP model requires interrogation of the parameters that control this correlation. A common correlation function employed for GP screening has the form

$$\text{cor}(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^d \exp(-\theta_i |x_i - x'_i|^{\alpha_i}), \quad \theta_i \geq 0, 0 < \alpha_i \leq 2. \quad (33.7)$$

Conditional on $\theta_1, \dots, \theta_d$, closed-form maximum likelihood or generalized least squares estimators for β and σ^2 are available. However, θ_i requires numerical estimation. Reliable estimation of these more sophisticated and flexible surrogate models for a large number of variables requires larger experiments and may incur an impractically large number of evaluations of the numerical model.

1.3 Screening without a Surrogate Model

The selection of active variables using a surrogate model relies on the model assumptions and their validation. An alternative model-free approach is the estimation of elementary effects [74]. The elementary effect for the i th input variable for a combination of input values $\mathbf{x}_0 \in \mathcal{X}$ is an approximation to the derivative of $Y(\mathbf{x}_0)$ in the direction of the i th variable. More formally,

$$\text{EE}_i(\mathbf{x}_0) = \frac{Y(\mathbf{x}_0 + \Delta \mathbf{e}_{id}) - Y(\mathbf{x}_0)}{\Delta}, \quad i = 1, \dots, d, \quad (33.8)$$

where \mathbf{e}_{id} is the i th unit vector of length d (the i th column of the $d \times d$ identity matrix) and $\Delta > 0$ is a given constant such that $\mathbf{x} + \Delta \mathbf{e}_{id} \in \mathcal{X}$. Repeated random draws of \mathbf{x}_0 from \mathcal{X} according to a chosen distribution enable an empirical, model-free distribution for the elementary effect of the i th variable to be estimated. The moments (e.g., mean and variance) of this distribution may be used to identify active effects, as discussed later.

In the remainder of the paper, a variety of screening methods are reviewed and discussed, starting with (regular and nonregular) factorial and fractional factorial designs in the next section. Later sections cover methods of screening groups of variables, such as factorial group screening and sequential bifurcation; random sampling plans and space-filling designs, including sampling plans for estimating elementary effects; and model selection methods. The paper finishes by comparing and contrasting the performance of six screening methods on two examples from the literature.

2 Factorial Screening Designs

In a full factorial design, each of the d input variables is assigned a fixed number of values or levels, and the design consists of one run of each of the distinct combinations of these values. Designs in which each variable has two values are mainly considered here, giving $n = 2^d$ runs in the full factorial design. For even moderate values of d , experiments using such designs may be infeasibly large due to the costs or computing resources required. Further, such designs can be wasteful as they allow estimation of all interactions among the d variables, whereas effect hierarchy suggests that low-order factorial effects (main effects and two-variable interactions) will be the most important. These problems may be overcome by using a carefully chosen subset, or *fraction*, of the combinations of variable values in the full factorial design. Such fractional factorial designs have a long history of use in

physical experiments [39] and, more recently, have also been used in the study of numerical models [36]. However, they bring the complication that the individual main effects and interactions cannot be estimated independently. Two classes of designs are discussed here.

2.1 Regular Fractional Factorial Designs

The most widely used two-level fractional factorial designs are $1/2^q$ fractions of the 2^d full factorial design, known as 2^{d-q} designs [112, ch. 5] ($1 \leq q < d$ is integer). As the outputs from all the combinations of variable values are not available from the experiment, the individual main effects and interactions cannot be estimated. However, in a *regular* fractional factorial design, 2^{d-q} linear combinations of the factorial effects can be estimated. Two factorial effects that occur in the same linear combination cannot be independently estimated and are said to be *aliased*. The designs are constructed by choosing which factorial effects should be aliased together.

The following example illustrates a full factorial design, the construction of a regular fractional factorial design, and the resulting aliasing among the factorial effects. Consider first a 2^3 factorial design in variables x_1, x_2, x_3 . Each run of this design is shown as a row across the columns 3–5 in Table 33.1. Thus, these three columns form the design matrix. The entries in these columns are the coefficients of the expected responses in the linear combinations that constitute the main effects, ignoring constants. Where interactions are involved, as in model (33.3), their corresponding coefficients are obtained as elementwise products of columns 3–5. Thus, columns 2–8 of Table 33.1 give the model matrix for model (33.3).

A 2^{4-1} regular fractional factorial design in $n = 8$ runs may be constructed from the 2^3 design by assigning the fourth variable, x_4 , to one of the interaction columns. In Table 33.1, x_4 is assigned to the column corresponding to the highest-order interaction, $x_1x_2x_3$. Each of the eight runs now has the property that $x_1x_2x_3x_4 = +1$, and hence, as each variable can only take values ± 1 , it follows that $x_1 = x_2x_3x_4$, $x_2 = x_1x_3x_4$, and $x_3 = x_1x_2x_4$. Similarly, $x_1x_2 = x_3x_4$, $x_1x_3 = x_2x_4$, and $x_1x_4 = x_2x_3$. Two consequences are: (i) each main effect is aliased with

Table 33.1 A 2^{4-1} fractional factorial design constructed from the 2^3 full factorial design showing the aliased effects

a three-variable interaction and (ii) each two-variable interaction is aliased with another two-variable interaction. However, for each variable, the *sum* of the main effect and the three-variable interaction not involving that variable can be estimated. These two effects are said to be aliased. The other pairs of aliased effects are shown in Table 33.1. The four-variable interaction cannot be estimated and is said to be aliased with the mean, denoted by $I = x_1x_2x_3x_4$ (column 2 of Table 33.1).

An estimable model for this 2^{4-1} design is

$$\tilde{\mathbf{h}}^T(\mathbf{x})\boldsymbol{\beta} = \beta_0 + \beta_1x_1 + \dots + \beta_4x_4 + \beta_{12}x_1x_2 + \beta_{13}x_1x_3 + \beta_{23}x_2x_3,$$

with model matrix \mathbf{H} given by columns 2–9 of Table 33.1. The columns of \mathbf{H} are mutually orthogonal, $\mathbf{h}(\mathbf{x}_j^n)^T\mathbf{h}(\mathbf{x}_k^n) = 0$ for $j \neq k$; $j, k = 1, \dots, 8$. The aliasing in the design will result in a biased estimator of $\boldsymbol{\beta}$. This can be seen by setting

$$\tilde{\mathbf{h}}^T(\mathbf{x})\tilde{\boldsymbol{\beta}} = \beta_{14}x_1x_4 + \beta_{24}x_2x_4 + \beta_{34}x_3x_4 + \sum_{1 \leq j < k < l \leq 4} \beta_{jkl}x_jx_kx_l + \beta_{1234}x_1x_2x_3x_4,$$

which leads to the alias matrix $\mathbf{A} = \sum_{j=1}^8 \mathbf{e}_{j8}\mathbf{e}_{(8-j+1)8}$, which is an anti-diagonal identity matrix, and

$$\mathbb{E}(\hat{\beta}_0) = \beta_0 + \beta_{1234},$$

$$\mathbb{E}(\hat{\beta}_1) = \beta_1 + \beta_{234}, \quad \mathbb{E}(\hat{\beta}_2) = \beta_2 + \beta_{134}, \quad \mathbb{E}(\hat{\beta}_3) = \beta_3 + \beta_{124}, \quad \mathbb{E}(\hat{\beta}_4) = \beta_4 + \beta_{123}$$

$$\mathbb{E}(\hat{\beta}_{12}) = \beta_{12} + \beta_{34}, \quad \mathbb{E}(\hat{\beta}_{13}) = \beta_{13} + \beta_{24}, \quad \mathbb{E}(\hat{\beta}_{23}) = \beta_{23} + \beta_{14}.$$

More generally, to construct a 2^{d-q} fractional factorial design, a set $\{v_1, \dots, v_q\}$ of *defining words*, such as $x_1x_2x_3x_4$, must be chosen and the corresponding factorial effects aliased with the mean. That is, the product of variable values defined by each of these words is constant in the design (and equal to either -1 or $+1$). As the product of any two columns of constants in the design must also be constant, there is a total of $2^q - 1$ effects aliased with the mean. The list of all effects aliased with the mean is called the defining relation and is written as $I = v_1 = \dots = v_q = v_1v_2 = \dots = v_1 \cdots v_q$. Products of the defining words are straightforward to calculate as $x_i^2 = 1$, so that $v_j^2 = 1$ ($i = 1, \dots, d$; $j = 1, \dots, 2^d$).

The aliasing scheme for a design is easily obtained from the defining relation. A factorial effect with corresponding word v_j is aliased with each factorial effect corresponding to the words $v_jv_1, v_jv_2, \dots, v_jv_1 \cdots v_q$ formed by the product of v_j with every word in the defining relation. Hence, the defining relation $I = x_1x_2x_3x_4$ results in $x_1 = x_2x_3x_4, x_2 = x_1x_3x_4$, and so on; see Table 33.1.

As demonstrated above, the impact of aliasing is bias in the estimators of the regression coefficients in (33.1) which can be formulated through the alias matrix. For a regular fractional factorial design, the columns of \mathbf{H} are mutually orthogonal and hence $\mathbf{A} = \frac{1}{n}\mathbf{H}^T\tilde{\mathbf{H}}$. If the functions in $\tilde{\mathbf{h}}$ correspond to those high-order interactions not included in \mathbf{h} , then the elements of A are all either 0 or ± 1 . This is because the aliasing of factorial effects ensures that each column of $\tilde{\mathbf{H}}$ is either

orthogonal to all columns in \mathbf{H} or identical to a column of \mathbf{H} up to a change of sign. Thus, \mathbf{A} identifies the aliasing among the factorial effects.

Crucial to fractional factorial design is the choice of a defining relation to ensure that effects of interest are not aliased together. Typically, this involves choosing defining words to ensure that only words corresponding to higher-order factorial effects are included in the defining relation.

Regular fractional factorial designs are classed according to their *resolution*. A *resolution III* design has at least one main effect aliased with a two-variable interaction. A *resolution IV* design has no main effects aliased with interactions but at least one pair of two-variable interactions aliased together. A *resolution V* design has no main effects or two-variable interactions aliased with any other main effects or two-variable interactions. A more detailed and informative classification of regular fractional factorial designs is obtained via the aberration criterion [25].

Although resolution V designs allow for the estimation of higher-fidelity surrogate models, they typically require too many runs for screening studies. The most common regular fractional factorial designs used in screening are resolution III designs as part of a main-effects screening strategy. The design in Table 33.1 has resolution IV.

2.2 Nonregular Fractional Factorial Designs

The regular designs discussed above require n to be a power of two, which limits their application to some experiments. Further, even resolution III regular fractional factorials may require too many runs to be feasible for large numbers of variables. For example, with 11 variables, a resolution III regular fractional design requires $n = 16$ runs. Smaller experiments with n not equal to a power of two can often be performed by using the wider class of *nonregular* fractional factorial designs [115] that cannot be constructed via a set of defining words. For 11 variables, a design with $n = 12$ runs can be constructed that can estimate all 11 main effects independently of each other. While these designs are more flexible in their run size, the cost is a more complex aliasing scheme that makes interpretation of experimental results more challenging and requires the use of more sophisticated modeling methods.

Many nonregular designs are constructed via *orthogonal arrays* [92]. A symmetric orthogonal array of strength t , denoted by $\text{OA}(n, s^d, t)$, is an $n \times d$ matrix of s different symbols such that all ordered t -tuples of the symbols occur equally often as rows of any $n \times t$ submatrix of the array. Each such array defines an n -run factorial design in d variables, each having s levels. Here, only arrays with $s = 2$ symbols, ± 1 , will be discussed. The strength of the array is closely related to the resolution of the design. An array of strength $t = 2$ allows estimation of all main effects independently of each other but not of the two-variable interactions (cf. resolution III); a strength 3 array allows estimation of main effects independently of two-variable interactions (cf. resolution IV). Clearly, the two-level regular fractional factorial designs are all orthogonal arrays. However, the class of orthogonal arrays is wider and includes many other designs that cannot be obtained via a defining relation.

Table 33.2 The $n = 12$ -run nonregular Plackett-Burman design

Run	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1	1	1	1	1	1	1
3	-1	-1	1	1	1	-1	-1	-1	1	1	1
4	-1	1	-1	1	1	-1	1	1	-1	-1	1
5	-1	1	1	-1	1	1	-1	1	-1	1	-1
6	-1	1	1	1	-1	1	1	-1	1	-1	-1
7	1	-1	1	1	-1	-1	1	1	-1	1	-1
8	1	-1	1	-1	1	1	1	-1	-1	-1	1
9	1	-1	-1	1	1	1	-1	1	1	-1	-1
10	1	1	1	-1	-1	-1	-1	1	1	-1	1
11	1	1	-1	1	-1	1	-1	-1	-1	1	1
12	1	1	-1	-1	1	-1	1	-1	1	1	-1

An important class of orthogonal arrays are constructed from Hadamard matrices [44]. A Hadamard matrix \mathbf{C} of order n is an $n \times n$ matrix with entries ± 1 such that $\mathbf{C}^T \mathbf{C} = n\mathbf{I}_n$, where \mathbf{I}_n is the $n \times n$ identity matrix. An $\text{OA}(n, 2^{n-1}, 2)$ is obtained by multiplying rows of \mathbf{C} by -1 as necessary to make all entries in the first column equal to $+1$ and then removing the first column. Such a design can estimate the main effects of all $d = n - 1$ variables independently, assuming negligible interactions. This class of designs includes the regular fractional factorials (e.g., for $n = 4, 8, 16, \dots$) but also other designs with n a multiple of four but not a power of two ($n = 12, 20, 24, \dots$). These designs were first proposed by Plackett and Burman [84]. Table 33.2 gives the $n = 12$ -run Plackett-Burman (PB) design, one of the most frequently used for screening.

The price paid for the greater economy of run size offered by nonregular designs is more complex aliasing. Although designs formed from orthogonal arrays, including PB designs, allow estimation of each main effect independently of all other main effects, these estimators will usually be *partially aliased* with many two-variable interactions. That is, the alias matrix \mathbf{A} will contain many entries with $0 < |a_{ij}| < 1$. For example, consider the aliasing between main effects and two-variable interactions for the 12-run PB design in Table 33.2, as summarized in the 11×55 alias matrix. The main effect of each variable is partially aliased with all 45 interactions that do not include that variable. That is, for the i th variable,

$$\mathbb{E}(\hat{\beta}_i) = \beta_i + \frac{1}{3} \sum_{j=1}^{11} \sum_{k>j}^{11} (-1)^{b_{ijk}} (1 - \mathbb{1}_{i=j \cap i=k}) \beta_{jk},$$

where $\mathbb{1}_A$ is the indicator function for the set A and $b_{ijk} = 0$ or 1 ($i, j, k = 1, \dots, 11$). For this design, each interaction is partially aliased with nine main effects. The competing 16-run resolution III 2^{11-7} regular fraction has each main effect aliased with at most four two-variable interactions, and each interaction aliased only with at most one main effect. Hence, while an active interaction would

Table 33.3 The definitive screening design for $d = 6$ variables

Run	x_1	x_2	x_3	x_4	x_5	x_6
1	0	1	-1	-1	-1	-1
2	0	-1	1	1	1	1
3	1	0	-1	1	1	-1
4	-1	0	1	-1	-1	1
5	-1	-1	0	1	-1	-1
6	1	1	0	-1	1	1
7	-1	1	1	0	1	-1
8	1	-1	-1	0	-1	1
9	1	-1	1	-1	0	-1
10	-1	1	-1	1	0	1
11	1	1	1	1	-1	0
12	-1	-1	-1	-1	1	0
13	0	0	0	0	0	0

bias only one main effect for the regular design, it would bias nine main effects for the PB design, albeit to a lesser extent.

However, an important advantage of partial aliasing is that it allows interactions to be considered through the use of variable selection methods (discussed later) without requiring a large increase in the number of runs. For example, the 12-run PB design has been used to identify important interactions [26, 46].

A wide range of nonregular designs can be constructed. An algorithm has been developed for constructing designs which allow orthogonal estimation of all main effects together with catalogues of designs for $n = 12, 16, 20$ [101]. Other authors have used computer search and criteria based on model selection properties to find nonregular designs [58]. A common approach is to use criteria derived from D -optimality [4, ch. 11] to find fractional factorial designs for differing numbers of variables and runs [35]. Designs from these methods may or may not allow independent estimation of the variable main effects dependent on the models under investigation and the number of runs available.

Most screening experiments use designs at two levels, possibly with the addition of one or more center points to provide a portmanteau test for curvature. Recently, an economic class of three-level screening designs have been proposed, called “definitive screening designs” (DSDs) [53], to investigate d variables, generally in as few as $n = 2d + 1$ runs. The structure of the designs is illustrated in Table 33.3 for $d = 6$. The design has a single center point and $2d$ runs formed from d mirrored pairs. The j th pair has the j th variable set to zero and the other $d - 1$ variables set to ± 1 . The second run in the pair is formed by multiplying all the elements in the first run by -1 . That is, the $2d$ runs form a *foldover* design [17]. This foldover property ensures that main effects and two-variable interactions are orthogonal and hence main effects are estimated independently from these interactions, unlike for resolution III or PB designs. Further, all quadratic effects are estimated independently of the main effects but not independently of the two-variable interactions. Finally, the two-variable interactions will be partially aliased

with each other. These designs are growing in popularity, with a sizeable literature available on their construction [79, 82, 113].

2.3 Supersaturated Designs for Main Effects Screening

For experiments with a large number of variables or runs that are very expensive or time consuming, *supersaturated* designs have been proposed as a low-resource (small n) solution to the screening problem [42]. Originally, supersaturated designs were defined as having too few runs to estimate the intercept and the d main effects in model (33.2), that is, $n < d + 1$. The resulting partial aliasing is more complicated than for the designs discussed so far, in that at least one main effect estimator is biased by one or more other main effects. Consequently, there has been some controversy about the use of these designs [1]. Recently, evidence has been provided for the effectiveness of the designs when factor sparsity holds and the active main effects are large [34, 67]. Applications of supersaturated designs include screening variables in numerical models for circuit design [65], extraterrestrial atmospheric science [27], and simulation models for maritime terrorism [114].

Supersaturated designs were first proposed in the discussion [14] of random balance designs [98]. The first systematic construction method [11] found designs via computer search that have pairs of columns of the design matrix \mathbf{X}^n as nearly orthogonal as possible through use of the $\mathbb{E}(s^2)$ design selection criterion (defined below). There was no further research in the area for more than 30 years until Lin [61] and Wu [111] independently revived interest in the construction of these designs. Both their methods are based on Hadamard matrices and can be understood, respectively, as (i) selecting a half-fraction from a Hadamard matrix (Lin) and (ii) appending one or more interaction columns to a Hadamard matrix and assigning a new variable to each of these columns (Wu).

Both methods can be illustrated using the $n = 12$ -run PB design in Table 33.2. To construct a supersaturated design for $d = 10$ variables in $n = 6$ runs by method (i), all six runs of the PB design with $x_{11} = -1$ are removed, followed by deletion of the x_{11} column. The resulting design is shown in Table 33.4. To obtain a design by method (ii) for $d = 21$ variables in $n = 12$ runs, ten columns are appended that correspond to the interactions of x_1 with variables x_2 to x_{11} , and variables x_{12} to x_{21} are assigned to these columns; see Table 33.5.

Table 33.4 An $n = 6$ -run supersaturated design for $d = 10$ variables obtained by the method of Lin [61]

Run	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	-1	-1	-1	-1	-1	1	1	1	1	1
2	-1	-1	1	1	1	-1	-1	-1	1	1
3	-1	1	-1	1	1	-1	1	1	-1	-1
4	1	-1	1	-1	1	1	1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	1	1	-1
6	1	1	-1	1	-1	1	-1	-1	-1	1

Table 33.5 An $n = 12$ -run supersaturated design for $d = 21$ obtained by the method of Wu [111]

Run	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1
4	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	-1	-1	-1	-1	-1	1	1	1	-1
5	-1	1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
6	-1	1	1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
7	1	-1	1	1	-1	-1	1	1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1
8	1	-1	1	-1	1	1	-1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	-1	-1	1
9	1	-1	1	1	1	-1	1	1	-1	-1	-1	-1	1	1	1	-1	1	1	-1	-1	-1
10	1	1	-1	-1	-1	-1	1	1	-1	1	1	-1	-1	-1	-1	1	1	1	-1	1	1
11	1	1	-1	1	-1	1	-1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	1
12	1	1	-1	-1	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	-1

Since 1993, there has been a substantial research effort on construction methods for supersaturated designs; see, for example, [62, 77, 78]. The most commonly used criterion for design selection in the literature is $\mathbb{E}(s^2)$ -optimality [11]. More recently, the Bayesian D -optimality criterion [35, 51] has become popular.

2.3.1 $\mathbb{E}(s^2)$ -Optimality

This criterion selects a *balanced* design, that is, a design with $n = 2m$ for some integer $m > 0$ where each column of \mathbf{X}^n contains m entries equal to -1 and m entries equal to $+1$. The $\mathbb{E}(s^2)$ -optimal design minimizes the average of the squared inner products between columns i and j of \mathbf{X}^n ($i, j = 1, \dots, d; i \neq j$),

$$\mathbb{E}(s^2) = \frac{2}{d(d-1)} \sum_{i < j} s_{ij}^2, \quad (33.9)$$

where s_{ij} is the ij th element of $(\mathbf{X}^n)^T \mathbf{X}^n$ ($i, j = 1, \dots, d$). A lower bound on $\mathbb{E}(s^2)$ is available [19, 94]. The designs in Tables 33.4 and 33.5 achieve the lower bound and hence are $\mathbb{E}(s^2)$ -optimal. For the design in Table 33.4, each $s_{ij}^2 = 4$. For the design in Table 33.5, $\mathbb{E}(s^2) = 6.857$ (to 3 dp), with 120 pairs of columns being orthogonal ($s_{ij}^2 = 0$) and the remaining 90 pairs of columns having $s_{ij}^2 = 16$. Recently, the definition of $\mathbb{E}(s^2)$ has been extended to unbalanced designs [52, 67] by including the inner product between each column of \mathbf{X}^n and the vector $\mathbf{1}_n$, the $n \times 1$ vector with every entry 1, which corresponds to the intercept term in model (33.1). This extension widens the class of available designs.

2.3.2 Bayesian D -Optimality

This criterion selects a design that maximizes the determinant of the posterior variance-covariance matrix for $(\beta_0, \boldsymbol{\beta}^T)^T$,

$$\Psi_D = |(\mathbf{H}^*)^T \mathbf{H}^* + \mathbf{K}/\tau^2|^{1/(d+1)}, \quad (33.10)$$

where $\mathbf{H}^* = [\mathbf{1}_n | \mathbf{X}^n]$, $\mathbf{K} = \mathbf{I}_{d+1} - \mathbf{e}_{1(d+1)} \mathbf{e}_{1(d+1)}^T$, $\tau^2 > 0$, and $\tau^2 \mathbf{K}^{-1}$ is the prior variance-covariance matrix for $\boldsymbol{\beta}$. Equation (33.10) results from assuming an informative prior distribution for each β_i ($i = 1, \dots, d$) with mean zero and small prior variance, to reflect factor sparsity, and a non-informative prior distribution for β_0 . The prior information can be regarded as equivalent to having sufficient additional runs to allow estimation of all parameters β_0, \dots, β_d , with the value of τ^2 reflecting the quantity of available prior information. However, the optimal designs obtained tend to be insensitive to the choice of τ^2 [67].

Both $\mathbb{E}(s^2)$ - and D -optimal designs may be found numerically, using algorithms such as columnwise-pairwise [59] or coordinate exchange [71]. From simulation studies, it has been shown that there is little difference in the performance of $\mathbb{E}(s^2)$ - and Bayesian D -optimal designs assessed by, for example, sensitivity and type I error rate [67].

Supersaturated designs have also been constructed that allow the detection of two-variable interactions [64]. Here, the definition of supersaturated has been widened to include designs that have fewer runs than the total number of factorial effects to be investigated. In particular, Bayesian D -optimal designs have been shown to be effective in identifying active interactions [34]. Note that under this expanded definition of supersaturated designs, all fractional factorial designs are supersaturated under model (33.1) when $n < p$.

2.4 Common Issues with Factorial Screening Designs

The analysis of unreplicated factorial designs commonly used for screening experiments has been a topic of much research [45, 56, 105]. In a physical experiment, the lack of replication to provide a model-free estimate of σ^2 can make it difficult to assess the importance of individual factorial effects. The most commonly applied method for orthogonal designs treats this problem as analogous to the identification of outliers and makes use of (half-) normal plots of the factorial effects. For many nonregular and supersaturated designs, more advanced analysis methods are necessary; see later. For studies on numerical models, provided all the input variables are controlled, the problem of assessing statistical significance does not occur as no unusually large observations can have arisen due to “chance.” Here, factorial effects may be ranked by size and those variables whose effects lead to a substantive change in the response declared active.

Biased estimators of factorial effects, however, are an issue for experiments on both numerical models and physical processes. Complex (partial) aliasing can produce two types of bias in the estimated parameters in model (33.1): upward bias so that a type I error may occur (amalgamation) or downward bias leading to missing active variables (cancellation). Simulation studies have been used to assess these risks [31, 34, 67].

Bias may also, of course, be induced by assuming a form of the surrogate model that is too simple, for example, through the surrogate having too few turning points (e.g., being a polynomial of too low order) or lacking the detail to explain the local behavior of the numerical model. This kind of bias is potentially the primary source of mistakes in screening variables in numerical models. When prior scientific knowledge suggests that the numerical model is highly nonlinear, screening methods should be employed that have fewer restrictions on the surrogate model or are model-free. Such methods, including designs for the estimation of elementary effects (33.8), are described later in this paper. Typically, they require larger experiments than the designs in the present section.

2.5 Systematic Fractional Replicate Designs

Systematic fractional replicate designs [28] enable expressions to be estimated that indicate the influence of each variable on the output, through main effects and

interactions, without assumptions in model (33.1) on the order of interactions that may be important. These designs have had considerable use for screening inputs to numerical models, especially in the medical and biological sciences [100, 116]. In these designs, each variable takes two levels and there are $n = 2d + 2$ runs.

The designs are simple to construct as (i) one run with all variables set to -1 , (ii) d runs with each variable in turn set to $+1$ and the other variables set to -1 , (iii) d runs with each variable in turn set to -1 and the other variables set to $+1$, and (iv) one run with all variables set to $+1$. Let the elements of vector \mathbf{Y}^n be such that $Y^{(1)}$ is the output from the run in (i), $Y^{(2)}, \dots, Y^{(d+1)}$ are the outputs from the runs in (ii), $Y^{(d+2)}, \dots, Y^{(2d+1)}$ are from the runs in (iii), and Y^{2d+2} is from the run in (iv). In such a design, each main effect can be estimated independently of all two-variable interactions. This can easily be seen from the alternative construction as a foldover from a *one-factor-at-a-time* (OFAAT) design with $n = d + 1$, that is, a design having one run with each variable set to -1 and d runs with each variable in turn set to $+1$ with all other variables set to -1 .

For each variable x_i ($i = 1, \dots, d$), two linear combinations, $S_o(i)$ and $S_e(i)$, of “odd order” and “even order” model parameters, respectively, can be estimated:

$$S_o(i) = \beta_i + \sum_{j=1}^d \sum_{\substack{k=1 \\ i \neq j \neq k}}^d \beta_{ijk} + \dots, \quad (33.11)$$

and

$$S_e(i) = \sum_{\substack{j=1 \\ i \neq j}}^d \beta_{ij} + \sum_{j=1}^d \sum_{\substack{k=1 \\ i \neq j \neq k}}^d \sum_{l=1}^d \beta_{ijkl} + \dots, \quad (33.12)$$

with respective unbiased estimators

$$C_0(i) = \frac{1}{4} \{ (Y^{(2d+2)} - Y^{(d+i+1)}) + (Y^{(i+1)} - Y^{(1)}) \},$$

and

$$C_e(i) = \frac{1}{4} \{ (Y^{(2d+2)} - Y^{(d+i+1)}) - (Y^{(i+1)} - Y^{(1)}) \}.$$

Under effect hierarchy, it may be anticipated that a large absolute value of $C_o(i)$ is due to a large main effect for the i th variable, and a large absolute value of $C_e(i)$ is due to large two-variable interactions. A design that also enables estimation of two-variable interactions independently of each other is obtained by appending $(d-1)(d-2)/2$ runs, each having two variables set to $+1$ and $d-2$ variables set to -1 [91].

For numerical models, where observations are not subject to random error, active variables are selected by ranking the sensitivity indices defined by

$$S(i) = \frac{M(i)}{\sum_{j=1}^d M(j)}, \quad i = 1, \dots, d, \quad (33.13)$$

where $M(i) = |C_o(i)| + |C_e(i)|$. This methodology is potentially sensitive to the cancellation or amalgamation of factorial effects, discussed in the previous section.

From (33.8), it can also be seen that use of a systematic fractional replicate design is equivalent to calculating two elementary effects (with $\Delta = 2$) for each variable at the extremes of the design region. Let $EE_{1i} = (Y^{(2d+2)} - Y^{(d+i+1)})/2$ and $EE_{2i} = (Y^{(i+1)} - Y^{(1)})/2$ be these elementary effects for the i th variable. Then, it follows directly that $S(i) \propto \max(|EE_{1i}|, |EE_{2i}|)$, and the above method selects as active those variables with elementary effects that are large in absolute value.

3 Screening Groups of Variables

Early work on group screening used pooled blood samples to detect individuals with a disease as economically as possible [33]. The technique was extended, almost 20 years later, to screening large numbers of two-level variables in factorial experiments where a main effects only model is assumed for the output [108]. For an overview of this work and several other strategies, see [75].

In group screening, the set of variables is partitioned into groups, and the values of the variables within each group are varied together. Smaller designs can then be used to experiment on these groups. This strategy deliberately aliases the main effects of the individual variables. Hence, follow-up experimentation is needed on those variables in the groups found to be important in order to detect the individual active variables. The main screening techniques that employ grouping of variables are described below.

3.1 Factorial Group Screening

The majority of factorial group screening methods apply to variables with two levels and use two stages of experimentation. At the first stage, the d variables are partitioned into g groups, where the j th group contains $g_j \geq 1$ variables ($j = 1, \dots, g$). High and low levels for each of the g grouped variables are defined by setting all the individual variables in a group to either their high level or their low level simultaneously. The first experiment with n_1 runs is performed on the relatively small number of grouped variables. *Classical group screening* then estimates the main effects for each of the grouped variables and takes those variables involved in groups that have large estimated main effects through to a second-stage experiment. Individual variables are investigated at this stage, and their main effects, and possibly interactions, are estimated.

For sufficiently large groups of variables, highly resource-efficient designs can be employed at stage 1 of classical group screening for even very large numbers of factors. Under the assumption of negligible interactions, orthogonal nonregular designs, such as PB designs, may be used. For screening variables from a deterministic numerical model, designs in which the columns corresponding to the grouped main effects are not orthogonal can be effective [9] provided $n_1 > g + 1$, as the precision of factorial effect estimators is not a concern.

Effective classical group screening depends on strong effect heredity, namely, that important two-variable interactions occur only between variables both having important main effects. More recently, strategies for group screening that also investigate interactions at stage 1 have been developed [57]. In *interaction group screening*, both main effects and two-variable interactions between the grouped variables are estimated at stage 1. The interaction between two grouped variables is the summation of the interactions between all pairs of variables where one variable comes from each group; interactions between two variables in the same group are aliased with the mean. Variables in groups found to have large estimated main effects or to be involved in large interactions are carried forward to the second stage. From the second-stage experiment, main effects and interactions are examined between the individual variables within each group declared active. Where the first stage has identified a large interaction between two grouped variables, the interactions between pairs of individual variables, one from each group, are also investigated. For this strategy, larger resolution V designs, capable of independently estimating all grouped main effects and two-variable interactions, have so far been used at stage 1, when decisions to drop groups of variables are made.

Group screening experiments can be viewed as supersaturated experiments in the individual variables. However, when orthogonal designs are used for the stage 1 experiment, decisions on which groups of variables to take forward can be made using t -tests on the grouped main effects and interactions. When smaller designs are used, particularly if n_1 is less than the number of grouped effects of interest, more advanced modeling methods are required, in common with other supersaturated designs (see later). Incorrectly discarding active variables at stage 1 may result in missed opportunities to improve process control or product quality. Hence, it is common to be conservative in the choice of design at stage 1, for example, in the number of runs, and also to allow a higher type I error rate.

In the two-stage process, the design for the second experiment cannot be decided until the stage 1 data have been collected and the groups of factors deemed active have been identified. In fact, the size, N_2 , of the second-stage experiment required by the group screening strategy is a random variable. The distribution of N_2 is determined by features under the experimenter's control, such as d , g , g_1, \dots, g_g , n_1 , the first-stage design, and decision rules for declaring a grouped variable active at stage 1. It also depends on features outside the experimenter's control, such as the number of active individual variables and the size and nature of their effects, and the signal-to-noise ratio if the process is noisy. Given prior knowledge of these uncontrollable features, the grouping strategy, designs, and analysis methods can be tailored, for example, to produce a smaller expected experiment size, $n_1 + \mathbb{E}(N_2)$,

or to minimize the probability of missing active variables [57, 104]. Of course, these two goals are usually in conflict and hence a trade-off has to be made. In practice, the design used at stage 2 depends on the number of variables brought forward and the particular effects requiring estimation; options include regular or nonregular fractional factorial designs and D -optimal designs.

Original descriptions of classical group screening made the assumption that all the active variable main effects have the same sign to avoid the possibility of cancellation of the main effects of two or more active variables in the same group. As discussed previously, cancellation can affect any fractional factorial experiment. Group screening is often viewed as particularly susceptible due to the complete aliasing of main effects of individual variables and the screening out of whole groups of variables at stage 1. Often, particularly for numerical models, prior knowledge makes reasonable the assumption of active main effects having the same sign. Otherwise, the risks of missing active variables should be assessed by simulation [69], and, in fact, the risk can be modest under factor sparsity [34].

3.2 Sequential Bifurcation

Screening groups of variables is also used in sequential bifurcation, proposed originally for deterministic simulation experiments [10]. The technique can investigate a very large number of variables, each having two levels, when a sufficiently accurate surrogate for the output is a first-order model (33.2). It is assumed that each parameter β_i ($i = 1, \dots, d$) is positive (or can be made positive by interchanging the variable levels) to avoid cancellation of effects.

The procedure starts with a single group composed of all the variables which is split into two new groups (bifurcation). For a deterministic numerical model, the initial experiment has just two runs: all variables set to the low levels ($\mathbf{x}^{(1)}$) and all variables set to the high levels ($\mathbf{x}^{(2)}$). If the output $Y^{(2)} > Y^{(1)}$, then the group is split, with variables x_1, \dots, x_{d_1} placed in group 1 and x_{d_1+1}, \dots, x_d placed in group 2. At the next stage, a single further run $\mathbf{x}^{(3)}$ is made which has all group 1 variables set to their high levels and all group 2 variables set low. If $Y^{(3)} > Y^{(1)}$, then group 1 is split further, and group 2 is split if $Y^{(2)} > Y^{(3)}$. These comparisons can be replaced by $Y^{(3)} - Y^{(1)} > \delta$ and $Y^{(2)} - Y^{(3)} > \delta$, where δ is an elicited threshold. This procedure of performing one new run and assessing the split of each subsequent group continues until singleton groups, containing variables deemed to be active, have been identified. Finally, these individual variables are investigated. If the output variable is stochastic, the replications of each run are made, and a two-sample t -test can be used to decide whether or not to split a group.

Typically, if $d = 2^k$ for some integer $k > 0$, then at each split, half the variables are assigned to one of the groups, and the other half are assigned to the second group. Otherwise, use of unequal group sizes can increase the efficiency (in terms of overall experiment size) of sequential bifurcation when there is prior knowledge of effect sizes. Then, at each split, the first new group should have size equal to the largest possible power of 2. For example, if the group to be split contains m variables, then

the first new group should contain 2^l variables such that $2^l < m$. The remaining $m - 2^l$ variables are assigned to the second group. If variables have been ordered by an a priori assessment of increasing importance, the most important variables will be in the second, smaller group, and hence more variables can be ruled out as unimportant more quickly.

The importance of two-variable interactions may be investigated by using the output from the following runs to assess each split. The first is run \mathbf{x} used in the standard sequential bifurcation method; the second is the mirror image of \mathbf{x} in which each variable is set low that is set high in \mathbf{x} and vice versa. This foldover structure ensures that any two-variable interactions will not bias estimators of grouped main effects at each stage. This alternative design also permits standard sequential bifurcation to be performed and, if the variables deemed active differ from those found via the foldover, then the presence of active interactions is indicated. Again, successful identification of the active variables relies on the principle of strong effect heredity.

A variety of adaptations of sequential bifurcation have been proposed, including methods of controlling type I and type II error rates [106, 107] and a procedure to identify dispersion effects in robust parameter design [3]. For further details, see [55, ch. 4].

3.3 Iterated Fractional Factorial Designs

These designs [2] also group variables in a sequence of applications of the same fractional factorial design. Unlike factorial group screening and sequential bifurcation, the variables are assigned at random to the groups at each stage. Individual variables are identified as active if they are in the intersection of those groups having important main effects at each stage.

Suppose there are $g = 2^l$ groups, for integer $l > 0$. The initial design has $2g$ runs obtained as a foldover of a $g \times g$ Hadamard matrix; for details, see [23]. This construction gives a design in which main effects are not aliased with two-variable interactions. The $d \geq g$ variables are assigned at random to the groups, and each grouped variable is then assigned at random to a column of the design. The experiment is performed and analyzed as a stage 1 group screening design. Subsequent stages repeat this procedure, using the same design but with different, independent assignments of variables to groups and groups to columns. Individual variables which are common to groups of variables found to be active across several stages of experimentation are deemed to be active. Estimates of the main effects using data from all the stages can also be constructed.

There are two further differences from the other grouping methods discussed in this section. First, for a proportion of the stages, the variables are set to a mid-level value (0), rather than high (+1) or low (-1). These runs allow an estimate of curvature to be made and some screening of quadratic effects to be undertaken. Second, to mitigate cancellation of main effects, the coding of the high and low levels may be swapped at random, that is, the sign of the main effect reversed.

The use of iterated fractional factorial designs requires a larger total number of runs than other group screening methods, as a sequence of factorial screening designs is implemented. However, the method has been suggested for use when there are many variables (thousands) arranged in a few large groups. Simulation studies [95, 96] have indicated that it can be effective here, provided there are very few active variables.

3.4 Two-Stage Group Screening for Gaussian Process Models

More recently, methodology for group screening in two stages of experimentation using Gaussian process modeling to identify the active variables has been developed for numerical models [73]. At the first stage, an initial experiment that employs an orthogonal space-filling design (see the next section) is used to identify variables to be grouped together. Examples are variables that are inert or those having a similar effect on the output, such as having a common sign and a similarly-sized linear or quadratic effect. A sensitivity analysis on the grouped variables is then performed using a Gaussian process model, built from the first-stage data. Groups of variables identified as active in this analysis are investigated in a second-stage experiment in which the variables found to be unimportant are kept constant. The second-stage data are then combined with the first-stage data and a further sensitivity analysis performed to make a final selection of the active variables. An important advantage of this method is the reduced computational cost of performing a sensitivity study on the grouped variables at the first stage.

4 Random Sampling Plans and Space Filling

4.1 Latin Hypercube Sampling

The most common experimental design used to study deterministic numerical models is the Latin hypercube sample (LHS) [70]. These designs address the difficult problem of space filling in high dimensions, that is, when there are many controllable variables. Even when adequate space filling in d dimensions with n points may be impossible, an LHS design offers n points that have good one-dimensional space-filling properties for a chosen distribution, usually a uniform distribution. Thus, use of an LHS at least implicitly invokes the principle of factor sparsity and hence is potentially suited for use in screening experiments.

Construction of a standard d -dimensional LHS is straightforward: generate d random permutations of the integers $1, \dots, n$ and arrange them as an $n \times d$ matrix \mathbf{D} (one permutation forming each column); transform each element of \mathbf{D} to obtain a sample from a given distribution $F(\cdot)$, that is, define the coordinates of the design points as $x_j^{(i)} = F^{-1} \left\{ (d_j^{(i)} - 1)/(n - 1) \right\}$, where $d_j^{(i)}$ is the ij th element of \mathbf{D} ($i = 1, \dots, n$; $j = 1, \dots, d$). Typically, a (small) random perturbation is added to

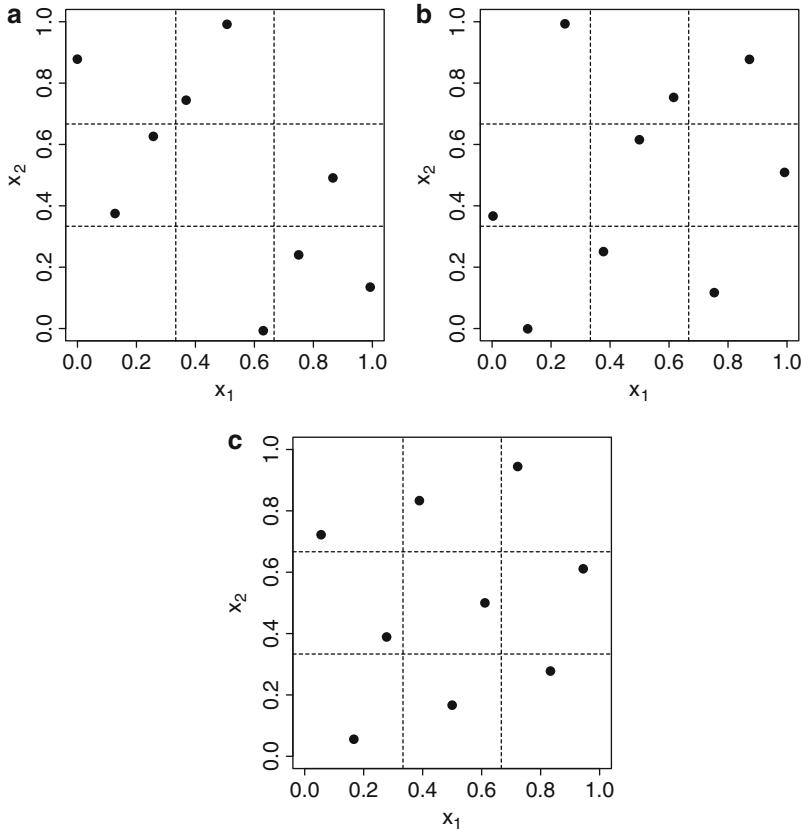


Fig. 33.1 Latin hypercube samples with $n = 9$ and $d = 2$: (a) random LHS, (b) random LHS generated from an orthogonal array, (c) maximin LHS

each $d_j^{(i)}$ or some equivalent operation performed, prior to transformation to $x_j^{(i)}$. An LHS design generated by this method is shown in Fig. 33.1a.

There may be great variation in the overall space-filling properties of LHS designs. For example, the LHS design in Fig. 33.1a clearly has poor two-dimensional space-filling properties. Hence, a variety of extensions to Latin hypercube sampling have been proposed. Most prevalent are orthogonal array-based and maximin Latin hypercube sampling.

To generate an orthogonal array-based LHS [81, 102], the matrix \mathbf{D} is formed from an orthogonal array. Hence, the columns of \mathbf{D} are no longer independent permutations of $1, \dots, d$. For simplicity, assume \mathbf{O} is a symmetric OA(n, s^d, t) with symbols $1, \dots, s$ and $t \geq 2$. The j th row of \mathbf{D} is formed by mapping the n/s occurrences of each symbol in the j th column of \mathbf{O} to a random permutation, α , of n/s new symbols, i.e., $1 \rightarrow \alpha(1, \dots, n/s)$, $2 \rightarrow \alpha(n/s + 1, \dots, 2n/s)$, ..., $s \rightarrow \alpha((s-1)n/s + 1, \dots, n)$, where $\alpha(1, \dots, a)$ is a permutation of the

integers $1, \dots, a$. Figure 33.1b shows an orthogonal array-based LHS, constructed from an OA(9, 3^2 , 2). Notice the improved two-dimensional space filling compared with the randomly generated LHS. The two-dimensional projection properties of more general space-filling designs have also been considered by other authors [29], especially for *uniform* designs minimizing specific L^2 -discrepancies [38].

In addition to the orthogonal array-based LHS, there has been a variety of work on generating space-filling designs that directly minimize the correlation between columns of \mathbf{X}^n [47], including algorithmic [103] and analytic [117] construction methods. Such designs have good two-dimensional space-filling properties and also provide near-independent estimators of the β_i ($i = 1, \dots, d$) in Eq. (33.2), a desirable property for screening.

A maximin LHS [76] achieves a wide spread of design points across the design region by maximizing the minimum distance between pairs of design points within the class of LHS designs with n points and d variables. The Euclidean distance between two points $\mathbf{x} = (x_1, \dots, x_d)^\top$ and $\mathbf{x}' = (x'_1, \dots, x'_d)^\top$ is given by

$$\text{dist}(\mathbf{x}, \mathbf{x}') = \left\{ \sum_{j=1}^d (x_j - x'_j)^2 \right\}^{1/2}. \quad (33.14)$$

Rather than maximizing directly the minimum of (33.14), most authors [6, 76] find designs by minimization of

$$\phi(\mathbf{X}^n) = \left\{ \sum_{1 \leq i < j \leq n} \left[\text{dist}(\mathbf{x}_i^n, \mathbf{x}_j^n) \right]^{-q} \right\}^{1/q}, \quad (33.15)$$

where, for $q \rightarrow \infty$, minimization of (33.15) is equivalent to maximizing the minimum of (33.14) across all pairs of design points; see also [85]. Figure 33.1c shows a maximin LHS, found by this method with $q = 15$. This design displays better two-dimensional space filling than the random LHS and a more even distribution of the design points than the orthogonal array-based LHS.

Maximin LHS can be found using the R packages DiceDesign [40] and SLHD [5]. More general classes of distance-based space-filling designs, without the projection properties of the Latin hypercubes, can also be found [see 13, 50]. Studies of the numerical efficiencies of optimization algorithms for LHS designs are available [29, 49].

Construction of LHS designs is an active area of research, and many further extensions to the basic methods have been suggested. Recently, maximum projection space-filling designs have been found [54] that minimize

$$\psi(\mathbf{X}^n) = \left\{ \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \frac{1}{\prod_{l=1}^d (x_i^{(l)} - x_j^{(l)})^2} \right\}. \quad (33.16)$$

Such a design promotes good space-filling properties, as measured by (33.15), in all projections of the design into subspaces of variables. Objective function (33.16) arises from the use of a weighted Euclidean distance; see also [13].

Another important recent development is *sliced* LHS designs [6, 88, 89], where the design can be partitioned into sets of runs or slices, each of which is an orthogonal or maximin LHS. The overall design, composed of the runs from all the slices, is also an LHS. Such designs may be used to study multiple numerical models having the same inputs where one slice is used to investigate each model, for example, to compare results from different model implementations. They are also useful for experiments on quantitative and qualitative variables when each slice is combined with one combination of levels of the qualitative variables. This latter application is the most important for screening where the resulting data may be used to estimate a surrogate linear model with dummy variables or a GP model with an appropriate correlation structure [90]. Supersaturated LHS designs, for $d \geq n$, have also been developed [21]. See *Maximin sliced Latin hypercube designs, with application to cross validating prediction error* for more on space-filling and Latin hypercube designs.

4.2 Sampling Plans for Estimating Elementary Effects (Morris' Method)

As an alternative to estimation of a surrogate model, Morris [74] suggested a model-free approach that uses the elementary effect (33.8) to measure the sensitivity of the response $Y(\mathbf{x})$ to a change in the i th variable at point \mathbf{x} . Each $\text{EE}_i(\mathbf{x})$ may be exactly or approximately (a) zero for all \mathbf{x} , implying a negligible influence of the i th variable on $Y(\mathbf{x})$; (b) a nonzero constant for all \mathbf{x} , implying a linear, additive effect; (c) a nonconstant function of x_i , implying nonlinearity; or (d) a nonconstant function of x_j for $j \neq i$, implying the presence of at least one interaction involving x_i .

In practice, active variables are usually selected using data from a relatively small experiment, and therefore it is not possible to reconstruct $\text{EE}_i(\mathbf{x})$ as a continuous function of \mathbf{x} . The use of r “trajectory vectors”, $\mathbf{x}_1, \dots, \mathbf{x}_r$, enables the following sensitivity indices to be defined for $i = 1, \dots, d$:

$$\mu_i = \frac{1}{r} \sum_{j=1}^r \text{EE}_i(\mathbf{x}_j) \quad (33.17)$$

and

$$\sigma_i = \sqrt{\sum_{j=1}^r \frac{(\text{EE}_i(\mathbf{x}_j) - \mu_i)^2}{r-1}}. \quad (33.18)$$

A large value of the mean μ_i suggests the i th input variable is active. Nonlinear and interaction effects are indicated by large values of σ_i . Plots of the sensitivity indices may be used to decide which variables are active and, among these variables, which have complex (nonadditive and nonlinear) effects.

In addition to (33.17) and (33.18), an additional measure [22] of the individual effect of the i th variable has been proposed that overcomes the possible cancellation of elementary effects in (33.17) due to non-monotonic variable effects, namely,

$$\mu_i^* = \frac{1}{r} \sum_{j=1}^r |\text{EE}_i(\mathbf{x}_j)|, \quad (33.19)$$

where $|\cdot|$ denotes the absolute value. Large values of both μ_i and μ_i^* suggest that the i th variable is active and has a linear effect on $Y(\mathbf{x})$; large values of μ_i^* and small values of μ_i indicate cancellation in (33.17) and a nonlinear effect for the i th variable.

Four examples of the types of effects which use of (33.17), (33.18) and (33.19) seeks to identify are shown in Fig. 33.2. These are linear, nonlinear, and interaction effects corresponding to nonzero values of (33.17) and (33.19) and both zero and nonzero values of (33.18). It is not possible to use these statistics to distinguish between nonlinear and interaction effects; see Fig. 33.2b, c.

The future development of a surrogate model can be simplified by separation of the active variables into two vectors, \mathbf{x}_{S_1} and \mathbf{x}_{S_2} , where the variables in \mathbf{x}_{S_1} have linear effects and the variables in \mathbf{x}_{S_2} have nonlinear effects or are involved in interactions [12]. For example, model (33.1) might be fitted in which $\mathbf{h}(\mathbf{x}_{S_1})$ consists of linear functions and $\varepsilon(\mathbf{x}_{S_2})$ is modeled via a Gaussian process with correlation structure dependent only on variables in \mathbf{x}_{S_2} .

In the elementary effect literature, the design region is assumed to be $\mathcal{X} = [0, 1]^d$, after any necessary scaling, and is usually approximated by a d -dimensional grid, \mathcal{X}_G , having f equally spaced values, $0, 1/(f - 1), \dots, 1$, for each input variable. The design of a screening experiment to allow the computation of the sample moments (33.17), (33.18) and (33.19) has three components: the trajectory vectors $\mathbf{x}_1, \dots, \mathbf{x}_r$, the m -run sub-design used to calculate $\text{EE}_i(\mathbf{x}_j)$ ($i = 1, \dots, d$) for each \mathbf{x}_j ($j = 1, \dots, r$), and stepsize Δ . Choices for these components are now discussed.

1. Morris [74] chose $\mathbf{x}_1, \dots, \mathbf{x}_r$ at random from \mathcal{X}_G , subject to the constraint that $\mathbf{x}_j + \Delta \mathbf{e}_{id} \in \mathcal{X}_G$ for $i = 1, \dots, d$; $j = 1, \dots, r$. Alternative suggestions include choosing the trajectory vectors as the r points of a space-filling design [22] found, for example, by minimizing (33.15). Larger values of r result in (33.17), (33.18) and (33.19) being more precise estimators of the corresponding moments of the elementary effect distribution. Morris used $r = 3$ or 4 ; more recently, other authors [22] have discussed the use of larger values ($r = 10\text{--}50$).
2. An OFAAT design with $m = d + 1$ runs may be used to calculate $\text{EE}_1(\mathbf{x}_j), \dots, \text{EE}_d(\mathbf{x}_j)$ for $j = 1, \dots, r$. The design matrix is

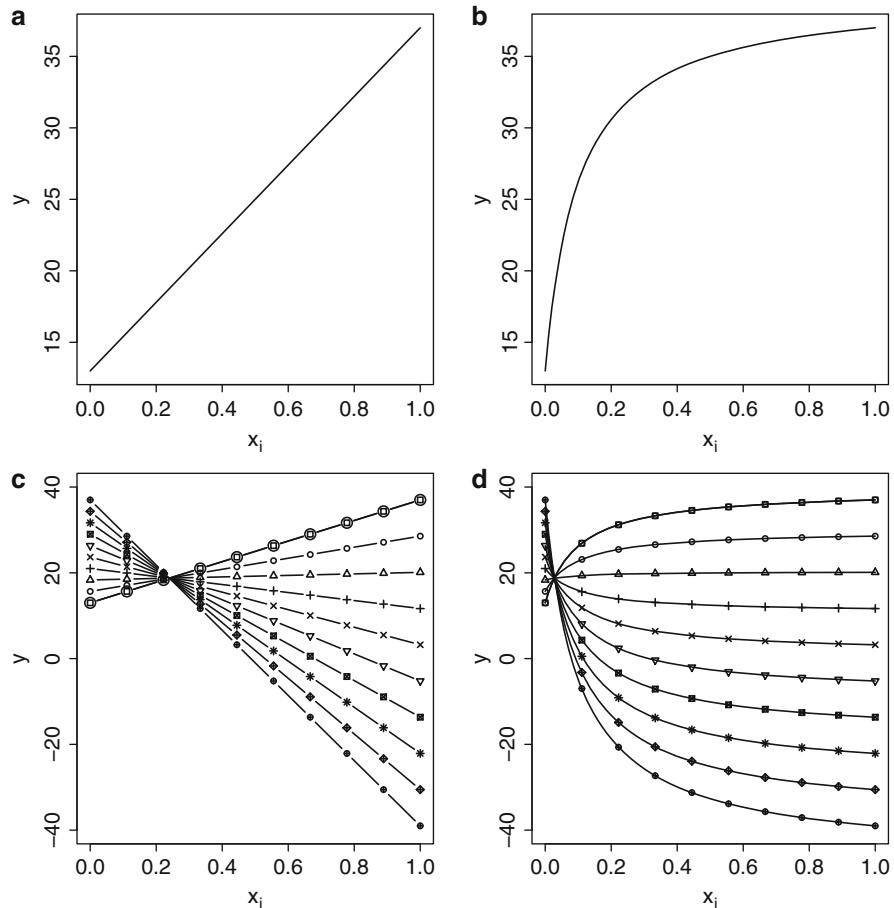


Fig. 33.2 Illustrative examples of effects for an active variable x_i with values of μ_i , μ_i^* , and σ_i . In plots (c) and (d), the plotting symbols correspond to the ten levels of a second variable. (a) Linear effect: $\mu_i > 0$, $\mu_i^* > 0$; $\sigma_i = 0$. (b) Nonlinear effect: $\mu_i^* > 0$; $\sigma_i > 0$. (c) Interaction: $\sigma_i > 0$. (d) Nonlinear effect and interaction: $\mu_i^* > 0$, $\sigma_i > 0$

$\mathbf{X}_j = \mathbf{1}_{d+1}\mathbf{x}_j + \Delta\mathbf{B}$, where $\mathbf{1}_m$ is a column m -vector with all entries equal to 1 and $\mathbf{B} = \sum_{l=2}^{d+1} \sum_{k=1}^{l-1} \mathbf{e}_{l(d+1)} \mathbf{e}_{k(d+1)}^T$. That is, \mathbf{B} is the $(d+1) \times d$ matrix

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}.$$

Designs are generated by swapping 0's and 1's at random within each column of \mathbf{B} and randomizing the column order. The overall design $\mathbf{X}^n = (\mathbf{X}_1^T, \dots, \mathbf{X}_r^T)^T$ then has $n = (d + 1)r$ runs. It can be shown that this choice of \mathbf{B} , combined with the randomization scheme, minimizes the variability in the number of times in \mathbf{X}^n that each variable takes each of the f possible values [74].

Such OFAAT designs have the disadvantage of poor projectivity onto subsets of (active) variables, compared with, for example, LHS designs. Projection of a d -dimensional OFAAT design into $d - 1$ dimensions reduces the number of distinct points from $d + 1$ to d . This is a particular issue when the projection of the screening design onto the active variables is used to estimate a detailed surrogate model. Better projection properties may be obtained by replacing an OFAAT design by a rotated simplex [86] at the cost of less precision in the estimators of the elementary effects.

3. The choice of the “stepsize” Δ in (33.8) is determined by the choice of design region \mathcal{X}_G . Recommended values are $f = 2g$ for some integer $g > 0$ and $\Delta = f/2(f - 1)$. This choice ensures that all $f^d/2$ elementary effects are equally likely to be selected for each variable when the trajectory vectors \mathbf{x} are selected at random from \mathcal{X}_G .

Further extensions of the elementary effect methodology include the application to group screening for numerical models with hundreds or thousands of input variables through the study of μ^* (33.19) [22], and a sequential experimentation strategy to reduce the number of runs of the numerical model required [12]. This latter approach performs r OFAAT experiments, one for each trajectory vector, in turn. For the j th experiment, elementary effects are calculated only for those variables that have not already been identified as having a nonlinear or interaction effect. That is, if $\sigma_i > \sigma_0$, for some threshold σ_0 , when σ_i is calculated from $r_1 < r$ trajectory vectors, the i th elementary effect is no longer calculated for $\mathbf{x}_{r_1+1}, \dots, \mathbf{x}_r$. The threshold σ_0 can be elicited directly from subject experts or by using prior knowledge about departures from linearity for the effect of each variable [12]. An obvious generalization of the elementary effect method is to compute sensitivity indices directly from the (averaged local) derivatives (see ▶ Chap. 33, “Design of Experiments for Screening”).

Methodology for the design and analysis of screening experiments using elementary effects is available in the R package `sensitivity` [87].

5 Model Selection Methods

The selection and estimation of a surrogate model (33.1) from an application of a design discussed in this paper generally requires advanced statistical methods. An exception is a regular fractional factorial design. For these designs, standard linear modeling methods can be used provided that only one effect from each alias string is included in the model and it is recognized that $\hat{\beta}$ may be biased. A brief description

is now given of variable selection methods for (a) linear models with nonregular and supersaturated designs and (b) Gaussian process models.

5.1 Variable Selection for Nonregular and Supersaturated Designs

For designs with complex partial aliasing such as supersaturated and nonregular fractional factorial designs, a wide range of model selection methods have been proposed. An early suggestion was forward stepwise selection [72], but this was shown to have low sensitivity in many situations [1, 67]. More recently, evidence has been provided for the effectiveness of shrinkage regression for the selection of active effects using data from these more complex designs [34, 67, 83], particularly use of the Dantzig selector [24]. For this method, estimators $\hat{\beta}$ of the parameters in model (33.1) are chosen to satisfy

$$\min_{\hat{\beta} \in \mathbb{R}^p} \sum_{u=1}^p |\hat{\beta}_u| \quad \text{subject to } \|\mathbf{H}^T(\mathbf{Y}^n - \mathbf{H}\hat{\beta})\|_\infty \leq s, \quad (33.20)$$

with s a tuning constant and $\|\mathbf{a}\|_\infty = \max |a_i|$, $\mathbf{a}^T = (a_1, \dots, a_p)$. This equation balances the desire for a parsimonious model with the need for models that adequately describe the data. The value of s can be chosen via an information criterion [20] (e.g., AIC, AICc, BIC). The solution to (33.20) may be obtained via linear programming; computationally efficient algorithms exist for calculating coefficient paths for varying s [48].

The Dantzig selector is applied to choose a subset of potentially active variables, and then standard least squares is used to fit a reduced linear model. The terms in this model whose coefficient estimates exceeded a threshold t , elicited from subject experts, are declared active. This procedure is known as the Gauss-Dantzig selector [24].

Other methods for variable selection that have been effective for these designs include Bayesian methods that use mixture prior distributions for the elements of β [26, 41] and the application of stochastic optimization algorithms [110].

5.2 Variable Selection for Gaussian Process Models

Screening for a Gaussian process model (33.1), with constant mean ($\mathbf{h}(\mathbf{x}) = 1$) and $\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x})'$ having correlation (33.7), may be performed by deciding which θ_i in (33.7) are “large” using data obtained from an LHS or other space-filling design. Two approaches to this problem are outlined below: stepwise Gaussian process variable selection (SGPVS) [109] and reference distribution variable selection (RDVS) [63].

In the first method, screening is via stepwise selection of θ_i, α_i in (33.7), analogous to forward stepwise selection in linear regression models. The SGPVS

algorithm identifies those variables that differ from the majority in their impact on the response in the following steps:

- (i) Find the maximized log-likelihood for model (33.1) subject to $\theta_i = \theta$ and $\alpha_i = \alpha$ for all $i = 1, \dots, d$; denote this by l_0 .
- (ii) Set $\mathcal{E} = \{1, \dots, d\}$.
- (iii) For each $j \in \mathcal{E}$, find the maximized log-likelihood subject to $\theta_k = \theta$ and $\alpha_k = \alpha$ for all $k \in \mathcal{E} \setminus \{j\}$; denote the maximized log-likelihood by l_j .
- (iv) Let $j^* = \arg \max_{j \in \mathcal{E}} l_j$. If $l_{j^*} - l_0 > c$, set $\mathcal{E} = \mathcal{E} \setminus \{j^*\}$, $l_0 = l_{j^*}$ and go to step (iii). Otherwise, stop.

In step (iv), the value $c \approx 6$ (the 5% critical value for a χ_2 distribution) has been suggested [109].

The algorithm starts by assuming an isotropic correlation function, and, at each iteration, at most one variable is allocated individual correlation parameters. The initial model contains only four parameters and the largest model considered has $4d + 2$ parameters. However, factor sparsity suggests that the algorithm usually terminates before models of this size are considered. Hence, smaller experiments can be used than are normally employed for GP regression in d variables (e.g., $d = 20$ and $n = 40$ or 50 [109]). This approach essentially adds one variable at a time to the model. Hence, it has, potentially, similar issues as stepwise regression for linear models; in particular, the space of possible GP models is not very thoroughly explored.

The second method, RDVS, is a fully Bayesian approach to the GP variable selection problem. A Bayesian treatment of model (33.1) with correlation function (33.7) requires numerical methods, such as Markov Chain Monte Carlo (MCMC), to obtain an approximate joint posterior distribution of $\theta_1, \dots, \theta_d$ (in RDVS, $\alpha_i = 2$ for all i). Conjugate prior distributions can be used for β_0 and σ^2 to reduce the computational complexity of this approximation. In RDVS, a prior distribution, formed as a mixture of a standard uniform distribution on $[0, 1]$ and a point mass at 1, is assigned to $\rho_i = \exp(-0.25\theta_i)$. This reparameterization of θ_i aids the implementation of MCMC algorithms and provides an intuitive interpretation: small $0 < \rho_i \leq 1$ corresponds to an active variable with the response changing rapidly with respect to the i th variable.

To screen variables using RDVS, the design matrix \mathbf{X}^n for the experiment is augmented by a $(d + 1)$ th column corresponding to an inert variable (having no substantive effect on the response) whose values are set at random. The posterior median for the correlation parameter, θ_{d+1} , of the inert variable is computed for b different randomly generated design matrices, formed by sampling values for the inert variable, to obtain an empirical *reference distribution* for the median θ_{d+1} . The percentiles of this reference distribution can be used to assess the importance of the “real” variables via the size of the corresponding correlation parameters.

For methods that also incorporate variable selection into the Gaussian process mean function, i.e., incorporating the choice of functions in $\mathbf{h}(\mathbf{x})$, see [68, 80].

6 Examples and Comparisons

In this section, six combinations of the design and modeling strategies discussed in this paper are demonstrated and compared for variable screening using two test functions from the literature having $d = 20$ variables and $\mathcal{X} = [-1, 1]^d$. The functions differ in the number of active variables and the strength of influence of these variables on the output.

Example 1. A function used to demonstrate stepwise Gaussian process variable selection [109]:

$$\begin{aligned} Y(\mathbf{x}) = & \frac{5w_{12}}{1+w_1} + 5(w_4 - w_{20})^2 + w_5 + 40w_{19}^3 - 5w_{19} \\ & + 0.05w_2 + 0.08w_3 - 0.03w_6 + 0.03w_7 - 0.09w_9 - 0.01w_{10} \\ & - 0.07w_{11} + 0.25w_{13}^2 - 0.04w_{14} + 0.06w_{15} - 0.01w_{17} - 0.03w_{18}, \end{aligned} \quad (33.21)$$

where $w_i = 0.5x_i$ ($i = 1, \dots, 20$). There are six active variables, $x_1, x_4, x_5, x_{12}, x_{19}$, and x_{20} .

Example 2. A function used to demonstrate the elementary effect method [74]:

$$\begin{aligned} Y(\mathbf{x}) = & \beta_0 + \sum_{j=1}^{20} \beta_j v_j + \sum_{1 \leq j < k}^{20} \beta_{jk} v_j v_k + \sum_{1 \leq j < k < l}^{20} \beta_{jkl} v_j v_k v_l \\ & + \sum_{1 \leq j < k < l < u} \beta_{jklu} v_j v_k v_l v_u, \end{aligned} \quad (33.22)$$

where $v_i = x_i$ for $i \neq 3, 5, 7$ and $v_i = 11(x_i + 1)/(5x_i + 6) - 1$ otherwise; $\beta_j = 20$ ($j = 1, \dots, 10$), $\beta_{jk} = -15$ ($j, k = 1, \dots, 6$), $\beta_{jkl} = -10$ ($j, k, l = 1, \dots, 5$), and $\beta_{jklu} = 5$ ($j, k, l, u = 1, \dots, 4$). The remaining β_j and β_{jk} values are independently generated from a $N(0, 1)$ distribution, and these are used in all the analyses; all other coefficients are set to 0. There are ten active variables, x_1, \dots, x_{10} .

There are some important differences between these two examples: function (33.21) has a proportion of active variables (0.3) in line with factor sparsity, but the influence of many of these active variables on the response is only small; function (33.22) is not effect sparse (with 50% of the variables being active), but the active variables have much stronger influence on the response. This second function also contains many more interactions. Thus, the examples present different screening challenges. For these two deterministic functions, a single data set was generated for each design employed.

The screening strategies employ experiment sizes chosen to allow comparison of the different methods. They fall into three classes:

1. Methods using Gaussian processes and space-filling designs:
 - (a) Stepwise Gaussian process variable selection (SGPVS)
 - (b) Reference distribution variable selection (RDVS)

These two variable selection methods use $n = 16, 41, 84, 200$ runs, where $n = 200$ follows the standard guidelines of $n = 10d$ runs for estimating a Gaussian process model [66]. For each value of n , two designs are found: a maximin Latin hypercube sampling design and a maximum projection space-filling design. These designs were generated from the R packages `SLHD` [5] and `MaxPro` [7] using simulated annealing and quasi-Newton algorithms. For both methods, $\alpha_i = 2$ in (33.7) for all $i = 1, \dots, d$; that is, for SGPVS, stepwise selection is performed for θ_i only.
2. One-factor-at-a-time methods:
 - (a) Elementary effect (EE) method
 - (b) Systematic fractional replicate designs (SFRD)

The elementary effects were calculated using the R package `sensitivity` [87], with each variable taking $f = 4$ levels, $\Delta = 2/3$ and $r = 2, 4, 10$ randomly generated trajectory vectors $\mathbf{x}_1, \dots, \mathbf{x}_r$, giving design sizes of $n = 42, 84, 210$, respectively. An $n = 42$ -run SFRD was used to calculate the sensitivity indices $S(i)$, Eq. (33.13), for $i = 1, \dots, 20$.
3. Linear model methods:
 - (a) Supersaturated design (SSD)
 - (b) Definitive screening design (DSD)

The designs used are an SSD with $n = 16$ runs and a DSD with $n = 41$ runs. For each design, variable selection is performed using the Dantzig selector as implemented in the R package `f1are` [60] with shrinkage parameter s chosen using AICc. Note that these designs are tailored to screening in linear models and hence may not perform well when the output is inadequately described by a linear model.

Tables 33.6 and 33.7 summarize the results for Examples 1 and 2, respectively, and present the sensitivity (ϕ_s), type I error rate (ϕ_I), and false discovery rate (ϕ_{fdr}) for the five methods. The summaries reported in these tables use automatic selection of tuning parameters in each method; see below. More nuanced screening, for example, using graphical methods, may produce different trade-offs between sensitivity, type I error rate, and false discovery rate. In general, with the exception of the EE method, results are better for Example 1, which obeys factor sparsity, than for Example 2, which does not.

For the Gaussian process methods (SGPVS and RDVS), the assessments presented in Tables 33.6 and 33.7 are the better of the results obtained from the maximin Latin hypercube sampling design and the maximum projection space-filling design. In Example 1, for $n = 16, 41, 84$ the maximin LHS design gave the better performance, while for $n = 200$ the maximum projection design was preferred. In Example 2, the maximum projection design was preferred for $n = 16, 200$

Table 33.6 Sensitivity (ϕ_s), type I error rate (ϕ_I), and false discovery rate (ϕ_{fdr}) for Example 1

	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}
Gaussian processes and space-filling designs												
	$n = 16$			$n = 41$			$n = 84$			$n = 200$		
SGPVS	0.33	0.07	0.33	1	0	0	1	0	0	0.67 (1) [†]	0	0
RVDS	0.17	0	0	0.33	0	0	1	0	0	1	0	0
One-factor-at-a-time designs				$n = 42$			$n = 84$			$n = 210$		
EE	—	—	—	0.5	0	0	0.83	0	0	0.83	0	0
SFRD	—	—	—	0.83 ^a (1) ^b	0	0	—	—	—	—	—	—
Nonregular fractional factorial designs and linear models												
	$n = 16$			$n = 41$								
SSD	0.50	0.14	0.40	—	—	—	—	—	—	—	—	—
DSD	—	—	—	0.17 (0.33) ^c	0	0	—	—	—	—	—	—

[†]Using (33.7) with $\alpha_i = 1$

^aUsing a threshold of 5% on the sensitivity indices

^bUsing a threshold of 1% on the sensitivity indices

^cSensitivity for a main effects only model

Table 33.7 Sensitivity (ϕ_s), type I error rate (ϕ_I), and false discovery rate (ϕ_{fdr}) for Example 2

	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}	ϕ_s	ϕ_I	ϕ_{fdr}
Gaussian processes and space-filling designs												
	$n = 16$			$n = 41$			$n = 84$			$n = 200$		
SGPVS	0	0	0	0.40	0	0	1	0	0	1	0	0
RVDS	0	0	0	0.30	0	0	0.80	0	0	1	0	0
One-factor-at-a-time designs				$n = 42$			$n = 84$			$n = 210$		
EE	—	—	—	0.80	0	0	1	0	0	1	0	0
SFRD	—	—	—	0.60 ^a (1) ^b	0	0	—	—	—	—	—	—
Nonregular fractional factorial designs and linear models												
	$n = 16$			$n = 41$								
SSD	0.60	0.90	0.60	—	—	—	—	—	—	—	—	—
DSD	—	—	—	0.10 (0) ^c	0	0	—	—	—	—	—	—

^aUsing a threshold of 5% on the sensitivity indices

^bUsing a threshold of 1% on the sensitivity indices

^cSensitivity for a main effects only model

and the maximin LHS design for $n = 41, 84$. For Example 1, note that, rather counterintuitively, for correlation function (33.7) with $\alpha_i = 2$, SGPVS has lower sensitivity for $n = 200$ than for $n = 41$ or $n = 84$. A working hypothesis to explain this result is that larger designs, with points closer together in the design space, can present challenges in estimating the parameters $\theta_1, \dots, \theta_d$ when the correlation function is very smooth. Setting $\alpha_i = 1$, so that the correlation function is less smooth, resulted in better screening for $n = 200$. The choice of design for Gaussian process screening is an area for further research, as is the application to screening of extensions of the Gaussian process model to high-dimensional inputs [37, 43].

SGPVS and RDVS performed well when used with larger designs ($n = 84, 200$) for both examples. For Example 1, the effectiveness of SGPVS has already been demonstrated [109] for a Latin hypercube design with $n = 50$ runs. In the current study, the method was also effective when $n = 41$. Neither RDVS or SGPVS provided reliable screening when $n = 16$. Both methods found Example 2, with a greater proportion of active variables, more challenging; effective screening was only achieved when $n = 84, 200$.

For RVDS, recall that the empirical distribution of the posterior median of the correlation parameter for the inert variable is used as a reference distribution to assess the size of the correlation parameters for the actual variables. For the assessments in Tables 33.6 and 33.7, a variable was declared active if the posterior median of its correlation parameter exceeded the 90th percentile of the reference distribution. A graphical analysis can provide a more detailed assessment of the method. Figure 33.3 shows the reference distribution and the posterior median of the correlation parameter for each of the 20 variables. Variables declared active have their posterior medians in the right-hand tail of the reference distribution. For both examples, the greater effectiveness of RDVS for larger n is clear. It is interesting to note that choosing a smaller percentile as the threshold to declare a variable active, for example, 80%, would have resulted in considerably higher type I error and false discovery rates for $n = 16, 41$.

For the EE method, performance was assessed by visual inspection of plots of μ_i^* against σ_i ($i = 1, \dots, 20$); see Fig. 33.4. A number of different samples of trajectory vectors $\mathbf{x}_1, \dots, \mathbf{x}_r$ were used and similar results obtained for each. For Example 1, where active variables have a smaller influence on the response, the EE method struggled to identify all the active variables. Variable x_{19} was consistently declared active, having a nonlinear effect. For larger n , variables x_1, x_4, x_{12} , and x_{20} were also identified. For Example 2, with larger active effects, the performance was better. All active variables are identified when $n = 84$ (as also demonstrated by Morris [74]) and when $n = 200$. Performance was also strong for $n = 42$, with only x_3 and x_7 not identified.

For both examples, relatively effective screening was achieved through use of an SFRD to estimate sensitivity indices (33.13). These estimated indices are displayed in Fig. 33.5. Using a threshold of $S(i) > 0.05$ leads to a small number of active variables being missed (one in Example 1 and four in Example 2); the choice of the lower threshold of $S(i) > 0.01$ results in all active variables being identified in both examples and no type I errors being made.

A study of the two true functions provides some understanding of the strong performance of the SFRD. Both functions can be reasonably well approximated (via Taylor series expansions) by linear models involving main effect and interaction terms, with no cancellation of main effects with three-variable interactions or of two variable with four-variable interactions. It is not difficult to modify the two functions to achieve a substantial reduction in the effectiveness of the SFRD. In Example 1, replacement of the term $5(w_4 - w_{20})^2$ by $5w_4^2 - 5w_{20}^2$ produces a function which is highly nonlinear in w_4 and w_{20} . Screening for this function resulted in these two variables being no longer declared active when the SFRD was used. For Example 2,

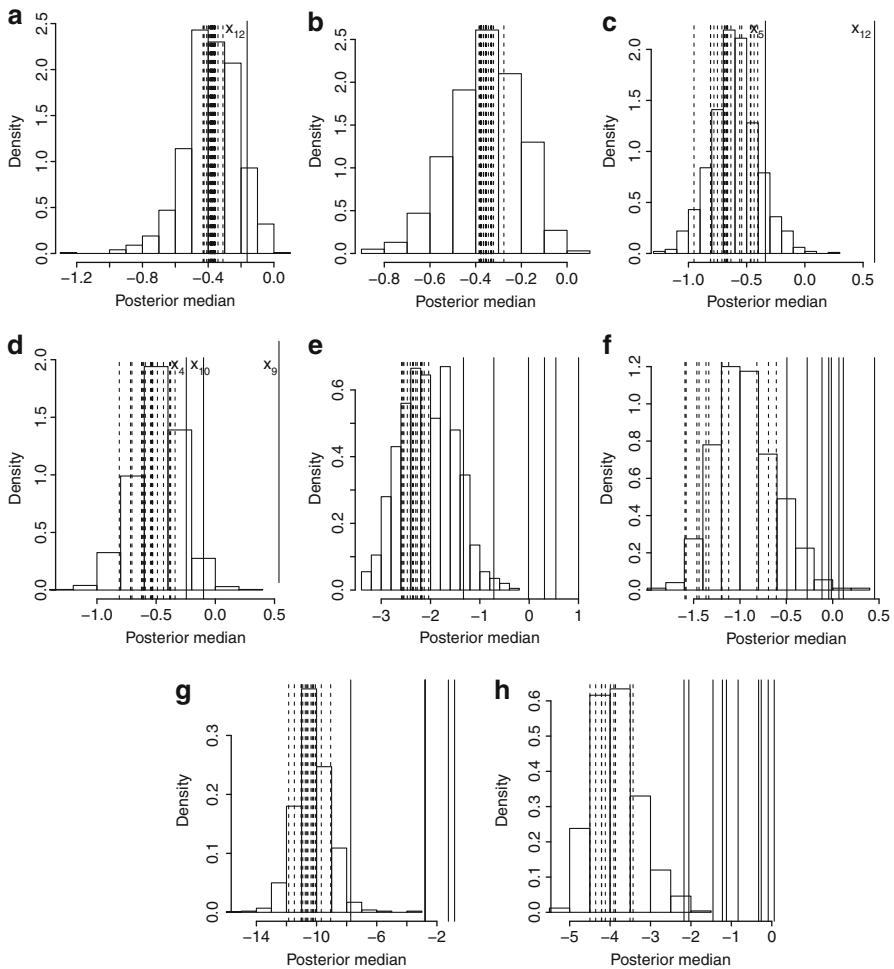


Fig. 33.3 RDVS results: histograms of the empirical distribution of the posterior median of the correlation parameter for 1000 randomly generated inert variables. The posterior medians of the correlation parameters for the 20 variables are marked as *vertical lines* (unbroken, declared active; broken, declared inactive). If there are fewer than five variables declared active, the variable names are also given. **(a)** Example 1: $n = 16$. **(b)** Example 2: $n = 16$. **(c)** Example 1: $n = 41$. **(d)** Example 2: $n = 41$. **(e)** Example 1: $n = 84$. **(f)** Example 2: $n = 84$. **(g)** Example 1: $n = 200$. **(h)** Example 2: $n = 200$

if $\beta_j = 20$ for $j = 1, \dots, 10$, $\beta_{jkl} = 0$ for $j, k, l = 1, \dots, 5$, and $\beta_{jkl} = -5$ for $jkl = 6, \dots, 10$, then use of an SFRD failed to detect variables $x_7 - x_{10}$, even when the threshold $S(i) > 0.01$ was applied, due to cancellation of main effects and three-variable interactions. The performance of the EE method for both these modified functions was the same as that achieved for the original functions.

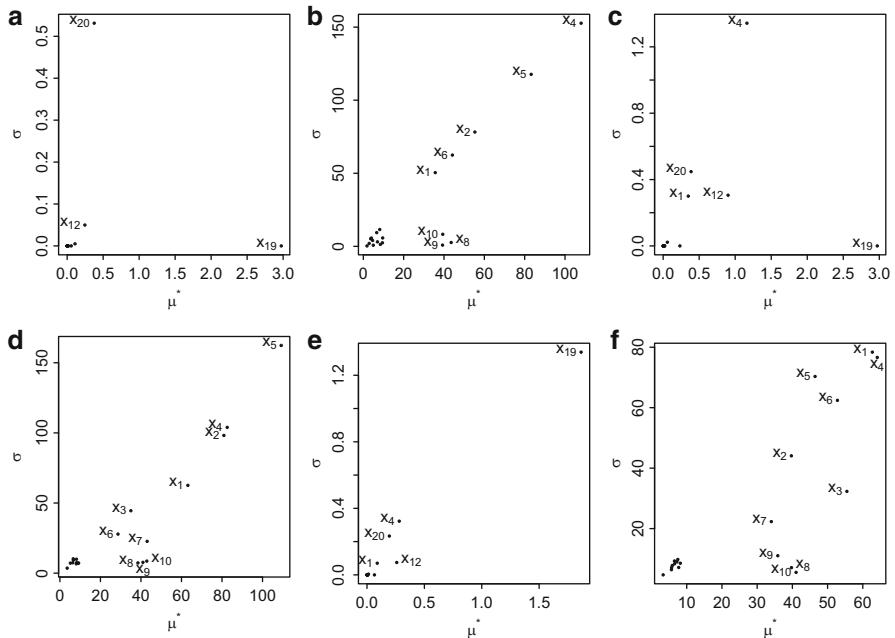


Fig. 33.4 EE results: plots of μ_i^* (33.19) against σ_i (33.18) for Examples 1 and 2. Labels indicate variables declared active by visual inspection. (a) Example 1: $n = 42$. (b) Example 2: $n = 42$. (c) Example 1: $n = 84$. (d) Example 2: $n = 84$. (e) Example 1: $n = 210$. (f) Example 2: $n = 210$

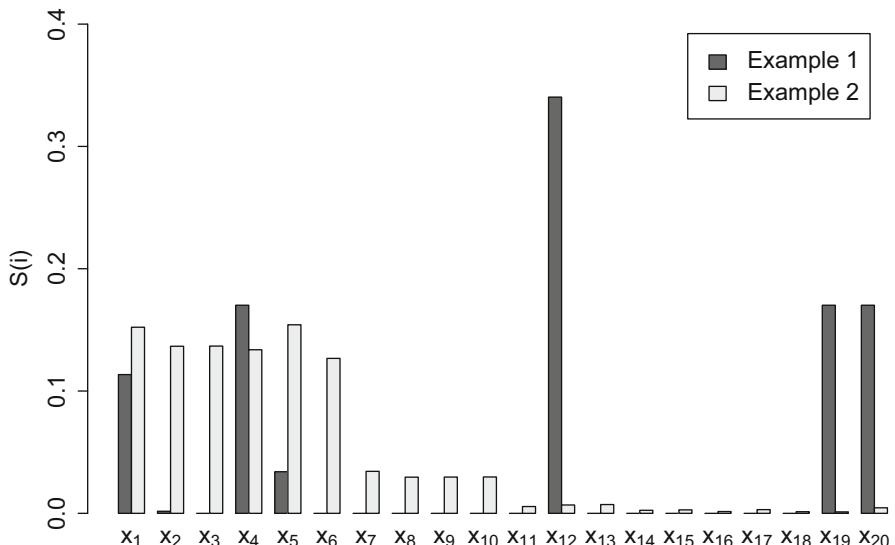


Fig. 33.5 Sensitivity indices (33.13) from the SFRD for Examples 1 and 2 for variables $x_1 - x_{20}$

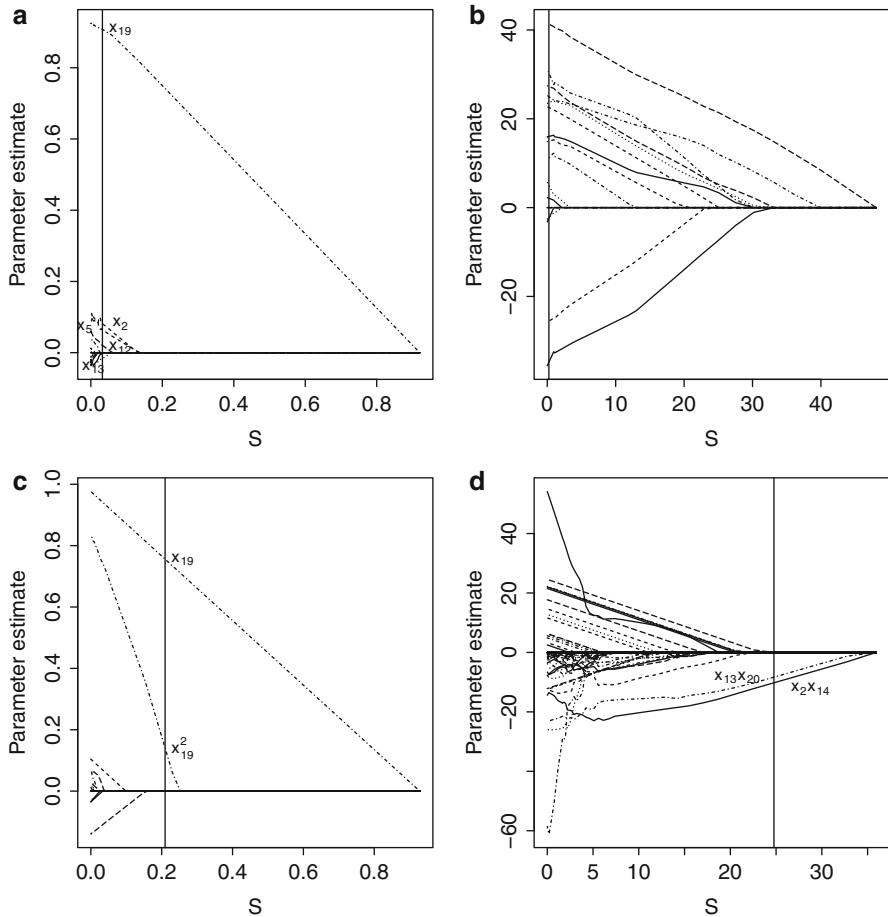


Fig. 33.6 Supersaturated (SSD) and definitive screening design (DSD) results: plots of penalty parameter s against parameter estimates. In each plot, the variables and effects declared active are labeled, with the exception of plot (b) where 15 variables are declared active. (a) Example 1: SSD $n = 16$. (b) Example 2: SSD $n = 16$. (c) Example 1: DSD $n = 41$. (d) Example 2: DSD $n = 41$

The nonregular designs (SSD and DSD) provide an interesting contrast to the other methods, all of which are tailored to, or have been suggested for, variable screening for nonlinear numerical models. For the SSD, only main effects are considered; for the DSD, the surrogate model can include main effects, two-variable interactions, and quadratic terms. Figure 33.6 gives shrinkage plots for each of the SSD and DSD which show the estimated model parameters against the shrinkage parameter s in (33.20). As $s \rightarrow 0$, the shrinkage of the estimated parameters is reduced; at $s = 0$, there would be no shrinkage which is not possible for designs with $n < p$. These plots may be used to choose the active variables, namely, those involved in at least one effect whose corresponding estimated model parameter is

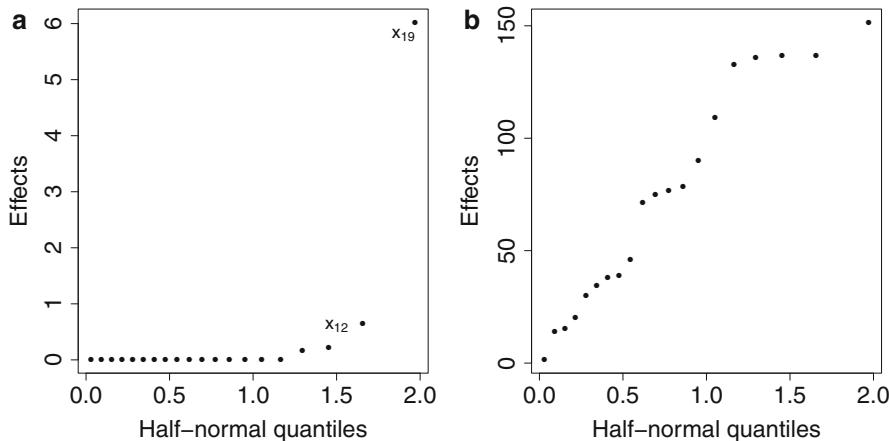


Fig. 33.7 Definitive screening design (DSD) results for main effects only: half-normal plots. (a) Example 1: DSD $n = 41$ (main effects). (b) Example 2: DSD $n = 41$ (main effects)

nonzero for larger values of s . In each plot, the value of s chosen by AICc is marked by a vertical line. Figure 33.6a, c shows shrinkage plots for Example 1 from which the dominant active variables are easily identified, although a number of smaller active variables are missed. For Example 2, Fig. 33.6b, d is much harder to interpret, because they have a number of moderately large estimated parameters. This reflects the larger number of active variables in Example 2. Clearly, the effectiveness of both methods for this second example is limited.

To provide a further comparison between the SSD and DSD, data from the latter design were also analyzed using a main effects only surrogate model (33.2). For these models, the DSD is an orthogonal design and hence standard linear model analyses are appropriate. To summarize the results, Fig. 33.7 gives half-normal plots [30] for each example. Here, the ordered absolute values of the estimated main effects are plotted against theoretical half-normal quantiles. Variables whose estimated main effects stand away from a straight line are declared active, such as x_{12} and x_{19} in Example 1; see Fig. 33.7a. No variables are identified as active for Example 2. These results agree with t-tests on the estimated model parameters.

7 Conclusions

Screening with numerical models is a challenging problem, due to the large number of input variables that are typically under investigation and the complex nonlinear relationships between these variables and the model outputs.

The results from the study in the previous section highlight these challenges and the dangers of attempting screening using experiments that are too small or are predicated on linear model methodology. For this study of $d = 20$ variables and six screening methods, a sample size of at least $n = 40$ was required for effective screening, with more runs needed when factor sparsity did not hold. The

EE method was the most effective and robust method for screening, with the highly resource-efficient SSD and DSD being the least effective here. Of course, these two nonregular designs were not developed for the purpose of screening variables in nonlinear functions; in contrast to the SFRD, neither explicitly incorporates higher-order interactions, and the SSD suffers from partial aliasing between main effects. The two Gaussian process methods, RDVS and SSD, required a greater number of runs to provide sensitive screening.

Methods that use Gaussian process models have the advantage of also providing predictive models for the response. Building such models with the EE or SFRD methods is likely to require additional experimentation. In common with screening via physical experiments, a sequential screening strategy, where possible, is likely to be more effective. Here, a small initial experiment could be run, for example, using the EE method, with more targeted follow-up experimentation and model building focused on a subset of variables using a Gaussian process modeling approach.

Acknowledgements D. C. Woods was supported by a fellowship from the UK Engineering and Physical Sciences Research Council (EP/J018317/1). The authors thank Dr Antony Overstall (University of Glasgow, UK) and Dr Maria Adamou (University of Southampton, UK) for providing code for the RDVS and SGPVS methods, respectively.

Cross-References

- ▶ [Derivative-Based Global Sensitivity Measures](#)
 - ▶ [Maximin Sliced Latin Hypercube Designs with Application to Cross Validating Prediction Error](#)
 - ▶ [Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes](#)
 - ▶ [Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms](#)
-

References

1. Abraham, B., Chipman, H., Vijayan, K.: Some risks in the construction and analysis of supersaturated designs. *Technometrics*, **41**, 135–141 (1999).
2. Andres, T.H., Hajas, W.C.: Using iterated fractional factorial design to screen parameters in sensitivity analysis of a probabilistic risk assessment model. In: Proceedings of Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications, Karlsruhe, pp. 328–340 (1993)
3. Ankenman, B.E., Cheng, R.C.H., Lewis, S.M.: Screening for dispersion effects by sequential bifurcation. *ACM Trans. Model. Comput. Simul.* **25** pages 2:1 - 2:27 (2014)
4. Atkinson, A.C., Donev, A.N., Tobias, R.D.: Optimum Experimental Designs, with SAS, 2nd edn. Oxford University Press, Oxford (2007)
5. Ba, S.: SLHD: Maximin-Distance (Sliced) Latin Hypercube Designs. <http://CRAN.R-project.org/package=SLHD> (2015). R package version 2.1-1
6. Ba, S., Brenneman, W.A., Myers, W.R.: Optimal sliced Latin hypercube designs. *Technometrics* **57**, 479–487 (2015)

7. Ba, S., Joseph, R.: MaxPro: Maximum Projection Designs. [http://CRAN.R-project.org/
package=MaxPro](http://CRAN.R-project.org/package=MaxPro) (2015). R package version 3.1-2
8. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B* **57**, 289–300 (1995)
9. Bettonvil, B.: Factor screening by sequential bifurcation. *Commun. Stat. Simul. Comput.* **24**, 165–185 (1995)
10. Bettonvil, B., Kleijnen, J.P.C.: Searching for important factors in simulation models with many factors: sequential bifurcation. *Eur. J. Oper. Res.* **96**, 180–194 (1996)
11. Booth, K.H.V., Cox, D.R.: Some systematic supersaturated designs. *Technometrics* **4**, 489–495 (1962)
12. Boukouvalas, A., Gosling, J.P., Maruri-Aguilar, H.: An efficient screening method for computer experiments. *Technometrics* **56**, 422–431 (2014)
13. Bowman, V.E., Woods, D.C.: Weighted space-filling designs. *J. Simul.* **7**, 249–263 (2013)
14. Box, G.E.P.: Discussion of the papers of Satterthwaite and Budne. *Technometrics* **1**, 174–180 (1959)
15. Box, G.E.P., Hunter, J.S., Hunter, W.G.: *Statistics for Experimenters: Design, Discovery and Innovation*, 2nd edn. Wiley, Hoboken (2005)
16. Box, G.E.P., Meyer, R.D.: An analysis for unreplicated fractional factorials. *Technometrics* **28**, 11–18 (1986)
17. Box, G.E.P., Wilson, K.B.: On the experimental attainment of optimum conditions. *J. R. Stat. Soc. B* **13**, 1–45 (1951)
18. Brenneman, W.A.: Comment: simulation used to solve tough practical problems. *Technometrics* **56**, 19–20 (2014)
19. Bulutoglu, D.A., Cheng, C.S.: Construction of $E(s^2)$ -optimal supersaturated designs. *Ann. Stat.* **32**, 1162–1178 (2004)
20. Burnham, K.P., Anderson, D.R.: *Model Selection and Multimodel Inference*, 2nd edn. Springer, New York (2002)
21. Butler, N.A.: Supersaturated Latin hypercube designs. *Commun. Stat. Theory Methods* **34**, 417–428 (2005)
22. Campolongo, F., Cariboni, J., Saltelli, A.: An effective screening design for sensitivity analysis of large models. *Environ. Model. Softw.* **22**, 1509–1518 (2007)
23. Campolongo, F., Kleijnen, J.P.C., Andres, T.H.: Screening methods. In: Saltelli, A., Chan, K., Scott, E.M. (eds.) *Sensitivity Analysis*, chap. 4. Wiley, Chichester (2000)
24. Candes, E.O., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n . *Ann. Stat.* **35**, 2313–2351 (2007)
25. Cheng, C.S., Tang, B.: A general theory of minimum aberration and its applications. *Ann. Stat.* **33**, 944–958 (2005)
26. Chipman, H.A., Hamada, M.S., Wu, C.F.J.: A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics* **39**, 372–381 (1997)
27. Claeys-Bruno, M., Dobrijevic, M., Cela, R., Phan-Tan-Luu, R., Sergent, M.: Supersaturated designs for computer experiments: comparison of construction methods and new methods of treatment adopted to the high dimensional problem. *Chemom. Intell. Lab. Syst.* **105**, 137–146 (2011)
28. Cotter, S.C.: A screening design for factorial experiments with interactions. *Biometrika* **66**, 317–320 (1979)
29. Damblin, G., Couplet, M., Iooss, B.: Numerical studies of space-filling designs: optimization of Latin Hypercube Samples and subprojection properties. *J. Simul.* **7**, 276–289 (2013)
30. Daniel, C.: Use of half-normal plots in interpreting factorial two-level experiments. *Technometrics* **1**, 311–341 (1959)
31. Dean, A.M., Lewis, S.M.: Comparison of group screening strategies for factorial experiments. *Comput. Stat. Data Anal.* **39**, 287–297 (2002)
32. Dean, A.M., Lewis, S.M. (eds.): *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*. Springer, New York (2006)

33. Dorfman, R.: The detection of defective members of large populations. *Ann. Math. Stat.* **14**, 436–440 (1943)
34. Draguljić, D., Woods, D.C., Dean, A.M., Lewis, S.M., Vine, A.E.: Screening strategies in the presence of interactions (with discussion). *Technometrics* **56**, 1–28 (2014)
35. DuMouchel, W., Jones, B.A.: A simple Bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics* **36**, 37–47 (1994)
36. Dupuy, D., Corre, B., Claeys-Bruno, M., Sergeant, M.: Comparison of different screening methods. *Case Stud. Bus. Ind. Gov. Stat.* **5**, 115–125 (2014)
37. Durrande, N., Ginsbourger, D., Roustant, O.: Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse* **21**, 481–499 (2012)
38. Fang, K.T., Lin, D.K.J., Winker, P., Zhang, Y.: Uniform design: theory and application. *Technometrics* **42**, 237–248 (2000)
39. Finney, D.J.: The fractional replication of factorial arrangements. *Ann. Eugen.* **12**, 291–301 (1943)
40. Franco, J., Dupuy, D., Roustant, O., Damblin, G., Iooss, B.: DiceDesign: Design of Computer Experiments. <http://CRAN.R-project.org/package=DiceDesign> (2014). R package version 1.6
41. George, E.I., McCulloch, R.E.: Variable selection via Gibbs sampling. *J. Am. Stat. Assoc.* **88**, 881–889 (1993)
42. Gilmour, S.G.: Factor screening via supersaturated designs. In: Dean, A.M., Lewis, S.M. (eds.) *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*, pp. 169–190. Springer, New York (2006)
43. Gramacy, R.B., Lee, H.K.H.: Bayesian treed Gaussian process models with an application to computer modeling. *J. Am. Stat. Assoc.* **103**, 1119–1130 (2008)
44. Hall, M.J.: *Combinatorial Theory*. Blaisdell, Waltham (1967)
45. Hamada, M., Balakrishnan, N.: Analyzing unreplicated factorial experiments: a review with some new proposals. *Statistica Sinica* **8**, 1–41 (1998)
46. Hamada, M., Wu, C.F.J.: Analysis of designed experiments with complex aliasing. *J. Qual. Technol.* **24**, 130–137 (1992)
47. Iman, R.L., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat. Simul. Comput.* **11**, 311–334 (1982)
48. James, G.M., Radchenko, P., Lv, J.: DASSO: connections between the Dantzig selector and lasso. *J. R. Stat. Soc. B* **71**, 127–142 (2009)
49. Jin, R., Chen, W., Sudjianto, A.: An efficient algorithm for constructing optimal design of computer experiments. *J. Stat. Plan. Inference* **134**, 268–287 (2005)
50. Johnson, M., Moore, L.M., Ylvisaker, D.: Minimax and maximin distance design. *J. Stat. Plan. Inference* **26**, 131–148 (1990)
51. Jones, B.A., Lin, D.K.J., Nachtsheim, C.J.: Bayesian D-optimal supersaturated designs. *J. Stat. Plan. Inference* **138**, 86–92 (2008)
52. Jones, B.A., Majumdar, D.: Optimal supersaturated designs. *J. Am. Stat. Assoc.* **109**, 1592–1600 (2014)
53. Jones, B.A., Nachtsheim, C.J.: A class of three-level designs for definitive screening in the presence of second-order effects. *J. Qual. Technol.* **43**, 1–15 (2011)
54. Joseph, R., Gul, E., Ba, S.: Maximum projection designs for computer experiments. *Biometrika* **102**, 371–380 (2015)
55. Kleijnen, J.P.C.: *Design and Analysis of Simulation Experiments*, 2nd edn. Springer, New York (2015)
56. Lenth, R.V.: Quick and easy analysis of unreplicated factorials. *Technometrics* **31**, 469–473 (1989)
57. Lewis, S.M., Dean, A.M.: Detection of interactions in experiments on large numbers of factors (with discussion). *J. R. Stat. Soc. B* **63**, 633–672 (2001)
58. Li, W.: Screening designs for model selection. In: Dean, A.M., Lewis, S.M. (eds.) *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*, pp. 207–234. Springer, New York (2006)

59. Li, W.W., Wu, C.F.J.: Columnwise-pairwise algorithms with applications to the construction of supersaturated designs. *Technometrics* **39**, 171–179 (1997)
60. Li, X., Zhao, T., Wong, L., Yuan, X., Liu, H.: flare: Family of Lasso Regression. <http://CRAN.R-project.org/package=flare> (2014). R package version 1.5.0
61. Lin, D.K.J.: A new class of supersaturated designs. *Technometrics* **35**, 28–31 (1993)
62. Lin, D.K.J.: Generating systematic supersaturated designs. *Technometrics* **37**, 213–225 (1995)
63. Linkletter, C., Bingham, D., Hengartner, N., Hidgon, D., Ye, K.Q.: Variable selection for Gaussian process models in computer experiments. *Technometrics* **48**, 478–490 (2006)
64. Liu, Y., Dean, A.M.: K-circulant supersaturated designs. *Technometrics* **46**, 32–43 (2004)
65. Liu, M., Fang, K.T.: A case study in the application of supersaturated designs to computer experiments. *Acta Mathematica Scientia* **26B**, 595–602 (2006)
66. Loeppky, J.L., Sacks, J., Welch, W.J.: Choosing the sample size of a computer experiment: a practical guide. *Technometrics* **51**, 366–376 (2009)
67. Marley, C.J., Woods, D.C.: A comparison of design and model selection methods for supersaturated experiments. *Comput. Stat. Data Anal.* **54**, 3158–3167 (2010)
68. Marrel, A., Iooss, B., Van Dorpe, F., Volkova, E.: An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput. Stat. Data Anal.* **52**, 4731–4744 (2008)
69. Mauro, C.A., Smith, D.E.: The performance of two-stage group screening in factor screening experiments. *Technometrics* **24**, 325–330 (1982)
70. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245 (1979)
71. Meyer, R.K., Nachtsheim, C.J.: The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics* **37**, 60–69 (1995)
72. Miller, A.: *Subset Selection in Regression*, 2nd edn. Chapman & Hall/CRC, Boca Raton (2002)
73. Moon, H., Dean, A.M., Santner, T.J.: Two-stage sensitivity-based group screening in computer experiments. *Technometrics* **54**, 376–387 (2012)
74. Morris, M.D.: Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**, 161–174 (1991)
75. Morris, M.D.: An overview of group factor screening. In: Dean, A.M., Lewis, S.M. (eds.) *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*, pp. 191–206. Springer, New York (2006)
76. Morris, M.D., Mitchell, T.J.: Exploratory designs for computer experiments. *J. Stat. Plan. Inference* **43**, 381–402 (1995)
77. Nguyen, N.K.: An algorithmic approach to constructing supersaturated designs. *Technometrics* **38**, 69–73 (1996)
78. Nguyen, N.K., Cheng, C.S.: New $E(s^2)$ -optimal supersaturated designs constructed from incomplete block designs. *Technometrics* **50**, 26–31 (2008)
79. Nguyen, N.K., Stylianou, S.: Constructing definitive screening designs using cyclic generators. *J. Stat. Theory Pract.* **7**, 713–724 (2012)
80. Overstall, A.M., Woods, D.C.: Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model (2016). *J. Roy. Statist. Soc. C*, in press (DOI:[10.1111/rssc.12141](https://doi.org/10.1111/rssc.12141)).
81. Owen, A.B.: Orthogonal arrays for computer experiments, integration and visualisation. *Statistica Sinica* **2**, 439–452 (1992)
82. Phoa, F.K.H., Lin, D.K.J.: A systematic approach for the construction of definitive screening designs. *Statistica Sinica* **25**, 853–861 (2015)
83. Phoa, F.K.H., Pan, Y.H., Xu, H.: Analysis of supersaturated designs via the Dantzig selector. *J. Stat. Plan. Inference* **139**, 2362–2372 (2009)
84. Plackett, R.L., Burman, J.P.: The design of optimum multifactorial experiments. *Biometrika* **33**, 305–325 (1946)

85. Pronzato, L., Müller, W.G.: Design of computer experiments: space filling and beyond. *Stat. Comput.* **22**, 681–701 (2012)
86. Pujol, G.: Simplex-based screening designs for estimating meta-models. *Reliab. Eng. Syst. Saf.* **94**, 1156–1160 (2009)
87. Pujol, G., Iooss, B., Janon, A.: Sensitivity: Sensitivity Analysis. [http://CRAN.R-project.org/
package=sensitivity](http://CRAN.R-project.org/package=sensitivity) (2015). R package version 1.11
88. Qian, P.Z.G.: Sliced Latin hypercube designs. *J. Am. Stat. Assoc.* **107**, 393–399 (2012)
89. Qian, P.Z.G., Wu, C.F.J.: Sliced space-filling designs. *Biometrika* **96**, 945–956 (2009)
90. Qian, P.Z.G., Wu, H., Wu, C.F.J.: Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**, 383–396 (2008)
91. Qu, X., Wu, C.F.J.: One-factor-at-a-time designs of resolution V. *J. Stat. Plan. Inference* **131**, 407–416 (2005)
92. Rao, C.R.: Factorial experiments derivable from combinatorial arrangements of arrays. *J. R. Stat. Soc. Suppl.* **9**, 128–139 (1947)
93. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT, Cambridge (2006)
94. Ryan, K.J., Bulutoglu, D.A.: $E(s^2)$ -optimal supersaturated designs with good minimax properties. *J. Stat. Plan. Inference* **137**, 2250–2262 (2007)
95. Saltelli, A., Andres, T.H., Homma, T.: Sensitivity analysis of model output. An investigation of new techniques. *Comput. Stat. Data Anal.* **15**, 211–238 (1993)
96. Saltelli, A., Andres, T.H., Homma, T.: Sensitivity analysis of model output. Performance of the iterated fractional factorial design method. *Comput. Stat. Data Anal.* **20**, 387–407 (1995)
97. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2003)
98. Satterthwaite, F.: Random balance experimentation. *Technometrics* **1**, 111–137 (1959)
99. Scinto, P.R., Wilkinson, R.G., Wang, Z., Rose, A.D.: Comment: need for guidelines on appropriate screening designs for practitioners. *Technometrics* **56**, 23–24 (2014)
100. Scott-Drechsel, D., Su, Z., Hunter, K., Li, M., Shandas, R., Tan, W.: A new flow co-culture system for studying mechanobiology effects of pulse flow waves. *Cytotechnology* **64**, 649–666 (2012)
101. Sun, D.X., Li, W., Ye, K.Q.: An algorithm for sequentially constructing non-isomorphic orthogonal designs and its applications. Technical report SUNYSB-AMS-02-13, Department of Applied Mathematics, SUNY at Stony Brook, New York (2002)
102. Tang, B.: Orthogonal array-based Latin hypercubes. *J. Am. Stat. Assoc.* **88**, 1392–1397 (1993)
103. Tang, B.: Selecting Latin hypercubes using correlation criteria. *Statistica Sinica* **8**, 965–977 (1998)
104. Vine, A.E., Lewis, S.M., Dean, A.M.: Two-stage group screening in the presence of noise factors and unequal probabilities of active effects. *Statistica Sinica* **15**, 871–888 (2005)
105. Voss, D.T., Wang, W.: Analysis of orthogonal saturated designs. In: Dean, A.M., Lewis, S.M. (eds.) Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics, pp. 268–286. Springer, New York (2006)
106. Wan, H.B.E., Ankenman, B.E., Nelson, B.L.: Controlled sequential bifurcation: a new factor screening method for discrete-event simulation. *Oper. Res.* **54**, 743–755 (2006)
107. Wan, H., Ankenman, B.E., Nelson, B.L.: Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening. *INFORMS J. Comput.* **3**, 482–492 (2010)
108. Watson, G.S.: A study of the group screening method. *Technometrics* **3**, 371–388 (1961)
109. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D.: Screening, predicting and computer experiments. *Technometrics* **34**, 15–25 (1992)
110. Wolters, M.A., Bingham, D.R.: Simulated annealing model search for subset selection in screening experiments. *Technometrics* **53**, 225–237 (2011)

111. Wu, C.F.J.: Construction of supersaturated designs through partially aliased interactions. *Biometrika* **80**, 661–669 (1993)
112. Wu, C.F.J., Hamada, M.: Experiments: Planning, Analysis and Optimization, 2nd edn. Wiley, Hoboken (2009)
113. Xiao, L., Lin, D.K.J., Bai, F.: Constructing definitive screening designs using conference matrices. *J. Qual. Technol.* **44**, 2–8 (2012)
114. Xing, D., Wan, H., Yu Zhu, M., Sanchez, S.M., Kaymal, T.: Simulation screening experiments using Lasso-optimal supersaturated design and analysis: a maritime operations application. In: Pasupathy, R., Kim, S.H., Tolk, A., Hill, R., Kuhl, M.E. (eds.) Proceedings of the 2013 Winter Simulation Conference, Washington, DC, pp. 497–508 (2013)
115. Xu, H., Phoa, F.K.H., Wong, W.K.: Recent developments in nonregular fractional factorial designs. *Stat. Surv.* **3**, 18–46 (2009)
116. Yang, H., Butz, K.D., Duffy, D., Niebur, G.L., Nauman, E.A., Main, R.P.: Characterization of cancellous and cortical bone strain in the *in vivo* mouse tibial loading model using microct-based finite element analysis. *Bone* **66**, 131–139 (2014)
117. Ye, K.Q.: Orthogonal column Latin hypercubes and their application in computer experiments. *J. Am. Stat. Assoc.* **93**, 1430–1439 (1998)

Weights and Importance in Composite Indicators: Mind the Gap

34

William Becker, Paolo Paruolo, Michaela Saisana, and Andrea Saltelli

Abstract

Multidimensional measures (often termed composite indicators) are popular tools in the public discourse for assessing the performance of countries on human development, perceived corruption, innovation, competitiveness, or other complex phenomena. These measures combine a set of variables using an aggregation formula, which is often a weighted arithmetic average. The values of the weights are usually meant to reflect the variables importance in the index. This paper uses measures drawn from global sensitivity analysis, specifically the Pearson correlation ratio, to discuss to what extent the importance of each variable coincides with the intentions of the developers. Two nonparametric regression approaches are used to provide alternative estimates of the correlation ratios, which are compared with linear measures. The relative advantages of different estimation procedures are discussed. Three case studies are investigated: the Resource Governance Index, the Good Country Index, and the Financial Secrecy Index.

Keywords

Composite indicators • Nonlinear regression • Sensitivity analysis

W. Becker (✉) • P. Paruolo • M. Saisana

European Commission Joint Research Centre, Ispra (VA), Italy

e-mail: william.becker@jrc.ec.europa.eu; paolo.paruolo@jrc.ec.europa.eu;
michaela.saisana@jrc.ec.europa.eu

A. Saltelli

Centre for the Study of the Sciences and the Humanities (SVT), University of Bergen (UIB),
Bergen, Norway

Institut de Ciencia i Tecnologia Ambientals (ICTA), Universitat Autonoma de Barcelona (UAB),
Barcelona, Spain

e-mail: andrea.saltelli@jrc.ec.europa.eu

Contents

1	Introduction	1188
2	Measures of Importance and Transformations	1191
2.1	Transformations and Weighting	1191
2.2	Importance Measures	1192
2.3	Relation to Sensitivity Analysis	1195
3	Estimating Main Effects	1196
3.1	Polynomial Regression	1196
3.2	Penalized Splines	1197
3.3	Local Polynomial Regression	1198
3.4	Remarks	1199
4	Case Studies	1199
4.1	Resource Governance Index (RGI)	1200
4.2	Financial Secrecy Index (FSI)	1204
4.3	Good Country Index	1209
5	Discussion on Estimation Approaches	1212
6	Conclusions	1213
	References	1215

1 Introduction

Composite indicators are aggregations of observable variables (indicators) that aim to quantify underlying concepts that are not directly observable, such as competitiveness, freedom of press, or climate hazards. Composite indicators are employed for many purposes, including policy monitoring, and they are called in several different ways; for instance, they are also referred to as “performance indices.”

Hardly any newspaper can resist the temptation of commenting on an international country ranking. The popularity of rankings is due to two main reasons. First, their popularity: they provide a summary picture of the multiple facets or dimensions of complex, multidimensional phenomena in a way that facilitates evaluation and comparison. Second, rankings force institutions and governments to question their standards; rankings are drivers of behavior and of change [10].

Hence, it comes as no surprise that there has been a turbulent growth of performance indices over the past two decades. [3] provides a comprehensive inventory of over 400 country-level indices monitoring complex phenomena from economic progress to educational quality. Similarly, a more recent inventory by the United Nations [27] details 101 composite measures of human well-being and progress, covering a broad range of themes from happiness-adjusted income to environmentally adjusted income, from child development to information and communication technology development. Several of those indices have been cited online more than a million times.

The construction of a composite indicator requires several choices; it involves a number of steps in which the developer must make decisions regarding which variables to include in the composite index and how to aggregate them. These steps involve, first, developing a conceptual framework and then selecting and treating

data sets. Next, a multivariate analysis is often performed to identify principal components and correlations. The variables are then normalized, for example, by scaling onto the unit interval or adjusting by mean and variance.

The step discussed in this chapter is the aggregation step, where typically the variables are combined in a weighted average to give the resulting value of the composite indicator. Apart from the decision of which kind of weighted average to use (e.g., arithmetic, geometric), the developer must select values of weights to apply to each variable. The values of these weights can have a large impact on the subsequent rankings of the composite indicator. Understanding the impact of weights on the output composite indicator is hence important.

A possible misconception is that the weight assigned to a variable can be directly interpreted as a measure of importance of the variable to the resulting value of the composite indicator. However, this is true only under very restrictive assumptions; different variances and correlations among variables, for instance, prevent the weights from corresponding to the variables' importance.

A common practice in composite indicator construction is to set the weights of each input variable to be equal, with the intention to make each input variable contribute equally to the value of the composite indicator. This is usually done because choosing weights other than equal would impose questionable assumptions on the relative importance of each variable. However, as shown in this chapter, nominal weights rarely coincide with variable importance, because of dependence between input variables or – when the variables have not been standardized to have the same variance – simply because of variance differences among inputs.

In this chapter, tools from sensitivity analysis are applied to address the question: how dependent is the composite indicator, considered as an *output* variable, with respect to each single measured variable (the *input* variable) which is used to build it? This question concerns the relative importance of input variables in the composite indicator, where the terms *input* and *output* follow the terminology used in uncertainty and sensitivity analysis which is the wider subject of this book.

This chapter highlights how the relative importance of input variables should not be confused with the nominal weights that are used to construct the composite indicator. Other aspects that influence the issue of the importance of input variables are the dependence between variables and the possible nonlinear transformations applied to input variables in the aggregation. The chapter reviews an earlier proposal of some of the present authors (see [13]) to measure the relative importance with the Pearson correlation ratio between the composite indicator and the input variables. Indeed, this measure of importance differs from the nominal weights used in the aggregation.

The chosen measure of importance in this chapter is the Pearson's *correlation ratio*, which is a variance-based measure that accounts for (possibly nonlinear) dependence between input variables and the output. It is exactly equivalent to the *main effect index* or *first-order sensitivity index* (as it is more widely referred to), but is termed the correlation ratio here, first, to avoid confusion of the term "index" with composite indices and, second, to emphasize that it is used as a measure of

nonlinear dependence, rather than uncertainty. This distinction is explained in a little more detail below.

According to the authors' experience, the fact that weights are not measures of importance is rather counterintuitive at first sight. The proof of this fact is that most developers do indeed use weights with the intention to tune the importance of variables in the composite index. As shown in [13], a simple glance at the scatterplots of the composite indicator versus its input variables is sufficient to convince the reader that the relative importance of variables may be quite far from what the weights would imply. Denote the output of a composite indicator as y , which is a function of several input variables x_1, x_2, \dots . Now, consider an example in which inputs x_1 and x_2 are correlated with each other, but both independent from a third input x_3 . In this case, the scatterplots of y against each input variable would show (qualitatively) that – even if the three variables have equal weights and variances – the importance of x_1 and x_2 (in terms of the effect on y) would be larger than that of x_3 due to the dependence between x_1 and x_2 .

This example is important because, in general, the variables in a composite indicator are correlated; they need to be, as one assumes that they concur to describe a unique latent phenomenon. In spite of this rule, the reader is now asked to entertain the following thought experiment, where a composite indicator is built using just two uncorrelated variables (or pillars) with the same variance. The purpose of the experiment is to show the counterintuitiveness of the relation between weights and importance. As a rule, weights in a composite indicator are set by their developers to add up to one. Thus, if one wishes to have, for example, variable x_1 more important than x_2 , one could assign weight $w_1 = 0.9$ to x_1 and $w_2 = 0.1$ to the x_2 . This would imply that x_1 drives y much more than x_2 .

Would this translate into a quantitative statement such as " x_1 accounts for 90% of the variance of y , while x_2 accounts for 10%"? No, because if one decomposes the variance of y according its uncorrelated inputs, the fractional importance of a given variable x_i (i.e., the correlation ratio just mentioned) is $w_i^2 / \sum_i w_i^2$ for the case where all variables have the same variance (this is an exercise in the sensitivity analysis handbook [23]). In the case above, this implies fractional importance measures of x_1 and x_2 equal to 99% and 1%, respectively. Thus, in order for the weights to add up to one, and have target importance 90% and 10%, weights should in fact be equal to 3/4 and 1/4, respectively.

Further justification for using the correlation ratio as the right importance measure to use for variables importance is provided in the remainder of this chapter, where the idea of target fractional variance to be achieved is discussed by judicious assignment of weights for the realistic case where the variables are not independent.

The correlation ratio can be estimated noninvasively (i.e., with only a set of input values and corresponding output values and no explicit modeling of uncertainties as in [19]), using relatively simple, nonlinear regression tools; [13] proposed to use nonparametric, local linear, kernel regression. This method is similar to the one in [7], but does not require to use an independent sample from the marginal distribution of x to estimate the variance of the conditional expectation. In this chapter, the use of penalized splines is also reviewed in this context; see [17]. The relative merits of

different estimation methods for the conditional expectations are discussed. Results obtained for nonlinear regression are compared to with those obtained by linear regression, and the added value of nonlinearity is assessed in concrete cases studies.

The approach of this chapter lends itself to the possibility of selecting nominal weights which imply the intended importance (correlation ratios) of each input variable by searching through the “weight space” (for a short discussion on reverse engineering the weights, see [13]). An additional feature of this approach is that, as a by-product of the analysis, it provides an estimate of the conditional expectation of the composite indicator as a function of a single input variable. The local slope of this conditional expectation answers the related research question: “how much would the composite indicator increase for a marginal increase of the input variable (averaged over variations in other variables)?”. This question may be of interest when discussing alternative policy measures geared to influence different input variables.

The remainder of this chapter is organized as follows: the construction of composite indicators and some measures of variables’ importance are reviewed first, including the correlation ratio. A description is then given of linear and nonlinear approaches to estimation of the correlation ratio. Three case studies are used to illustrate the relative merits of each estimation approach: the Resource Governance Index, which aims to measure transparency and accountability in the oil, gas, and mining sectors; the Financial Secrecy Index, which measures secrecy and scope for abuse in the financial sector for each country; and the Good Country Index, which aims to measure to what extent a given country contributes to some preestablished normative “goods” for humanity. In these case studies, the relative strengths of the correlation ratio compared to linear measures of dependence are also discussed.

2 Measures of Importance and Transformations

2.1 Transformations and Weighting

Consider the case of a composite indicator y (output) calculated aggregating over d input variables x_i . The most common aggregation scheme is the weighted arithmetic average, i.e.:

$$y_j = \sum_{i=1}^d w_i x_{ji}, \quad j = 1, 2, \dots, n \quad (34.1)$$

where x_{ji} is the normalized score of individual j (e.g., country) based on the value X_{ji} of the i th raw variable $i = 1, \dots, d$ and w_i is the nominal weight assigned to the i th variable X_i or x_i . Input variables are usually normalized according to the min-max normalization method (see [3]):

$$x_{ji} = \frac{X_{ji} - X_{\min,i}}{X_{\max,i} - X_{\min,i}}, \quad (34.2)$$

where $X_{\max,i}$ and $X_{\min,i}$ are the upper and lower values, respectively, for the variable X_i ; in this case all scores x_{ji} vary in $[0, 1]$. $X_{\max,i}$ and $X_{\min,i}$ could be replaced by maximal and minimal values for the X_i that do not depend on the sample observations.

A popular alternative to the min-max normalization in (34.2) is given by the standardization:

$$x_{ji} = \frac{X_{ji} - \mathbb{E}(X_i)}{\sqrt{\mathbb{V}(X_i)}}, \quad (34.3)$$

where $\mathbb{E}(X_i)$ and $\mathbb{V}(X_i)$ are the mean and variances of the original variables X_i . Here $\mathbb{E}(X_i)$ and $\mathbb{V}(X_i)$ can be estimated by the sample mean and variance. Note that (34.3) guarantees equal variances, while transformation (34.2) does not.

In fact, transformation (34.2) scales each input variable onto $[0, 1]$, but allows for different means and variances. On the other hand, (34.3) transforms all variables such that they have a mean of zero and a variance of one. Importantly, however, both transformations do not affect the correlation structure of variables, because both are linear transformations of the original variables and correlations are invariant with respect to linear transformations.

The weight, w_i , attached to each variable, x_i , is usually chosen so as to reflect the importance of that variable with respect to the concept being measured. The ratios w_i/w_ℓ can be taken to be the “revealed target relative importance” of inputs i and ℓ because they measure the substitution effect between x_i and x_ℓ , i.e., how much x_ℓ must be increased to offset or balance a unit decrease in x_i (see [8]).

One of the central messages in this chapter is that the target importance does not usually coincide with the “true” importance of each variable, as defined by the correlation ratio, as shown above for the trivial case of two uncorrelated variables. In order to better understand the contribution of each input variable to the output of the composite indicator, measures of linear and nonlinear dependence may be used; these are reviewed in the following section.

2.2 Importance Measures

In the following, the sample index subscript j (usually representing the country or region) is dropped unless it is needed for clarity. Let also the expected value and variance of y be $\mathbb{E}(y) = \mu_y$ and $\mathbb{V}(y) = \sigma_y^2$. Similarly, the means and variances of each x_i are denoted as $\{\mu_i\}_{i=1}^d$ and $\{\sigma_i^2\}_{i=1}^d$, respectively.

For any given composite indicator, one can define measures of importance of each of the input variables with respect to the output values of the composite indicator. One approach for assessing the influence of each input variable x_i on the composite indicator is to measure the dependence of y on x_i , where, from here on, variables are assumed to be normalized – see Eqs. (34.2) and (34.3).

Assume that

$$y_j = f_i(x_{ji}) + \varepsilon_j \quad (34.4)$$

where $f_i(x_{j,i})$ is an appropriate function, possibly nonlinear, that models $E(y_j|x_i)$ – the conditional expectation of y given x_i – and ε_j is an error term. Dependence of y on x_i can be measured in a number of ways. The covariance and correlation between x_i and y , for example, are defined as

$$\text{cov}(y, x_i) := E[(y - \mu_y)(x_i - \mu_i)], \quad R_i := \text{corr}(y, x_i) := \frac{\text{cov}(y, x_i)}{\sigma_y \sigma_i}. \quad (34.5)$$

Remark here that $\text{corr}(y, x_i)$ is a standardized version of the covariance, called the *Pearson product-moment correlation coefficient*, which scales the covariance onto the interval $[-1, 1]$. In the case of a simple linear regression of y on x_i , the square of the correlation coefficient gives the well-known linear *coefficient of determination* R^2 , i.e., $R_i^2 := \text{corr}^2(y, x_i)$, which takes values in $[0, 1]$.

The coefficient of determination is used to measure the goodness of fit of an ordinary linear regression: as such R_i^2 is a measure of linear dependence. Because of (34.5), the covariance $\text{cov}(y, x_i)$, the correlation $\text{corr}(y, x_i)$, and the coefficient of determination R_i^2 are all related measures of linear dependence. In sample, the coefficient of determination R_i^2 can be computed as

$$SS_{\text{reg},i} / SS_{\text{tot}}, \quad (34.6)$$

where $SS_{\text{reg},i} = \sum_{j=1}^n (\hat{f}_i(x_{i,j}) - \bar{y})^2$ is the sum of squares explained by the linear regression, $\bar{y} := n^{-1} \sum_{j=1}^n y_j$ is the sample average, $\hat{f}_i(x_{i,j}) = \hat{\beta}_0 + \hat{\beta}_1 x_{i,j}$ is the linear fit for observation y_j , and $SS_{\text{tot}} = \sum_j (y_j - \bar{y})^2$ is the total sum of squares. R_i^2 can hence be seen as the ratio of the sum of squares explained by the linear regression of y on x_i and the total sum of squares of y .

If the relationship between y on x_i is nonlinear, R_i^2 may underestimate the degree of dependence. The proposed measure in this chapter is the *correlation ratio*, S_i , $i = 1, 2, \dots, d$ also widely known as the *first-order sensitivity index*, or *main effect index* (see ▶ Chap. 35, “[Variance-based Sensitivity Analysis: Theory and Estimation Algorithms](#)”). This measure is meant to measure the (possibly nonlinear) influence of each variable on the composite indicator. The correlation ratio can be interpreted as the expected variance reduction of the composite indicator, if a given variable were fixed. The correlation ratio is traditionally denoted as η_i^2 and was introduced by [14]; to follow the wider literature on sensitivity analysis, it is referred to here as S_i . Both the correlation ratio and the first-order sensitivity index are defined as

$$S_i \equiv \eta_i^2 := \frac{V_{x_i} (E_{x \sim i} (y | x_i))}{V(y)}, \quad (34.7)$$

where $x_{\sim i}$ is defined as the vector containing all the variables (x_1, \dots, x_d) except variable x_i and $E_{x \sim i} (y | x_i)$ denotes the conditional expectation of y given x_i , e.g., with x_i fixed at one value in its interval of variation. The notation employed here stresses that the expectation in $E_{x \sim i} (y | x_i)$ is computed with respect to the

distribution of $\mathbf{x}_{\sim i}$, i.e., with respect to all other input variables; while the subscript x_i used as the outer variance signifies that the variance is taken over all possible values of x_i . In the following the variance and expected value subscripts are dropped to economize on notation.

The conditional expectation $E(y | x_i)$ is known as the *main effect* of x_i on y , and it describes the expected value of y (the composite indicator) averaged over all input variables except x_i , keeping x_i fixed. As such, $E(y | x_i)$ is a function of x_i and is here denoted as $f_i(x_i)$. This function is not typically known; however, it can be estimated by performing a (nonlinear) regression of y on x_i . Various approaches for this problem are discussed in the following section; any of them delivers a fitted value $m_j := \hat{f}_i(x_{ij})$ corresponding to the prediction of y_j . The correlation ratio S_i can then be computed in sample as

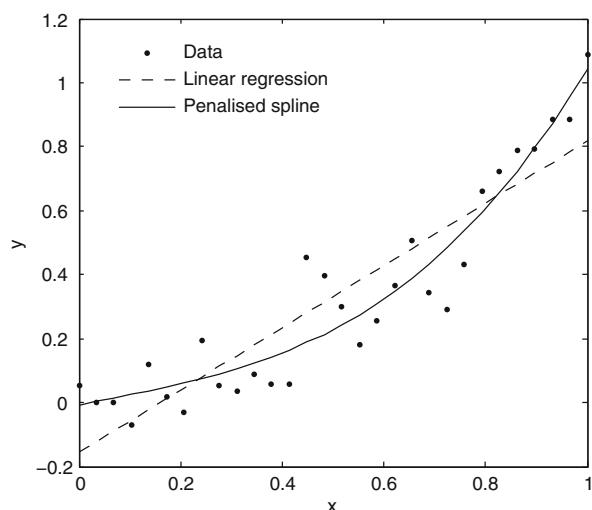
$$\sum_{j=1}^n (m_j - \bar{m})^2 / \sum_{j=1}^n (y_j - \bar{y})^2 \quad (34.8)$$

where $\bar{m} := n^{-1} \sum_{j=1}^n m_j$, $m_j := \hat{m}(x_{ij})$, and $\hat{m}(\cdot)$ is the estimate of $m(\cdot) := f_i(\cdot)$. Equation (34.8) mimics (34.6).

The correlation ratio S_i is closely connected with the measure R_i^2 of linear dependence discussed previously. More specifically, R_i^2 equals S_i when $f_i(x_i)$ is linear; note that the main effect is dependent on the functional form of the composite indicator *and* the dependence structure between variables. This shows that, in the linear case, S_i is also related to the covariance and correlation coefficients by the associations discussed earlier.

In order to illustrate why linear measures of importance would not be sufficient for nonlinear dependence, Fig. 34.1 plots data generated using the relationship

Fig. 34.1 Parabolic conditional mean with linear and penalized spline fits. Here $\text{corr}(x_1, y) = 0.91$, $R_i^2 = 0.82$, and $S_1 = 0.91$ (using the penalized spline estimate)



$y = 0.2 + x_1^2 + 0.1u$, with $u \sim \mathcal{N}(0, 1)$ independent of x_1 , which is generated drawing from a uniform distribution on $[0, 1]$. The conditional mean $E(y | x_1) = x_1^2$ is here nonlinear (a branch of a parabola). The linear regression through the data in Fig. 34.1 gives $R^2 = 0.82$, while the correlation ratio S_i is 0.91, as estimated by a nonlinear (penalized spline) regression.

In this case the strong dependency between y and x_1 would be underestimated by the linear measure R_i^2 , but it is captured by the correlation ratio S_i : about 91% of the variance in y is explained by the correlation ratio (using a nonlinear conditional mean specification), and about 9% of it would be missed by a linear regression model.

2.3 Relation to Sensitivity Analysis

The correlation ratio is widely used in the discipline of variance-based sensitivity analysis of computer models, typically under the name of “first-order sensitivity index,” “main effect index,” or “Sobol index.” It is important to recognize that sensitivity analysis, as it is commonly performed in physical sciences and engineering, is concerned with analyzing the effects of *uncertainty* in model input variables on the model output. Each input variable therefore has an associated probability distribution $p(x_i)$ which attempts to characterize the uncertainty in the value of x_i , either as a result of lack of knowledge or inherent variability. The “sensitivity analysis” described in this chapter, on the other hand, does not deal with uncertainty in the inputs of a composite indicator (although this is also an important area of research – see, e.g., [20]). Instead, it quantifies the contribution of each input to the composite indicator, as a result of the weight and aggregation of the indicator and the distribution and interdependence of each input variable, by measuring the nonlinear dependence of the composite indicator on each of its input variables. The distributions $p(x_i)$ of each input variable do not therefore characterize uncertainty in x_i – rather, they simply represent the distribution of entities (e.g., countries, institutions) in the variable x_i . Indeed, these distributions are often discrete: they represent a finite number of entities, such as countries or regions. It is therefore to emphasize that the application here is distinct from uncertainty analysis that the term “The correlation ratio is therefore used (as opposed to “first-order/main effect index,” which is more widely used elsewhere) to emphasize that the application here is distinct from uncertainty analysis.”

Usually in sensitivity analysis, as it is applied in physical models in science and engineering, one finds $\sum_{i=1}^d S_i \leq 1$ when studying a generic nonlinear output function with independent input variables; see [11, 21]. In this setting, the variance of the model output can be decomposed into portions that can be uniquely attributed to each variable and subset of variables. In the case of composite indicators, the input variables are almost certainly (positively) correlated, and it is usually the case that $\sum_{i=1}^d S_i > 1$. Furthermore, the correlation ratios here are not interpreted in terms of a variance decomposition – they are simply used as nonlinear measures of

correlation. However, even for the case of non-independent inputs, the S_i measure preserves its meaning of expected fractional reduction of the output variance that would be achieved if a variable could be fixed [22].

In order to estimate $f_i(x_i)$, a wide number of approaches are available. Two nonlinear regression approaches are discussed in the next section; both offer a flexible framework for estimating nonlinear main effects.

3 Estimating Main Effects

This section reviews the estimation of the conditional expectation $E(y | x_i)$ using polynomial (linear) regression, penalized splines, and nonparametric regression. There are many other ways of modeling $E(y | x_i)$, such as Gaussian processes [16]; see also [25]. The methods reviewed here are chosen for comparison purposes and because they have some attractive properties. Specifically, the linear regression model is considered here as the simplest model of the conditional expectation; kernel nonparametric regression is considered as the opposite polar leading example of nonlinear models for the conditional expectations; this method was employed in [13] in the context of composite indicators. As an additional method to estimate the nonlinear regression function, penalized splines are also investigated here, which share similar properties in smoothing nonlinear data with kernel nonparametric regression; moreover, they are computationally fast (and so can cope with large data sets). Note that in this section the variable subscript i is dropped, since the techniques here all refer to regression of y on a single variable x .

3.1 Polynomial Regression

Consider a set of bivariate data consisting of n pairs of observations $\{x_j, y_j\}$, $j = 1, 2, \dots, n$, an integer p (the degree of the polynomial), and the model

$$y_j = \beta_0 + \beta_1 x_j + \dots + \beta_p x_j^p + \epsilon_j \quad j = 1, 2, \dots, n \quad (34.9)$$

in which the β_h , $h = 0, \dots, p$ are coefficients and ϵ_j is an error term. This can be rewritten in matrix form using the $p + 1$ column vectors $\mathbf{x}_j = (1, x_j, x_j^2, \dots, x_j^p)^\top$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^\top$ and the n -dimensional column vectors $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \dots, \epsilon_n)^\top$. Next, one can define the $n \times (p + 1)$ design matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$ so as to rewrite (34.9) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (34.10)$$

Minimizing the sum of squared residuals with respect to $\boldsymbol{\beta}$ gives the well-known expression

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (34.11)$$

In case $p = 1$, (34.11) gives the linear regression coefficients $\hat{\beta}_0, \hat{\beta}_1$, which can be used to calculate R^2 . Note that in cases where $p > 1$, (34.10) is nonlinear with respect to x , but is still linear with respect to its parameters $\boldsymbol{\beta}$. Therefore, quadratic, cubic, and higher-order regressions (and indeed other basis functions that can be used with (34.10)) are often referred to as linear regression, depending on the context.

3.2 Penalized Splines

One approach to smoothing nonlinear data is the *penalized spline*. Penalized splines are also referred to as *semiparametric regression*, given that they are an extension of linear parametric regression (linear in the parameters), but also have the capabilities of nonparametric regression (i.e., local polynomial regression), such as the flexibility to accommodate nonlinear trends in the data [17].

The basis function which is the heart of the spline model is given by $(x - \kappa_h)_+^p$, where the “+” subscript denotes the positive part; in other words, for any number u , $u_+ = u$ if u is positive and equals zero otherwise. The κ_h parameter is called the “knot” of the basis function; it is a value of x at which the spline is “split.” Splines are constructed by using a number $h = 0, 1, 2, \dots, H$ of spline basis functions with different knots κ_h . Specifically, the polynomial $\beta_0 + \beta_1 x_j + \dots + \beta_p x_j^p$ in (34.9) is extended to give

$$\beta_0 + \beta_1 x + \dots + \beta_p x^p + \sum_{k=1}^K \beta_{pk} (x - \kappa_k)_+^p \quad (34.12)$$

where again β are coefficients to be estimated.

Since the model is linear with respect to its coefficients, it is possible to write it in the same form as (34.9) replacing $\mathbf{x}_j = (1, x_j, x_j^2, \dots, x_j^p)^\top$ with $\mathbf{x}_j = (1, x_j, x_j^2, \dots, x_j^p, (x_j - \kappa_1)_+^p, \dots, (x_j - \kappa_K)_+^p)^\top$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^\top$ with $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p, \beta_{p1}, \dots, \beta_{pK})^\top$. When $p = 1$, this is known as a *linear spline*. In practice, quadratic or cubic splines are normally used (corresponding to $p = 2$ or 3, respectively), because they result in a smooth fit to the data. In the applications here, a value of $p = 3$ is used.

The coefficients may be directly estimated by using (34.11); however, this tends to result in a fit which is too “rough” – in other words, it fluctuates too much and is drawn too much to individual data points rather than following the smoother underlying trend. This is known as overfitting. To overcome this problem, the estimator (34.11) can be constrained to limit the influence of the spline basis terms, resulting in a smoother fit.

This results in what is known as *penalized splines*. To do this, $\boldsymbol{\beta}$ is chosen to minimize the sum of squared residuals, under the constraint that $\sum_{h=1}^H \beta_{ph}^2 = \boldsymbol{\beta}^\top \mathbf{D} \boldsymbol{\beta} < C$, where C is a positive constant and $\mathbf{D} = \text{diag}(\mathbf{0}_{(p+1) \times (p+1)}, \mathbf{I}_H)$. The

constraint reduces the influence of the last additional H terms $\sum_{h=1}^H \beta_{ph}(x - \kappa_h)_+^p$ in (34.12) to avoid overfitting the data and results in the *penalized spline* model. The constraining of regression parameters shares similarities with other approaches such as *ridge regression* (see, e.g., [9]) and *LASSO* [26].

The constrained minimization problem is solved by

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{D})^{-1} \mathbf{X}^\top \mathbf{y} \quad (34.13)$$

where λ plays the role of a Lagrange multiplier for the penalization. High λ values result in a very smooth fit, whereas low values give a rough fit which is closer to interpolating the data points. The smoothing parameter λ needs to be optimized, and this is done using a cross-validation approach, in which a search algorithm is used to find the value of λ that minimizes the average squared error resulting from removing a point from the set of training data and predicting at that point. The fact that penalized splines are only a short step from linear regression means that they can exploit well-known properties to give fast order- n algorithms for the calculation of cross-validation measures; therefore, the fitting of a penalized spline can be very fast. More details on this can be found in [17] Section 5.3.

The spline model requires the specification of the knots κ_h ; here they are placed at equally spaced quantiles of the unique x_i values, and the number of knots is chosen from a set of candidate H values, as the number of knots which results in the best fit according to a measure of cross-validation. This procedure is described in more detail in [17] Section 5.5.

In the applications in this chapter, the spline models are deliberately constrained to be quite simple fits to the data. To do this, the spline is allowed to have $K = 0, 1, 2, 3, 4$ knots, where the optimal number is chosen as that which minimizes the cross-validation measure. In the case of $H = 0$, the spline is reduced to the polynomial model, which in this work is cubic, since p is set to 3.

Finally, note that in the following work, “penalized splines” will sometimes be referred to simply as “splines.” All splines used in this investigation are penalized cubic splines.

3.3 Local Polynomial Regression

Another approach to nonlinear regression is *local polynomial regression*, also referred to as *kernel smoothing* – see, e.g., [4]. Kernel smoothing works by averaging a number of weighted polynomial regressions, centered at different values of x . The regression $\hat{f}(x)$ is chosen here to be *local linear*, and the estimation is performed minimizing the weighted sum of squares, where weights are proportional to a kernel function with bandwidth b . The kernel function gives the strongest weight to squared residuals corresponding to points close to x_j for observation j , which reflects the belief that the closer two points are to each other in x , the more likely they are to have similar values in y .

A commonly used kernel function is the Gaussian density function with standard deviation b . Local-linear regression is used here (as opposed to, e.g., local mean regression) since it is generally regarded as a good choice, due to its good properties near the edges of the data cloud. The smoothing parameter b can be optimized by cross-validation; this is the method that is adopted in the applications in this chapter. The local polynomial approach is used by [7] for estimation of first-order sensitivity indices of uncertainty for models with correlated inputs.

3.4 Remarks

The choice of whether to use penalized splines, kernel regression, linear regression, or another nonlinear regression approach entirely is a working decision of the analyst. By construction, linear regression (i.e., linear with respect to x) only reveals linear dependencies between variables and so should be used with this caveat in mind. Higher-order polynomial regression or other basis functions within a linear regression might improve the fit, but also risk overfitting. The penalized splines and local-linear regression allow for nonlinearities when they are present in the data, but can also model near-linear data when required. In most applications, these two latter approaches are expected to give comparable results. However, in the presence of highly skewed and/or heteroskedastic data, the two fits may be quite different.

While the fits are likely to be similar, splines may have a slight advantage over local-linear regression in some other respects. First, they are less computationally demanding than kernel regression, due to the fast computation of cross-validation measures discussed previously. While the difference is small enough to be negligible for running a few regressions on small data sets, the difference may be sizable if one wants to attempt an optimization of weights or if a very large number of regressions need to be run. Large data sets of the order of thousands or millions of points are common in environmental science. A second advantage of splines is that the computation of their derivatives is just as easy.

In any case, a prudent strategy would be to run both analyses with splines and kernel regression and compare the results – this is the approach taken in the following examples where both methods are applied and compared.

4 Case Studies

This section delves into the conceptual and statistical properties of three composite indicators selected for their interesting structure as well as for their popularity. These examples allow a practical demonstration of the methods described in previous sections, as well as providing an interesting analysis of three well-known composite indicators. The case studies are chosen because:

- They are transparent – namely, the values of each input variable are publicly available, and the index can be reproduced given the methodological notes provided by the developers.

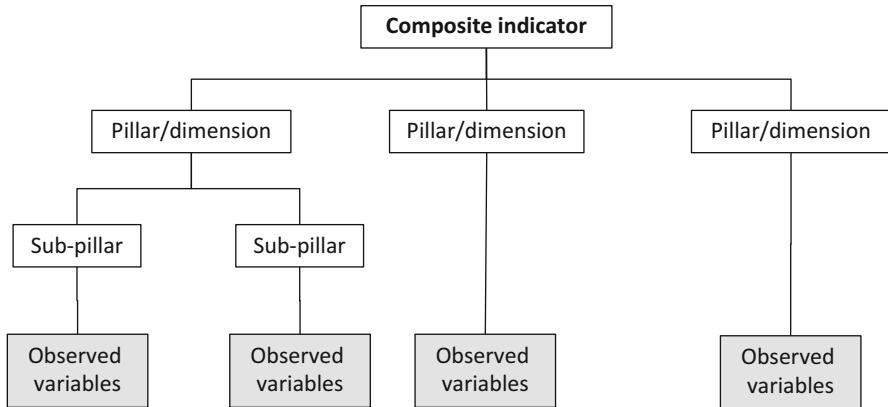


Fig. 34.2 A (fictional) example of the hierarchy of a composite indicator

- They use three different aggregation formulas, which allows the effect of the different measures of importance described here, and estimation methods, to be assessed.
- They deal with issues of governance and accountability and hence offer interesting narratives on the misconception of what matters when developing an index.

Often, composite indicators are built on a number of hierarchical “levels.” Rather than having d measured variables x_1, x_2, \dots, x_d as direct inputs to the composite indicator, variables (indicators) are usually put together into groups, known as “pillars” or “dimensions,” which share similar conceptual characteristics (see Fig. 34.2). Variables in each pillar are aggregated in a weighted sum, such that each pillar is itself a composite indicator characterizing one aspect of the greater theme. The composite values of each pillar are then used as the inputs of the composite indicator itself. Indeed, pillars may also consist of sub-pillars if the developers deem it appropriate. For simplicity however, and to be consistent with the sensitivity analysis literature and previous sections, the direct inputs to the composite indicators here are referred to as “variables” in the analytical parts of the following section, even though they might represent pillars.

4.1 Resource Governance Index (RGI)

4.1.1 Aim

The Resource Governance Index (RGI) is developed by the Revenue Watch Institute in order to measure the transparency and accountability in the oil, gas, and mining sectors in 58 countries [15]. These nations produce 85% of the world’s petroleum, 90% of diamonds, and 80% of copper, generating trillions of dollars in annual profits. The future of these countries depends on how well they manage their natural resources.

4.1.2 Sources

To evaluate the quality of governance in the extractive sector, the Resource Governance Index employs a 173-item questionnaire that is based on the standards put forward by the International Monetary Fund's 2007 *Guide on Resource Revenue Transparency* and the Extractive Industries Transparency Initiative, among others. The answers to the 173 questions are grouped into 45 indicators that are then mapped into three (of the four) RGI dimensions: Institutional and Legal Setting, Reporting Practices, and Safeguards and Quality Controls. The fourth dimension, Enabling Environment, consists of five additional indicators that describe a country's broader governance environment; it uses data compiled from over 30 external sources by the Economist Intelligence Unit, International Budget Partnership, Transparency International, and Worldwide Governance Indicators. The Index is therefore a hybrid, with three dimensions based on the questionnaire specifically assessing the extractive sector, and the fourth rating the country's overall governance.

4.1.3 Main Dimensions

The RGI's four dimensions cover the following topics:

1. Institutional and Legal Setting (20% weight): ten indicators that assess whether the laws, regulations, and institutional practices enable comprehensive disclosures, open and fair competition, and accountability.
2. Reporting Practices (40% weight): 20 indicators that evaluate the actual disclosure of information and reporting practices by government agencies.
3. Safeguards and Quality Controls (20% weight): 15 indicators that measure the checks and oversight mechanisms that guard against conflicts of interest and undue discretion, such as audits.
4. Enabling Environment (20% weight): five indicators of the broader governance environment generated using over 30 external measures of accountability, government effectiveness, rule of law, corruption, and democracy.

The RGI score is a weighted arithmetic average of the four dimensions, i.e., of the form of (34.1), where the dimensions here are treated as the x_1, x_2, x_3, x_4 input variables.

Because actual disclosure constitutes the core of transparency, developers assigned a greater weight to the Reporting Practices dimension. This choice also reflects a belief that without reporting information, rules and safeguards ring hollow. Therefore, Reporting Practices are assigned a weight equal to 40% of the final country score, and the other three dimensions (Institutional and Legal Setting, Safeguards and Quality Controls, and Enabling Environment) account for 20% each.

On the inclusion of the Enabling Environment dimension, there have been arguments for and against. Against its inclusion, the arguments are:

1. The Enabling Environment dimension dilutes the focus of the index on the oil, gas, and mining sector by incorporating measures of overall governance.

2. The Enabling Environment dimension can have an undue effect on the index scores, driving scores up or down, inflating or depressing performances beyond what countries actually show in their extractive sector.

In favor of its inclusion, the arguments are:

1. External governance indicators reflect the influence of the broader country environment on the quality of natural resource governance. When considering the quality of transparency and accountability in a certain area, it does matter whether a country also has an authoritarian regime, a high risk of corruption, or respect for basic freedoms.
2. As an expert-based index, the accuracy and consistency of its findings suffer from the bias introduced by researchers and by peer and Research Watch Institute reviewers. Including this dimension as an external measure reduces this margin of error.

Given these last two arguments, the developers chose to include the Enabling Environment dimension and to allocate a 20% weight to this dimension. As part of the Index website, the developers provide a tool that allows users to change the weights for the different dimensions, creating different composite scores that reflect their own sense of prioritization. This is a direct example of how weights are often (erroneously) interpreted as measures of importance.

4.1.4 Results

Before examining the relationships between each variable (dimension/pillar) and the output of the composite indicator, a basic analysis of the structure of the data was performed. There are no outliers (absolute skewness is less than 0.63) in the four dimensions' distributions that could bias the subsequent interpretations of the correlation structure. The four dimensions of the Resource Governance Index have moderate to high correlations that range from 0.41 (between Institutional and Legal Setting and Enabling Environment) to 0.82 (between Reporting Practices and Safeguards and Quality Controls) and an overall good average bivariate correlation of 0.65. Principal component analysis suggests that there is indeed a single latent phenomenon underlying the four index dimensions. This first principal component captures 74% of the variation in the four dimensions.

Moving now to the importance measures, Table 34.1 shows the estimates of the correlation ratios obtained both with penalized splines and local-linear (LL) regression, as well linear correlations, for the four input variables of the index. The correlation ratios S_i (penalized splines and LL approach) and the Pearson correlation coefficients R_i^2 confirm that the Reporting Practices component has indeed the highest impact on the index. This was the intention of the RGI developers on the grounds that actual disclosure constitutes the core of transparency. This choice also reflects a belief that without reporting information, rules and safeguards are inconsequential. In fact, the correlation between Reporting Practices and Safeguards and Quality Control is very high (0.82, the highest among the

Table 34.1 Measures of dependence of the Resource Governance Index on its input variables: $R_i = \text{corr}(x_i, y)$ (correlation), $S_{i,\text{spl}}$ (correlation ratio, penalized spline), $S_{i,\text{LL}}$ (correlation ratio, local linear)

Resource Governance Index ($n = 58$)	x_i	w_i	R_i	R_i^2	$S_{i,\text{spl}}$	$S_{i,\text{LL}}$
Institutional and legal setting	x_1	0.2	0.79	0.63	0.65	0.67
Reporting practices	x_2	0.4	0.95	0.90	0.90	0.94
Safeguards and quality controls	x_3	0.2	0.91	0.82	0.83	0.83
Enabling Environment	x_4	0.2	0.77	0.59	0.65	0.70

components). If one could fix the Reporting Practices variable, the variance of the RGI scores across the 58 countries would on average be reduced by 94% (local-linear estimate). It is worth noting that despite the equal weights assigned to the other three components, their impact on the RGI variation differs: by fixing any of the other variables, the variance reduction would be 83% for Safeguards and Quality Control, 70% for Enabling Environment, and 67% for Institutional and Legal Setting, using the estimates of the local-linear regression.

As per the developers intention to have Reporting Practices twice as important as any of the other three dimensions, i.e. Institutional and Legal Setting, Safeguards and Quality Controls, and Enabling Environment, it is easy to see that this was not achieved. The strong correlation of Reporting Practices with Safeguards and Quality Controls results in these two pillars being almost equally important (0.90 and 0.82), while the importance of Reporting Practices relative to the remaining two pillars is of about 3/2 rather than 2.

Aside from the implications of this analysis from the point of view of the RGI, it is interesting to look at the differences between the measures of importance in Table 34.1. The linear R_i^2 measure consistently gives the lowest estimates of importance, the LL estimate of correlation ratio is the highest, and the penalized spline estimate of correlation ratio is somewhere in between. To see in a little more detail what is happening, Fig. 34.3 shows the scatterplots of the four input variables and the simple linear, penalized spline and local-linear estimations of main effects. For the Institutional and Legal Setting variable, the two nonlinear fits are significantly different to the linear fit, but fairly similar to each other. However, the structure of the data is such that the gradient of the linear fit (and the overall trend of the nonlinear fits) is not very steep, resulting in relatively low importance estimates compared to other variables. The Reporting Practices variable shows quite similar fits between all three methods, although the local-linear curve has a slightly higher variance. Safeguards and Quality gives a near-linear fit for both nonlinear regression approaches, showing a strong agreement in measures of importance. The Enabling Environment variable is the most interesting here from a regression point of view: the penalized spline fits a roughly cubic model, whereas the local-linear curve fluctuates more strongly with the data. Whether the more parsimonious penalized spline curve or the more variable local-linear curve is a better estimate of $E(y|x_i)$ is not intuitively clear from visual inspection.

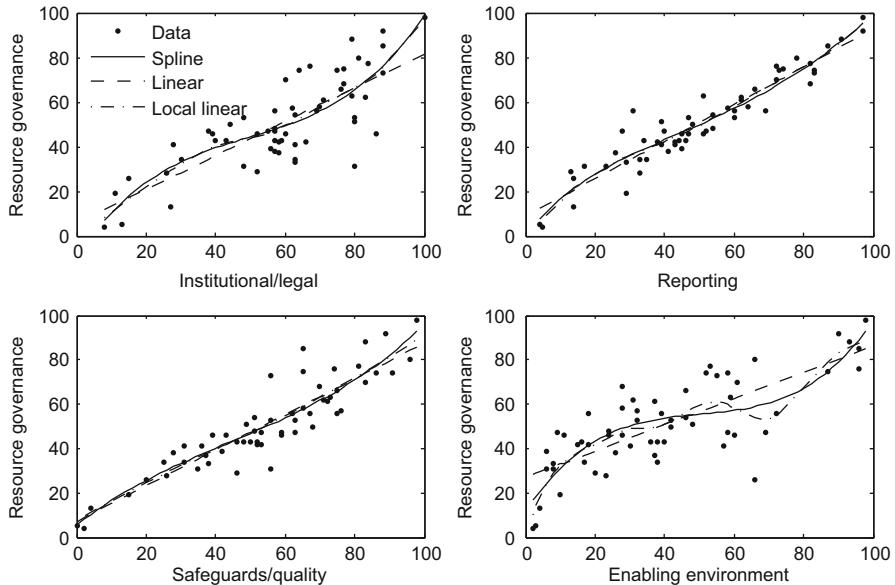


Fig. 34.3 Penalized spline, local linear, and linear fits to the Resource Governance Index

As a further analysis of the effect of each variable, Fig. 34.4 shows the first derivatives of $E(y|x_i)$ with respect to each input variable, as estimated by the penalized splines. First, the nonlinearities of the spline fits are evident from the nonconstant derivatives. The derivative plots also have implications for the index itself – they effectively summarize the expected change in the RGI that a country would achieve if it moved a given amount in each variable. Two example countries, Libya and India, have their scores in each indicator marked as dotted vertical lines. In the case of Libya, it is clear that to move up the rankings in the RGI, it would be better to invest efforts in improving the Institutional and Legal Setting, and Enabling Environment dimensions, whereas gains in Reporting and Safeguards and Quality would yield lesser gains. India, a country ranked overall 12th in the RGI, would on the other hand stand to gain very little from small improvements in Enabling Environment, and immediate efforts would be better directed toward either Reporting or Safeguards and Quality.

4.2 Financial Secrecy Index (FSI)

4.2.1 Aim

The Financial Secrecy Index (FSI) is developed by the Tax Justice Network (TJN) and aims to measure the contribution to the global problem of financial secrecy in 80 jurisdictions worldwide [5]. In other words, the Index attempts to provide an answer to the question: by providing offshore financial services in combination

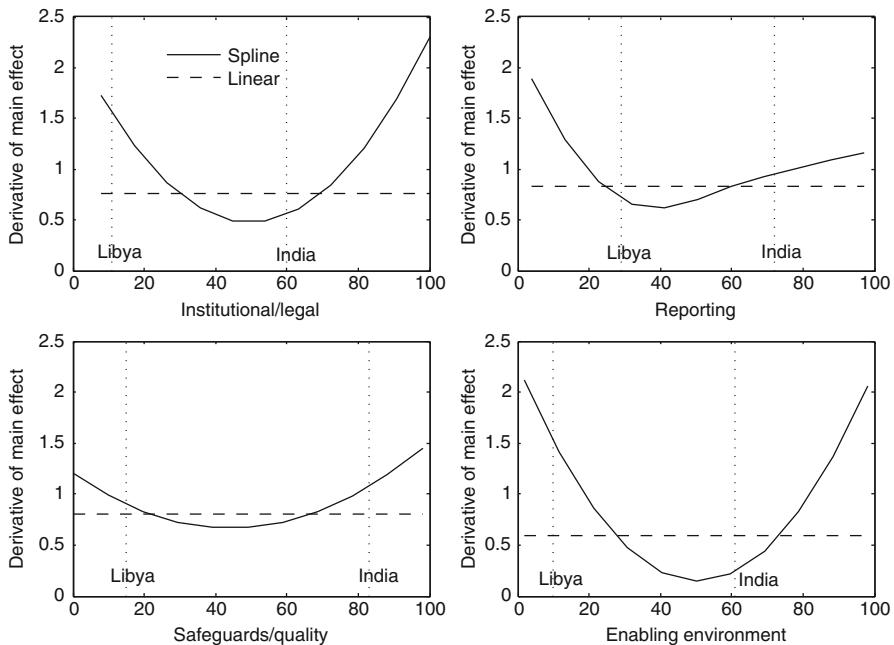


Fig. 34.4 Derivatives $d[E(y|x_i)]/dx_i$ using penalized splines and linear regression fits to the Resource Governance Index. Indicator values of Libya and India marked as *vertical dotted lines*

with a lack of transparency, how much damage is each secrecy jurisdiction actually responsible for?

To give an example of what this implies, the home page of the FSI informs us that because of capital flights dwarfing the inflow of aid money, Africa is in fact a net creditor to the rest of the world since the seventies. As per the money stashed away, the TJN informs us that “those assets are in the hands of a few wealthy people, protected by offshore secrecy, while the debts are shouldered by broad African populations.”

4.2.2 Sources

The index combines qualitative data and quantitative data. Qualitative data are based on laws, regulations, cooperation with information exchange processes, and other verifiable data sources and are used to calculate a secrecy score for each jurisdiction. Secrecy jurisdictions with the highest secrecy scores are more opaque in the operations they host, less engaged in information sharing with other national authorities, and less compliant with international norms relating to combating money laundering. Lack of transparency and unwillingness to engage in effective information exchange makes a secrecy jurisdiction a more attractive location for routing illicit financial flows and for concealing criminal and corrupt activities.

Quantitative data is used to create a global scale score, for each jurisdiction, according to its share of offshore financial service activity in the global total.

Publicly available data about the trade in international financial services of each jurisdiction are used. Where necessary because of missing data, the developers follow International Monetary Fund methodology to extrapolate from stock measures to generate flow estimates. Jurisdictions with the largest weightings are those that play the biggest role in the market for financial services offered to nonresidents.

4.2.3 Main Dimensions

The first dimension of the FSI is the Financial Secrecy score, which is calculated from a set of fifteen qualitative key financial secrecy indicators that assess the degree to which the legal and regulatory systems (or their absence) of a country contribute to the secrecy that enables illicit financial flows. These indicators can be grouped around four dimensions of secrecy: 1) knowledge of beneficial ownership (3 indicators), 2) corporate transparency (3 indicators), 3) efficiency of tax and financial regulation (4 indicators), and 4) international standards and cooperation (5 indicators). Taken together, these indicators result in one aggregate secrecy score for each jurisdiction.

The second dimension of the FSI is the Global Scale score, which is calculated based on quantitative data (publicly available) about the trade in international financial services, and captures the potential for a jurisdiction to contribute to the global problem of financial secrecy. Data on international trade in financial services come from the IMF's Balance of Payments statistics. Data on stocks of portfolio assets and liabilities are taken from two IMF sources: the Coordinated Portfolio Investment Survey and the International Investment Position statistics.

At the final step, the Financial Secrecy score is cubed and the Global Scale is cube rooted before being multiplied to produce the FSI scores for each j th jurisdiction, i.e.:

$$FSI_j = \text{Secrecy}_j^3 \cdot \sqrt[3]{\text{GlobalScale}_j}.$$

Critics have argued that the Global Scale dimension unfairly points to large financial centers. However, the developers' response is that "to dispense with the scale risks ignoring the big elephants in the room." While large players may be slightly less secretive than other jurisdictions, the extraordinary size of their financial sectors offers far more opportunities for illicit financial flows to hide. Therefore, the larger an international financial sector becomes, the better its regulations and transparency ought to be. This logic is reflected in the FSI which aims to avoid the conceptual pitfalls of the "usual suspects" – lists of tax havens which are often remote islands whose overall share in global financial markets is tiny. In the FSI a jurisdiction with a larger share of the offshore finance market, and a moderate degree of opacity, may receive the same overall ranking as a smaller but more secretive jurisdiction.

Due to a significantly greater skew in the Global Scale scores compared to the Financial Secrecy scores, the developers transform the two to generate series that are more evenly distributed. The choice of the transformation has been guided by the 90/10% and the 75/25% quantile ratios in each of the two series. In the original

Table 34.2 Importance measures of the variables of the Financial Secrecy Index. $R_i = \text{corr}(x_i, y)$ (correlation), $S_{i,\text{spl}}$ (correlation ratio, penalized spline), $S_{i,\text{LL}}$ (correlation ratio, local linear)

Financial Secrecy Index (n=80)	x_i	w_i	R_i	R_i^2	$S_{i,\text{spl}}$	$S_{i,\text{LL}}$
Global Scale	x_1	0.5	0.57	0.32	0.63	0.79
Financial Secrecy	x_2	0.5	0.09	0.01	0.13	0.09

series, the 90/10% ratio is more than five thousand times higher for the Global Scale than the Secrecy variable, the 75/25 ratio nearly a hundred times higher. If one squares the Secrecy Score and takes the square root of the Global Scale, these ratios fall to below 26 and 6, respectively; and if one cubes the Secrecy Score and takes the cube root of the Global Scale, they fall below 3 and 2, respectively. Finally, looking at fourth and fifth roots and powers, the variation of the Global Scale series becomes disproportionately small. Hence, the cube root/cube combination was adopted by the developers on the grounds that “these transformations are sufficient to ensure that neither secrecy nor scale alone determine the FSI” – see [6].

4.2.4 Results

Despite the intentions of the developers, and the reasoning based on the quantiles, the correlation ratios S_i (splines and LL approach) and the Pearson correlation coefficients reveal a notably unbalanced impact of the two components on the FSI (see Table 34.2). The greatest difference between the estimates is in the correlation ratios provided by the local-linear regression, in which the values are 0.79 for Global Scale and 0.09 for Secrecy. This is in stark contrast to the intended influence, which should be roughly equal for both variables. Examining the scatterplots in Fig. 34.5 however, the data looks quite challenging to smooth, and the three regression approaches have markedly different fits. In particular, the local-linear regression of the Global Scale variable has a large spike in the fit at a value of around 7, which does not appear to be justified by the data. Therefore, the LL importance estimates ought to be treated with caution. The spline fit is slightly more convincing, but seems to be quite heavily biased by the outlying points above a score of around 5. However, given that even the linear fit yields a far higher importance measure for the Global Scale variable than the Secrecy variable (0.32 and 0.01, respectively), the evidence seems fairly compelling that the Global scale variable dominates the FSI by a significant margin. The analysis illustrates vividly that the cube root/cube transformation of the FSI components and the equal weights assigned to the two components are not a sufficient condition to ensure equal importance, at least according to the correlation ratio measure.

The Global scale scores are particularly skewed (skewness = 4.6 and kurtosis = 23). Yet, after considering the cube root, the distribution is no longer problematic (skewness = 1.7 and kurtosis = 3.0). Nevertheless, what remains problematic is the strong negative association between the cubed distribution of the Financial Secrecy scores and the cube-rooted distribution of the Global Scale scores ($\text{corr} = -.536$).

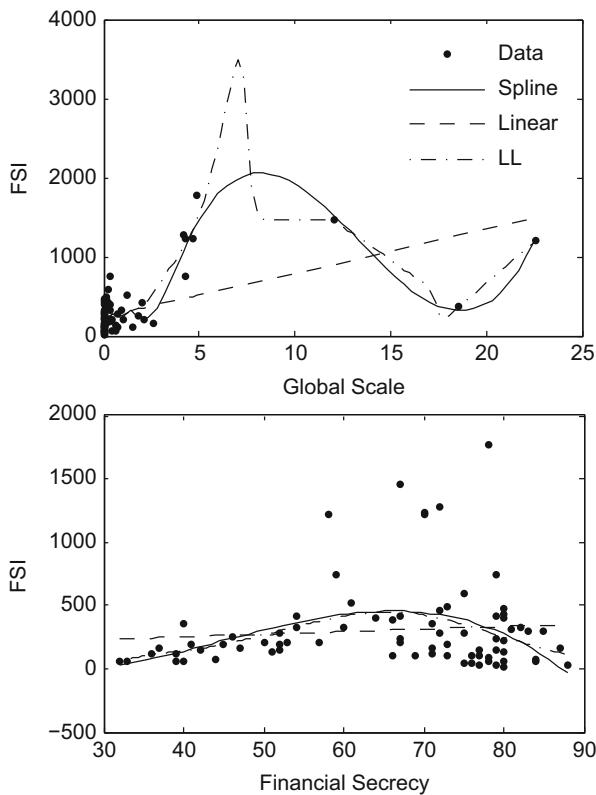


Fig. 34.5 Scatter plots of Financial Secrecy Index versus its components

Table 34.3 Importance measures of the variables of the Financial Secrecy Index with the N_{trim} points with the highest Global Scale scores removed. $R_i = \text{corr}(x_i, y)$ (correlation), $S_{i,\text{spl}}$ (correlation ratio, spline), $S_{i,\text{LL}}$ (correlation ratio, local-linear)

N_{trim}	Financial Secrecy Index (n=50)	R_i^2	$S_{i,\text{spl}}$	$S_{i,\text{LL}}$
3	Global Scale	0.616	0.798	0.754
	Financial Secrecy	0.017	0.112	0.074
8	Global Scale	0.028	0.310	0.077
	Financial Secrecy	0.015	0.119	0.093

If the points with the highest Global Scale scores are treated as outliers, the problem lessens somewhat but does not disappear. Table 34.3 shows the results of rerunning the analysis, trimming the points with three highest or eight highest Global Scale values. By visual inspection, these seem to be two possible reasonable definitions of outliers depending on where the cutoff value is set. Trimming the top three points, the disparity is very similar, with the Global Scale correlation ratios far higher than the Secrecy values. After trimming the top eight points, the importance

scores are much closer; however, in the linear and spline estimates, the importance of the Global Scale variable is still at least twice that of the Secrecy variable. In the case of the LL regression, the Secrecy variable is now rated as slightly more influential than the Global Scale, but given that this is the only (mild) contradiction to an otherwise compelling trend, the conclusion that the variables are not equally influential appears to stand.

In defense of the developers, one may note that this gross unbalance refers to a specific definition of importance (how much the variance of the index would be reduced on average if one could fix one dimension), and this definition appears to condemn the FSI as problematic. One might argue however that the use of variance of the main effect $E(y|x_i)$ as a measure of importance might not tell the whole story and that interactions between the two variables should also be accounted for. Following this line of thought, a useful tool might be the “total effect index” S_{Ti} , which is defined as $E_{\mathbf{x}_{\sim i}}[V_{x_i}(y | \mathbf{x}_{\sim i})]/V(y)$, where $\mathbf{x}_{\sim i}$ denotes the vector of all variables but x_i . This measure also accounts for variance due to interactions. In fact the scatterplot of Financial Secrecy in Fig. 34.5 looks like a textbook example of a variable with a low S_i and a high S_{Ti} , with the points displaying a rather constant mean and an increasing variance while moving over the horizontal axis.

An investigation using this measure in the context of composite indicators is outside of the scope of this work, but further information can be found in the section on Variance-Based Sensitivity Analysis: Theory and Applications.

4.3 Good Country Index

4.3.1 Aim

The Good Country Index (GCI) is developed by the Good Country Party with a view to measure what a country contributes to the common good of humanity, and what it takes away [1], following its developers’ normative framework and world view. In total, 125 countries are included in the Index. In contrast to the majority of similar composite indicators, the Good Country Index does not measure what countries do at home; rather, it aims to start a global discussion about how countries can balance their duty to their own citizens with their responsibility to the wider world.

This reflects the consideration that the biggest challenges facing humanity today are global and borderless: problems such as climate change, economic crisis, terrorism, drug trafficking, slavery, pandemics, poverty and inequality, population growth, food and water shortages, energy, species loss, human rights, and migration.

All of these problems stretch across national borders, so they can be properly tackled through international efforts. Hence, the concept of the “Good Country” is about encouraging populations and their governments to be more outward looking and to consider the international consequences of their national behavior.

4.3.2 Sources

The GCI builds upon 35 indicators that are produced by the United Nations and other international agencies and a few by NGOs and other organizations. Most of

the indicators used are direct measurements of world-friendly or world-unfriendly behavior (such as signing of international treaties, pollution, acts of terrorism, wars) and some are rather indirect (such as Nobel prizes, exports of scientific journals). By adding them up, the developers aim to get a reasonable picture of whether each country is effectively a net creditor to the rest of humanity in each of the seven categories or whether it is a “freeloader” on the global system and ought to be recognized as such.

4.3.3 Main Dimensions

The 35 indicators are split in seven groups of five indicators each. These seven dimensions, which closely mirror the dimensions of the United Nations Charter, are:

1. Science, Technology, and Knowledge, which includes foreign students studying in the country; exports of periodicals, scientific journals, and newspapers; articles published in international journals; Nobel prize winners; and International Patent Cooperation Treaty applications.
2. Culture, which measures exports and imports of creative goods; UNESCO dues in arrears (a negative indicator); countries and territories that citizens can enter without a visa; and freedom of the press (based on the mean score of the Reporters without Borders and Freedom House indices as a negative indicator).
3. International Peace and Security, which aggregates peacekeeping troops sent overseas; dues in arrears to financial contribution to UN peacekeeping missions (negative indicator); casualties of international organized violence (negative indicator); exports of weapons and ammunition (negative indicator); and the Global Cyber Security Index (negative indicator).
4. World Order, which measures population that gives to charity as proxy for cosmopolitan attitude; refugees hosted; refugees overseas (negative indicator); population growth rate (negative indicator); and treaties signed as proxy for diplomatic action and peaceful conflict resolution.
5. Planet and Climate, which measures the National Footprint Accounts Biocapacity reserve; exports of hazardous waste (negative indicator); organic water pollutant emissions (negative indicator); CO₂ emissions (negative indicator); and methane + nitrous oxide + other greenhouse gases (HFC, PFC, and SF6) emissions (negative indicator).
6. Prosperity and Equality, which aggregates trading across borders; aid workers and volunteers sent overseas; fair trade market size; foreign direct investment outflow; and development cooperation contributions (aid).
7. Health and Wellbeing, which includes wheat-tonne-equivalent food aid shipments; exports of pharmaceuticals; voluntary excess contributions to the World Health Organization; humanitarian aid contributions; and drug seizures.

A ranking is calculated for each of the seven dimensions. The Good Country Index is then calculated by taking the arithmetic average of the seven ranks in Science, Technology, and Knowledge; Culture; International Peace and Security;

World Order; Planet and Climate; Prosperity and Equality; and, finally, Health and Wellbeing. This aggregation scheme has been selected by the developers because of its simplicity and relative robustness to outliers. Beyond what is stated by the developers, a further argument in favor of this aggregation scheme would be the “imperfect substitutability across all seven index components,” i.e., the reduction of the fully compensatory nature of an arithmetic average of the seven scores.

4.3.4 Results

Before looking at the correlation ratios describing the importance of each variable to the output, it is useful to look at the correlations between input variables, as it is good practice in the construction/evaluation of composite indicators. Six of the seven dimensions of the Good Country Index have low to moderate correlations that range from 0.20 (between several pairs of dimensions, mostly those involving Prosperity and Equality) to 0.78 (between Science and Technology and Culture) and an overall moderate average bivariate correlation of 0.37. Principal component analysis suggests that there is indeed a single latent phenomenon underlying the six dimensions and that the first principal component captures almost 50% of the variation in these dimensions. However, the International Peace and Security dimension has a negative correlation to both the Science and Technology and to the Culture variables (-0.48) and almost random correlation to all remaining dimensions. This point is a strong concern for the validity of the GCI. The negative correlations between International Peace and Security on one hand and either Science and Technology or Culture or Health and Wellbeing on the other are undesirable in a composite indicator context, as they suggest the presence of trade-offs, and are a reminder of the danger of compensability between components.

Turning now to the correlation coefficients, Table 34.4 shows that, unlike the equal weighting scheme of the seven components would suggest, the impact of the seven components on the index is uneven. Three of the seven components, namely, Culture, World Order, and Science and Technology, account for over 50% of the variation in the index scores (up to 63% for Culture). Instead, by fixing either of the three components – Health and Wellbeing, Planet and Climate, and Prosperity and Equality – the variance reduction would be between 25–37%. What is striking

Table 34.4 $R_i = \text{corr}(x_{i,}, y)$ (correlation), $S_{i,\text{spl}}$ (correlation ratio, spline), $S_{i,\text{ker}}$ (correlation ratio, local linear)

Good Country Index (n=125)	x_i	w_i	R_i	R_i^2	$S_{i,\text{spl}}$	$S_{i,\text{LL}}$
Science and Technology	x_1	1/7	0.71	0.50	0.50	0.50
Culture	x_2	1/7	0.79	0.63	0.63	0.66
International Peace and Security	x_3	1/7	-0.17	0.03	0.05	0.03
World Order	x_4	1/7	0.78	0.62	0.64	0.63
Planet and Climate	x_5	1/7	0.57	0.32	0.34	0.33
Prosperity and Equality	x_6	1/7	0.49	0.24	0.27	0.25
Health and Wellbeing	x_7	1/7	0.55	0.30	0.35	0.37

is that the International Peace and Security component is practically cosmetic in the framework: by fixing this component, the index variance would be reduced by merely 3%. Moreover, it actually has a negative correlation with the GCI output, meaning that countries that rank low in International Peace and Security, on average, actually stand out as “good countries,” with a higher GCI score. This effect, likely due to the negative correlations with other variables, suggests a weakness in the GCI which ought to be addressed.

This conclusion can be of value in a general sense because it indicates that, aside from the choice of weights and aggregation formulas based on subjective considerations, the impact of components on the variance of the index is driven by the statistical properties of the components, and this latter fact is often overlooked during the index development process.

5 Discussion on Estimation Approaches

The tables showing correlation ratios and correlation coefficients from the case studies in the previous section (i.e., Tables 34.1, 34.2, and 34.4) help to understand the relative merits of the measures used in this study. The correlation coefficient R_i and the coefficient of determination R_i^2 give the linear correlation of the composite indicator with each of its inputs. The R_i^2 measure can be interpreted as the fraction of variance accounted for by the linear regression (similar to S_i), but it is also useful to see the R_i value in order to understand whether the relation is positive or negative. As shown in the Good Country Index (Table 34.4), this measure revealed that one of the input indicators is in fact negatively correlated with the composite indicator – this is an undesirable property as discussed previously, at least in the context of linearly or geometrically aggregated composite indicators.

The spline and local-linear estimates of S_i are all higher than or equal to the R_i^2 values. In most cases the difference is small – this reflects the fact that the main effects are close to linear. In these cases (see, e.g., the “Reporting Practices” variable of the Resource Governance Index in Table 34.1), the spline and LL regression fits are very close to straight lines. However, in some cases, such as the Enabling Environment pillar (also from the Resource Governance Index), S_i estimates are significantly higher than R_i^2 – this is due to the nonlinear main effect, which is captured by the spline and LL regression, but missed by the linear regression. In this instance, $R^2 = 0.59$ and S_i is estimated as 0.65 or 0.70 by the spline and LL regression, respectively. Looking at all three tables, one can see that in the majority of cases, R_i^2 would be a sufficient approximation of the effect of each variable, but there are several exceptions. This shows the added value of nonlinear regression – it can approximate linear and near-linear main effects, which appear in the majority of cases, but can generalize to nonlinear fits when required.

An obvious question at this point is to ask whether splines or LL regression provides a better estimate of S_i , given that the fits are slightly different in some cases. The short answer is that there is no way to know which is better, given that we do not know the “true” values of S_i or the “true” main effects. The three tables

show however that in the majority of cases, the estimates are very similar and in the cases where the estimates differ, neither the splines nor the LL regression has a tendency to provide higher estimates than the other overall.

It is tempting to think that higher estimates might be better estimates, given that they “capture” more of the total variance. However, both the splines and the LL regression could be easily adjusted to capture 100% of the variance; this would result in interpolation rather than smoothing. But this does not in general provide a good approximation of the main effects and results in *overfitting*. Instead, the aim of nonlinear regression is to find a balance between interpolation and simplicity – this is known as the “bias-variance tradeoff” in machine learning; see, e.g., [9].

Given then that neither splines nor local-linear regression provides consistently higher or lower estimates than the other, the best strategy would be to estimate main effects using both methods and then compare the results. It can also be helpful to visually examine the fits – for example, in Fig. 34.5, the Global Scale variable of the Financial Secrecy Index gives a data set that results in very different main effect estimates between the linear, spline, and local-linear approaches. Although none of the fits seem extremely convincing (the data is strongly skewed and heteroskedastic), the LL fit in particular has a strange “spike” that does not appear to be justified by the data – in this case one might treat the estimate of the local-linear regression with caution. A clear avenue of research here would be to incorporate confidence intervals into the estimates of main effects and subsequently into the estimates of S_i . Some approaches to doing this within the context of splines can be found in [17]. An alternative tool might be to use a Bayesian implementation of a Gaussian process, which would formally propagate uncertainties in main effect estimation to estimates of correlation ratios.

Aside from the fitting of the data, splines may offer some small advantages over LL regression. The first is that, being closely related to linear regression, they can provide fast analytic estimates of derivatives. This property is used in Fig. 34.4, to illustrate the gain in Resource Governance Index that a country or entity would achieve if its value of a given variable changed by a given amount along its axis. Furthermore, splines are able to exploit properties of linear regression (such as calculation of the cross-validation measures) to allow very fast fitting to a given data set. In the examples here, which feature relatively small data sets, computational time is not an issue. However, some composite indicators based on physical maps measure concepts such as ecosystem service indices at resolutions as high as 1km, over the whole of Europe [12]. In such cases, the data set may feature millions of points, and splines offer a significant advantage over the slower local-linear regression.

6 Conclusions

The leitmotif of the present chapter is to “mind the gap” between the two stories which can be told about the weights of composite indicators. On one hand weights appear to “belong” to developers, who are entitled to set them by analysis and

negotiation based on their norms and values. On the other hand, most forms of aggregation – and notably all linear aggregations – are particularly inept at translating these weights into “effects.” The proposal here is to look at the problem by choosing a statistically defensible definition of importance, one that derives from the theory of global sensitivity analysis, which in turn originates from the theory of experimental design and of analysis of variance (ANOVA-like decomposition). It is therefore possible to compare the importance of different variables against what their weights would purport them to be. The discussion of the Financial Secrecy Index makes it clear that all inferences made here are conditional on the definition of “importance.” There is nothing surprising in this. The same holds in sensitivity analysis. In fact “in sensitivity analysis [an] analyst [must] specify what is meant by “factor’s importance” for a particular application. Leaving instead the concept of importance undefined could result in several tests being thrown at the problem and in several rankings of factor importance being obtained, without a basis to decide which one to believe” [24]. The situation is similar in the analysis of the coherence of the weights with the behavior of the index.

Still, within the limits of this analysis, it is perhaps possible to say something useful to the developers. As far as the Resource Governance Index is concerned, it can be said that developers were successful in making Reporting Practices the most important dimension. They were less successful in making the other three dimensions equally important (Table 34.1). Further, in the intention of the developers, each of these three dimensions, “Institutional and Legal Setting,” “Safeguards and Quality Controls,” and “Enabling Environment,” was supposed to be half important as “Reporting Practices.” The correlation structure of the problem did not allow this to happen, at least with the selected weights.

For the Financial Secrecy Index, the measures of importance point to a problematic property of the index, whereby one dimension, “Global Scale,” apparently far more important than the other, “Financial Secrecy” (with the exact ratio varying somewhat between various estimates), while the two are intended to be equally important. At the same time, the aggregation formula chosen generates an important interaction term in the index. This challenges the correlation ratio as a sufficient measure of importance for this particular application. More analysis is needed on this challenging test case.

In the case of the Good Country Index, it is clear that the ambition to capture several dimensions normatively labeled as equally important was not rewarded by the results. Not only are the dimensions unequal, but one important dimension, “International Peace and Security,” has a very small influence on the index score and in fact has a negative correlation with the output.

Note that the approach presented in this chapter can be used to actually improve the indices by tuning the weights as to obtain the desired importance [13]. The approach has been implemented in two important indices: the INSEAD-WIPO Global Innovation Index [18] and the Environmental Performance Index developed by Yale University [2].

References

1. Anholt, S., Govers, R.: The good country index. Tech. rep., The Good Country Party. <http://www.goodcountry.org/> (2014)
2. Athanasoglou, S., Weziak-Bialowolska, D., Saisana, M.: Environmental performance index 2014: Jrc analysis and recommendations. Tech. rep., European Commission, Joint Research Centre (2014)
3. Bandura, R.: Composite indicators and rankings: inventory 2011. Tech. rep., United Nations Development Programme – Office of Development Studies (2011)
4. Bowman, A., Azzalini, A.: Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations, vol 18. Oxford University Press, New York (1997)
5. Cobham, A., Jansky, P., Christensen, J., Eichenberger, S.: Financial Secrecy Index 2013: Methodology. Tech. rep., The Tax Justice Network. <http://www.financialsecrecyindex.com/> (2013)
6. Cobham, A., Janský, P., Meinzer, M.: The financial secrecy index: shedding new light on the geography of secrecy. *Econ. Geogr.* **91**(3), 281–303 (2015)
7. Da Veiga, S., Wahl, F., Gamboa, F.: Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics* **51**(4), 452–463 (2009)
8. Decancq, K., Lugo, M.A.: Weights in multidimensional indices of wellbeing: an overview. *Econ. Rev.* **32**(1), 7–34 (2013)
9. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Data Mining, Inference and Prediction. Springer, New York (2001)
10. Kelley, J.G., Simmons, B.A.: Politics by number: indicators as social pressure in international relations. *Am. J. Pol. Sci.* **59**(1), 55–70 (2015)
11. Li, G., Rabitz, H., Yelvington, P.E., Oluwole, O.O., Bacon, F., Kolb, C.E., Schoendorf, J.: Global sensitivity analysis for systems with independent and/or correlated inputs. *J. Phys. Chem. A* **114**(19), 6022–6032 (2010)
12. Paracchini, M.L., Zulian, G., Koppenroinen, L., Maes, J., Schägner, J.P., Termansen, M., Zandersen, M., Perez-Soba, M., Scholefield, P.A., Bidoglio, G.: Mapping cultural ecosystem services: a framework to assess the potential for outdoor recreation across the EU. *Ecol. Indic.* **45**, 371–385 (2014)
13. Paruolo, P., Saisana, M., Saltelli, A.: Ratings and rankings: voodoo or science? *J. R. Stat. Soc. Ser. A (Stat. Soc.)* **176**(3), 609–634 (2013)
14. Pearson, K.: On the General Theory of Skew Correlation and Non-linear Regression. Volume XIV of Mathematical Contributions to the Theory of Evolution, Drapers' Company Research Memoirs. Dulau & Co., London (1905). Reprinted in: Early Statistical Papers, Cambridge University Press, Cambridge (1948)
15. Quiroz, J.C., Lintzer, M.: The 2013 resource governance index. Tech. rep., The Revenue Watch Institute (2013)
16. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
17. Ruppert, D., Wand, M., Carroll, R.: Semiparametric Regression, vol. 12. Cambridge University Press, Cambridge (2003)
18. Saisana, M., Saltelli, A.: Joint Research Centre statistical audit of the 2014 Global Innovation Index. Tech. rep., European Commission, Joint Research Centre (2014)
19. Saisana, M., Saltelli, A., Tarantola, S.: Uncertainty and sensitivity analysis techniques as tools for the quality assessment of composite indicators. *J. R. Stat. Soc. A* **168**(2), 307–323 (2005)
20. Saisana, M., d'Hombres, B., Saltelli, A.: Ricketty numbers: volatility of university rankings and policy implications. *Res. Policy* **40**(1), 165–177 (2011)
21. Saltelli, A., Tarantola, S., Campolongo, F.: Sensitivity analysis as an ingredient of modelling. *Stat. Sci.* **15**(4), 377–395 (2000)

22. Saltelli, A., Tarantola, S.: On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. *J. Am. Stat. Assoc.* **97**, 702–709 (2002)
23. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global Sensitivity Analysis – The Primer*. John Wiley & Sons, Hoboken (2008)
24. Saltelli, A., Ratto, M., Tarantola, S., Campolongo, F.: Update 1 of: Sensitivity analysis for chemical models. *Chem. Rev.* **112**(5), 1–25 (2012)
25. Storlie, C., Helton, J.: Multiple predictor smoothing methods for sensitivity analysis: Description of techniques. *Reliab. Eng. Syst. Saf.* **93**(1), 28–54 (2008)
26. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **58**, 267–288 (1996)
27. Yang, L.: An inventory of composite measures of human progress. Tech. rep., United Nations Development Programme Human Development Report Office (2014)

Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms

35

Clémentine Prieur and Stefano Tarantola

Abstract

This section aims at presenting an overview of variance-based approaches for global sensitivity analysis. Starting from functional ANOVA, Sobol' indices are first defined and then estimation algorithms are provided. The performance of these algorithms is theoretically and practically discussed. The review includes recent results on the topic.

Keywords

FANOVA • Sobol' sensitivity indices • Global sensitivity analysis • Monte Carlo sampling • Quasi-Monte Carlo sampling • Sampling design • Replication • Latin hypercube sampling • Orthogonal arrays (OA) • Spectral methods • Fourier amplitude sensitivity test • Random balance design • Effective algorithm for sensitivity indices • Polynomial chaos expansion • effective dimension • Sensitivity indices with given data

Contents

1	Introduction	1218
2	General Definition of FANOVA Representation	1218
3	Definition of Sobol' Sensitivity Indices	1220
4	Available Estimation Techniques	1221

C. Prieur (✉)

Laboratoire Jean Kuntzmann (LJK), University of Grenoble Alpes, INRIA, Grenoble, France
e-mail: clementine.prieur@imag.fr

S. Tarantola

Statistical Indicators for Policy Assessment, Joint Research Centre of the European Commission,
Ispra (VA), Italy
e-mail: stefano.tarantola@jrc.ec.europa.eu

4.1 Monte Carlo-Based Techniques	1221
4.2 Spectral Approaches	1227
5 Test Functions	1232
5.1 The g -Sobol' Function and Effective Dimension	1232
5.2 A Discontinuous Test Function	1234
6 Conclusions	1236
References	1236

1 Introduction

Many mathematical models involve input parameters and variables (simply named inputs from now on), which are not precisely known. Global sensitivity analysis aims to identify the inputs whose uncertainty has the largest impact on the variability of a quantity of interest (output of the model). One of the statistical tools used to quantify the influence of each input on the output are the Sobol' sensitivity indices. These indices measure the part of the output's variance due to one or more inputs. To define the so-called Sobol' sensitivity indices, a first step is the decomposition of the model, as a d -variate function $G(x)$ into a finite hierarchical expansion of component functions in terms of the input $x = (x_1, x_2, \dots, x_d)$. A way to do that is the functional ANOVA (FANOVA) decomposition also known as the high-dimensional model representation (HDMR) technique.

2 General Definition of FANOVA Representation

Consider a numerical model $G(x)$, depending on d inputs x_1, \dots, x_d , which may be scalar or vectorial: for any $j \in \{1, \dots, d\}$, $x_j \in \mathbb{R}^{k_j}$, where $k_j \in \mathbb{Z}_{>0}$, where $\mathbb{Z}_{>0}$ denotes the set of positive integers. In the global sensitivity analysis framework, the uncertainty of the inputs is modeled by random vectors. More precisely, the uncertainty of the inputs is modeled by the random vector $\mathbf{X} = (X_1, \dots, X_d)$ whose distribution is a product distribution $P_{\mathbf{X}} = P_{X_1} \otimes \dots \otimes P_{X_d}$, with P_{X_j} a probability measure on \mathbb{R}^{k_j} whose support is denoted by $I_j \subset \mathbb{R}^{k_j}$. It means that the random vectors X_j , $j = 1, \dots, d$ are mutually independent. Note that no further assumption is made on the dependence structure of each random vector X_j , $j = 1, \dots, d$.

Definition 1 (FANOVA decomposition; see, e.g., [9, 37, 48]). Let $\mathbf{x} = (x_1, \dots, x_d) \in I_1 \times \dots \times I_d$. Let $G \in \mathbb{L}_{\mathbb{R}}^2(P_{\mathbf{X}}) = \{f : I_1 \times \dots \times I_d \rightarrow \mathbb{R} \mid \int f^2(\mathbf{x}) P_{\mathbf{X}}(d\mathbf{x}) < +\infty\}$. One can decompose $G(\mathbf{x}) = G(x_1, \dots, x_d)$ as the sum of increasing dimension functions:

$$\begin{aligned} G(\mathbf{x}) &= G_{\emptyset} + \sum_{j=1}^d G_j(x_j) + \sum_{1 \leq j < k \leq d} G_{j,k}(x_j, x_k) + \dots + G_{1,\dots,d}(\mathbf{x}) \\ &= \sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})} G_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}), \end{aligned}$$

with $\mathbf{x}_\mathbf{u} = (x_j, j \in \mathbf{u})$. This expansion exists and is unique under the constraint

$$\int G_\mathbf{u}(\mathbf{x}_\mathbf{u}) P_{X_j}(dx_j) = 0 \quad \forall j \in \mathbf{u}, \quad \forall \mathbf{u} \in \mathcal{P}(\{1, \dots, d\}).$$

Remark. A consequence of Definition 1 is that for any $\mathbf{u} \neq \mathbf{v} \in \mathcal{P}(\{1, \dots, d\})$, $\int G_\mathbf{u}(\mathbf{x}_\mathbf{u}) G_\mathbf{v}(\mathbf{x}_\mathbf{v}) P_{\mathbf{X}}(d\mathbf{x}) = 0$.

From Definition 1, the following corollary follows:

Corollary 1. *From the FANOVA decomposition of G , one gets*

$$\begin{aligned} \text{Var}(G(\mathbf{X})) &= \sum_{j=1}^d \text{Var}(G_j(X_j)) \\ &\quad + \sum_{1 \leq j < k \leq d} \text{Var}(G_{j,k}(X_j, X_k)) + \dots + \text{Var}(G_{1,\dots,d}(\mathbf{X})) \\ &= \sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})} \text{Var}(G_\mathbf{u}(\mathbf{X}_\mathbf{u})). \end{aligned}$$

Proof. The FANOVA decomposition defined in Definition 1 is applied to the random vector \mathbf{X} , and the expectation of the square

$$\mathbb{E}(G(\mathbf{X}) - G_\emptyset)^2 = \mathbb{E}\left(\sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})} G_\mathbf{u}(\mathbf{X}_\mathbf{u})\right)^2$$

is considered. From the orthogonality of the $G_\mathbf{u}$, $\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})$ (see the remark just below Definition 1), one can deduce:

$$\mathbb{E}(G(\mathbf{X}) - G_\emptyset)^2 = \sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})} \mathbb{E}(G_\mathbf{u}(\mathbf{X}_\mathbf{u}))^2.$$

Then, under the constraints in Definition 1, one gets that $G_\emptyset = \mathbb{E}(G(\mathbf{X}))$ and that the $G_\mathbf{u}(\mathbf{X}_\mathbf{u})$ are centered. Thus, $\mathbb{E}(G(\mathbf{X}) - G_\emptyset)^2 = \text{Var}(G(\mathbf{X}))$ and $\mathbb{E}(G_\mathbf{u}(\mathbf{X}_\mathbf{u}))^2 = \text{Var}(G_\mathbf{u}(\mathbf{X}_\mathbf{u}))$. This concludes the proof of Corollary 1. \square

Remark. The elements of the decomposition can be computed recursively with the formula

$$G_\mathbf{u}(\mathbf{X}_\mathbf{u}) = \sum_{\mathbf{v} \subset \mathbf{u}} (-1)^{|\mathbf{u}| - |\mathbf{v}|} \mathbb{E}(G(\mathbf{X}) | \mathbf{X}_\mathbf{i}, \mathbf{i} \in \mathbf{v}).$$

3 Definition of Sobol' Sensitivity Indices

Following [48], Sobol' sensitivity indices are defined based on the functional ANOVA decomposition introduced in the previous section.

Definition 2. For any $\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})$, if $G \in \mathbb{L}_{\mathbb{R}}^2(P_{\mathbf{X}})$ and $\text{Var}(G(\mathbf{X})) \neq 0$, define

- the Sobol' index associated to \mathbf{u} : $S_{\mathbf{u}} = \frac{\text{Var } G_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}})}{\text{Var } G(\mathbf{X})}$.
- the closed Sobol' index associated to \mathbf{u} : $S_{\mathbf{u}}^{\text{clo}} = \frac{\text{Var } (\mathbb{E}(G(\mathbf{X})|X_j, j \in \mathbf{u}))}{\text{Var } G(\mathbf{X})}$.
- the total Sobol' index associated to \mathbf{u} : $S_{\mathbf{u}}^{\text{tot}} = 1 - \frac{\text{Var } (\mathbb{E}(G(\mathbf{X})|X_j, j \notin \mathbf{u}))}{\text{Var } G(\mathbf{X})}$.

From Corollary 1, one gets

$$1 = \sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\}), \mathbf{u} \neq \emptyset} S_{\mathbf{u}}.$$

Interpretation. An intuitive interpretation of these indices is given in the case where $\mathbf{u} = \{j\}$ or $\{j, k\}$, $1 \leq j \neq k \leq d$.

- The first-order Sobol' index $S_{\{j\}}$, also denoted by S_j , measures the main effect of the group of inputs $X_j \in \mathbb{R}^{k_j}$. The second-order Sobol' index $S_{\{j,k\}}$ measures the effect due to the second-order interaction between groups X_j and X_k , once the main effect of each group has been removed.
- The closed second-order Sobol' index $S_{\{j,k\}}^{\text{clo}}$ measures the effect due to both groups X_j and X_k . One has $S_{\{j,k\}}^{\text{clo}} = S_{\{j,k\}} + S_{\{j\}} + S_{\{k\}}$. Moreover, $S_{\{j\}}^{\text{clo}} = S_{\{j\}}$.
- At last, the total Sobol' index $S_{\{j\}}^{\text{tot}}$ measures the effect due to X_j and its interactions with all the other X_k , $k = 1, \dots, j-1, j+1, \dots, d$ [15]. One has $S_{\{j\}}^{\text{tot}} = \sum_{\mathbf{u} \in \mathcal{P}(\{1, \dots, d\}), j \in \mathbf{u}} S_{\mathbf{u}}$. Note that $S_{\{j\}}^{\text{tot}} = 1 - S_{-j}^{\text{clo}}$ where $-j = \{1, \dots, j-1, j+1, \dots, d\}$.

Remark.

- The origin of Sobol' indices probably goes back to the correlation ratio [39]. In the seminal paper [48], the X_j are assumed to be uniformly distributed on $[0, 1]$. However, the definitions in [48] easily generalize to the ones proposed in Definition 2.
- The generalization of the total Sobol' index $S_{\mathbf{u}}^{\text{tot}}$, for any $\mathbf{u} \in \mathcal{P}(\{1, \dots, d\})$, is also known as total variation associated with input set \mathbf{u} (see, e.g., [45]). It is also possible to consider the importance superset measure of the group of variables

$\mathbf{X}_{\mathbf{u}}$, as introduced in [31] for uniform distributions: $\sum_{\mathbf{u} \subset \mathbf{v}} S_{\mathbf{v}}$. This last measure was also investigated in the data-mining framework by [16], and in the case where \mathbf{u} represents a pair of inputs, it was used for screening purposes in [11]. In these last papers, the authors work with the unnormalized versions of the indices, that is, they consider the above indices multiplied by the overall variance $\text{Var } G(\mathbf{X})$.

4 Available Estimation Techniques

4.1 Monte Carlo-Based Techniques

4.1.1 Derivation of the MC-Type Estimators

The evaluation of the sensitivity indices defined above is usually based on Monte Carlo integrations. Several approaches have been proposed in the literature to numerically compute such sensitivity indices [15, 19, 44, 48]. This work present the most efficient formulas available today, as recommended by [44]. For the sake of clarity in the presentation, it is assumed that for any $j = 1, \dots, d$, X_j is a one-dimensional random vector, that is, $k_j = 1$. However, the estimation techniques presented below can easily be generalized to the estimation of the general indices defined in Definition 2.

The closed Sobol' index associated to \mathbf{u} , $S_{\mathbf{u}}^{\text{clo}}$, is derived by expressing its numerator as

$$\text{Var} [\mathbb{E}(G(\mathbf{X})|\mathbf{X}_{\mathbf{u}})] = \int \mathbb{E}^2(G(\mathbf{X})|\mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}_{\mathbf{u}}}(d\mathbf{x}_{\mathbf{u}}) - \left(\int \mathbb{E}(G(\mathbf{X})|\mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}_{\mathbf{u}}}(d\mathbf{x}_{\mathbf{u}}) \right)^2 \quad (35.1)$$

The second term in (35.1) reduces to $\mathbb{E}^2(G(\mathbf{X}))$ since $\mathbb{E}[\mathbb{E}(G(\mathbf{X})|\mathbf{X}_{\mathbf{u}})] = \mathbb{E}(G(\mathbf{X}))$, whereas the first term can be written as

$$\begin{aligned} \int \mathbb{E}^2(G(\mathbf{X})|\mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}_{\mathbf{u}}}(d\mathbf{x}_{\mathbf{u}}) &= \int \mathbb{E}(G(\mathbf{X})|\mathbf{x}_{\mathbf{u}}) \mathbb{E}(G(\mathbf{X})|\mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}_{\mathbf{u}}}(d\mathbf{x}_{\mathbf{u}}) \\ &= \int \left(\int \int G(\mathbf{x}_{\bar{\mathbf{u}}}, \mathbf{x}_{\mathbf{u}}) G(\mathbf{x}'_{\bar{\mathbf{u}}}, \mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}_{\bar{\mathbf{u}}}}(d\mathbf{x}_{\bar{\mathbf{u}}}) P_{\mathbf{X}_{\bar{\mathbf{u}}}}(d\mathbf{x}'_{\bar{\mathbf{u}}}) \right) P_{\mathbf{X}_{\mathbf{u}}}(d\mathbf{x}_{\mathbf{u}}) \\ &= \int \int G(\mathbf{x}_{\bar{\mathbf{u}}}, \mathbf{x}_{\mathbf{u}}) G(\mathbf{x}'_{\bar{\mathbf{u}}}, \mathbf{x}_{\mathbf{u}}) P_{\mathbf{X}}(d\mathbf{x}) P_{\mathbf{X}'_{\bar{\mathbf{u}}}}(d\mathbf{x}'_{\bar{\mathbf{u}}}) \end{aligned} \quad (35.2)$$

where $\mathbf{X}_{\bar{\mathbf{u}}}$ denotes the input variables complementary to $\mathbf{X}_{\mathbf{u}}$ and \mathbf{X}' is an independent copy of \mathbf{X} . Writing $\mathbb{E}^2(G(\mathbf{X}))$ as

$$\mathbb{E}^2(G(\mathbf{X})) = \mathbb{E}(G(\mathbf{X})) \mathbb{E}(G(\mathbf{X}')) \quad (35.3)$$

and replacing (35.2) and (35.3) into (35.1), one obtains

$$\text{Var} [\mathbb{E}(G(\mathbf{X})|\mathbf{X}_{\mathbf{u}})] = \int G(\mathbf{x}) \left(G(\mathbf{x}'_{\bar{\mathbf{u}}}, \mathbf{x}_{\mathbf{u}}) - G(\mathbf{x}') \right) P_{\mathbf{X}}(d\mathbf{x}) P_{\mathbf{X}'_{\bar{\mathbf{u}}}}(d\mathbf{x}'). \quad (35.4)$$

Now, if \mathbf{C} denotes a $N \times d$ sampling matrix, $G(\mathbf{C})$ is the N -dimensional vector whose i th coordinate $G(\mathbf{C})^{(i)}$ is the evaluation of G on the i th row of \mathbf{C} .

Then, the integral in the right-hand side of (35.4) can be computed numerically by Monte Carlo:

$$S_{\mathbf{u}}^{\text{clo}} \cdot \text{Var } G(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N G(\mathbf{B})^{(i)} (G(\mathbf{A}_{\mathbf{B}(\mathbf{u})}^{(i)}) - G(\mathbf{A})^{(i)}) \quad (35.5)$$

where N is the sample size (typically of the order of 500–1,000, although larger values might be needed to achieve stable estimates of sensitivity for more complex $G(\mathbf{X})$); \mathbf{A} and \mathbf{B} are two independent $N \times d$ sampling matrices where each row of the matrix is a sample point in the d -dimensional space of the inputs. In (35.5), $\mathbf{A}_{\mathbf{B}(\mathbf{u})}$ is a resampled matrix, where columns \mathbf{u} come from matrix \mathbf{B} and all other $d - \text{card}(\mathbf{u})$ columns come from matrix \mathbf{A} . Here, $\text{card}(\mathbf{u})$ is the cardinality of subset \mathbf{u} . The term $\text{Var } G(\mathbf{X})$ can be computed efficiently using $2N$ sample points, i.e., the N rows of matrix \mathbf{A} and the N rows of matrix \mathbf{B} :

$$\text{Var}(G(\mathbf{X})) = \frac{1}{2N} \sum_{i=1}^N [(G(\mathbf{A})^{(i)} - \mu(\mathbf{A}))^2 + (G(\mathbf{B})^{(i)} - \mu(\mathbf{B}))^2] \quad (35.6)$$

where $\mu(\mathbf{A}) = \frac{1}{N} \sum_{i=1}^N G(\mathbf{A})^{(i)}$ and $\mu(\mathbf{B}) = \frac{1}{N} \sum_{i=1}^N G(\mathbf{B})^{(i)}$.

In the particular case where $\mathbf{u} = \{j\}$, (35.5) reduces to (35.7a), i.e., to the first-order Sobol' index for a single input:

$$S_j \cdot \text{Var } G(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N G(\mathbf{B})^{(i)} (G(\mathbf{A}_{\mathbf{B}(\{j\})})^{(i)} - G(\mathbf{A})^{(i)}). \quad (35.7a)$$

4.1.2 An Alternative to Formulas (35.6) and (35.7a)

An alternative to formulas (35.6) and (35.7a) was proposed in [36]. For $i = 1, \dots, N$ and $j = 1, \dots, d$, define

$$Y^{(i)} = G(\mathbf{B})^{(i)} \text{ and } Y_j^{(i)} = G(\mathbf{A}_{\mathbf{B}(\{j\})})^{(i)}.$$

$\text{Var}(G(\mathbf{X}))$ is estimated by

$$\frac{1}{N} \sum_{i=1}^N Y^{(i)} Y_j^{(i)} - \left(\frac{1}{N} \sum_{i=1}^N \frac{Y^{(i)} + Y_j^{(i)}}{2} \right)^2 \quad (35.7b)$$

and define $S_j \cdot \text{Var } G(\mathbf{X})$ by

$$\frac{1}{N} \sum_{i=1}^N \frac{(Y^{(i)})^2 + (Y_j^{(i)})^2}{2} - \left(\frac{1}{N} \sum_{i=1}^N \frac{Y^{(i)} + Y_j^{(i)}}{2} \right)^2. \quad (35.7c)$$

Remark. It was proven in [18] that the estimator based on (35.7b) and (35.7c) is asymptotically efficient, in the sense that it has an optimal asymptotic variance among a given class of estimators containing, e.g., the estimator defined by formulas (35.6) and (35.7a). However, in the form of (35.7b) and (35.7c), the numerical stability is far from being guaranteed. In Remark 1.4 of [18], the authors propose mathematically equivalent formulas to (35.7b) and (35.7c) which enable greater numerical stability (i.e., less error due to roundoffs).

Remark. Formulas (35.7b) and (35.7c) can be extended for the estimation of closed Sobol' indices of any order as well (see [18] for more details).

Remark. Owen [38] extends (35.5) by drawing points from a third $N \times d$ sample matrix obtaining more accurate estimates of small sensitivity indices. However, no additional evaluations are necessary when applying the strategy of simultaneous estimation of all indices described in [43], Theorem 1 (see also the comments after Theorem 1).

4.1.3 Estimation of Total Sobol' Indices

The Monte Carlo formula for the total Sobol' index S_j^{tot} was derived by [20]

$$S_j^{\text{tot}} \cdot \text{Var } G(\mathbf{X}) = \frac{1}{2N} \sum_{i=1}^N (G(\mathbf{A})^{(i)} - G(\mathbf{A}_{\mathbf{B}(\{j\})})^{(i)})^2. \quad (35.7d)$$

Remark. The total number of model runs required to compute $S_j^{\text{tot}}, j = 1, 2, \dots, d$ (see (35.7d)) is $N_T = N(d + 1)$, while if $S_j, j = 1, 2, \dots, d$ are also computed (see (35.7a)), the total cost increases up to $N_T = N(d + 2)$, due to the extra runs corresponding to the elements of the matrix \mathbf{B} . Likewise, in [38] the cost of the analysis is $N(3d + 2)$ model runs, with a good accuracy for the estimation of small indices. The cost for estimating $S_{\mathbf{u}}^{\text{clo}}, \mathbf{u} \in \mathcal{P}(\{1, \dots, d\})$ (see 35.5) is much larger as there are $2^d - 1$ possible measures of closed Sobol' indices. Therefore, $S_{\mathbf{u}}^{\text{clo}}$ are rarely estimated as the number of model runs could be prohibitive. When used, $S_{\mathbf{u}}^{\text{clo}}$ are generally estimated only for pairs of input variables, i.e., $\text{card}(\mathbf{u}) = 2$. However, in that case, the approach by Saltelli [43] is preferred, which is based on a tricky combinatoric argument.

4.1.4 A Tricky Combinatoric Argument Introduced in [43]

Theorem 1 (Theorem 2 in [43]). *Allowing for $N_T = N \times (2d + 2)$ model evaluations, one can obtain:*

- (1) double estimates for each S_j and S_j^{tot} index
- (2) double estimates for all $\binom{d}{2}$ indices $S_{j_1 j_2}^{\text{clo}}$
- (3) double estimates for all $\binom{d}{2}$ indices $S_{-\{j_1, j_2\}}^{\text{clo}}$

where $\{j_1, j_2\} \subset \{1, \dots, d\}$ and $-\{j_1, j_2\}$ means all variables except $\{j_1, j_2\}$.

Another theorem (Theorem 1 in [43]) shows that one can obtain single estimates of all first-order, second-order, and total sensitivity indices using $N_T = N \times (d + 2)$ model evaluations.

To explain the procedure yielding to Theorem 1 above, an illustration in dimension $d = 4$ is given. The following notation is used:

$$\begin{aligned} U_{\mathbf{u}} &= \mathbb{E}^2(G(\mathbf{X})|\mathbf{X}_{\mathbf{u}}), \\ V_{\mathbf{u}} &= \text{Var}[\mathbb{E}(G(\mathbf{X})|\mathbf{X}_{\mathbf{u}})] = U_{\mathbf{u}} - \mathbb{E}^2(G(\mathbf{X})). \end{aligned}$$

One then has

$$\begin{aligned} S_j &= \frac{U_j - \mathbb{E}^2(G(\mathbf{X}))}{\text{Var } G(\mathbf{X})}, \\ S_j^{\text{tot}} &= \frac{U_{-j} - \mathbb{E}^2(G(\mathbf{X}))}{\text{Var } G(\mathbf{X})}, \\ S_{j_1 j_2}^{\text{clo}} &= \frac{U_{j_1 j_2} - U_{j_1} - U_{j_2} + \mathbb{E}^2(G(\mathbf{X}))}{\text{Var } G(\mathbf{X})}. \end{aligned}$$

Let $\mathbf{a} = \mathbf{A}$, $\mathbf{b} = \mathbf{B}$, and $\mathbf{b}_{\mathbf{u}} = \mathbf{B}_{\mathbf{A}(\mathbf{u})}$.

Saltelli [43] suggests to estimate $\mathbb{E}^2(G(\mathbf{X}))$ by $\frac{1}{N} \sum_{i=1}^N G(\mathbf{b})^{(i)} G(\mathbf{a})^{(i)}$. The following table shows how the terms $U_{\mathbf{u}}$, $\mathbf{u} \subset \{1, \dots, d\}$ can be estimated.

One can see that only $N_T = N \times (2d + 2) = 10N$ model evaluations are needed to estimate twice all indices: S_j , S_j^{tot} , $j = 1, 2, 3, 4$, $S_{j_1 j_2}^{\text{clo}}$, $S_{-\{j_1, j_2\}}^{\text{clo}}$, $1 \leq j_1 \neq j_2 \leq 4$. This combinatoric trick can be applied to 35.5, 35.7a, and 35.7d to obtain double estimates of $S_{\mathbf{u}}$, $S_{\mathbf{u}}^{\text{tot}}$, $S_{\mathbf{u}}^{\text{clo}}$, and $S_{-\mathbf{u}}^{\text{clo}}$ indices with $\mathbf{u} = \{j_1, j_2\} \subset \{1, \dots, d\}$.

4.1.5 Sampling Strategies

There exist various ways to generate the matrices \mathbf{A} and \mathbf{B} . Some of the designs proposed in the following are also discussed in the paper ▶ Chap. 33, “Design of Experiments for Screening” of the present chapter. The more classical sampling scheme is to generate \mathbf{A} and \mathbf{B} from two *i.i.d.* (independent and identically distributed) $N \times d$ samples.

However, particularly in the case where N is small, one should prefer a “space-filling” sampling strategy.

The sampling matrices \mathbf{A} and \mathbf{B} can be generated from quasi-random (also called *low-discrepancy*) sequences. Indeed, quasi-Monte Carlo (QMC) can speed up the convergence rate of sensitivity estimates. Loosely speaking, the estimation of Sobol’ indices can be summarized by a problem of approximating multidimensional integrals. The efficiency of QMC approximations of integrals can be measured by the QMC error bound (see, e.g., [25, 56] for more details on the QMC error

	\mathbf{b}	$\mathbf{b}_{\{1\}}$	$\mathbf{b}_{\{2\}}$	$\mathbf{b}_{\{3\}}$	$\mathbf{b}_{\{4\}}$	$\mathbf{b}_{\{2,3,4\}}$	$\mathbf{b}_{\{1,3,4\}}$	$\mathbf{b}_{\{1,2,4\}}$	$\mathbf{b}_{\{1,2,3\}}$	$\mathbf{b}_{\{1,2,3,4\}}$
\mathbf{b}										
$\mathbf{b}_{\{1\}}$		U_{-1}								
$\mathbf{b}_{\{2\}}$			$U_{-2} U_{-12}$							
$\mathbf{b}_{\{3\}}$				$U_{-3} U_{-13} U_{-23}$						
$\mathbf{b}_{\{4\}}$					$U_{-4} U_{-14} U_{-24} U_{-34}$					
$\mathbf{b}_{\{2,3,4\}}$						$U_1 U_{12} U_{13} U_{14}$				
$\mathbf{b}_{\{1,3,4\}}$							$U_2 U_{12} U_{23} U_{24} U_{-12}$			
$\mathbf{b}_{\{1,2,4\}}$								$U_3 U_{13} U_{23} U_{34} U_{-13} U_{-23}$		
$\mathbf{b}_{\{1,2,3\}}$									$U_4 U_{14} U_{24} U_{34} U_{-14} U_{-24} U_{-34}$	
$\mathbf{b}_{\{1,2,3,4\}}$										$U_1 U_2 U_3 U_4 U_{-1} U_{-2} U_{-3} U_{-4}$

Fig. 35.1 Illustration in dimension $d = 4$ of all possible combinations of \mathbf{b}_u to obtain double estimates of S_j , S_j^{tot} , $S_{j_1 j_2}^{\text{clo}}$, and $S_{-\{j_1, j_2\}}^{\text{clo}}$ indices

bound). If the model has some regularity, in the sense that it has bounded Hardy-Krause variation, the Koksma-Hlawka inequality gives an upper bound for the QMC integration error. The rate of convergence is then in $\mathcal{O}\left(\frac{\log^d(N)}{N}\right)$ where N is the sample size and d the dimension. However, this bound is a weak bound for the integration error, as in practice it was proved that QMC may outperform MC even for large dimension d . Indeed, the key parameter is not the model nominal dimension but rather its effective dimension [2]. The effective dimension in the truncation sense d_T roughly speaking is equal to the number of important factors in the model, while a model has the effective dimension in the superposition sense d_S if it is almost a sum of s -dimensional components, with $s \leq d_S$, in the FANOVA decomposition. These dimensions d_T and d_S have a much more important impact on the rate of convergence of QMC than the nominal dimension d . This property will be illustrated on numerical experiments later in the chapter.

There exist two main families of constructions for low-discrepancy point sets and sequences: lattices and digital nets/sequences (see [28] for more details on these constructions and their properties). It is usual to use Sobol' sequences [21] as particular cases of *quasi-random* (or *low-discrepancy*) sequences of size (N, d) . The low-discrepancy properties of Sobol' sequences deteriorate with increasing dimension of the input space. If the important inputs are located in the last components of X , then the rate of convergence is negatively affected. Therefore, if a preliminary ranking of inputs in order of importance is available, the convergence

of sensitivity estimates could benefit by sampling the inputs in decreasing order of importance. The interested reader is referred to Chapter 5 in [28] for other low-discrepancy sequences, such as Halton, Faure, or Niederreiter sequences.

Some analysts prefer generating matrices \mathbf{A} and \mathbf{B} using Latin hypercube sampling (LHS) (see, e.g., [24]).

Both Sobol' quasi-random sequences and LHS allow additional sample points to be added to matrices \mathbf{A} and \mathbf{B} (i.e., increasing the number of rows N), and equations (35.7a) and (35.7d) can be updated until a desired target accuracy is achieved. Refer to [42] for the presentation of nested LHS. For LHS, at each update, the size of the sample is at least doubled, whereas Sobol' sequences can be updated point by point. Note that Sobol' sequences and LHS can be generated using freely available packages (e.g., the package DiceDesign [51] available on CRAN or the link [22] for MATLAB codes).

4.1.6 A Trick to Reduce the Cost: The Use of Replicated Designs

It is possible to reduce the cost of estimation, in terms of number of model evaluations, for the estimation of first-order Sobol' indices. The trick was first introduced in [32] and then further investigated in [55].

One starts as previously by constructing a $N \times d$ sampling matrix \mathbf{A} . The sampling matrix \mathbf{B} , of size $N \times d$ also, is then constructed from \mathbf{A} by applying a random permutation π_j to each column of \mathbf{A} , $j = 1, \dots, d$. For any $j = 1, \dots, d$, one then constructs a new $N \times d$ array by reordering the lines in \mathbf{B} in such a way that the j th column in this new array $\tilde{\mathbf{B}}_j$ coincides with the j th column of \mathbf{A} .

For each $j = 1, \dots, d$, one then estimates S_j with formula (35.7a) by replacing the design matrix $\mathbf{A}_{\mathbf{B}(\{j\})}$ with $\tilde{\mathbf{B}}_j$. It is important to note that the sampling matrices \mathbf{B} and $\tilde{\mathbf{B}}_j$, $j = 1, \dots, d$ are identical when considered as a set of N points in \mathbb{R}^d , by construction. Thus, the estimation of all first-order Sobol' indices only requires $2N$ model evaluations.

This approach can be extended to the estimation of closed second-order indices, using randomized orthogonal arrays (see [55]), but not to the estimation of total indices.

4.1.7 Estimating Sensitivity Indices Without a Specific Sampling Design or Given Data Estimators

Within the realm of Monte Carlo-based techniques, it is possible to estimate the first-order indices S_j (i.e., $S_{\mathbf{u}}$ when $\mathbf{u} = \{j\}$) without the need to use the sampling design proposed above. This approach is very simple because it uses the definition of S_j . However, the approach is not applicable in practice for the computation of $S_{\mathbf{u}}^{\text{clo}}$ and $S_{\mathbf{u}}^{\text{tot}}$. The starting point can either be a generic random sample of N points of the input \mathbf{X} , on which one can evaluate the model output $\mathbf{G}(\mathbf{X})$, or a precomputed set of model output $(\mathbf{X}^{(i)}, G(\mathbf{X})^{(i)})$, $i = 1, \dots, N$. From the N pairs $(\mathbf{X}^{(i)}, G(\mathbf{X})^{(i)})$, $i = 1, \dots, N$, the quantity $G_j(X_j)$, $j = 1, \dots, d$ can be estimated by partitioning the values of X_j into classes \mathcal{C}_m , $m = 1, \dots, M$ and computing the expectation of $G(\mathbf{X})$ conditional on all the values of X_j that belong to each class.

Finally, the variance of all such conditional expectations yields the numerator of S_j , $\text{Var } G_j(X_j)$. Although several partition strategies are possible, an effective way is to form classes of nearly the same size $N_m = (N/M)$. A de-biased formula has been proposed by [41]

$$S_j = \frac{\sum_{m=1}^M N_m (\bar{G}_m - \bar{G})^2}{\sum_{i=1}^N (G(\mathbf{X})^{(i)} - \bar{G})^2} \quad (35.8)$$

where

$$\bar{G}_m = \frac{1}{N_m} \sum_{i:X_j^{(i)} \in C_m} G(\mathbf{X})^{(i)}$$

and

$$\bar{G} = \frac{1}{N} \sum_{i=1}^N G(\mathbf{X})^{(i)}.$$

Remark. Numerical experiments have shown that increasing such equally sized partitions beyond 50 classes has negligible effect on the estimation accuracy [41].

Remark. The form of the estimate in (35.8) is very close to the one proposed by McKay [34] (see also [35]) based on permuted column sampling. However, the one described here is more general as it allows using any sampling design.

4.2 Spectral Approaches

In case the model G has some regularity, one might prefer spectral approaches rather than Monte Carlo approaches to estimate Sobol' indices.

4.2.1 Notation, Definitions

For this purpose, consider, for all $j = 1, \dots, d$, an orthonormal Hilbert basis of $\mathbb{L}_{\mathbb{R}}^2(P_{X_j})$, denoted by $(\Phi_{j,k}, k \in \mathbb{Z})$ such that $\Phi_{j,0} \equiv 1$.

Remark. Considering that $\mathbb{L}_{\mathbb{R}}^2(P_{X_j}) \subset \mathbb{L}_{\mathbb{C}}^2(P_{X_j})$, one can also replace the basis $(\Phi_{j,k}, k \in \mathbb{Z})$ by a Hilbert basis of $\mathbb{L}_{\mathbb{C}}^2(P_{X_j})$. It will have a practical interest when considering, e.g., the Fourier basis.

Consider the tensorized basis of $\mathbb{L}_{\mathbb{K}}^2(P_{\mathbf{X}})$, $\mathbb{K} = \mathbb{R}$, or \mathbb{C} , whose elements are indexed by $\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{Z}^d$ and defined as the products $\Phi_{1,k_1}(\cdot) \times \dots \times \Phi_{d,k_d}(\cdot)$. $G(\mathbf{X})$ can be decomposed as follows:

$$Y = G(\mathbf{X}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} c_{\mathbf{k}}(G) \Phi_{1,k_1}(X_1) \dots \Phi_{d,k_d}(X_d).$$

Now define

$$\begin{aligned}\mathbb{Z}_{\mathbf{u}, \neq 0} &:= \{\mathbf{k} \in \mathbb{Z}^d \mid \forall j \in \mathbf{u}, k_j \in \mathbb{Z}_{\neq 0} \text{ and } \forall j \notin \mathbf{u}, k_j = 0\} \\ \text{and } (\mathbb{Z}^d)_{\neq 0} &= \{\mathbf{k} \in \mathbb{Z}^d \mid \mathbf{k} \neq (0, \dots, 0)\}.\end{aligned}$$

Thanks to the orthonormality of the bases $(\Phi_{j,k}, k \in \mathbb{Z})$, it is possible to identify the components in the FANOVA decomposition as

$$G_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) = \sum_{\mathbf{k} \in \mathbb{Z}_{\mathbf{u}, \neq 0}} c_{\mathbf{k}}(G) \Phi_{1,k_1}(X_1) \dots \Phi_{d,k_d}(X_d). \quad (35.9)$$

Proposition 1. Assume that $G \in \mathbb{L}_{\mathbb{K}}^2(P_{\mathbf{X}})$ and that $\text{Var}(G(\mathbf{X})) \neq 0$. Then, for any nonempty subset $\mathbf{u} \subset \{1, \dots, d\}$,

$$S_{\mathbf{u}} = \frac{\sum_{\mathbf{k} \in \mathbb{Z}_{\mathbf{u}, \neq 0}} |c_{\mathbf{k}}(G)|^2}{\text{Var}(G(\mathbf{X}))}. \quad (35.10)$$

Proof. Endow the space $\mathbb{L}_{\mathbb{K}}^2(P_{\mathbf{X}})$ with the scalar product associated to $P_{\mathbf{X}}$: $\langle f, g \rangle = \int f(\mathbf{x}) \overline{g(\mathbf{x})} P_{\mathbf{X}}(d\mathbf{x})$. Equality (35.10) is a straightforward consequence of the identification equality (35.9) and of Parseval's identity. \square

Recall that for all $j = 1, \dots, d$, I_j is the support of the probability distribution of X_j . For $n \in \mathbb{Z}_{>0}$, let $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be any finite subset of n points in $I_1 \times \dots \times I_d$. Let $\mathbf{p} = (p_1, \dots, p_n) \in (\mathbb{R}_{\geq 0})^n$ such that $\sum_{i=1}^n p_i = 1$. In the following, integrals in (35.10) are approximated by quadrature formulas associated to the set of nodes D and to the weights \mathbf{p} . $c_{\mathbf{k}}(G) = \langle G, \Phi_{\mathbf{k}} \rangle$ can be approximated by

$$\hat{c}_{\mathbf{k}}(G, D, \mathbf{p}) = \sum_{i=1}^n G(\mathbf{x}_i) \overline{\Phi_{\mathbf{k}}(\mathbf{x}_i)} p_i$$

and $\text{Var}(G(\mathbf{X}))$ by

$$\widehat{V}(D, \mathbf{p}) = \sum_{i=1}^n \left(G(\mathbf{x}_i) - \sum_{j=1}^n G(\mathbf{x}_j) p_j \right)^2 p_i.$$

Let now $K_{\mathbf{u}}$ be any finite subset of $\mathbb{Z}_{\mathbf{u}, \neq 0}$ and define the following intuitive estimator for $S_{\mathbf{u}}$:

$$\widehat{S}_{\mathbf{u}}(K_{\mathbf{u}}, D, \mathbf{p}) = \frac{\sum_{\mathbf{k} \in K_{\mathbf{u}}} \hat{c}_{\mathbf{k}}(G, D, \mathbf{p})}{\widehat{V}(D, \mathbf{p})}. \quad (35.11)$$

Remark. As the infinite set $\mathbb{Z}_{\mathbf{u}, \neq 0}$ is truncated, an important requirement under this way of estimating Sobol' indices is the fast decrease of the coefficients $c_{\mathbf{k}}(G)$. This is why spectral approaches are particularly well suited for regular models G , for which they outperform Monte Carlo-type approaches. However, the choice of the truncation set $K_{\mathbf{u}}$ is very important. Applying a too drastic truncation may lead to nonconvergent estimates.

4.2.2 FAST and RBD Particular Cases

Consider the particular case where $\forall j \in \{1, \dots, d\}$, X_j is uniformly distributed on $[0, 1]$. A possible choice for the basis is then the Fourier basis: $\mathbf{x} \mapsto e^{i\mathbf{k}\cdot\mathbf{x}}$, $\mathbf{k} \in \mathbb{Z}^d$, with $\mathbf{k}\cdot\mathbf{x} = \sum_{j=1}^d k_j x_j$. For any $p \in \mathbb{Z}_{>0}$, let

$$\begin{cases} r_p : [0, 1] \longrightarrow [0, 1] \\ x \longmapsto \begin{cases} 2\{px\} & \text{if } 0 \leq \{px\} < \frac{1}{2} \\ 2 - 2\{px\} & \text{if } \frac{1}{2} \leq \{px\} \leq 1 \end{cases} \end{cases}$$

and for any $\varphi \in [0, 2\pi)$, let

$$\begin{cases} t_{\varphi} : [0, 1] \longrightarrow [0, 1] \\ x \longmapsto \{x + \tilde{\varphi}\} \quad \text{with } \tilde{\varphi} = \frac{1}{4} + \frac{\varphi}{2\pi}, \end{cases}$$

with for any real number z , $\{z\} = z - \lfloor z \rfloor$, the fractional part of $z \in \mathbb{R}$ and $\lfloor \cdot \rfloor$ the floor function.

Then, define the linear operators \mathcal{R}_p and \mathcal{T}_{φ} on $L^2_{\mathbb{C}}([0, 1]^d)$ (see Fig. 35.2 below) such that for all $\mathbf{x} \in [0, 1]^d$,

$$\mathcal{R}_p G(\mathbf{x}) = G(r_p(x_1), \dots, r_p(x_d)) \quad \text{and} \quad \mathcal{T}_{\varphi} G(\mathbf{x}) = G(t_{\varphi_1}(x_1), \dots, t_{\varphi_d}(x_d)).$$

with $\mathcal{R}_p = \underbrace{\mathcal{R}_1 \circ \cdots \circ \mathcal{R}_1}_{p \text{ times}}$.

Lemma 1 (Lemma 1 in [54]). *For any $p \in \mathbb{N}^*$ and any $\varphi \in [0, 2\pi)^d$, the variance decomposition is \mathcal{R}_p and \mathcal{T}_{φ} -invariant on $L^2_{\mathbb{C}}([0, 1]^d)$.*

Proof. The interested reader is referred to Appendix A.3 in [54] for a detailed proof of Lemma 1. \square

Two classical designs of experiments are introduced. For any $\omega \in (\mathbb{Z}_{>0})^d$, denote

$$\mathcal{G}(\omega) = \left\{ \left(\left\{ \frac{i}{n} \omega_1 \right\}, \dots, \left\{ \frac{i}{n} \omega_d \right\} \right), i \in \{0, \dots, n-1\} \right\}.$$

The cyclic subgroup – of order $n/gcd(\omega_1, \dots, \omega_d, n)$ – of the torus $\mathbb{T}^d = (\mathbb{R}/\mathbb{Z})^d \simeq [0, 1]^d$ is generated by $(\{\frac{\omega_1}{n}\}, \dots, \{\frac{\omega_d}{n}\})$.

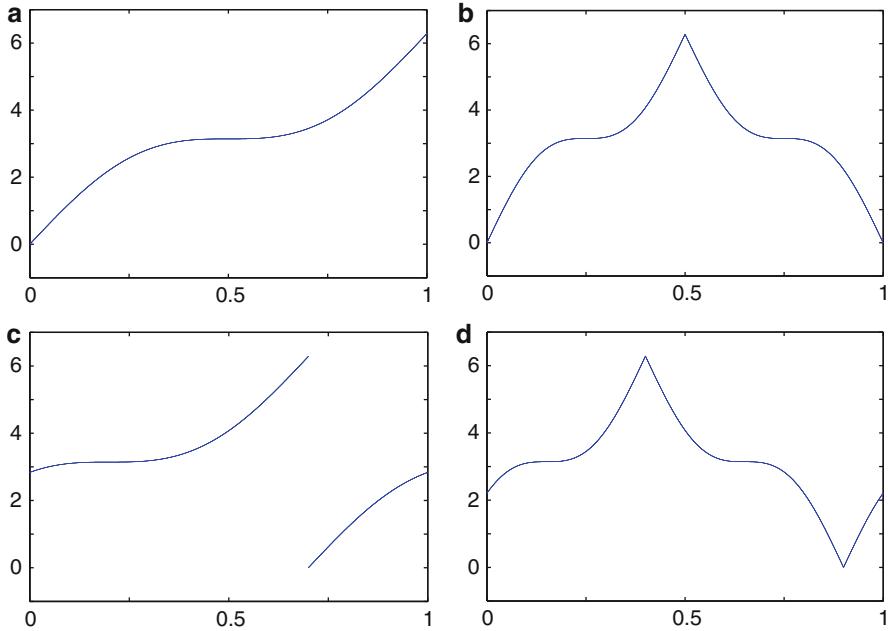


Fig. 35.2 Examples of operators \mathcal{R}_p and \mathcal{T}_φ in dimension 1. (a) Plot of $f : x \mapsto x + \sin(x)$. (b) Plot of $\mathcal{R}_1 f$. (c) Plot of $\mathcal{T}_{\frac{\pi}{10}} f$. (d) Plot of $(\mathcal{T}_{\frac{\pi}{30}} \circ \mathcal{R}_1) f$

Let $\sigma_1, \dots, \sigma_d$ be random permutations on $\{0, \dots, n - 1\}$ and \mathfrak{S} denote the set of all possible $\sigma = (\sigma_1, \dots, \sigma_d)$. For any $\sigma \in \mathfrak{S}$, denote with

$$A(\sigma) = \left\{ \left(\frac{\sigma_1(i)}{n}, \dots, \frac{\sigma_d(i)}{n} \right), i \in \{0, \dots, n - 1\} \right\}$$

the orthogonal array of strength 1 and index unity with elements taken from $\{0, \frac{1}{n}, \dots, \frac{n-1}{n}\}$ and based on the permutation σ .

Let \mathbf{p}_0 be the set of weights defined by $\mathbf{p}_0 = (\underbrace{n^{-1}, \dots, n^{-1}}_{n \text{ times}})$.

Fourier amplitude sensitivity test (FAST) and random balance design (RBD) methods for estimating Sobol' indices are then defined by Definition 3.

Definition 3. For any nonempty subset $\mathbf{u} \subset \{1, \dots, d\}$, and any finite subset $K_{\mathbf{u}} \subset \mathbb{Z}_{\mathbf{u}, \neq 0}$, for any $\varphi \in [0, 2\pi]^d$ and any $\omega \in (\mathbb{Z}_{>0})^d$, one has

$$\hat{S}_{\mathbf{u}}^{\text{FAST}}(G, K_{\mathbf{u}}, \mathbf{x}^*) = \hat{S}_{\mathbf{u}}((\mathcal{T}_\varphi \circ \mathcal{R}_1)G, K_{\mathbf{u}}, \mathcal{G}(\omega), \mathbf{p}_0).$$

For any nonempty subset $\mathbf{u} \subset \{1, \dots, d\}$, and any finite subset $K_{\mathbf{u}} \subset \mathbb{Z}_{\mathbf{u}, \neq 0}$, for any $\sigma \in \mathfrak{S}$ and any $\omega \in \mathbb{Z}_{>0}$, one has

$$\hat{S}_{\mathbf{u}}^{\text{RBD}}(G, K_{\{\mathbf{u}\}}, \mathbf{x}^*) = \hat{S}_{\mathbf{u}}((\mathcal{T}_\omega \circ \mathcal{R}_\omega)G, \omega K_{\{\mathbf{u}\}}, A(\sigma), \mathbf{p}_0)$$

where $\tilde{\omega} = \left(\frac{(1-\omega)\pi}{2\omega}, \dots, \frac{(1-\omega)\pi}{2\omega} \right)$ and $\omega K_{\{u\}} = \{(\omega k_1, \dots, \omega k_d), \mathbf{k} \in K_{\{u\}}\}$.

Remark. Note that the linear operator \mathcal{R}_p “regularizes” the function G in the sense that the coefficients $c_{\mathbf{k}}(\mathcal{R}_p G)$ decrease faster to zero than the coefficients $c_{\mathbf{k}}(G)$ as $\sum_{j=1}^d |k_j|$ tends to infinity. The other operator \mathcal{T}_{φ} essentially allows to define randomized estimators in FAST.

Remark. FAST was introduced in [6–8] and RBD for first-order Sobol’ indices in [50] and for higher-order Sobol’ indices in [57]. Both methods were revisited recently in [54], leading to Definition 3. Under regularity assumptions, the estimators in RBD are asymptotically unbiased.

Remark. A variant of RBD called quasi-random balance design (QRBD) has recently been proposed to increase the precision and accuracy of first-order indices. The method differs from RBD in the way the sample is created. QRBD uses multidimensional quasi-random sequences to construct the permutations which are used to create the realizations of the inputs. A working paper describing the methodology is being prepared by Plischke, Tarantola, and Mara.

4.2.3 Choice of Parameters ω and n in FAST

There exist various criteria for the choice of these parameters. Refer to [54] for a recent review. In [47], the authors propose a criterion which aims at choosing $\omega_1, \dots, \omega_d$ free of interferences up to order $M \in \mathbb{Z}_{>0}$:

$$(\mathbf{k} - \mathbf{k}') \cdot \omega \neq 0 \text{ for all } \mathbf{k}, \mathbf{k}' \in \mathbb{Z}^d, \mathbf{k} \neq \mathbf{k}', \text{ s.t. } \sum_{i=1}^d |k_i - k'_i| \leq M + 1 \quad (35.12)$$

and n sufficiently large

$$n \approx M \max(\omega_1, \dots, \omega_d).$$

More recently, referring to the classical information theory, the authors in [46] suggest to replace the previous formula with Nyquist-Shannon sampling theorem

$$n > 2M \max(\omega_1, \dots, \omega_d).$$

Choosing $K_u = \mathbb{Z}_u \cap (-M, M]^d$, it seems more natural to replace (35.12) by

$$(\mathbf{k} - \mathbf{k}') \cdot \omega \neq 0 \text{ for all } \mathbf{k}, \mathbf{k}' \in \mathbb{Z}^d, \mathbf{k} \neq \mathbf{k}', \text{ s.t. } \sum_{i=1}^d |k_i| \leq M \text{ and } \sum_{i=1}^d |k'_i| \leq M.$$

Remark. In the RBD method, the parameter ω is usually set equal to 1. The RBD approach is known to be biased (even if asymptotically unbiased), and recently a partial correction of the bias was proposed in [53] and generalized in [54].

4.2.4 The EASI Method

An effective algorithm for computing sensitivity indices (EASI) has been proposed in [40] where first-order sensitivity indices are estimated from a set of precomputed model evaluations $(\mathbf{X}^{(i)}, G(\mathbf{X})^{(i)}), i = 1, \dots, N$ using the fast Fourier transform. The article also discusses ideas for the estimation of higher-order sensitivity indices. The sampled values of the j th input $X_j^{(i)}, i = 1, \dots, N$ are ordered increasingly. Then, the sampled values that correspond to all odd indices i are ordered increasingly followed by all even indices in decreasing order. With this sorting and shuffling procedure, one obtains a reordered input that is triangular shaped and can be thought of as a periodic curve with frequency 1. By permuting the output values according to this rule, one looks out for resonances of period 1 and its higher harmonics in the power spectrum of the permuted output using the standard procedure implemented in both FAST and RBD. By repeating the sorting and shuffling of X_j for all other j on the same set of N sampled values, one obtains all the other first-order sensitivity indices. Estimates produced by EASI are biased with respect to the maximal number of harmonics M , defined in Eq. (35.12). A bias correction formula, derived in [23], is proposed in [40].

4.2.5 Polynomial Chaos Expansion

Consider the case where the vector $\mathbf{X} = (X_1, \dots, X_d)$ is d -dimensional with independent components. A pertinent choice for the basis $(\Phi_{j,k}, k \in \mathbb{Z})$ is then a basis of orthogonal polynomials, e.g., the Legendre basis if X_j is uniformly distributed on $[0, 1]$ or the Hermite polynomials if X_j is normally distributed. There exists a wide literature on PC expansions (see, e.g., [1, 49]).

5 Test Functions

5.1 The g -Sobol' Function and Effective Dimension

The first test case is the g -Sobol' function. It is a classical test case, which is parameter dependent. Depending on the choice of the parameter, the effective dimension of this model can be varied. Moreover, an analytical expression for Sobol' indices is available.

5.1.1 Model

$Y = f_1(X_1) \times \dots \times f_d(X_d)$ with $(X_1, \dots, X_d) \sim \mathcal{U}([0, 1]^d)$ and

$$f_j(X_j) = \frac{|4X_j - 2| + a_j}{1 + a_j}, \quad a_j \geq 0, \quad j = 1, \dots, d.$$

The example provided in [43] with $a = (0, 0.5, 3, 9, 99, 99)$ has been chosen in dimension $d = 6$.

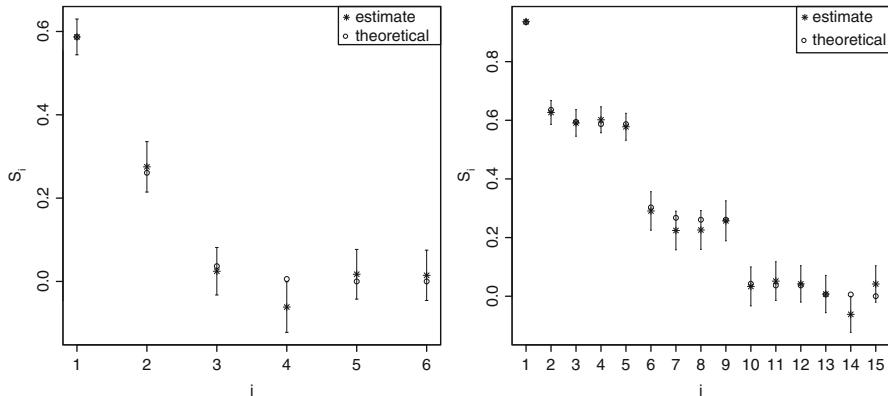


Fig. 35.3 Estimation of (left) first-order and (right) closed second-order indices for the g-Sobol' function with $a = (0, 0.5, 3, 9, 99, 99)$, using replicated designs and formulas (35.7b) and (35.7c), with replicated LHS (left) of level 1024 and replicated randomized orthogonal arrays (right) of level 31. The confidence bounds are provided by the asymptotic central limit theorem stated in [18]

The estimation of first-order Sobol' indices is given below using replicated designs [32, 55]. Closed second-order Sobol' indices are also provided using the extension of formulas (35.7b) and (35.7c) given in [18].

$N = 1024$ has been chosen. The number of model evaluations for the estimation of all d first-order Sobol' indices is $N_T = 2 \times N = 2048$, and the cost to estimate all closed second-order Sobol' indices is $N'_T = 2 \times 31^2 = 1922$. This number is explained hereafter. Replicated designs for the estimation of closed second-order indices are based on the construction of an orthogonal array of strength 2. To construct this orthogonal array, a prime number has to be chosen for the level. The largest prime number p such that $p^2 \leq 1024$ was chosen; thus, $p = 31$.

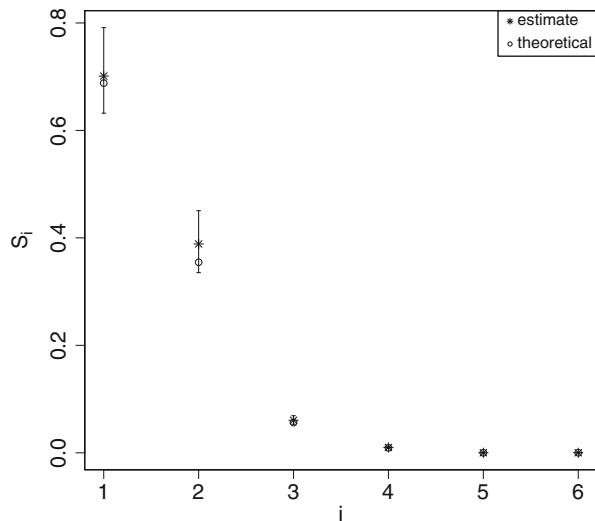
The results are presented in Fig. 35.3. Closed second-order Sobol' indices are labeled from 1 to 15 from $S_{\{1,2\}}^{\text{clo}}$ to $S_{\{5,6\}}^{\text{clo}}$.

Note that the confidence bounds are small enough at this sample size. The estimation is better for influential factors. However, the replicated approach does not allow one to estimate total indices. If needed, and if a budget for that is available, one can use the estimator defined by formula (35.7d) and initially derived by [20]. The results are presented in Fig. 35.4. Again, $N = 1024$ has been chosen; thus, the number of model evaluations which are required for the estimation of all d total Sobol' indices is $N_T = (d + 2) \times N = 8192$.

If one is interested in estimating first-order, second-order and total sensitivity indices altogether, the combinatoric trick proposed by [43] is suggested, as it requires $N_T = N(d + 2)$ model evaluations (see Theorem 1 in [43]).

The numerical tests above have been performed by using the package `sensitivity` [52] available on CRAN.

Fig. 35.4 Estimation of total Sobol' indices for the g -Sobol' function with $a = (0, 0.5, 3, 9, 99, 99)$, using random sampling and formula (35.7d). The confidence bounds are computed with $B = 100$ bootstrap replications. The number of model evaluations is $N = (d + 2) \times 1024 = 8192$



5.2 A Discontinuous Test Function

The test function in d dimensions is defined as follows:

$$G(\mathbf{X}) = \sum_{i=1}^d X_i - \gamma/2$$

where

$$\begin{aligned}\gamma &= 1 \text{ if } \exists X_i : 0 \leq X_i < 1/2, \\ \gamma &= 2 \text{ if } \exists i \neq j \exists X_i, X_j : \{0 \leq X_i < 1/2, 0 \leq X_j < 1/2\}, \\ \gamma &= 3 \text{ if } \exists i, j, k \text{ distinct } \exists X_i, X_j, X_k : \{0 \leq X_i < 1/2, 0 \leq X_j < 1/2, 0 \leq X_k < 1/2\},\end{aligned}$$

and so on until $\gamma = d$.

Table 35.1 contains the results of applying the Sobol' method and all the spectral methods described in this chapter, for the estimation of first-order indices to the discontinuous function defined above in two dimensions. Table 35.1 shows quite interesting results. Firstly, the Sobol' method provides very accurate estimates even at $N = 32$, although the precision of such estimates (expressed by the standard deviation over the 100 replicates) is quite weak and becomes comparable to that of the other techniques only at $N = 512$. Secondly, RBD is biased at small sample size, although this bias tends to disappear as N increases. At small N , the de-biased version of RBD is very accurate, but its precision is lower than that of RBD. QRBD is both accurate and precise even at small N . However, its de-biased version underestimates the true values. This is because the assumption of white noise

Table 35.1 First-order indices obtained by the method of Sobol'; de-biased and non-de-biased versions of RBD, QRBD, and EASI; as well as the definition of first-order index with "no specific design." The tests have been made on the discontinuous function, defined above, in the case of two inputs. The analytic first-order indices are 0.5 and 0.5, for the symmetric nature of the test function. N indicates the total sample size. The analysis is replicated 100 times. In each cell, the average sensitivity indices and the standard deviations are shown. The "no specific design" approach uses simple random sample with 16 equally sized bins. In these tests, 16 bins allow the estimation of sensitivity indices only for sample sizes 128, 256, and 512. At lower sample size, estimations are not possible due to few empty bins

N	32	64	128	256	512
Sobol'	.48 .50 ± .40	.49 .48 ± .21	.47 .49 ± .13	.49 .50 ± .08	.49 .49 ± .04
RBD	.68 .67 ± .18	.58 .58 ± .12	.54 .53 ± .08	.51 .51 ± .06	.51 .51 ± .04
RBD de-biased	.51 .53 ± .26	.49 .50 ± .15	.49 .49 ± .10	.49 .49 ± .06	.50 .49 ± .04
QRBD	.56 .56 ± .04	.52 .52 ± .03	.50 .50 ± .01	.50 .50 ± .01	.49 .49 ± .00
QRBD de-biased	.28 .28 ± .07	.42 .41 ± .04	.45 .44 ± .01	.47 .47 ± .01	.48 .48 ± .00
EASI	.68 .67 ± .09	.58 .57 ± .06	.54 .54 ± .05	.51 .51 ± .03	.50 .50 ± .02
EASI de-biased	.47 .48 ± .16	.48 .48 ± .07	.50 .50 ± .05	.50 .50 ± .03	.49 .49 ± .02
No specific design [41]	—	—	.56 .56 ± .05	.53 .52 ± .04	.51 .51 ± .03

Table 35.2 First-order indices obtained using the de-noised version of the EASI method for different values of frequency truncation; see Equation (35.12). The tests have been made on the discontinuous function, defined above, in the case of two inputs. Each analytic first-order index is 0.5, due to the symmetric nature of the test function. The total sample size $N = 10,000$ is chosen sufficiently large so as to minimize sampling error and highlight the underestimation error due to the truncation of the Fourier coefficients. The analysis is replicated 100 times. In each cell, the average sensitivity indices over the 100 replicates are shown. The standard deviations across the 100 replicates are all equal to 0.005. As one can see, the negative bias is progressively reduced as the truncation parameter increases

Truncation	S_1	S_2
4	.4870	.4865
6	.4925	.4915
8	.4942	.4942
10	.4951	.4953
20	.4978	.4981
50	.4985	.4987
100	.4993	.4998

underlying the de-biasing algorithm does not hold for quasi-random permutations. Therefore, it is recommended not to use this procedure. Finally, the EASI technique, which does not require a specific design but works for any given input dataset, is very precise even at $N = 32$, although less accurate. However, at large sample size, its estimates are very accurate. The de-biased version of EASI proves very effective in both accuracy and precision (Table 35.1).

6 Conclusions

This section provides an overview of variance-based sensitivity measures. Conceptually, the variance of the output of a computer model is decomposed into parts (called partial variances) representing the fraction of the output's variance that is accounted for by each input to the model. Such partial variances provide a measure of relative importance of the model inputs, either taken singularly or considered by groups. The section introduces first-order, higher-order, and total sensitivity measures. Of these, the analyst is generally interested in first-order, second-order, and total indices, as higher-order indices require too many runs of the model and hence too much computational time. The section splits the available estimation techniques in two classes, those based on Monte Carlo and those based on spectral approaches. After illustrating their properties, the described techniques are tested on few functions for which the true sensitivity measures can be calculated analytically, so as to compare their relative performances. Monte Carlo-type approaches are particularly flexible in the sense that no regularity assumption on the model is required. For first- and second-order indices, replicated designs seem particularly well suited. However, if one wishes to compute total indices, one has to pay a price which is linear in the dimension, and the recommended estimator is given by Saltelli [43] which allows to get simultaneously first-order, second-order, and total indices. However, in case the model has some regularity, spectral approaches perform very well and particularly the EASI approach.

Scalar (and independent sets of) inputs as well as univariate outputs have been considered in the present section. Sometimes, this is not the case.

- Various approaches have been proposed to handle the case of correlated inputs. The interested reader is referred to [3–5, 26, 29, 33] (see also **Introduction** of the present chapter).
- In many instances, one has to deal with multivariate outputs. Two interesting approaches are described in [13, 27]. The reader can also refer to [14] for an overview of different approaches to the analysis of multivariate output.
- It is also possible to generalize global sensitivity analysis to computer codes with functional inputs. In [17, 30], or [10], the authors propose approaches which result in one sensitivity index for each functional input as a whole. More recently, the authors in [12] proposed a method which provides insight into the sensitivity with respect to changes at specific intervals of the functional domain (see also the paper ► [Chap. 39, “Sensitivity Analysis of Spatial and/or Temporal Phenomena”](#) of the present chapter which presents a review on sensitivity analysis for functional inputs and/or functional outputs).

References

1. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys.* **230**(6), 2345–2367 (2011)

2. Caflisch, R., Morokoff, W., Owen, A.: Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *J. Comput. Fin.* **1**(1), 27–46 (1997)
3. Champion, M., Chastaing, G., Gadat, S., Prieur, C.: L2-Boosting for sensitivity analysis with dependent inputs. *Stat. Sin.* **25**(4), (2015). doi:[10.5705/ss.2013.310](https://doi.org/10.5705/ss.2013.310)
4. Chastaing, G., Gamboa, F., Prieur, C.: Generalized Hoeffding-Sobol decomposition for dependent variables. Application to sensitivity analysis. *Electron. J. Stat.* **6**, 2420–2448 (2012)
5. Chastaing, G., Gamboa, F., Prieur, C.: Generalized Sobol sensitivity indices for dependent variables: numerical methods. *J. Stat. Comput. Simul.* (ahead-of-print), 1–28 (2014)
6. Cukier, H., Fortuin, C.M., Shuler, K., Petschek, A.G., Schaibly, J.H.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients: theory. *J. Chem. Phys.* **59**, 3873–3878 (1973)
7. Cukier, H., Levine, R.I., Shuler, K.: Nonlinear sensitivity analysis of multiparameter model systems. *J. Comput. Phys.* **26**, 1–42 (1978)
8. Cukier, H., Schaibly, J.H., Shuler, K.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients: analysis of the approximations. *J. Chem. Phys.* **63**, 1140–1149 (1975)
9. Efron, B., Stein, C.: The jackknife estimate of variance. *Ann. Stat.* **9**(3), 586–596 (1981)
10. Fort, J.-C., Klein, T., Lagnoux, A., Laurent, B.: Estimation of the Sobol indices in a linear functional multidimensional model. *J. Stat. Plan. Inference* **143**(9), 1590–1605 (2013)
11. Fruth, J., Roustant, O., Kuhnt, S.: Total interaction index: a variance-based sensitivity index for second-order interaction screening. *J. Stat. Plann. Inference* **147**, 212–223 (2014)
12. Fruth, J., Roustant, O., Kuhnt, S.: Sequential designs for sensitivity analysis of functional inputs in computer experiments. *Reliab. Eng. Syst. Saf.* **134**, 260–267 (2015)
13. Gamboa, F., Janon, A., Klein, T., Lagnoux, A.: Sensitivity analysis for multidimensional and functional outputs. *Electron. J. Stat.* **8**(1), 575–603 (2014)
14. Garcia-Cabrejo, O., Valocchi, A.: Global sensitivity analysis for multivariate output using polynomial chaos expansion. *Reliab. Eng. Syst. Saf.* **126**, 25–36 (2014)
15. Homma, T., Saltelli, A.: Importance measures in global sensitivity analysis of nonlinear models. *Reliab. Eng. Syst. Saf.* **52**(1), 1–17 (1996)
16. Hooker, G.: Discovering additive structure in black box functions. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’04), Seattle, pp. 575–580. ACM (2004)
17. Iooss, B., Ribatet, M.: Global sensitivity analysis of computer models with functional inputs. *Reliab. Eng. Syst. Saf.* **94**(7), 1194–1204 (2009)
18. Janon, A., Klein, T., Lagnoux-Renaudie, A., Nodet, M., Prieur, C.: Asymptotic normality and efficiency of two Sobol index estimators. *ESAIM: Probab. Stat.* **18**, 342–364 (2014)
19. Jansen, M.J.W.: Analysis of variance designs for model output. *Comput. Phys. Commun.* **117**(1), 35–43 (1999)
20. Jansen, M.J.W., Rossing, W.A.H., Daamen, R.A.: Monte carlo estimation of uncertainty contributions from several independent multivariate sources. In: Gasman, J., van Straten, G. (eds.) *Predictability and Nonlinear Modelling in Natural Sciences and Economics*, pp. 334–343. Kluwer Academic, Dordrecht (1994)
21. Joe, S., Kuo, F.Y.: Remark on algorithm 659: Implementing Sobol’s quasirandom sequence generator. *ACM Trans. Math. Softw.* **29**(1), 49–57 (2003)
22. Joint Research Centre of the European Commission: Sensitivity analysis software. <http://ipsc.jrc.ec.europa.eu/index.php?id=756> (2014)
23. Kelley, T.L.: An unbiased correlation ratio measure. *Proc. Natl. Acad. Sci. U. S. A.* **21**(9), 554–559 (1935)
24. Kishen, K.: On latin and hyper-graeco cubes and hypercubes. *Curr. Sci.* **11**(3), 98–99 (1942)
25. Kucherenko, S., Balazs, F., Nilay, S., Mauntz, W.: The identification of model effective dimensions using global sensitivity analysis. *Reliab. Eng. Syst. Saf.* **96**, 440–449 (2011)
26. Kucherenko, S., Tarantola, S., Annoni, P.: Estimation of global sensitivity indices for models with dependent variables. *Comput. Phys. Commun.* **183**(4), 937–946 (2012)

27. Lamboni, M., Monod, H., Makowski, D.: Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliab. Eng. Syst. Saf.* **96**(4), 450–459 (2011)
28. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer Series in Statistics. Springer, New York (2009)
29. Li, G., Rabitz, H.: General formulation of HDMR component functions with independent and correlated variables. *J. Math. Chem.* **50**(1), 99–130 (2012)
30. Lilburne, L., Tarantola, S.: Sensitivity analysis of spatial models. *Int. J. Geogr. Inf. Sci.* **23**(2), 151–168 (2009)
31. Liu, R., Owen, A.: Estimating mean dimensionality of analysis of variance decompositions. *J. Am. Stat. Assoc.* **101**(474), 712–721 (2006)
32. Mara, T.A., Joseph, O.R.: Comparison of some efficient methods to evaluate the main effect of computer model factors. *J. Stat. Comput. Simul.* **78**, 167–178 (2008)
33. Mara, T.A., Tarantola, S.: Variance-based sensitivity indices for models with dependent inputs. *Reliab. Eng. Syst. Saf.* **107**, 115–121 (2012)
34. McKay, M.D.: Evaluating prediction uncertainty. Technical Report NUREG/CR-6311, US Nuclear Regulatory Commission and Los Alamos National Laboratory, pp. 1–79 (1995)
35. Moore, L.M., Morris, M.D., McKay, M.D.: Using orthogonal arrays in the sensitivity analysis of computer models. *Technometrics* **50**, 205–215 (2008)
36. Monod, H., Naud, C., Makowski, D.: Uncertainty and sensitivity analysis for crop models. In: Wallach, D., Makowski, D., Jones, J.W. (eds.) Working with Dynamic Crop Models: Evaluation, Analysis, Parameterization, Applications, chap. 4, pp. 55–99. Elsevier (2006)
37. Owen, A.B.: Orthogonal arrays for computer experiments, integration and visualization. *Stat. Sin.* **2**, 439–452 (1992)
38. Owen, A.B.: Better estimation of small Sobol' sensitivity indices. *ACM Trans. Model. Comput. Simul.* **23**, 11–17 (2013)
39. Pearson, K.: Mathematical contributions to the theory of evolution. *Proc. R. Soc. Lond.* **71**, 288–313 (1903)
40. Plischke, E.: An effective algorithm for computing global sensitivity indices (easi). *Reliab. Eng. Syst. Saf.* **95**, 354–360 (2010)
41. Plischke, E., Borgonovo, E., Smith, C.L.: Global sensitivity measures from given data. *Eur. J. Oper. Res.* **226**, 536–550 (2013)
42. Qian, P.Z.G.: Nested Latin hypercube designs. *Biometrika* **96**(4), 957–970 (2009)
43. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002)
44. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**(2), 259–270 (2010)
45. Saltelli, A., Tarantola, S., Campolongo, F.: Sensitivity analysis as an ingredient of modeling. *Stat. Sci.* **15**(4), 377–395 (2000)
46. Saltelli, A., Tarantola, S., Chan, K.: A quantitative, model-independent method for global sensitivity analysis of model output. *Technometrics* **41**, 39–56 (1999)
47. Schaibly, J.H., Shuler, K.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients: applications. *J. Chem. Phys.* **59**, 3879–3888 (1973)
48. Sobol', I.M.: Sensitivity analysis for nonlinear mathematical models. *Math. Model. Comput. Exp.* **1**, 407–414, 1993.
49. Sudret, B.: Global sensitivity analysis using polynomial chaos expansion. *Reliab. Eng. Syst. Saf.* **93**, 964–979 (2008)
50. Tarantola, S., Gatelli, D., Mara, T.A.: Random balance designs for the estimation of first-order global sensitivity indices. *Reliab. Eng. Syst. Saf.* **91**, 717–727 (2006)
51. The Comprehensive R Archive Network: DiceDesign package. <http://cran.r-project.org/web/packages/DiceDesign/index.html>
52. The Comprehensive R Archive Network: Sensitivity package. <http://cran.r-project.org/web/packages/sensitivity/>

53. Tissot, J.Y., Prieur, C.: A bias correction method for the estimation of sensitivity indices based on random balance designs. *Reliab. Eng. Syst. Saf.* **107**, 205–213 (2012)
54. Tissot, J.Y., Prieur, C.: Variance-based sensitivity analysis using harmonic analysis. Technical report (2012) <http://hal.archives-ouvertes.fr/hal-00680725>.
55. Tissot, J.Y., Prieur, C.: A randomized orthogonal array-based procedure for the estimation of first- and second-order Sobol' indices. *J. Stat. Comput. Simul.* **85**(7), 1358–1381 (2015)
56. Wang, X., Fang, K.-T.: The effective dimension and quasi-Monte Carlo integration. *J. Complex.* **19**(2), 101–124 (2003)
57. Xu, C., Gertner, G.: Understanding and comparisons of different sampling approaches for the Fourier amplitudes sensitivity test (fast). *Comput. Stat. Data Anal.* **55**(1), 184–198 (2011)

Sergey Kucherenko and Bertrand Iooss

Abstract

The method of derivative-based global sensitivity measures (DGSM) has recently become popular among practitioners. It has a strong link with the Morris screening method and Sobol' sensitivity indices and has several advantages over them. DGSM are very easy to implement and evaluate numerically. The computational time required for numerical evaluation of DGSM is generally much lower than that for estimation of Sobol' sensitivity indices. This paper presents a survey of recent advances in DGSM concerning lower and upper bounds on the values of Sobol' total sensitivity indices S_i^{tot} . Using these bounds it is possible in most cases to get a good practical estimation of the values of S_i^{tot} . Several examples are used to illustrate an application of DGSM.

Keywords

Sensitivity analysis • Sobol' indices • Morris method • Model derivatives • DGSM • Poincaré inequality

Contents

1	Introduction	1242
2	From Morris Method to DGSM	1244
2.1	Basics of the Morris Method	1244
2.2	The Local Sensitivity Measure	1245

S. Kucherenko (✉)

Department of Chemical Engineering, Imperial College London, London, UK
e-mail: s.kucherenko@imperial.ac.uk

B. Iooss

Industrial Risk Management Department, EDF R&D, Chatou, France

Institut de Mathématiques de Toulouse, Université Paul Sabatier, Toulouse, France
e-mail: bertrand.iooss@edf.fr, biooss@yahoo.fr

2.3	DGSM for Uniformly Distributed Variables	1245
2.4	DGSM for Randomly Distributed Variables	1245
3	Sobol' Global Sensitivity Indices	1246
3.1	Definitions	1246
3.2	Useful Relationships	1247
3.3	A First Direct Link Between Total Sobol' Sensitivity Indices and Partial Derivatives	1248
4	DGSM-Based Bounds for Uniformly and Normally Distributed Variables	1249
4.1	Uniformly Distributed Variables	1249
4.2	Normally Distributed Variables	1251
5	DGSM-Based Bounds for Groups of Variables	1253
5.1	Importance Criterion τ_i	1254
5.2	Normally Distributed Random Variables	1254
6	DGSM-Based Upper Bounds in the General Case	1255
7	Computational Costs	1257
8	Test Cases	1258
9	Conclusions	1261
	References	1261

1 Introduction

Global sensitivity analysis (SA) offers a comprehensive approach to the model analysis. Unlike local SA, global SA methods evaluate the effect of a factor while all other factors are varied as well, and thus they account for interactions between variables and do not depend on the choice of a nominal point. Reviews of different global SA methods can be found in [30] and [37]. The method of global sensitivity indices suggested by [33, 34] and then further developed by [11] is one of the most efficient and popular global SA techniques. It belongs to the class of variance-based methods. These methods provide information on the importance of different subsets of input variables to the output variance. There are two types of Sobol' sensitivity indices: the main effect indices, which estimate the individual contribution of each input parameter to the output variance, and the total sensitivity indices, which measure the total contribution of a single input factor or a group of inputs. The total sensitivity indices are used to identify non-important variables which can then be fixed at their nominal values to reduce model complexity. This approach is known as “factors' fixing setting” [30]. For high-dimensional models, the direct application of variance-based global SA measures can be extremely time-consuming and impractical.

A number of alternative SA techniques have been proposed. One of them is the screening method by [21]. It can be regarded as global as the final measure is obtained by averaging local measures (the elementary effects). This method is considerably cheaper than the variance-based methods in terms of computational time. The Morris method can be used for identifying unimportant variables. However, the Morris method has two main drawbacks. Firstly, it uses random sampling of points from the fixed grid (levels) for averaging elementary effects which are calculated as finite differences with the increment delta comparable with the range of uncertainty. For this reason it cannot correctly account for the effects

with characteristic dimensions much less than delta. Secondly, it lacks the ability of the Sobol' method to provide information about main effects (contribution of individual variables to uncertainty), and it cannot distinguish between low- and high-order interactions.

This paper presents a survey of derivative-based global sensitivity measures (DGSM) and their link with Sobol' sensitivity indices. DGSM are based on averaging local derivatives using Monte Carlo or quasi-Monte Carlo sampling methods. This technique is much more accurate than the Morris method as the elementary effects are evaluated as strict local derivatives with small increments compared to the variable uncertainty ranges. Local derivatives are evaluated at randomly or quasi-randomly selected points in the whole range of uncertainty and not at the points from a fixed grid.

The so-called alternative global sensitivity estimator defined as a normalized integral of partial derivatives was firstly introduced by [36]. Kucherenko et al. [17] introduced some other DGSM and coined the acronym DGSM. They showed that DGSM can be seen as the generalization of the Morris method [21]. Kucherenko et al. [17] also established empirically the link between DGSM and Sobol' sensitivity indices. They showed that the computational cost of numerical evaluation of DGSM can be much lower than that for estimation of Sobol' sensitivity indices.

Sobol and Kucherenko [38] proved theoretically that, in the cases of uniformly and normally distributed input variables, there is a link between DGSM and the Sobol' total sensitivity index S_i^{tot} for the same input. They showed that DGSM can be used as an upper bound on total sensitivity index S_i^{tot} . Small values of DGSM imply small S_i^{tot} and hence unessential factors x_i . However, ranking influential factors using DGSM can be similar to that based on S_i^{tot} only for the case of linear and quasi-linear models. For highly nonlinear models, two rankings can be very different. They also introduced modified DGSM which can be used for both a single input and groups of inputs [39]. From DGSM, [16] have also derived lower bounds on total sensitivity index. Lamboni et al. [19] extended results of Sobol' and Kucherenko for models with input variables belonging to the general class of continuous probability distributions. In the same framework, [28] have defined crossed-DGSM, based on second-order derivatives of model output, in order to bound the total Sobol' indices of an interaction between two inputs.

All these DGSM can be applied for problems with a high number of input variables to reduce the computational time. Indeed, the numerical efficiency of the DGSM method can be improved by using the automatic differentiation algorithm for calculation DGSM as was shown in [15]. However, the number of required function evaluations still remains to be proportional to the number of inputs. This dependence can be greatly reduced using an approach based on algorithmic differentiation in the adjoint or reverse mode [9] (see ▶ Chap. 32, “Variational Methods”). It allows estimating all derivatives at a cost at most 4–6 times of that for evaluating the original function [13].

This paper is organized as follows: the Morris method and DGSM are firstly described in the following section. Sobol' global sensitivity indices and useful relationships are then introduced. Therefore, DGSM-based lower and upper bounds

on total Sobol' sensitivity indices for uniformly and normally distributed random variables are presented, followed by DGSM for groups of variables and their link with total Sobol' sensitivity indices. Another section presents the upper bound results in the general case of variables with continuous probability distributions. Then, computational costs are considered, followed by some test cases which illustrate an application of DGSM and their links with total Sobol' sensitivity indices. Finally, conclusions are presented in the last section.

2 From Morris Method to DGSM

2.1 Basics of the Morris Method

The Morris method is traditionally used as a screening method for problems with a high number of variables for which function evaluations can be CPU time-consuming (see ▶ Chap. 33, “Design of Experiments for Screening”). It is composed of individually randomized “one-factor-at-a-time” (OAT) experiments. Each input factor may assume a discrete number of values, called levels, which are chosen within the factor range of variation.

The sensitivity measures proposed in the original work of [21] are based on what is called an elementary effect. It is defined as follows. The range of each input variable is divided into p levels. Then the elementary effect (incremental ratio) of the i -th input factor is defined as

$$EE_i(\mathbf{x}^*) = \frac{[G(x_1^*, \dots, x_{i-1}^*, x_i^* + \Delta, x_{i+1}^*, \dots, x_d^*) - G(\mathbf{x}^*)]}{\Delta}, \quad (36.1)$$

where Δ is a predetermined multiple of $1/(p-1)$ and point $\mathbf{x}^* = (x_1^*, \dots, x_d^*) \in H^d$ is such that $x_i^* + \Delta \leq 1$. One can see that the elementary effects are finite difference approximations of the model derivative with respect to x_i and using a large perturbation step Δ .

The distribution of elementary effects EE_i is obtained by randomly sampling R points from H^d . Two sensitivity measures are evaluated for each factor: μ_i an estimate of the mean of the distribution EE_i , and σ_i an estimate of the standard deviation of EE_i . A high value of μ_i indicates an input variable with an important overall influence on the output. A high value of σ_i indicates a factor involved in interaction with other factors or whose effect is nonlinear. The computational cost of the Morris method is $N_F = R(d+1)$.

The revised version of the $EE_i(\mathbf{x}^*)$ measure and a more effective sampling strategy, which allows a better exploration of the space of the uncertain input factors, were proposed by [3]. To avoid the canceling effect which appears in non-monotonic functions, [3] introduced another sensitivity measure μ_i^* based on the absolute value of $EE_i(\mathbf{x}^*)$: $|EE_i(\mathbf{x}^*)|$. It was also noticed that μ_i^* has similarities with the total sensitivity index S_i^{tot} in that it can give a ranking of the variables similar to that based on the S_i^{tot} , but no formal proof of the link between μ_i^* and S_i^{tot} was given [3].

Finally, other extensions of the initial Morris method have been introduced for the second-order effects' analysis [2, 4, 6], for the estimation of Morris measures with any type of design [26, 32], and for building some 3D Morris graph [26].

2.2 The Local Sensitivity Measure

Consider a differentiable function $G(\mathbf{x})$, where $\mathbf{x} = (x_1, \dots, x_d)$ is a vector of input variables defined in the unit hypercube H^d ($0 \leq x_i \leq 1$, $i = 1, \dots, d$). Local sensitivity measures are based on partial derivatives

$$E_i(\mathbf{x}^*) = \frac{\partial G(\mathbf{x}^*)}{\partial x_i}. \quad (36.2)$$

This measure E_i is the limit version of the elementary effect EE_i defined in (36.2) when Δ tends to zero. It is its generalization in this sense. In SA, using the partial derivative $\partial G / \partial x_i$ is well known as a local method (see ▶ Chap. 32, “Variational Methods”). In this paper, the goal is to take advantage of this information in global SA.

The local sensitivity measure $E_i(\mathbf{x}^*)$ depends on a nominal point \mathbf{x}^* , and it changes with a change of \mathbf{x}^* . This deficiency can be overcome by averaging $E_i(\mathbf{x}^*)$ over the parameter space H^d . This is done just below, allowing to define new sensitivity measures, called DGSM for derivative-based global sensitivity measures.

2.3 DGSM for Uniformly Distributed Variables

Assume that $\partial G / \partial x_i \in L_2$. Three different DGSM are defined:

$$v_i = \int_{H^d} \left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 d\mathbf{x}, \quad (36.3)$$

$$w_i^{(m)} = \int_{H^d} x_i^m \frac{\partial G(\mathbf{x})}{\partial x_i} d\mathbf{x}, \quad (36.4)$$

where $m > 0$ is a constant, and

$$\varsigma_i = \frac{1}{2} \int_{H^d} x_i(1 - x_i) \left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 d\mathbf{x}. \quad (36.5)$$

2.4 DGSM for Randomly Distributed Variables

Consider a function $G(X_1, \dots, X_d)$, where X_1, \dots, X_d are independent random variables, defined in the Euclidian space R^d , with cumulative density functions (cdfs) $F_1(x_1), \dots, F_d(x_d)$. The following DGSM was introduced in [38]:

$$v_i = \int_{R^d} \left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 dF(\mathbf{x}) = \mathbb{E} \left[\left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 \right], \quad (36.6)$$

with F the joint cdf. A new measure is also introduced:

$$w_i = \int_{R^d} \frac{\partial G(\mathbf{x})}{\partial x_i} dF(\mathbf{x}) = \mathbb{E} \left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right). \quad (36.7)$$

In (36.3) and (36.6), v_i is in fact the mean value of $(\partial G / \partial x_i)^2$. In the following and in practice, it will be the most useful DGSM.

3 Sobol' Global Sensitivity Indices

3.1 Definitions

The method of global sensitivity indices developed by Sobol' (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”) is based on ANOVA decomposition [32, 33]. Consider a square integrable function $G(\mathbf{x})$ defined in the unit hypercube H^d . It can be expanded in the following form:

$$G(\mathbf{x}) = g_0 + \sum_i g_i(x_i) + \sum_{i < j} g_{ij}(x_i, x_j) + \dots + g_{12\dots d}(x_1, x_2, \dots, x_d). \quad (36.8)$$

This decomposition is unique if conditions $\int_0^1 g_{i_1\dots i_s} dx_{i_k} = 0$ for $1 \leq k \leq s$ are satisfied. Here $1 \leq i_1 < \dots < i_s \leq d$.

The variances of the terms in the ANOVA decomposition add up to the total variance of the function

$$V = \sum_{s=1}^d \sum_{i_1 < \dots < i_s} V_{i_1\dots i_s},$$

where $V_{i_1\dots i_s} = \int_0^1 g_{i_1\dots i_s}^2(x_{i_1}, \dots, x_{i_s}) dx_{i_1}, \dots, x_{i_s}$ are called partial variances.

Sobol' defined the global sensitivity indices as the ratios

$$S_{i_1\dots i_s} = V_{i_1\dots i_s} / V.$$

All $S_{i_1\dots i_s}$ are nonnegative and add up to one:

$$\sum_{i=1}^d S_i + \sum_i \sum_j S_{ij} + \sum_i \sum_j \sum_k S_{ijk} \dots + S_{1,2,\dots,d} = 1.$$

Sobol' also defined sensitivity indices for subsets of variables. Consider two complementary subsets of variables y and z :

$$\mathbf{x} = (y, z).$$

Let $y = (x_{i_1}, \dots, x_{i_m})$, $1 \leq i_1 < \dots < i_m \leq d$, $K = (i_1, \dots, i_m)$. The variance corresponding to the set y is defined as

$$V_y = \sum_{s=1}^m \sum_{(i_1 < \dots < i_s) \in K} V_{i_1 \dots i_s}.$$

V_y includes all partial variances V_{i_1} , V_{i_2}, \dots , $V_{i_1 \dots i_s}$ such that their subsets of indices $(i_1, \dots, i_s) \in K$.

The total sensitivity indices were introduced by [11]. The total variance V_y^{tot} is defined as

$$V_y^{\text{tot}} = V - V_z.$$

V_y^{tot} consists of all $V_{i_1 \dots i_s}$ such that at least one index $i_p \in K$ while the remaining indices can belong to the complimentary K set \bar{K} . The corresponding global sensitivity indices are defined as

$$\begin{aligned} S_y &= V_y/V, \\ S_y^{\text{tot}} &= V_y^{\text{tot}}/V. \end{aligned} \quad (36.9)$$

The important indices in practice are S_i and S_i^{tot} , $i = 1, \dots, d$:

$$\begin{aligned} S_i &= V_i/V, \\ S_i^{\text{tot}} &= V_i^{\text{tot}}/V. \end{aligned} \quad (36.10)$$

Their values in most cases provide sufficient information to determine the sensitivity of the analyzed function to individual input variables. Variance-based methods generally require a large number of function evaluations (see Variance-Based Methods: Theory and Algorithms) to achieve reasonable convergence and can become impractical for large engineering problems.

3.2 Useful Relationships

To present further results on lower and upper bounds of S_i^{tot} , new notations and useful relationships have to be firstly presented. Denote $u_i(\mathbf{x})$ the sum of all terms in the ANOVA decomposition (36.8) that depend on x_i :

$$u_i(\mathbf{x}) = g_i(x_i) + \sum_{j=1, j \neq i}^d g_{ij}(x_i, x_j) + \dots + g_{12 \dots d}(x_1, \dots, x_d). \quad (36.11)$$

From the definition of ANOVA decomposition, it follows that

$$\int_{H^d} u_i(\mathbf{x}) d\mathbf{x} = 0. \quad (36.12)$$

It is obvious that

$$\frac{\partial G}{\partial x_i} = \frac{\partial u_i}{\partial x_i}. \quad (36.13)$$

Denote $\mathbf{z} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$ the vector of all variables but x_i , then $\mathbf{x} \equiv (x_i, \mathbf{z})$ and $G(\mathbf{x}) \equiv G(x_i, \mathbf{z})$. The ANOVA decomposition of $G(\mathbf{x})$ (36.8) can be presented in the following form:

$$G(\mathbf{x}) = u_i(x_i, \mathbf{z}) + v(\mathbf{z}),$$

where $v(\mathbf{z})$ is the sum of terms independent of x_i . Because of (36.12) it is easy to show that $v(\mathbf{z}) = \int_0^1 G(\mathbf{x}) dx_i$. Hence

$$u_i(x_i, \mathbf{z}) = G(\mathbf{x}) - \int_0^1 G(\mathbf{x}) dx_i. \quad (36.14)$$

This equation can be found in [18]. The total partial variance V_i^{tot} can be computed as

$$V_i^{\text{tot}} = \int_{H^d} u_i^2(\mathbf{x}) d\mathbf{x} = \int_{H^d} u_i^2(x_i, z) dx_i d\mathbf{z}.$$

Then the total sensitivity index S_i^{tot} (36.10) is equal to

$$S_i^{\text{tot}} = \frac{1}{V} \int_{H^d} u_i^2(\mathbf{x}) d\mathbf{x}. \quad (36.15)$$

3.3 A First Direct Link Between Total Sobol' Sensitivity Indices and Partial Derivatives

Consider continuously differentiable function $G(\mathbf{x})$ defined in the unit hypercube $H^d = [0, 1]^d$. This section presents a theorem that establishes links between the index S_i^{tot} and the limiting values of $|\partial G / \partial x_i|$.

In the case when $\mathbf{y} = (x_i)$, Sobol'-Jansen formula [14, 31, 35] for D_i^{tot} can be rewritten as

$$D_i^{\text{tot}} = \frac{1}{2} \int_{H^d} \int_0^1 \left[G(\mathbf{x}) - G(\overset{\circ}{\mathbf{x}}) \right]^2 d\mathbf{x} dx'_i, \quad (36.16)$$

where $\overset{\circ}{\mathbf{x}} = (x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_n)$.

Theorem 1. Assuming that $c \leq \left| \frac{\partial G}{\partial x_i} \right| \leq C$, then

$$\frac{c^2}{12V} \leq S_i^{\text{tot}} \leq \frac{C^2}{12V}. \quad (36.17)$$

Proof. Consider the increment of $G(\mathbf{x})$ in (36.16):

$$G(\mathbf{x}) - G\left(\overset{\circ}{\mathbf{x}}\right) = \frac{\partial G(\hat{\mathbf{x}})}{\partial x_i} (x_i - x'_i), \quad (36.18)$$

where $\hat{\mathbf{x}}$ is a point between \mathbf{x} and $\overset{\circ}{\mathbf{x}}$. Substituting (36.18) into (36.16) leads to

$$V_i^{\text{tot}} = \frac{1}{2} \int_{H^d} \int_0^1 \left(\frac{\partial G(\hat{\mathbf{x}})}{\partial x_i} \right)^2 (x_i - x'_i)^2 d\mathbf{x} dx'_i. \quad (36.19)$$

In (36.19) $c^2 \leq (\partial G / \partial x_i)^2 \leq C^2$, while the remaining integral is

$$\int_0^1 \int_0^1 (x'_i - x_i)^2 dx'_i dx_i = \frac{1}{6}.$$

Thus obtained inequalities are equivalent to (36.17). Consider the function $G = g_0 + c(x_i - 1/2)$. In this case $C = c$, $V = 1/12$, and $S_i^{\text{tot}} = 1$ and the inequalities in (36.17) become equalities.

4 DGSM-Based Bounds for Uniformly and Normally Distributed Variables

In this section, several theorems are listed in order to define useful lower and upper bounds of the total Sobol' indices. The proofs of these theorems come from previous works and papers and are not recalled here. Two cases are considered: variables \mathbf{x} following uniform distributions and variables \mathbf{x} following Gaussian distributions. The general case will be seen in a subsequent section.

4.1 Uniformly Distributed Variables

4.1.1 Lower Bounds on S_i^{tot}

Theorem 2. There exists the following lower bound between DGSM (36.3) and the Sobol' total sensitivity index:

$$\frac{\left(\int_{H^d} [G(1, \mathbf{z}) - G(0, \mathbf{z})] [G(1, \mathbf{z}) + G(0, \mathbf{z}) - 2G(\mathbf{x})] d\mathbf{x} \right)^2}{4v_i V} < S_i^{\text{tot}} \quad (36.20)$$

Proof. The proof of this theorem is given in [16] and is based on Eq. (36.15) and a Cauchy-Schwarz inequality applied on $\int_{H^d} u_i(\mathbf{x}) \frac{\partial u_i(\mathbf{x})}{\partial x_i} d\mathbf{x}$.

The lower bound number one (LB1) is defined as

$$\frac{\left(\int_{H^d} [G(1, \mathbf{z}) - G(0, \mathbf{z})] [G(1, \mathbf{z}) + G(0, \mathbf{z}) - 2G(\mathbf{x})] d\mathbf{x} \right)^2}{4v_i V}.$$

Theorem 3. *There exists the following lower bound, denoted $\gamma(m)$, between DGSM (36.4) and the Sobol' total sensitivity index:*

$$\gamma(m) = \frac{(2m+1) \left[\int_{H^d} (G(1, \mathbf{z}) - G(\mathbf{x})) d\mathbf{x} - w_i^{(m+1)} \right]^2}{(m+1)^2 V} < S_i^{\text{tot}}. \quad (36.21)$$

Proof. The proof of this theorem is given in [16] and is based on Eq. (36.15) and a Cauchy-Schwarz inequality applied on $\int_{H^d} x_i^m u_i(\mathbf{x}) d\mathbf{x}$.

In fact, Theorem 3 gives a set of lower bounds depending on parameter m . The value of m at which $\gamma(m)$ attains its maximum is of particular interest. Further, star (*) is used to denote such a value m : $m^* = \arg \max(\gamma(m))$. $\gamma(m^*)$ is called the lower bound number two (LB2):

$$\gamma(m^*) = \frac{(2m^*+1) \left[\int_{H^d} (G(1, \mathbf{z}) - G(\mathbf{x})) d\mathbf{x} - w_i^{(m^*+1)} \right]^2}{(m^*+1)^2 V} \quad (36.22)$$

The maximum lower bound LB* is defined as

$$\text{LB}^* = \max(\text{LB1}, \text{LB2}). \quad (36.23)$$

Both lower and upper bounds can be estimated by a set of derivative-based measures:

$$\Upsilon_i = \{v_i, w_i^{(m)}, \xi_i\}, \quad m > 0. \quad (36.24)$$

4.1.2 Upper Bounds on S_i^{tot}

Theorem 4. *There exists the following upper bound between DGSM (36.3) and the Sobol' total sensitivity index:*

$$S_i^{\text{tot}} \leq \frac{v_i}{\pi^2 V}. \quad (36.25)$$

Proof. The proof of this theorem is given in [38]. It is based on inequality

$$\int_0^1 u^2(x) dx \leq \frac{1}{\pi^2} \int_0^1 \left(\frac{\partial u}{\partial x} \right)^2 dx$$

and relationships (36.13) and (36.15).

Consider the set of values v_1, \dots, v_d , $1 \leq i \leq d$. One can expect that smaller v_i correspond to less influential variables x_i . This importance criterion is similar to the modified Morris importance measure μ^* , whose limiting values are

$$\mu_i^* = \int_{H^d} \left| \frac{\partial G(\mathbf{x})}{\partial x_i} \right| d\mathbf{x}.$$

From a practical point of view, the criteria μ_i and v_i are equivalent: they are evaluated by the same numerical algorithm and are linked by relations $v_i \leq C\mu_i$ and $\mu_i \leq \sqrt{v_i}$.

The right term in (36.25) is further called the upper bound number one (UB1).

Theorem 5. *There exists the following upper bound between DGSM (36.5) and the Sobol' total sensitivity index:*

$$S_i^{\text{tot}} \leq \frac{\varsigma_i}{V}. \quad (36.26)$$

Proof. The following inequality [10] is used:

$$0 \leq \int_0^1 u^2 dx - \left(\int_0^1 u dx \right)^2 \leq \frac{1}{2} \int_0^1 x(1-x)u'^2 dx. \quad (36.27)$$

The inequality is reduced to an equality only if u is constant. Assuming that u is given by (36.11), then $\int_0^1 u dx = 0$. From (36.27), Eq. (36.26) is obtained.

Further ς_i/D is called the upper bound number two (UB2). Note that $\frac{1}{2}x_i(1-x_i)$ for $0 \leq x_i \leq 1$ is bounded: $0 \leq \frac{1}{2}x_i(1-x_i) \leq 1/8$. Therefore, $0 \leq \varsigma_i \leq v_i/8$.

4.2 Normally Distributed Variables

4.2.1 Lower Bound on S_i^{tot}

Theorem 6. *If X_i is normally distributed with a mean μ_i and a finite variance σ_i^2 , there exists the following lower bound between DGSM (36.7) and the Sobol' total sensitivity index:*

$$\frac{\sigma_i^4}{(\mu_i^2 + \sigma_i^2)V} w_i^2 \leq S_i^{\text{tot}}. \quad (36.28)$$

Proof. Using Eq. (36.15) and Cauchy-Schwarz inequality applied on $\int_{R^d} x_i u_i(\mathbf{x}) dF(\mathbf{x})$ (with F the joint cdf), [16] give the proof of this inequality when $\mu_i = 0$ (omitting to mention this condition). The general proof, obtained by [25], is given below.

Consider a univariate function $g(X)$, with X a normally distributed variable with mean μ , finite variance σ^2 , and cdf F . With adequate conditions on g , the following equality is obtained by integrating by parts:

$$\begin{aligned}\mathbb{E}[g'(X)] &= \int_{-\infty}^{\infty} g'(x) dF(x) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} g'(x) \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx \\ &= \frac{1}{\sigma \sqrt{2\pi}} \left[g(x) \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \right]_{-\infty}^{+\infty} \\ &\quad + \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \frac{x-\mu}{\sigma^2} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx \\ &= \frac{1}{\sigma^2} \int_{-\infty}^{\infty} x g(x) dF(x) - \mu \int_{-\infty}^{\infty} g(x) dF(x).\end{aligned}$$

In this equation, replacing $g(x)$ by $u_i(\mathbf{x})$ with x_i normally distributed, the w_i DGSM writes

$$w_i = \int_{R^d} \frac{\partial G(\mathbf{x})}{\partial x_i} dF(\mathbf{x}) = \int_{R^d} \frac{\partial u_i(\mathbf{x})}{\partial x_i} dF(\mathbf{x}) = \frac{1}{\sigma_i^2} \int_{R^d} x_i u_i(\mathbf{x}) dF(\mathbf{x}),$$

because $\int_{R^d} u_i(\mathbf{x}) dF(\mathbf{x}) = 0$ (due to the ANOVA decomposition condition). Moreover, the Cauchy-Schwarz inequality applied on $\int_{R^d} x_i u_i(\mathbf{x}) dF(\mathbf{x})$ gives

$$\left[\int_{R^d} x_i u_i(\mathbf{x}) dF(\mathbf{x}) \right]^2 \leq \int_{R^d} x_i^2 dF(\mathbf{x}) \int_{R^d} [u_i(\mathbf{x})]^2 dF(\mathbf{x}).$$

Combining the two latter equations leads to the expression

$$w_i^2 \leq \frac{1}{\sigma_i^4} (\mu_i^2 + \sigma_i^2) VS_i^{\text{tot}},$$

which is equivalent to Eq. (36.28).

4.2.2 Upper Bounds on S_i^{tot}

The following Theorem 7 is a generalization of Theorem 1.

Theorem 7. If X_i has a finite variance σ_i^2 and $c \leq \left| \frac{\partial G}{\partial x_i} \right| \leq C$, then

$$\frac{\sigma_i^2 c^2}{V} \leq S_i^{\text{tot}} \leq \frac{\sigma_i^2 C^2}{V}. \quad (36.29)$$

The constant factor σ_i^2 cannot be improved.

Theorem 8. If X_i is normally distributed with a finite variance σ_i^2 , there exists the following upper bound between DGSM (36.6) and the Sobol' total sensitivity index:

$$S_i^{\text{tot}} \leq \frac{\sigma_i^2}{V} v_i. \quad (36.30)$$

The constant factor σ_i^2 cannot be reduced.

Proof. The proofs of these theorems are presented in [38].

5 DGSM-Based Bounds for Groups of Variables

Let $\mathbf{x} = (x_1, \dots, x_d)$ be a point in the d -dimensional unit hypercube with Lebesgue measure $d\mathbf{x} = dx_1 \cdots dx_d$. Consider an arbitrary subset of the variables $y = (x_{i_1}, \dots, x_{i_s})$, $1 \leq i_1 \leq \dots \leq i_s \leq d$, and the set of remaining complementary variables z , so that $\mathbf{x} = (y, z)$, $d\mathbf{x} = dy dz$. Further all the integrals are written without integration limits, by assuming that each integration variable varies independently from 0 to 1.

Consider the following DGSM τ_y :

$$\tau_y = \sum_{p=1}^s \int \left(\frac{\partial G(\mathbf{x})}{\partial x_{i_p}} \right)^2 \frac{1 - 3x_{i_p} + 3x_{i_p}^2}{6} d\mathbf{x}. \quad (36.31)$$

Theorem 9. If $G(\mathbf{x})$ is linear with respect to x_{i_1}, \dots, x_{i_s} , then $V_y^{\text{tot}} = \tau_y$, or in other words $S_y^{\text{tot}} = \frac{\tau_y}{V}$.

Theorem 10. The following general inequality holds: $V_y^{\text{tot}} \leq (24/\pi^2) \tau_y$, or in other words $S_y^{\text{tot}} \leq \frac{24}{\pi^2 V} \tau_y$.

Proof. The proofs of these theorems are given in [39]. The second theorem shows that small values of τ_y imply small values of S_y^{tot} , and this allows identification of a set of unessential factors y (usually defined by a condition of the type $S_y^{\text{tot}} < \epsilon$, where ϵ is small).

5.1 Importance Criterion τ_i

Considering the one-dimensional case when the subset y consists of only one variable $y = (x_i)$, then measure $\tau_y = \tau_i$ has the form

$$\tau_i = \int \left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 \frac{1 - 3x_i + 3x_i^2}{6} d\mathbf{x}. \quad (36.32)$$

It is easy to show that $v_i/24 \leq \tau_i \leq v_i/6$. From UB1 it follows that

$$S_i^{\text{tot}} \leq \frac{24}{\pi^2 V} \tau_i. \quad (36.33)$$

Thus small values of τ_i imply small values of S_i^{tot} that are characteristic for non-important variables x_i . At the same time, the following corollary is obtained from Theorem 9: if $G(\mathbf{x})$ depends linearly on x_i , then $S_i^{\text{tot}} = \tau_i/V$. Thus τ_i is closer to V_i^{tot} than v_i .

Note that the constant factor $1/\pi^2$ in (36.25) is the best possible. But in the general inequality for τ_i (36.33), the best possible constant factor is unknown.

There is a general link between importance measures τ_i , ς_i , and v_i :

$$\tau_i = -\varsigma_i + \frac{1}{6}v_i,$$

then

$$\varsigma_i = \frac{1}{6}v_i - \tau_i.$$

5.2 Normally Distributed Random Variables

Consider independent normal random variables X_1, \dots, X_d with parameters $(\mu_i, \sigma_i)_{i=1\dots d}$. Define τ_i as

$$\tau_i = \frac{1}{2} \mathbb{E} \left[\left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 (x'_i - x_i)^2 \right].$$

The expectation over x'_i can be computed analytically. Then

$$\tau_i = \frac{1}{2} \mathbb{E} \left[\left(\frac{\partial G(\mathbf{x})}{\partial x_i} \right)^2 \frac{(x_i - \mu_i)^2 + \sigma_i^2}{2} \right].$$

Theorem 11. If X_1, \dots, X_d are independent normal random variables, then for an arbitrary subset y of these variables, the following inequality is obtained:

$$S_y^{\text{tot}} \leq \frac{2}{V} \tau_y.$$

Proof. The proof is given in [39].

6 DGSM-Based Upper Bounds in the General Case

As previously, consider the function $G(X_1, \dots, X_d)$, where X_1, \dots, X_d are independent random variables, defined in the Euclidian space R^d , with cdfs $F_1(x_1), \dots, F_d(x_d)$. Assume further that each X_i admits a probability density function (pdf), denoted by $f_i(x_i)$. In the following, all the integrals are written without integration limits.

The developments in this section are based on the classical L^2 -Poincaré inequality:

$$\int G(\mathbf{x})^2 dF(\mathbf{x}) \leq C(F) \int \|\nabla G(\mathbf{x})\|^2 dF(\mathbf{x}) \quad (36.34)$$

where F is the joint cdf of (X_1, \dots, X_d) . Equation (36.34) is valid for all functions G in $L^2(F)$ such that $\int G(\mathbf{x}) dF(\mathbf{x}) = 0$ and $\|\nabla f\| \in L^2(F)$. The constant $C(F)$ in Eq. (36.34) is called a Poincaré constant of F . In some cases, it exists an optimal Poincaré constant $C_{\text{opt}}(F)$ which is the best possible constant. In measure theory, the Poincaré constants are expressed as a function of so-called Cheeger constants [1] which are used for SA in [19] (see [28] for more details).

A connection between total indices and DGSM has been established by [19] for variables with continuous distributions (called Boltzmann probability measures in their paper).

Theorem 12. Let F_i and f_i be, respectively, the cdf and the pdf of X_i ; the following inequality is obtained:

$$S_i^{\text{tot}} \leq \frac{C(F_i)}{V} v_i, \quad (36.35)$$

with v_i the DGSM defined in Eq. (36.6) and

$$C(F_i) = 4 \left[\sup_{x \in \mathbb{R}} \frac{\min(F_i(x), 1 - F_i(x))}{f_i(x)} \right]^2. \quad (36.36)$$

Proof. This result comes from the direct application of the L^2 -Poincaré inequality (36.34) on $u_i(\mathbf{x})$ (see Eq. (36.11)).

In [19] and [28], the particular case of log-concave probability distribution has been developed. It includes classical distributions as, for instance, the normal, exponential, beta, gamma, and Gumbel distributions. In this case, the constant writes

$$C(F_i) = \frac{1}{f_i(\tilde{m}_i)^2} \quad (36.37)$$

with \tilde{m}_i the median of the distribution F_i . This allows to obtain analytical expressions for $C(F_i)$ in several cases [19]. In the case of a log-concave truncated distribution on $[a, b]$, the constant writes [28]

$$(F_i(b) - F_i(a))^2 / f_i \left(q_i \left(\frac{F_i(a) + F_i(b)}{2} \right) \right)^2 \quad (36.38)$$

with $q_i(\cdot)$ the quantile function of X_i . Table 36.1 gives some examples of Poincaré constants for several well-known and often used probability distributions in practice.

For studying second-order interactions, [28] have derived a similar inequality to (36.35) based on the squared crossed derivatives of the function. Assuming that second-order derivatives of G are in $L^2(F)$, it uses the so-called crossed-DGSM

$$v_{ij} = \int \left(\frac{\partial^2 G(\mathbf{x})}{\partial x_i \partial x_j} \right)^2 dF(\mathbf{x}), \quad (36.39)$$

introduced by [7]. An inequality link is made with an extension of the total Sobol' sensitivity indices to general sets of variables (called superset importance or total interaction index) proposed by [20]. In the case of a pair of variables $\{X_i, X_j\}$, the superset importance is defined as

$$V_{ij}^{\text{super}} = \sum_{I \supseteq \{i, j\}} V_I. \quad (36.40)$$

The estimation methods of this total interaction index have also been studied by [8].

Table 36.1 Poincaré constants for a few probability distributions

Distribution	Poincaré constant	Optimal constant
Uniform $\mathcal{U}[a \ b]$	$(b - a)^2 / \pi^2$	Yes
Normal $\mathcal{N}(\mu, \sigma^2)$	σ^2	Yes
Exponential $\mathcal{E}(\lambda), \lambda > 0$	$\frac{4}{\lambda^2}$	Yes
Gumbel $\mathcal{G}(\mu, \beta)$, scale $\beta > 0$	$\left(\frac{2\beta}{\log 2} \right)^2$	No
Weibull $\mathcal{W}(k, \lambda)$, shape $k \geq 1$, scale $\lambda > 0$	$\left[\frac{2\lambda(\log 2)^{(1-k)/k}}{k} \right]^2$	No

Theorem 13. For all pairs $\{i, j\}$ ($1 \leq i < j \leq d$),

$$V_{ij} \leq V_{ij}^{\text{super}} \leq C(F_i)C(F_j)v_{ij}. \quad (36.41)$$

These inequalities with the corresponding Sobol' indices write

$$S_{ij} \leq S_{ij}^{\text{super}} \leq \frac{C(F_i)C(F_j)}{V}v_{ij}. \quad (36.42)$$

Roustant et al. [28] have shown on several examples how to apply this result in order to detect pairs of inputs that do not interact together (see also [22] and [8] which use Sobol' indices).

7 Computational Costs

All DGSM can be computed using the same set of partial derivatives $\frac{\partial G(\mathbf{x})}{\partial x_i}$, $i = 1, \dots, d$. Evaluation of $\frac{\partial G(\mathbf{x})}{\partial x_i}$ can be done analytically for explicitly given easily differentiable functions or numerically:

$$\frac{\partial G(\mathbf{x}^*)}{\partial x_i} = \frac{[G(x_1^*, \dots, x_{i-1}^*, x_i^* + \delta, x_{i+1}^*, \dots, x_n^*) - G(\mathbf{x}^*)]}{\delta}. \quad (36.43)$$

This is called a finite-difference scheme (see ▶ Chap. 32, “Variational Methods”) with δ which is a small increment. There is a similarity with the elementary effect formula (2) of the Morris method which is however computed with large Δ .

In the case of straightforward numerical estimations of all partial derivatives (36.43) and computation of integrals using MC or QMC methods, the number of required function evaluations for a set of all input variables is equal to $N(d + 1)$, where N is a number of sampled points. Computing LB1 also requires values of $G(0, \mathbf{z}), G(1, \mathbf{z})$, while computing LB2 requires only values of $G(1, \mathbf{z})$. In total, numerical computation of LB* for all input variables would require $N_G^{\text{LB}*} = N(d + 1) + 2Nd = N(3d + 1)$ function evaluations. Computation of all upper bounds requires $N_G^{\text{UB}} = N(d + 1)$ function evaluations. This is the same number that the number of function evaluations required for computation of S_i^{tot} which is $N_G^S = N(d + 1)$ [31].

However, the number of sampled points N needed to achieve numerical convergence can be different for DGSM and S_i^{tot} . It is generally lower for the case of DGSM. Moreover, the numerical efficiency of the DGSM method can be significantly increased by using algorithmic differentiation in the adjoint (reverse) mode [9] (see also ▶ Chap. 32, “Variational Methods”). This approach allows estimating all derivatives at a cost independent of d , at most 4–6 times of that for evaluating the original function $G(\mathbf{x})$ [13].

8 Test Cases

In this section, three test cases are considered, in order to illustrate application of DGSM and their links with S_i^{tot} .

Example 1. Consider a linear with respect to x_i function:

$$G(x) = a(\mathbf{z})x_i + b(\mathbf{z}).$$

For this function $S_i = S_i^{\text{tot}}$, $V_i^{\text{tot}} = \frac{1}{12} \int_{H^{d-1}} a^2(\mathbf{z}) d\mathbf{z}$, $v_i = \int_{H^{d-1}} a^2(\mathbf{z}) d\mathbf{z}$, $\text{LB1} = \left(\int_{H^d} (a^2(\mathbf{z}) - 2a^2(\mathbf{z})x_i) d\mathbf{z} dx_i \right)^2 = 0$, and $\gamma(m) = \frac{(2m+1)m^2 \left(\int_{H^{d-1}} a(\mathbf{z}) d\mathbf{z} \right)^2}{4(m+2)^2(m+1)^2 V}$.

A maximum value of $\gamma(m)$ is attained at $m^* = 3.745$, while $\gamma^*(m^*) = \frac{0.0401}{V} \left(\int a(\mathbf{z}) d\mathbf{z} \right)^2$. The lower and upper bounds are $\text{LB}^* \approx 0.48S_i^{\text{tot}}$, $\text{UB1} \approx 1.22S_i^{\text{tot}}$. $\text{UB2} = \frac{1}{12V} \int_0^1 a(\mathbf{z})^2 d\mathbf{z} = S_i^{\text{tot}}$.

For this test function, $\text{UB2} < \text{UB1}$.

Example 2. Consider the so-called g-function which is often used in global SA for illustration purposes:

$$G(x) = \prod_{i=1}^d v_i,$$

where $v_i = \frac{|4x_i - 2| + a_i}{1 + a_i}$, $a_i (i = 1, \dots, d)$ are constants. It is easy to see that for this function $g_i(x_i) = (v_i - 1)$, $u_i(x) = (v_i - 1) \prod_{j=1, j \neq i}^d v_j$, and as a result $\text{LB1} = 0$. The total variance is $V = -1 + \prod_{j=1}^d \left(1 + \frac{1/3}{(1+a_j)^2} \right)$. The analytical values of S_i , S_i^{tot} and LB2 are given in Table 36.2.

By solving equation $\frac{d\gamma(m)}{dm} = 0$, $m^* = 9.64$ and $\gamma(m^*) = \frac{0.0772}{(1+a_i)^2 V}$. It is interesting to note that m^* does not depend on a_i , $i = 1, 2, \dots, d$ and d . In the

Table 36.2 The analytical expressions for S_i , S_i^{tot} and LB2 for g-function

S_i	S_i^{tot}	$\gamma(m)$
$\frac{1/3}{(1+a_i)^2 V}$	$\frac{\frac{1/3}{(1+a_i)^2} \prod_{j=1, j \neq i}^d \left(1 + \frac{1/3}{(1+a_j)^2} \right)}{V}$	$\frac{(2m+1) \left[1 - \frac{4(1-(1/2)^{m+1})}{m+2} \right]^2}{(1+a_i)^2(m+1)^2 V}$

extreme cases, if $a_i \rightarrow \infty$ for all i , $\frac{\gamma(m^*)}{S_i^{\text{tot}}} \rightarrow 0.257$, $\frac{S_i}{S_i^{\text{tot}}} \rightarrow 1$, while if $a_i \rightarrow 0$ for all i , $\frac{\gamma(m^*)}{S_i^{\text{tot}}} \rightarrow \frac{0.257}{(4/3)^{d-1}}$, $\frac{S_i}{S_i^{\text{tot}}} \rightarrow \frac{1}{(4/3)^{d-1}}$. The analytical expression for S_i^{tot} , UB1, and UB2 are given in Table 36.3.

For this test function, $\frac{S_i^{\text{tot}}}{\text{UB1}} = \frac{\pi^2}{48}$, $\frac{S_i^{\text{tot}}}{\text{UB2}} = \frac{1}{4}$; hence $\frac{\text{UB2}}{\text{UB1}} = \frac{\pi^2}{12} < 1$.

Values of S_i , S_i^{tot} , UB1, UB2, and LB2 for the case of $\mathbf{a} = [0, 1, 4.5, 9, 99, 99, 99, 99]$, $d = 8$, are given in Table 36.4 and shown in Fig. 36.1. One can see that the knowledge

Table 36.3 The analytical expressions for S_i^{tot} , UB1, and UB2 for g-function

S_i^{tot}	UB1	UB2
$\frac{1/3}{(1+a_i)^2} \prod_{j=1, j \neq i}^d \left(1 + \frac{1/3}{(1+a_j)^2}\right)$	$\frac{16 \prod_{j=1, j \neq i}^d \left(1 + \frac{1/3}{(1+a_j)^2}\right)}{(1+a_i)^2 \pi^2 V}$	$\frac{4 \prod_{j=1, j \neq i}^d \left(1 + \frac{1/3}{(1+a_j)^2}\right)}{3(1+a_i)^2 V}$

Table 36.4 Values of LB*, S_i , S_i^{tot} , UB1, and UB1. Example 2, $\mathbf{a} = [0, 1, 4.5, 9, 99, 99, 99, 99]$, $d = 8$.

	x_1	x_2	x_3	x_4	$x_5 \dots x_8$
LB*	0.166	0.0416	0.00549	0.00166	0.000017
S_i	0.716	0.179	0.0237	0.00720	0.0000716
S_i^{tot}	0.788	0.242	0.0343	0.0105	0.000105
UB1	3.828	1.178	0.167	0.0509	0.00051
UB2	3.149	0.969	0.137	0.0418	0.00042

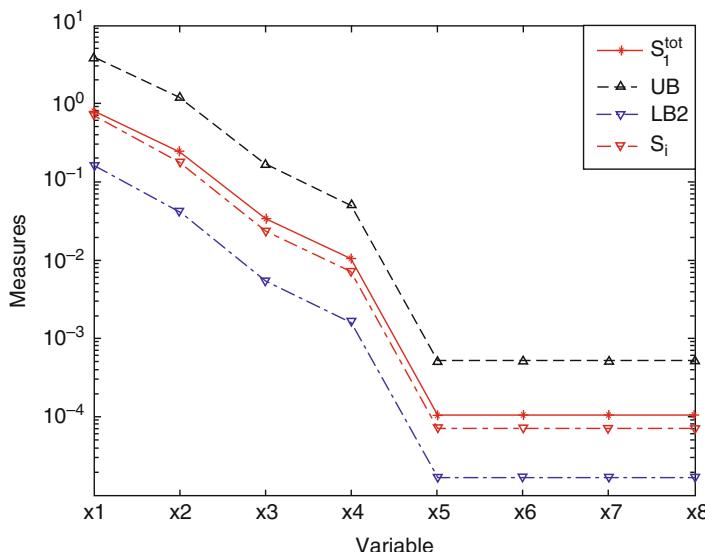


Fig. 36.1 Values of S_i , S_i^{tot} , LB2, and UB1 for all input variables. Example 2 with $\mathbf{a} = [0, 1, 4.5, 9, 99, 99, 99, 99]$, $d = 8$

of LB2 and UB1 allows to rank correctly all the variables in the order of their importance.

Example 3. Consider the reduced Morris test function with four inputs [3]:

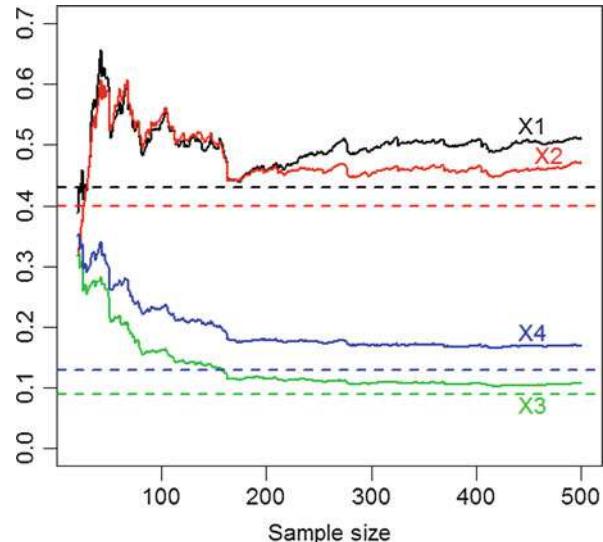
$$f(\mathbf{x}) = \sum_{i=1}^4 b_i x_i + \sum_{i \leq j}^4 b_{ij} x_i x_j + \sum_{i \leq j \leq k}^4 b_{ijk} x_i x_j x_k \quad (36.44)$$

$$\text{with } b_i = \begin{bmatrix} 0.05 \\ 0.59 \\ 10 \\ 0.21 \end{bmatrix}, \quad b_{ij} = \begin{bmatrix} 0 & 80 & 60 & 40 \\ 0 & 30 & 0.73 & 0.18 \\ 0 & 0 & 0.64 & 0.93 \\ 0 & 0 & 0 & 0.06 \end{bmatrix}, \quad b_{ij4} = \begin{bmatrix} 0 & 10 & 0.98 & 0.19 \\ 0 & 0 & 0.49 & 50 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The indices $b_{ijk} \forall k \neq 4$ are null.

The four input variables x_i ($i = 1, \dots, 4$) follow uniform distribution on $[0, 1]$. Sobol' indices are computed via the Monte Carlo scheme of [29] (using two initial matrices of size 10^5), while DGSM are computed with Monte Carlo sampling of size n (using derivatives computing by finite differences (36.43) with $\delta = 10^{-5}$), with n ranging from 20 to 500. Figure 36.2 shows that DGSM bounds UB_{1i} are greater than the total Sobol' indices S_{T_i} (for $i = 1, 2, 3, 4$) as expected, except for $n < 30$ which is a too small sample size. For small S_{T_i} , UB_{1i} is close to the S_{T_i} value. It confirms that DGSM bounds are first useful for screening exercises. Other

Fig. 36.2 For the *four* input variables of the reduced Morris test function: convergence of the DGSM bound estimates (*solid lines*) in function of the sample size and comparison to theoretical values of total Sobol' indices S_{T_i} (*dashed lines*)



numerical tests involving nonuniform and non-normal distributions for the inputs can be found in [19] and [8].

9 Conclusions

This paper has shown that using lower and upper bounds based on DGSM is possible in most cases to get a good practical estimation of the values of S_i^{tot} at a fraction of the CPU cost for estimating S_i^{tot} . Upper and lower bounds can be estimated using MC/QMC integration methods using the same set of partial derivative values. Most of the applications show that DGSM can be used for fixing unimportant variables and subsequent model reduction because small values of DGSM imply small values of S_i^{tot} . In a general case, variable ranking can be different for DGSM and variance-based methods, but for linear function and product function, DGSM can give the same variable ranking as S_i^{tot} .

Engineering applications of DGSM can be found, for instance, in [15] and [27] for biological systems modeling, [24] for structural mechanics, [12] for an aquatic prey-predator model, [25] for a river flood model, and [41] for a hydrogeological simulator of the oil industry. One of the main prospects in practical situations is to use algorithmic differentiation in the reverse (adjoint) mode on the numerical model, allowing to estimate efficiency of all partial derivatives of this model (see ► Chap. 32, “Variational Methods”). In this case, the cost of DGSM estimations would be independent of the number of input variables. Obtaining global sensitivity information in a reasonable CPU time cost is therefore possible even for large-dimensional model (several tens and spatially distributed inputs in the recent and pioneering attempt of [25]). When the adjoint model is not available, the DGSM estimation remains a problem in high dimension, and novel ideas have to be explored [23, 24]. Coupling DGSM with non-parametric regression techniques or metamodel-based technique (see ► Chap. 38, “Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes”) is another research prospect as first shown by [40] and [5].

The authors would like to thank Prof. I. Sobol’, Dr. S. Song, S. Petit, Dr. M. Lamboni, Dr. O. Roustant, and Prof. F. Gamboa for their contributions to this work. One of the authors (SK) gratefully acknowledges the financial support by the EPSRC grant EP/H03126X/1.

References

1. Bobkov, S.G.: Isoperimetric and analytic inequalities for log-concave probability measures. *Ann. Probab.* **27**(4), 1903–1921 (1999)
2. Campolongo, F., Braddock, R.: The use of graph theory in the sensitivity analysis of model output: a second order screening method. *Reliab. Eng. Syst. Saf.* **64**, 1–12 (1999)
3. Campolongo, F., Cariboni, J., Saltelli, A.: An effective screening design for sensitivity analysis of large models. *Environ. Model. Softw.* **22**, 1509–1518 (2007)
4. Cropp, R., Braddock, R.: The new Morris method: an efficient second-order screening method. *Reliab. Eng. Syst. Saf.* **78**, 77–83 (2002)

5. De Lozzo, M., Marrel, A.: Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel. *SIAM/ASA J. on Uncertain. Quantif.* (2016, Accepted for publication)
6. Fédou, J.M., Rendas, M.J.: Extending Morris method: identification of the interaction graph using cycle-equitable designs. *J. Stat. Comput. Simul.* **85**, 1398–1419 (2015)
7. Friedman, J., Popescu, B.: Predictive learning via rule ensembles. *Ann. Appl. Stat.* **2**(3), 916–954 (2008)
8. Fruth, J., Roustant, O., Kuhnt, S.: Total interaction index: a variance-based sensitivity index for second-order interaction screening. *J. Stat. Plan. Inference* **147**, 212–223 (2014)
9. Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Automatic Differentiation. SIAM, Philadelphia (2008)
10. Hardy, G., Littlewood, J., Polya, G.: Inequalities, 2nd edn. Cambridge University Press, London (1988)
11. Homma, T., Saltelli, A.: Importance measures in global sensitivity analysis of non linear models. *Reliab. Eng. Syst. Saf.* **52**, 1–17 (1996)
12. Iooss, B., Popelin, A.L., Blatman, G., Ciric, C., Gamboa, F., Lacaze, S., Lamboni, M.: Some new insights in derivative-based global sensitivity measures. In: Proceedings of the PSAM11 ESREL 2012 Conference, Helsinki, pp. 1094–1104 (2012)
13. Jansen, K., Leovey, H., Nube, A., Griewank, A., Mueller-Preussker, M.: A first look of quasi-Monte Carlo for lattice field theory problems. *Comput. Phys. Commun.* **185**, 948–959 (2014)
14. Jansen, M.: Analysis of variance designs for model output. *Comput. Phys. Commun.* **117**, 25–43 (1999)
15. Kiparissides, A., Kucherenko, S., Mantalaris, A., Pistikopoulos, E.: Global sensitivity analysis challenges in biological systems modeling. *J. Ind. Eng. Chem. Res.* **48**, 1135–1148 (2009)
16. Kucherenko, S., Song, S.: Derivative-based global sensitivity measures and their link with Sobol' sensitivity indices. In: Cools, R., Nuyens, D. (eds.) Proceedings of the Eleventh International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC 2014). Springer, Leuven (2015)
17. Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C., Shah, N.: Monte carlo evaluation of derivative-based global sensitivity measures. *Reliab. Eng. Syst. Saf.* **94**, 1135–1148 (2009)
18. Lamboni, M.: New way of estimating total sensitivity indices. In: Proceedings of the 7th International Conference on Sensitivity Analysis of Model Output (SAMO 2013), Nice (2013)
19. Lamboni, M., Iooss, B., Popelin, A.L., Gamboa, F.: Derivative-based global sensitivity measures: general links with sobol' indices and numerical tests. *Math. Comput. Simul.* **87**, 45–54 (2013)
20. Liu, R., Owen, A.: Estimating mean dimensionality of analysis of variance decompositions. *J. Am. Stat. Assoc.* **101**(474), 712–721 (2006)
21. Morris, M.: Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**, 161–174 (1991)
22. Muehlenstaedt, T., Roustant, O., Carraro, L., Kuhnt, S.: Data-driven Kriging models based on FANOVA-decomposition. *Stat. Comput.* **22**, 723–738 (2012)
23. Patelli, E., Pradlwarter, H.: Monte Carlo gradient estimation in high dimensions. *Int. J. Numer. Methods Eng.* **81**, 172–188 (2010)
24. Patelli, E., Pradlwarter, H.J., Schüller, G.I.: Global sensitivity of structural variability by random sampling. *Comput. Phys. Commun.* **181**, 2072–2081 (2010)
25. Petit, S.: Analyse de sensibilité globale du module MASCARET par l'utilisation de la différentiation automatique. Rapport de stage de fin d'études de Supélec, EDF R&D, Chatou (2015)
26. Pujol, G.: Simplex-based screening designs for estimating metamodels. *Reliab. Eng. Syst. Saf.* **94**, 1156–1160 (2009)
27. Rodriguez-Fernandez, M., Banga, J., Doyle, F.: Novel global sensitivity analysis methodology accounting for the crucial role of the distribution of input parameters: application to systems biology models. *Int. J. Robust Nonlinear Control* **22**, 1082–1102 (2012)

28. Roustant, O., Fruth, J., Iooss, B., Kuhnt, S.: Crossed-derivative-based sensitivity measures for interaction screening. *Math. Comput. Simul.* **105**, 105–118 (2014)
29. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002)
30. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global sensitivity analysis. The primer. Wiley, Chichester/Hoboken (2008)
31. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**, 259–270 (2010)
32. Santiago, J., Corre, B., Claeys-Bruno, M., Sergent, M.: Improved sensitivity through Morris extension. *Chemom. Intell. Lab. Syst.* **113**, 52–57 (2012)
33. Sobol, I.: Sensitivity estimates for non linear mathematical models (in Russian). *Matematicheskoe Modelirovaniye* **2**, 112–118 (1990)
34. Sobol, I.: Sensitivity estimates for non linear mathematical models. *Math. Model. Comput. Exp.* **1**, 407–414 (1993)
35. Sobol, I.: Global sensitivity indices for non linear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **55**, 271–280 (2001)
36. Sobol, I., Gershman, A.: On an alternative global sensitivity estimators. In: Proceedings of SAMO 1995, Belgirate, pp. 40–42 (1995)
37. Sobol, I., Kucherenko, S.: Global sensitivity indices for non linear mathematical models. *Rev. Wilmott Mag.* **1**, 56–61 (2005)
38. Sobol, I., Kucherenko, S.: Derivative based global sensitivity measures and their links with global sensitivity indices. *Math. Comput. Simul.* **79**, 3009–3017 (2009)
39. Sobol, I., Kucherenko, S.: A new derivative based importance criterion for groups of variables and its link with the global sensitivity indices. *Comput. Phys. Commun.* **181**, 1212–1217 (2010)
40. Sudret, B., Mai, C.V.: Computing derivative-based global sensitivity measures using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **134**, 241–250 (2015)
41. Touzany, S., Busby, D.: Screening method using the derivative-based global sensitivity indices with application to reservoir simulator. *Oil Gas Sci. Technol. Rev. IFP Energ. Nouvelles* **69**, 619–632 (2014)

Emanuele Borgonovo and Bertrand Iooss

Abstract

This chapter discusses the class of moment-independent importance measures. This class comprises density-based, cumulative distribution function-based, and value of information-based sensitivity measures. The chapter illustrates the definition and properties of these importance measures as they have been proposed in the literature, reviewing a common rationale that envelops them, as well as recent results that concern the general properties of global sensitivity measures. The final part of the chapter reviews importance measures developed in the context of reliability and structural reliability theories.

Keywords

Computer experiment • Global sensitivity analysis • Moment-independent importance measures • Reliability importance measures • Structural reliability • Value of information • Common rationale • Uncertainty

Contents

1	Introduction	1266
2	Density-Based Importance Measures	1267
3	Sensitivity Measures Based on Separations Between Cumulative Distribution Functions	1272
4	Value of Information-Based Sensitivity Measures	1274
5	A Common Rationale: Properties of Sensitivity Measures	1276

E. Borgonovo (✉)

Department of Decision Sciences, Bocconi University, Milan, Italy

e-mail: emanuele.borgonovo@unibocconi.it

B. Iooss

Industrial Risk Management Department, EDF R&D, Chatou, France

Institut de Mathématiques de Toulouse, Université Paul Sabatier, Toulouse, France

e-mail: bertrand.iooss@edf.fr, biooss@yahoo.fr

6 Given Data Estimation	1278
7 Sensitivity Analysis for Reliability Studies	1280
7.1 Reliability Importance Measures	1280
7.2 Sensitivity Indices for Structural Reliability Analysis	1280
8 Conclusions	1284
References	1285

1 Introduction

This chapter deals with the class of moment-independent importance measures, which comprises sensitivity measures based on discrepancies between density functions, cumulative distribution functions, and value of information. These sensitivity measures have been mainly developed in association with *factor prioritization* [44], one of the most important settings in global sensitivity analysis of model output (Section 7 in Ref. [9] reviews further sensitivity analysis settings). Factor prioritization is associated with the task of identifying key-uncertainty drivers.

The identification of key-uncertainty drivers can be performed using alternative methods. In previous chapters, sampling-based (see ► Chap. 34, “Weights and Importance in Composite Indicators: Mind the Gap”), variance-based (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”), and derivative-based methods (see ► Chap. 36, “Derivative-Based Global Sensitivity Measures”) have been illustrated. In this chapter, we explore a further class of sensitivity measures which are becoming increasingly popular among practitioners and subject of notable recent research interest, namely, moment-independent sensitivity measures. We also discuss that, as shown in recent findings, these sensitivity measures can be based on the same common rationale as variance-based and value of information-based sensitivity measures, which we also present in detail. This common rationale allows analysts to obtain a framework for analyzing several properties of sensitivity measures from a general standpoint, formulating corresponding if and only if conditions that concern important insights [12]. For instance, a null value of a probabilistic sensitivity measure is usually interpreted as an indication that the model output is independent of the model input (or group) of interest. However, this conclusion is not true in general, and recent literature has discussed the properties that reassure us that a null value of a sensitivity measure implies that the model output is statistically independent of the model input [12]. One can then examine whether variance-based, moment-independent, and value of information-based sensitivity measures possess these and other properties. The chapter concludes presenting sensitivity measures in the context of reliability studies.

The remainder of the chapter is organized as follows. First, density-based sensitivity measures are presented. The following section addresses transformation invariant global sensitivity measures. Then, value of information-based sensitivity measures is addressed. The subsequent section presents a common rationale enveloping the previously discussed sensitivity measures and discusses general properties. Another section discusses direct estimation from Monte Carlo sampling.

The last section introduces first classical reliability measures and then a recently introduced perturbation-based sensitivity measure for structural reliability.

2 Density-Based Importance Measures

This section describes the principles of the class of density-based sensitivity measures, which is timewise the first class of moment-independent sensitivity measures that have been introduced. The name moment independent communicates the intuition that these sensitivity measures take into account the change in the entire distribution (density) of the model output, instead of the variation of one of its particular moments (e.g., variance; [5, 23]). To characterize the intuition below density-based sensitivity measures, we start with an example. Consider the following input-output mapping:

$$Y = G(X_1, X_2, X_3) = X_1 X_2 + X_3 \quad (37.1)$$

and suppose that the analyst is uncertain about the model inputs. She assigns X_1 uniformly and independently distributed between -1 and 1 , and X_2 and X_3 uniformly distributed between 0 and 1 . Then, what our degree of belief about Y is displayed by the distribution of Y . If Y is a continuous random variable, then we can refer to its density, $f_Y(y)$, which is the blue curve in Fig. 37.1. Precisely, $f_Y(y)$ is the unconditional density of Y .

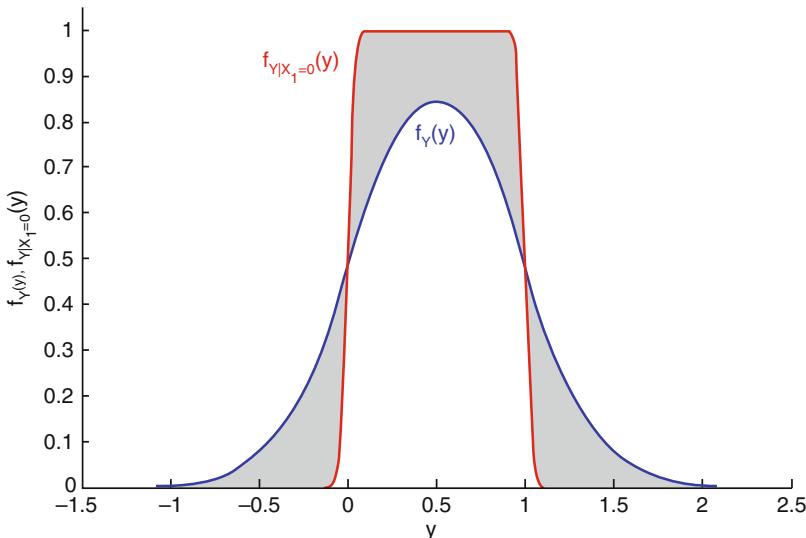


Fig. 37.1 Unconditional model output density ($f_Y(y)$, blue) and conditional model output density given that $X_1 = 0$ ($f_{Y|X_1=0}(y)$, red)

Consider now to inspect what happens if we are able to fix X_1 , say, to its mean value (which equals 0). Then, the input-output mapping becomes

$$G(0, X_2, X_3) = X_3 \quad (37.2)$$

and the conditional model output density given $X_1 = 0$ is a uniform distribution between 0 and 1. This distribution is plotted in the red curve of Fig. 37.1. Thus, receiving information that $X_1 = 0$ causes a change in the decision-maker's view about Y which is reflected by the change in the corresponding densities.

The shaded region in Fig. 37.1 displays the area enclosed between the two densities. Such area is given, in formulae, by

$$a_1(0) = \int_{\mathcal{Y}} |f_Y(y) - f_{Y|X_1=0}(y)| dy \quad (37.3)$$

where the subscript in $a_1(0)$ refers to the model input (X_1) and (0) reflects the value at which X_1 has been fixed. In our case, we register $a_1(0) = 0.4556$. In general, we shall write

$$a_i(x_i) = \int_{\mathcal{Y}} |f_Y(y) - f_{Y|X_i=x_i}(y)| dy \quad (37.4)$$

where $a_i(x_i)$ is the area enclosed between the conditional and unconditional densities, obtained by fixing the generic model input X_i at x_i .

However, because X_i is a continuous random variable, there is a finite probability that X_i assumes some other values in its support. Thus, $a_i(x_i)$ is a conditional separation for any value x_i . We can make the separation unconditional by taking the expectation over all possible values of X_i in accordance with the marginal distribution of X_i . We then write

$$\begin{aligned} \delta_i &= \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_Y(y) - f_{Y|X_i}(y)| dy \right] \\ &= \frac{1}{2} \int_{\mathcal{X}_i} f_{X_i}(x_i) dx_i \int_{\mathcal{Y}} |f_Y(y) - f_{Y|X_i}(y)| dy \end{aligned} \quad (37.5)$$

where the factor $\frac{1}{2}$ is inserted for normalization purposes. The sensitivity measure in Eq. (37.5) is the expected separation between the conditional and unconditional model output densities provoked by fixing X_i , with the separation measured using the L^1 -norm. The sensitivity measure in Eq. (37.5) has been introduced in Borgonovo [6] and is referred to as moment-independent importance measure [36] or Borgonovo's δ [51]. We shall refer to it as δ , for simplicity, in the remainder.

δ possesses several properties. The first is normalization, which is $0 \leq \delta_i \leq 1$. In particular, δ assumes the null value if and only if Y is independent of X_i . In fact, if Y and X_i are independent, fixing X_i leaves the distribution of Y unchanged. Conversely, note that δ_i can be rewritten as (see Plischke et al. [41]),

$$\delta_i = \frac{1}{2} \iint_{\mathcal{X}_i \times \mathcal{Y}} |f_Y(y) f_{X_i}(x_i) - f_{Y,X_i}(y, x_i)| dy dx_i \quad (37.6)$$

whence $\delta_i = 0$ implies that Y is independent of X_i . In particular, Eq. (37.6) shows that δ measures the distance between the product of the two marginals $f_Y(y)$, $f_{X_i}(x_i)$ and the joint $f_{Y,X_i}(y, x_i)$. Now, the fact that δ_i is null if and only if Y is independent of X_i is an important property, because it reassures the analyst that a null value of δ implies that the model output is independent of X_i . As shown in the literature, first-order variance-based and correlation ratios (both Pearson and Spearman) do not possess this property. For instance, for the model in Eq. (37.1) with the given model input distributions, we find that both the correlation coefficient between Y and X_1 ($\rho_1 = 0$) and the correlation ratio (first-order sensitivity index $S_1 = 0$) are null. However, fixing X_1 has an effect on the decision-maker's view about Y , as Fig. 37.1 shows.

δ is equal to unity when we resolve uncertainty in all model inputs. Formally, $\delta_{1,2,\dots,d} = 1$. In fact, suppose that we are able to fix the vector of all model inputs \mathbf{X} at $= \mathbf{x}^*$, where $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_d^*)$. The model output becomes the unique sure value $y^* = G(\mathbf{x}^*)$. The model output conditional density is, then, a Dirac- δ function centered at y^* . To illustrate, consider back our simple example in Eq. (37.1). Suppose that we know that $\mathbf{x}^* = \left(0, \frac{1}{2}, \frac{1}{2}\right)$. Then, Y is for sure equal to $G\left(0, \frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2}$, and the distribution of Y is a Dirac- δ function centered at $y^* = \frac{1}{2}$. Then, it can be proven that the L^1 -distance between any density function and the Dirac- δ density is equal to 2 (see Borgonovo [6]), which leads to

$$\delta_{1,2,\dots,d} = \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_Y(y) - f_{Y|\mathbf{X}^*=\mathbf{x}^*}(y)| dy \right] = \frac{1}{2} \int_{\mathcal{X}_i} 2 f_{X_i}(x_i) dx_i = 1 \quad (37.7)$$

A third property concerns sequential learning. Consider going from the present state of knowledge. Suppose that we are able to fix model input X_j after we determine model input X_i . Then, we register a first change in degree of belief from $f_Y(y)$ (where we have not fixed any model input) to $f_{Y|X_i}(y)$ (Fig. 37.2). This change is quantified by δ_i .

Once we get to know X_j , we have a second shift in degree of belief from the situation in which we know X_i and the situation in which we know X_i and X_j , that is, a shift from $f_{Y|X_i}(y)$ to $f_{Y|X_i X_j}(y)$. Then, the average shift from $f_{Y|X_i}(y)$ to $f_{Y|X_i X_j}(y)$ is given by

$$\delta_{j|i} = \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_{Y|X_i}(y) - f_{Y|X_i, X_j}(y)| dy \right]. \quad (37.8)$$

Now, after these two shifts, we are in a state in which we know both X_i and X_j . Then, our degree of belief about is now represented by $f_{Y|X_i X_j}(y)$. This state is

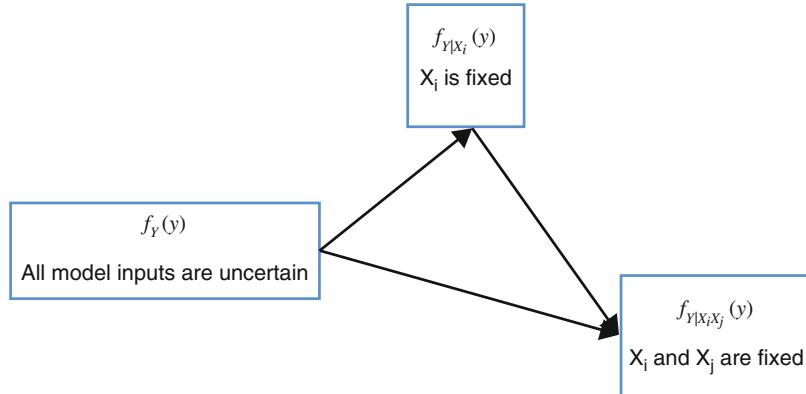


Fig. 37.2 A visual explanation of the triangular inequality for distance-based sensitivity measures

equivalently reached with a direct shift from $f_Y(y)$ to $f_{Y|X_iX_j}(y)$. This last change is quantified by

$$\delta_{i,j} = \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_Y(y) - f_{Y|X_i,X_j}(y)| dy \right]. \quad (37.9)$$

Then, based on the triangular inequality property of distances, it is readily seen that the three sensitivity measures δ_i , $\delta_{j|i}$, and $\delta_{i,j}$ are then related by (see Borgonovo [6])

$$\delta_{i,j} \leq \delta_i + \delta_{j|i} \quad (37.10)$$

A further property possessed by δ is transformation invariance. If $z(\cdot)$ is a monotonically increasing function, then we say that $Z(Y) = z(G(\mathbf{X}))$ is a monotonic transformation of the model output. To illustrate, suppose that $Z = \ln(\cdot)$, then a logarithmic transformation of the model in Eq. (37.1) is $Z = \ln(X_1 X_2 + X_3)$. Then, let δ_i^Z denote the sensitivity measure of X_i when the model output is transformed and δ_i^Y when the model output is not transformed. Then, as proven in [10], it is

$$\delta_i^Y = \delta_i^Z \quad (37.11)$$

In fact, for any density function, the change of variable rule says that if $Z = g(Y)$ is a monotonically increasing function, then

$$f_Y(y)dy = f_Z(z)dz \quad (37.12)$$

Hence, we have

$$\delta_i^Y = \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_Y(y)dy - f_{Y|X_i}(y)dy| \right] = \frac{1}{2} \mathbb{E} \left[\int_{\mathcal{Y}} |f_Z(z)dz - f_{Z|X_i}(z)dz| \right] = \delta_i^Z \quad (37.13)$$

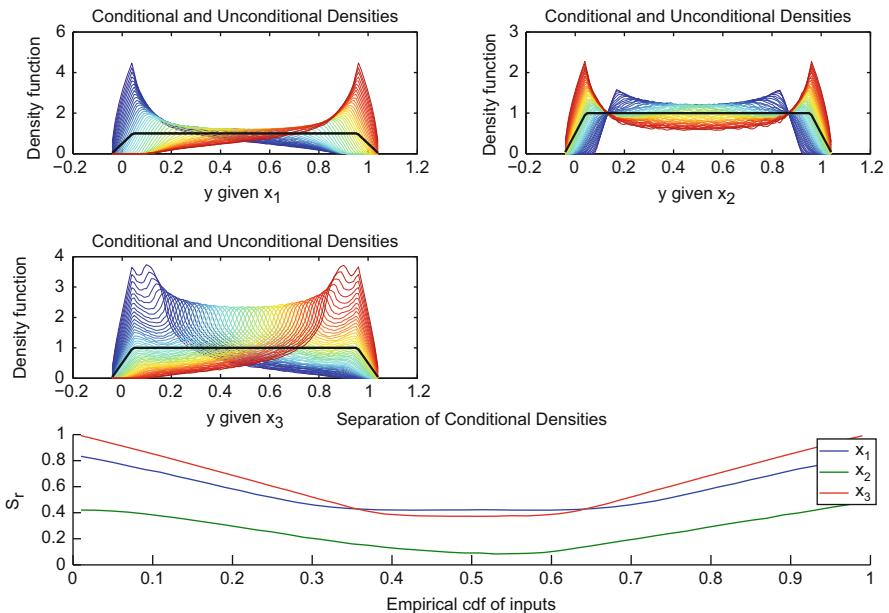


Fig. 37.3 First three graphs: unconditional model output density (black) and conditional densities given that X_1 , X_2 , and X_3 are fixed at several points in their domain, respectively. Fourth graph: the separations $a_i(x_i)$ as x_i varies between 0 and 1, for $i = 1, 2, 3$

The estimation of δ is a subject attracting attention in present research. It has been studied in a series of works [16–18, 23, 36, 41, 50, 51] that have contributed to abating computational burden. In particular, the most recent findings allow the estimation of δ at the same cost of nonparametric methods, i.e., at the cost of a single Monte Carlo loop. We illustrate by computing the moment-independent sensitivity measures for the three model inputs in the simple model of Eq. (37.1) based on the results in Plischke et al. [41]. The estimation is performed through the Matlab script deltamim.m (available upon request).

Figure 37.3 shows the results of the analysis. The first three graphs overlap the unconditional model output density $f_Y(y)$ (black) and conditional densities given that X_1 , X_2 , and X_3 are fixed at several points in their domain, respectively. These graphs visually show that all three model inputs are relevant, because they all provoke a change in model output density. They also show that X_3 and X_2 induce, in relative terms, higher fluctuations of the density when compared to X_1 .

The fourth graph displays the separations $a_i(x_i)$ as X_i varies between 0 and 1 for the three model inputs. They are symmetric and show that X_3 and X_2 are associated with separations higher in magnitude than X_1 . This information is then synthesized in the value of the δ sensitivity measures of the three model inputs, which we report in the second row of Table 37.1.

Table 37.1 shows that X_3 is the most important model input, followed by X_2 and X_1 .

Table 37.1 Numerical values of the sensitivity measures discussed in this chapter

Sensitivity measure	X_1	X_2	X_3
ρ_i Pearson correlation coefficient	0.655	0	0.655
V_i (First-order variance based)	0.412	0	0.445
δ_i (Density based)	0.29	0.13	0.312
β_i^{Ku} (Distance based)	0.29	0.13	0.312
ϵ_i (Value of information)	0.1250	0	0.1250

We conclude this section observing that the δ sensitivity measure presented above can also be seen as a representative of a general class of sensitivity measures that can be introduced by measuring some distance (or divergence) between cumulative distribution functions. Several authors make use of the Kullback-Leibler divergence to define probabilistic sensitivity measures. An early example in medical decision-making is [20]. The concept is then reconsidered in later works, such as [2, 30, 35, 39], in different fields of application. Formally, one writes

$$\theta_i = \mathbb{E} \left[\int_{\mathcal{Y}} f_{Y|X_i}(y) (\log f_{Y|X_i}(y) - \log f_Y(y)) dy \right] \quad (37.14)$$

The inner separation in Eq. (37.14) is

$$b_i(X_i) = \int_{\mathcal{Y}} f_{Y|X_i}(y) (\log f_{Y|X_i}(y) - \log f_Y(y)) dy \quad (37.15)$$

which represents the Kullback-Leibler divergence between the conditional and unconditional model output distributions. One has that $\theta_i \geq 0$, and it is equal to zero if and only if Y is independent of X_i . Also, θ_i is transformation invariant (see, among others, [12]).

Finally, while full details about estimation cannot be given due to space limitations, let us observe that the estimation of any density-based sensitivity measure involves the problem of estimating the empirical density. We refer to the works of [36, 41, 50], and [23] for a detailed discussion of the issues and for recent advances.

3 Sensitivity Measures Based on Separations Between Cumulative Distribution Functions

A second class of moment-independent sensitivity measures is represented by sensitivity measures that take into account the separation between cumulative distribution functions. In general, we write

$$\beta_i = \mathbb{E} [h(F_Y(y), F_{Y|X_i}(y))] \quad (37.16)$$

where $h(\cdot, \cdot)$ is some distance or divergence between cumulative distribution functions, $F_Y(y)$ is the unconditional model output cumulative distribution function, and

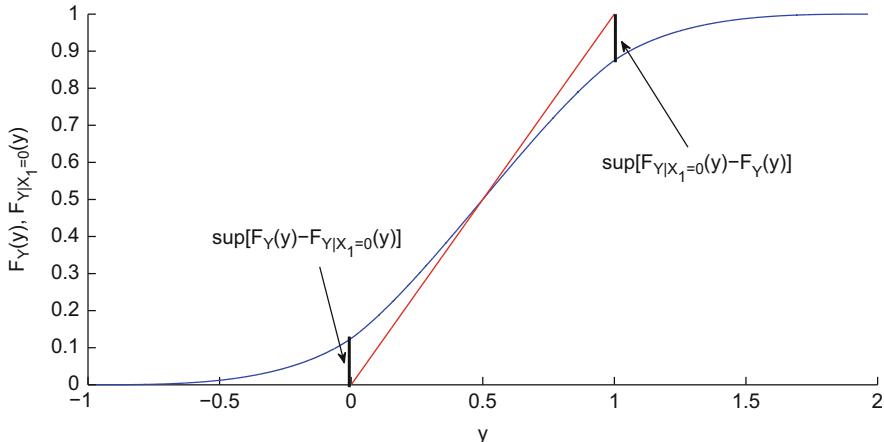


Fig. 37.4 Visual interpretation of $k_i(x_i)$ in Eq. (37.19). Kuiper's distance between cumulative distribution functions

$F_{Y|X_i}(y)$ is the cumulative distribution function conditional on fixing model input X_i . The expectation is taken in accordance with the marginal distribution of X_i .

In Baucells and Borgonovo [3], the choice of the metric is thoroughly discussed. If scale invariance is not a relevant property, one can use a metric based on a generic L_p -distance between cumulative distribution function, selecting

$$h \{F_Y(y), F_{Y|X_i}(y)\} = \left(\int (|F_Y(y) - F_{Y|X_i}(y)|)^p dy \right)^{1/p} \quad (37.17)$$

$$1 \leq p < \infty.$$

If transformation invariance is, instead, a convenient property, then one can base the sensitivity measure on the Kolmogorov-Smirnov distance and on its modifications. In Baucells and Borgonovo [3], Kuiper's distance is considered, because this metric is “equally sensitive for all values of y , and therefore puts all percentiles on an equal footing” [21, p. 140]. Kuiper's modification remedies a weakness associated with the Kolmogorov-Smirnov metric, which attributes more weight to deviations around the median rather than at distribution tails [19, 37]. A visualization of Kuiper's metric is in Fig. 37.4.

Formally, one has

$$k_i(X_i) = \sup_y \{F_Y(y) - F_{Y|X_i}(y)\} + \sup_y \{F_{Y|X_i}(y) - F_Y(y)\} \quad (37.18)$$

where $\sup_y \{F_Y(y) - F_{Y|X_i}(y)\}$ is the highest value of the difference between $F_Y(y)$ and $F_{Y|X_i}(y)$, while $\sup_y \{F_{Y|X_i}(y) - F_Y(y)\}$ is the highest value of the opposite difference.

Correspondingly, one obtains the distance-based sensitivity measure [3]:

$$\beta^{Ku} = \mathbb{E} \left[\sup_y \{F_Y(y) - F_{Y|X_i}(y)\} + \sup_y \{F_{Y|X_i}(y) - F_Y(y)\} \right] \quad (37.19)$$

As for the properties of β^{Ku} , we refer to [3] for technical details. However, we observe that β^{Ku} is nonnegative and lower than unity. It is also equal to zero if and only if Y is independent of X_i . Moreover, β^{Ku} in Eq. (37.19) is transformation invariant. It is also $\beta_{i,j}^{Ku} \leq \beta_{j|i}^{Ku} + \beta_i^{Ku}$.

Baucells and Borgonovo [3] report a thorough discussion about the selection of the metric. We limit ourselves to note that numerous modifications of the Kuiper and Kolmogorov-Smirnov metrics have been introduced. A general form is given by Mason and Shuenmeyer [37], who propose a metric of the type

$$h\{F_Y, F_{Y|X_i}\} = \sup_y \{w(F_Y(y), F_{Y|X_i}(y)) \cdot |F_Y(y) - F_{Y|X_i}(y)|\} \quad (37.20)$$

where $w[F_Y(y), F_{Y|X_i}(y)]$ is a general weighting function that allows us to assign a different relevance to different portions of the distribution. The weighting function proposed by Anderson and Darling [1] ($w(F_Y(y), F_{Y|X_i}(y)) = \sqrt{F_Y(y)[1 - F_Y(y)]}$) is one of the most well known. Anderson-Darling weight places emphasis in distribution tails. A weighting function well known in financial studies for values at risk, with focus on distribution tails, is proposed by Crnkovic and Drachman [21], who obtain a suitable modification of Kuiper's metric.

To illustrate, we conclude with the values of β^{Ku} for the model in Eq. (37.1). The sensitivity measures are computed using the subroutine betaKS2.m (available upon request). The numerical values are reported in the fourth row of Table 37.1. We observe that the values of β_i^{Ku} are equal to the values of δ_i for the three model inputs. This is not a coincidence, because, as proven in Plischke and Borgonovo [40], the two sensitivity measures β^{Ku} and δ_i are numerically identical when the conditional and unconditional distributions are unimodal.

4 Value of Information-Based Sensitivity Measures

In several applications of sensitivity analysis, the quantity estimated by a computer code is part of a broad decision-making process, in which the decision-maker is comparing a set of A alternatives. In particular, for each alternative a , the model computes a quantity Y_a of interest. In a climate change context, Y_a may represent the amount of CO₂ emissions produced under a given climate policy. In an economic context, Y_a may represent the overall loss or utility incurred when alternative a is implemented. Each Y_a is a function $G_a(\mathbf{X})$ of the n uncertain model inputs. Thus, the model produces a vector $\mathbf{y} = (y_1, y_2, \dots, y_A)$ of outputs.

The decision-maker selects the alternative that has the maximum expected utility, solving the problem

$$\max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a]\} = \max_{a=1,2,\dots,A} \{\mathbb{E}[g_a(\mathbf{X})]\}. \quad (37.21)$$

Equation (37.21) implies that the decision-maker is using the model to estimate the distribution of the decision criterion under each alternative, propagates uncertainty, and then selects the alternative that produces the highest expected value. The quantity $\max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a]\}$ is the value of the decision problem given the present state of information.

In this context, the effect of receiving information about X_i is twofold. On the one hand, the distributions of all Y_a 's change. We have discussed this effect in previous sections. On the other hand, the change in distributions might induce a change in the preferred alternative. In fact, after we are informed that $X_i = x_i$, the decision problem becomes to select the best alternative conditional on $X_i = x_i$. In other words, the optimal alternative is alternative a without the additional information but it might become a' when the new information has arrived. In this case, we register a change in both the preferred alternative and the value of the decision problem change. Then, the change in value of the decision problem is given by

$$e_i(X_i) = \max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a|X_i]\} - \max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a]\} \quad (37.22)$$

Taking the expectation with respect to X_i , we obtain the partial expected value of perfect information about X_i

$$\epsilon_i = \mathbb{E} \left[\max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a|X_i]\} \right] - \max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a]\} \quad (37.23)$$

The concept of value of information has been introduced by Howard [28]. The expression $\mathbb{E}[\max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a|X_i]\}]$ is called in Pratt et al. [42, p. 252] *prior expected value of action posterior to perfect information*. In the remainder, we shall refer to ϵ_i as information value. As observed in Howard [28], information value is a nonnegative quantity.

In sensitivity analysis, the intuition of using information value as global importance measure is due to Felli and Hazen [25]. It has then been studied in several works [38, 46] and [48].

As for the properties of information value, we note that ϵ_i is greater than or equal to zero. Its lower bound, $\epsilon_i = 0$, is reached when Y_a is independent of X_i . Its upper bound is represented by the total information value,

$$\epsilon^T = \mathbb{E} \left[\max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a|\mathbf{X}]\} \right] - \max_{a=1,2,\dots,A} \{\mathbb{E}[Y_a]\} \quad (37.24)$$

which is the value of information for resolving uncertainty in all the model inputs simultaneously.

As opposed to δ_i and β_i , a null information value does not reassure the analyst that Y is independent of X_i . In fact, it might be the case of knowing that fixing X_i never changes the preferred alternative even if Y_a is dependent on X_i . We refer to Borgonovo, Hazen, and Plischke [13] for further details. We conclude the section estimating the value of information for the model inputs of our running example. Suppose the decision-maker has two available alternatives, a and b . Alternative a leads to a sure $Y_a = \frac{1}{2}$, while alternative b leads to an uncertainty Y_b , where Y_b is in Eq. (37.1), with the usual model input distributions. The numerical values of ϵ_i are reported in Table 37.1. We observe that $\epsilon_1 = \epsilon_3 = \frac{1}{8}$, while $\epsilon_2 = 0$. This last result, as mentioned, does not mean that Y is independent of X_2 . Conversely, it indicates that fixing X_2 at any possible value does not cause the preferred alternative to change.

5 A Common Rationale: Properties of Sensitivity Measures

This section report some more recent theoretical developments. The discussion in the previous sections shows that density-based, distribution-based, and information value sensitivity measures have a common aspect. As underlined in Borgonovo, Hazen, and Plischke [13], they measure, using some form of discrepancy, the shift between the unconditional model output distribution and the conditional one, given that X_i is fixed. This common rationale is thoroughly investigated in Borgonovo, Hazen, and Plischke [13], to which we refer to full details. We succinctly report here the main findings. Let \mathbb{P}_Y be the distribution that reflects the present decision-maker degree of belief about the model output(s). Then, let $\mathbb{P}_{Y|X_i}$ be the distribution that reflects the present decision-maker degree of belief about the model output(s) when X_i is fixed. Then, consider a generic operator between distributions, say $\zeta(\cdot, \cdot)$. The operator goes from pairs of distributions $\zeta(\mathbb{P}', \mathbb{P}'')$ to \mathbb{R} . Borgonovo, Hazen, and Plischke [13] define inner operator as an operator $\zeta(\cdot, \cdot)$ that possesses the minimal property that $\zeta(\mathbb{P}, \mathbb{P}) = 0$. That is, if the two distributions coincide, then the operator returns a null value. Then, one can define the inner statistic based on inner operator $\zeta(\cdot, \cdot)$ the quantity

$$\gamma_i(X_i) = \zeta(\mathbb{P}_Y, \mathbb{P}_{Y|X_i}). \quad (37.25)$$

The inner statistic in Eq. (37.25) is a function that goes from the support of X_i into \mathbb{R} and whose numerical value depends on the value taken on by X_i . Requiring that $\gamma_i(X_i)$ is Riemann-Stieltjes integrable with respect to the marginal distribution of X_i , [13] defines the global sensitivity measure of X_i with respect to \mathbf{Y} as the quantity

$$\xi_i = \mathbb{E} [\zeta(\mathbb{P}_Y, \mathbb{P}_{Y|X_i})]. \quad (37.26)$$

Observe that the definition in Eq. (37.26) does not require independence among the model inputs. The definition also applies when a group of model inputs is considered (thus, the subscript i would denote a group of inputs). Reference [13] shows that this definition encompasses all sensitivity measures discussed in this chapter, as well as variance-based sensitivity measures. In fact, if we set

$$\gamma_i(X_i) = \mathbb{E}[(Y - \mathbb{E}[Y|X_i])^2] - \mathbb{E}[(Y - \mathbb{E}[Y])^2] \quad (37.27)$$

and average, we obtain first-order variance-based sensitivity measures. Similarly, setting $\gamma_i(X_i) = e_i(X_i)$ in Eq. (37.22) or $\gamma_i(X_i)$ equal to $a_i(x_i)$, $b_i(x_i)$, and $k_i(x_i)$ (respectively, in Eqs. (37.4), (37.15), and (37.18)) and averaging, we obtain the sensitivity measures ϵ , δ , β^{K_u} , and θ .

The choice of the inner statistic determines the properties of the corresponding global sensitivity measures. Starting from the general definition of inner operator and inner statistic, Ref. [15] proves a series if and only if conditions that link properties of the sensitivity measure to properties of the corresponding inner statistic. A property of interest in applications is transformation invariance. When the model output is sparse or ranges over several orders of magnitudes, numerical issues emerge in estimation. Analysts then resort to transformations for improving numerical accuracy and accelerating convergence. *The scaling problem most often can be overcome by performing uncertainty importance calculations based on a logarithmic scale [...] However, the log-based uncertainty importance calculations do not readily translate back to a linear scale* [29, p. 402]. The statement of Iman and Hora [29, p. 402] applies to other transformations as well. However, the problem of reverting back from the transformed to the original data is circumvented if one employs a sensitivity measure which is transformation invariant. In fact, if the sensitivity measure is transformation invariant, it assumes the same values both for the original and the transformed model output.

References [3] and [14] discuss families of transformation invariant sensitivity measures. In particular, a family of transformation invariant inner operators is obtained through the following definition [14]:

$$\zeta(\mathbb{P}, \mathbb{Q}) := \sup_{A \in \mathcal{A}} h(|\mathbb{P}(A) - \mathbb{Q}(A)|) \quad (37.28)$$

where \mathbb{P} and \mathbb{Q} are two probability measures on measurable space (Ω, \mathcal{A}) , with $h : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, $h(0) = 0$ and $\sup_t \frac{h(2t)}{h(t)} < \infty$. Then, if one lets $h(t) = t$, one obtains

$$\zeta(\mathbb{P}, \mathbb{Q}) := \sup_{A \in \mathcal{A}} (|\mathbb{P}(A) - \mathbb{Q}(A)|) = d\text{var}(\mathbb{P}, \mathbb{Q}) \quad (37.29)$$

which is the variational distance. Then, by Scheffè's theorem [45], we have

$$d\text{var}(\mathbb{P}, \mathbb{Q}) = \sup_{A \in \mathcal{A}} (|\mathbb{P}(A) - \mathbb{Q}(A)|) = \frac{1}{2} \int |p(y) - q(y)| dy \quad (37.30)$$

that is, if \mathbb{P} and \mathbb{Q} have corresponding densities, then $d_{\text{var}}(\mathbb{P}, \mathbb{Q})$ is the L^1 -norm between densities.

Borgonovo, Tarantola, Plischke, and Morris [14] also show that Eq.(37.28) contains the Birnbaum-Orlicz family of metrics

$$z(\mathbb{P}, \mathbb{Q}) = \sup_{y \in \mathbb{R}} h(|F(y) - G(y)|) = d_{\text{BO}}(\mathbb{P}, \mathbb{Q}). \quad (37.31)$$

which, in turn, setting $h(t) = t$ leads to the Kolmogorov-Smirnov metric. Finally, let $t(\cdot)$ be a convex function such that $t(0) = 1$; then the family of Csiszár divergences [22]:

$$d_h(\mathbb{P}, \mathbb{Q}) = \int p(y) t\left(\frac{q(y)}{p(y)}\right) dy \quad (37.32)$$

is invariant under monotonic transformation and comprises the Kullback-Leibler divergence as a member. Selecting any member of the families in Eq.(37.28) or (37.32) as inner operators, we obtain sensitivity measures which are transformation invariant. Da Veiga [23] also discusses the properties of the Csiszar f-divergences from a sensitivity analysis viewpoint. Da Veiga [23] shows that setting $t(s) = (\sqrt{s} - 1)^2$, we obtain a global sensitivity measure based on the Hellinger distance [23], while setting $t(s) = (s - 1)^2$ or $t(s) = \frac{(s - 1)^2}{s}$ one obtains sensitivity measures based on Pearson's χ^2 or Neyman's χ^2 divergences, respectively.

6 Given Data Estimation

The estimation of probabilistic sensitivity measures has been a subject of intensive research. The reason is that, when estimating sensitivity measures from computer codes, it is important to minimize the number of model runs or to make best use of the available ones. If one aims at estimating the sensitivity measures presented in this section (i.e., variance based, density based, distribution based, and value of information based), then, one has several choices. The first is to use a so-called brute force approach. In a brute force approach, we develop an algorithm that strictly replicates the definition of the quantity to be estimated. Then, we need to estimate the inner statistic and the outer expectation. For estimating the inner statistic, we sample a value x_i randomly from the marginal distribution of X_i . Then, we fix $X_i = x_i$ in the model and propagate uncertainty from the conditional distribution of the model inputs given that $X_i = x_i$. This distribution is denoted as $F_{\mathbf{X}|X_i}(\mathbf{x}_{\sim i}; x_i)$, where $\mathbf{x}_{\sim i}$ denotes a vector containing all model inputs but X_i . Numerically, we generate a sample of size N_{int} from this conditional distribution and evaluate the model N_{int} times, to get the conditional distribution given that $X_i = x_i$. We have then the samples necessary to estimate the inner statistics (Eq.37.25) of any sensitivity measure discussed in this section, from variance-

based to value of information. This procedure needs to be repeated for as many values of X_i as possible to obtain the expectation of the inner statistic and the corresponding global sensitivity measure (Eq. 37.26). In fact, we need to compute the outer expectation. Suppose that N_{ext} is a large enough sample for the external expectation. Then, the total number of model runs needed to estimate ξ_i is equal to $N_{\text{ext}}N_{\text{int}}$. This procedure then needs to be repeated for all model inputs, leading to a total computational cost of $C^{BF} = d \cdot N_{\text{ext}} \cdot N_{\text{int}}$, where C^{BF} stands for brute force cost. Hence, the cost is linear in the number of model inputs and depends on the square of the Monte Carlo sample. Such dependence makes the estimation rapidly unfeasible. For instance, consider a model that has $d = 10$ inputs and that $N_{\text{int}} = N_{\text{ext}}$ of at least 1000 is needed to obtain accurate estimates. Then, the cost becomes equal to $C^{BF} = 10 \cdot 1000^2 = 10,000,000$.

Several designs have been identified in the literature to abate this computational cost. For instance, the design of Saltelli [43] estimates first- and total-order variance-based sensitivity measures at $N(2d + 2)$, which is a notable reduction. Nonetheless, the cost can be abated to a minimum of N model runs, i.e., all the discussed probabilistic sensitivity measures can be estimated from a single Monte Carlo loop.

Two strategies are available. The first is to fit a metamodel on the given Monte Carlo sample [32, 49]. Then, because metamodel executions are computationally inexpensive, one can resort to a brute force approach to estimate sensitivity measures, substituting the metamodel for the original model. For moment-independent methods, a combination of smoothing spline metamodeling and brute force estimation is applied in Borgonovo [11] for the estimation of the δ importance measures, and a polynomial chaos expansion is used in Caniou and Sudret [16, 17]. For value of information sensitivity measures, [48] use Gaussian process metamodeling.

A second strategy is represented by using a given data approach. This approach is receiving increasing attention and is employed in a variety of works [3, 14, 41, 46, 47]. It consists of estimating the sensitivity measures directly from the Monte Carlo sample, but without fitting a metamodel. The main intuition is to relax the point condition $X_i = x_i$, replacing it with the bin condition $X_i \in [x_i - \Delta, x_i + \Delta]$, where Δ is a positive number, $[x_i - \Delta, x_i + \Delta]$ is an interval around x_i , and, more precisely, $[x_i - \Delta, x_i + \Delta]$ is a subset of a proper partition of the support of X_i . Then, by making Δ a decreasing function of the sample size, so that $\lim_{n \rightarrow \infty} \Delta = 0$, the bin condition reduces to the point condition. This latter step is achieved by means of a partition-refining strategy. Technical details are discussed in [13].

In general terms, Borgonovo, Hazen, and Plischke [13] obtain estimators of the type

$$\hat{\xi}_i = \frac{1}{M(N)} \sum_{m=1}^{M(N)} \xi \left(\widehat{F}_Y(y), \widehat{F}_{Y|X_i \in [x_i - \Delta, x_i + \Delta]}(y) \right) \quad (37.33)$$

where $M(N)$ is the number of partitions at size N and $\xi \left(\widehat{F}_Y(y), \widehat{F}_{Y|X_i \in [x_i - \Delta, x_i + \Delta]}(y) \right)$ is an estimator of the inner statistic in which the point condition has been replaced by the bin condition.

Borgonovo, Hazen, and Plischke [13] then prove a general result showing that under the assumptions that $\xi(\cdot, \cdot)$ is a continuous function of its arguments, that the inner statistic is Riemann-Stieltjes integrable, and that $M(N)$ is a monotonically increasing function such that $\lim_{N \rightarrow \infty} \frac{N}{M(N)} = \infty$, the estimator $\hat{\xi}_i$ in Eq. (37.33) is a consistent estimator of ξ_i .

The numerical values of the sensitivity measures presented in Table 37.1 are obtained using a given data estimation approach.

We finally note that the common rationale cited in this work is also related to the recent work by Fort et al. [26].

7 Sensitivity Analysis for Reliability Studies

As a complement, this section presents other moment-independent sensitivity measures, but devoted to reliability studies. In the first part, we present traditional importance measures, while in the second part, we present sensitivity indices for structural reliability analysis.

7.1 Reliability Importance Measures

An important realm for the application of sensitivity measures is reliability analysis. In fact, the first sensitivity measure that has been introduced for determining importance is the reliability importance measure of Birnbaum [4]. Birnbaum's work is concerned with the fact that not all components contribute to system reliability in the same way and Birnbaum aims at determining the importance of the failure of a component. Then, it turns out that, under appropriate conditions, the Birnbaum importance measure of component is a local sensitivity measure equal to the partial derivative of the reliability function with respect to the failure probability of interest. Several other importance measures have followed. Among the most widely known and used are the Fussell-Vesely, criticality, risk achievement worth, and risk reduction worth (we refer to [7, 8], and [31] for a review). For the purposes of this chapter, it is important to remark that these importance measures consider aleatory uncertainty, that is, uncertainty associated with the occurrence of an event. In structural reliability studies, often epistemic uncertainty enters the picture, where by epistemic uncertainty, we mean uncertainty in a parameter (model input) of a complex structural reliability code. The next section describes a recently introduced reliability importance measure which takes this aspect into account.

7.2 Sensitivity Indices for Structural Reliability Analysis

In structural reliability analysis [33], complex numerical codes help engineers in assessing system safety. Analysts are then often interested in quantifying the

contribution to uncertainty in system response by uncertain model inputs. Lemaitre et al. [34] have recently proposed a sensitivity index based upon the modification of the probability density function (pdf) of the random inputs, when the quantity of interest is a failure probability (probability that a model output exceeds a given threshold). This method can also be used when the quantity of interest is a quantile.

Let $y = G(\mathbf{x})$ be a mapping that summarizes the system structural properties as a function of given characteristics (\mathbf{x} is the vector of model inputs, in this context). Then, $G(\mathbf{x}) < 0$ denotes system failure, while the complementary event $G(\mathbf{x}) \geq 0$ denotes that the system is operating in a safe mode. Then, assume that the model inputs $X_i, i = 1, \dots, d$, are uncertain and independently distributed with marginal densities $f_i, i = 1, \dots, d$. The system response becomes a random variable and the model writes $Y = G(\mathbf{X})$. The quantity of interest to the analyst is then the failure probability:

$$P = Pr\{G(\mathbf{X}) < 0\} = \int \mathbf{1}_{\{G(\mathbf{x}) < 0\}} f(\mathbf{x}) d\mathbf{x}, \quad (37.34)$$

where $f(\mathbf{x})$ is the pdf of \mathbf{X} . The objective of the sensitivity analysis becomes the quantification of the influence of each X_i on this probability.

The proposed approach consists in perturbing the original pdf f_i for a given input variable X_i while keeping constant the pdfs for all the other input variables $\mathbf{X}_{\sim i} = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_d)$. Let us call $X_{i\zeta} \sim f_{i\zeta}$ the corresponding perturbed random input. This perturbed input takes the place of the real random input X_i , in a sense of modeling error: what if the real variable were $X_{i\zeta}$ instead of X_i ? In order to answer this question, a new value for the failure probability, denoted $P_{i\zeta}$, is computed. X_i can be said influential (resp. no influential) on the failure probability if the value $P_{i\zeta}$ is different (resp. similar) from the reference value P .

Following this idea, the perturbed failure probability of the system writes

$$P_{i\zeta} = \int \mathbf{1}_{\{G(\mathbf{x}) < 0\}} \frac{f_{i\zeta}(x_i)}{f_i(x_i)} f(\mathbf{x}) d\mathbf{x}. \quad (37.35)$$

The sensitivity index of X_i is then defined as the quantity:

$$S_{i\zeta} = \left[\frac{P_{i\zeta}}{P} - 1 \right] \mathbf{1}_{\{P_{i\zeta} \geq P\}} + \left[1 - \frac{P}{P_{i\zeta}} \right] \mathbf{1}_{\{P_{i\zeta} < P\}}. \quad (37.36)$$

This sensitivity index is called density modification-based reliability sensitivity index (DMBRSI) in [34]. However, the simpler acronym PLI, for perturbation law-based sensitivity indices, was then found. It is easy to see that $S_{i\zeta} = 0$ if $P_{i\zeta} = P$, as expected if X_i is a non-influential variable or if ζ expresses a negligible perturbation. The sign of $S_{i\zeta}$ indicates how the perturbation impacts the failure probability qualitatively. An example of PLI will be given at the end of this section.

The PLI can be estimated directly from the set of simulations used to compute the system failure probability, thus limiting the number of calls to the numerical model. Under mild support constraints, a consistent estimator of $P_{i\varsigma}$ is given by

$$\hat{P}_{i\varsigma N} = \frac{1}{N} \sum_{j=1}^N \mathbf{1}_{\{G(\mathbf{x}^j) < 0\}} \frac{f_{i\varsigma}(x_i^j)}{f_i(x_i^j)}. \quad (37.37)$$

where $(\mathbf{x}^1, \dots, \mathbf{x}^N)$ is a Monte Carlo sample. This property holds in the more general case when P is originally estimated by importance sampling (e.g., see Lemaire [33]) rather than simple Monte Carlo, which is more appealing in contexts when G is time-consuming. Asymptotical properties of the indices are derived in Lemaitre et al. [34], including a central limit theorem for the plug-in estimator of $S_{i\varsigma}$.

Concerning the perturbed input density $f_{i\varsigma}$, it is defined as the closest distribution to the original f_i in the entropy sense and under some constraints of perturbation. The Kullback-Leibler divergence (Eq. 37.15) is used in [34] to quantify the distance between the reference model input density and the perturbed density. The perturbed density is then found as the solution to a constrained optimization problem, as follows:

$$\begin{aligned} \min_{f_{\text{mod}}} & \{KL(f_{\text{mod}}, f_i)\} \\ \text{s.t.} & \int g_k(x_i) f_{\text{mod}}(x_i) dx_i = \varsigma_{k,i} \quad (k = 1 \dots K) \end{aligned} \quad (37.38)$$

where $KL(f_{\text{mod}}, f_i)$ is the Kullback-Leibler divergence between the original and the perturbed model input density, g_k are assigned univariate functions, and $\varsigma_{k,i}$ are real numbers, for $i = 1, 2, \dots, d$ and $k = 1, 2, \dots, K$.

Then, there exists a unique vector $\boldsymbol{\lambda}^* \in \mathbb{R}^K$ that solves the minimization problem in Eq. (37.38) and the explicit solution is

$$f_{i\varsigma}(x_i) = f_i(x_i) \exp \left[\sum_{k=1}^K \lambda_k^* g_k(x_i) - \psi_i(\boldsymbol{\lambda}^*) \right]. \quad (37.39)$$

where

$$\psi_i(\boldsymbol{\lambda}) = \log \int f_i(x) \exp \left[\sum_{k=1}^K \lambda_k g_k(x) \right] dx. \quad (37.40)$$

Alternative types of perturbations are considered in Lemaitre et al. [34]. The first is (*mean shifting*), i.e., a perturbation in the mean:

$$\int x_i f_{\text{mod}}(x_i) dx_i = \varsigma_i. \quad (37.41)$$

A second perturbation is *variance shifting*, when the analyst is perturbing the variance of X_i :

$$\begin{cases} \int x_i f_{\text{mod}}(x_i) dx_i = \mathbb{E}[X_i], \\ \int x_i^2 f_{\text{mod}}(x_i) dx_i = \varsigma_i + \mathbb{E}[X_i]^2. \end{cases} \quad (37.42)$$

One can also incorporate in this framework a *quantile shifting*, which is especially useful if the analyst is interested in modifications in the tails of the model input distributions:

$$\int 1_{]-\infty; q_r]}(x_i) f_{\text{mod}}(x_i) dx_i = \varsigma_i \quad (37.43)$$

where q_r is the reference quantile. Equation (37.43) has the meaning that f_{mod} is a density such that its ς_i -quantile is q_r .

To illustrate the PLI indices defined in Eq. (37.36), we report the example in [34] which uses the modified version of the Ishigami function:

$$G(\mathbf{X}) = \sin(X_1) + 7 \sin(X_2)^2 + 0.1 X_3^4 \sin(X_1) + 7$$

where $X_i \sim \mathcal{U}[-\pi, \pi]$, $i = 1, \dots, 3$. Under these numerical assumptions, the system failure probability is estimated at roughly $P \simeq 0.0061$, obtained with a sample of 10^5 points. The PLI of the three model inputs undergoing a mean perturbation are displayed in Fig. 37.5, which reproduces identical numerical results as Figure 5 in [34]. ς ranges from -3 to 3 , and the sensitivity is evaluated at 60 points.

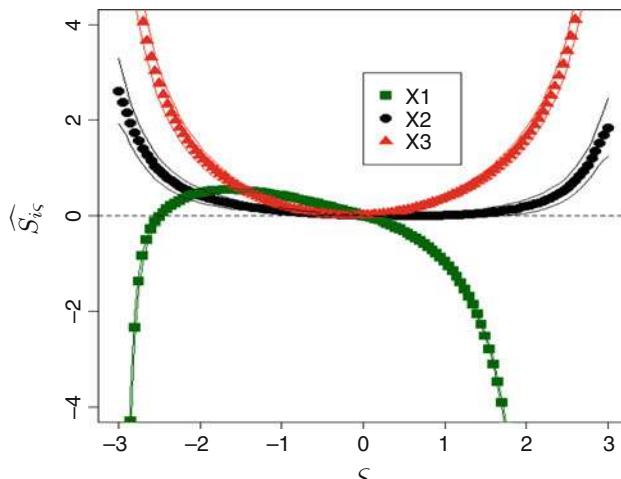


Fig. 37.5 Estimated PLI (indices \widehat{S}_i^ς) for the thresholded Ishigami function with a mean shifting

Figure 37.5 shows that a positive perturbation of the mean of X_2 or X_3 increases the failure probability for all values of ζ . Conversely, an increase in the mean or a large decrease in the mean of X_1 strongly diminishes the failure probability. Figure 37.5 also shows that the confidence intervals of the three sensitivity curves are well separated, except in the -1 to 1 zone.

8 Conclusions

We have reviewed the important class of moment-independent sensitivity measures. These sensitivity measures are subject of increasing interest among practitioners and researchers for their ability to produce reliable insights on the strength of the dependence of the model output on a given model input.

The chapter has also reviewed the fact that variance-based, moment-independent, and value of information-based sensitivity measures are particular cases of a common rationale, in which a global sensitivity measure is seen as the expected separation between the conditional and unconditional model output distribution.

The section has then reviewed estimation at minimal computational cost, namely, N model runs. We have then reported findings that concern both estimation through metamodels and, also, estimation through given data. In this second case, estimation is direct from the Monte Carlo sample and does not require the use of a metamodel.

We have completed the chapter discussing reliability importance measures. Traditional importance measures have been addressed first. We have then overviewed a perturbation-based sensitivity index for structural reliability analysis recently introduced.

We observe that moment-independent sensitivity measures, due to their more recent introduction with respect to other classes of global sensitivity measures, are a subject of increasing research interest, and, nowadays, several works are addressing the definition of new moment-independent sensitivity measures and/or studying alternative approaches to their estimation (we refer to [24, 27, 40] for recent works in progress). Also, some aspects concerning their interpretation are subject to future research. For instance, the interpretation of higher-order moment-independent sensitivity measures has not been compared to higher-order variance-based sensitivity measures. To illustrate, consider δ_{ij} in Eq. (37.9). This joint sensitivity index is clearly different from a second-order variance-based sensitivity index. In fact, δ_{ij} is the distance that we travel for getting to know both X_i and X_j . Thus, it represents the expected effect on the decision-maker's degree of belief of acquiring perfect knowledge on both model inputs. This interpretation holds for both dependent and independent model inputs. Conversely, the second variance-based sensitivity index S_{ij} represents a mathematical interaction, and, under independence, it tells us whether there is a residual term that makes the response of the input-output mapping not additive in X_i and X_j . Then, S_{ij} might be seen as closer in interpretation to $\delta_{j|i}$, which represents the additional (residual) effect on the decision-maker's degree of belief of getting to know X_j , after she has gotten to know X_i . However, $\delta_{j|i}$

addresses a different question than S_{ij} . A thorough discussion of these aspects is out of the scope of the present chapter and is part of future research.

References

1. Anderson, T.W., Darling, D.A.: Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes. *Ann. Math. Stat.* **23**, 193–212 (1952)
2. Auder, B., Iooss, B.: Global sensitivity analysis based on entropy. In: ESREL 2008 Conference, ESRA, Valencia, Sept 2008
3. Baucells, M., Borgonovo, E.: Invariant probabilistic sensitivity analysis. *Manag. Sci.* **59**(11), 2536–2549 (2013). <http://www.scopus.com/inward/record.url?eid=2-s2.0-8488863510&partnerID=tZotx3y1>
4. Birnbaum, L.: On the importance of different elements in a multielement system. In: Krishnaiah, P.R. (ed.) *Multivariate Analysis*, vol. 2, pp. 1–15. Academic, New York (1969)
5. Borgonovo, E.: Measuring uncertainty importance: investigation and comparison of alternative approaches. *Risk Anal.* **26**(5), 1349–1361 (2006)
6. Borgonovo, E.: A new uncertainty importance measure. *Reliab. Eng. Syst. Saf.* **92**(6), 771–784 (2007)
7. Borgonovo, E.: Differential, criticality and Birnbaum importance measures: an application to basic event, groups and SSCs in event trees and binary decision diagrams. *Reliab. Eng. Syst. Saf.* **92**(10), 1458–1467 (2007)
8. Borgonovo, E.: The reliability importance of components and prime implicants in coherent and non-coherent systems including total-order interactions. *Eur. J. Oper. Res.* **204**(3), 485–495 (2010)
9. Borgonovo, E., Plischke, E.: Sensitivity analysis for operational research. *Eur. J. Oper. Res.*, **3**(1), 869–887 (2016)
10. Borgonovo, E., Castaings, W., Tarantola, S.: Moment independent uncertainty importance: new results and analytical test cases. *Risk Anal.* **31**(3), 404–428 (2011)
11. Borgonovo, E., Castaings, W., Tarantola, S.: Model emulation and moment-independent sensitivity analysis: an application to environmental modeling. *Environ. Model. Softw.* **34**, 105–115 (2012)
12. Borgonovo, E., Hazen, G., Plischke, E.: A common rationale for global sensitivity analysis. In: Steenberg, R.D.M., Van Gelder, P., Miraglia, S., Vrouwenvelder, A.C.W.M.T. (eds.) *Proceedings of the 2013 ESREL Conference*, Amsterdam, pp. 3255–3260 (2013)
13. Borgonovo, E., Hazen, G., & Plischke, E. (2016). A Common Rationale for Global Sensitivity Measures and their Estimation. *Risk Analysis*, forthcoming, DOI: 10.1111/risa.12555, 1–24
14. Borgonovo, E., Tarantola, S., Plischke, E., Morris, M.: Transformation and invariance in the sensitivity analysis of computer experiments. *J. R. Stat. Soc. Ser. B* **76**(5), 925–947 (2014)
15. Borgonovo, E., Hazen, G., Jose, V., Plischke, E., 2016: Value of Information, Scoring Rules and Global Sensitivity Analysis, work in progress
16. Caniou, Y., Sudret, B.: Distribution-based global sensitivity analysis using polynomial chaos expansions. *Procedia – Social and Behavioral Sciences*, pp. 7625–7626 (2010)
17. Caniou, Y., Sudret, B.: Distribution-based global sensitivity analysis in case of correlated input parameters using polynomial chaos expansions. In: 11th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP11), Zurich (2011)
18. Castaings, W., Borgonovo, E., Tarantola, S., Morris, M.D.: Sampling strategies in density-based sensitivity analysis. *Environ. Model. Softw.* **38**, 13–26 (2012)
19. Chandra, M., Singpurwalla, N.D., Stephens, M.A.: Kolmogorov statistics for tests of fit for the extreme value and Weibull distributions. *J. Am. Stat. Assoc.* **76**(375), 729–731 (1981)
20. Critchfield, G.G., Willard, K.E.: Probabilistic analysis of decision trees using Monte Carlo simulation. *Med. Decis. Mak.* **6**(2), 85–92 (1986)
21. Crnkovic, C., Drachman, J.: Quality control. *RISK* **9**(9), 139–143 (1996)

22. Csiszár, I.: Axiomatic characterizations of information measures. *Entropy* **10**, 261–273 (2008)
23. Da Veiga, S.: Global sensitivity analysis with dependence measures. *J. Stat. Comput. Simul.* **85**, 1283–1305 (2015)
24. De Lozzo, M., Marrel, A.: New improvements in the use of dependence measures for sensitivity analysis and screening, pp. 1–21 (Dec 2014). arXiv:1412.1414v1
25. Felli, J., Hazen, G.: Sensitivity analysis and the expected value of perfect information. *Med. Decis. Mak.* **18**, 95–109 (1998)
26. Fort, J., Klein, T., Rachdi, N.: New sensitivity analysis subordinated to a contrast. *Commun. Stat. Theory Methods* (2014, in press)
27. Gamboa, F., Klein, T., Lagnoux, A.: Sensitivity analysis based on Cramer von Mises distance, pp. 1–20 (2015). arXiv:1506.04133 [math.PR]
28. Howard, R.A.: Decision analysis: applied decision theory. In: *Proceedings of the Fourth International Conference on Operational Research*. Wiley-Interscience, New York (1966)
29. Iman, R., Hora, S.: A robust measure of uncertainty importance for use in fault tree system analysis. *Risk Anal.* **10**, 401–406 (1990)
30. Krzykacz-Hausmann, B.: Epistemic sensitivity analysis based on the concept of entropy. In: *SAMO 2001*, Madrid, CIEMAT, pp. 31–35 (2001)
31. Kuo, W., Zhu, X.: *Importance Measures in Reliability, Risk and Optimization*. Wiley, Chichester (2012)
32. Le Gratiet, L., Cannamela, C., Iooss, B.: A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA J. Uncertain. Quantif.* **2**, 336–363 (2014)
33. Lemaire, M.: *Structural Reliability*. Wiley-ISTE, London/Hoboken (2009)
34. Lemaitre, P., Sergienko, E., Arnaud, A., Bousquet, N., Gamboa, F., Iooss, B.: Density modification based reliability sensitivity analysis. *J. Stat. Comput. Simul.* **85**, 1200–1223 (2015)
35. Liu, H., Chen, W., Sudjianto, A.: Relative entropy based method for probabilistic sensitivity analysis in engineering design. *ASME J. Mech. Des.* **128**, 326–336 (2006)
36. Luo, X., Lu, Z., Xu, X.: A fast computational method for moment-independent uncertainty importance measure. *Comput. Phys. Commun.* **185**, 19–27 (2014)
37. Mason, D.M., Shuenmeyer, J.H.: A modified Kolmogorov Smirnov test sensitive to tail alternatives. *Ann. Stat.* **11**(3), 933–946 (1983)
38. Oakley, J.: Decision-theoretic sensitivity analysis for complex computer models. *Technometrics* **51**(2), 121–129 (2009)
39. Park, C.K., Ahn, K.I.: A new approach for measuring uncertainty importance and distributional sensitivity in probabilistic safety assessment. *Reliability Engineering & System Safety* **46**, 253–261 (1994)
40. Plischke, E., Borgonovo, E.: Probabilistic Sensitivity Measures from Empirical Cumulative Distribution Functions: A Horse Race of Methods, 2016, Work in Progress.
41. Plischke, E., Borgonovo, E., Smith, C.: Global sensitivity measures from given data. *Eur. J. Oper. Res.* **226**(3), 536–550 (2013)
42. Pratt, J., Raiffa, H., Schlaifer, R.: *Introduction to Statistical Decision Theory*. MIT, Cambridge (1995)
43. Saltelli, A.: Making best use of model valuations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002)
44. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global Sensitivity Analysis – The Primer*. Wiley, Chichester (2008)
45. Scheffé, H.: A useful convergence theorem for probability distributions. *Ann. Math. Stat.* **18**(3), 434–438 (1947)
46. Strong, M., Oakley, J.: An efficient method for computing partial expected value of perfect information for correlated inputs. *Med. Decis. Mak.* **33**, 755–766 (2013)
47. Strong, M., Oakley, J.E., Chilcott, J.: Managing structural uncertainty in health economic decision models: a discrepancy approach. *J. R. Stat. Soc. Ser. C* **61**(1), 25–45 (2012)

48. Strong, M., Oakley, J., Brennan, A.: Estimating multiparameter partial expected value of perfect information from a probabilistic sensitivity analysis sample: a nonparametric regression approach. *Med. Decis. Mak.* **34**, 311–326 (2014)
49. Sudret, B.: Global sensitivity analysis using polynomial chaos expansion. *Reliab. Eng. Syst. Saf.* **93**, 964–979 (2008)
50. Xu, X., Lu, Z., Luo, X.: Stable approach based on asymptotic space integration for moment-independent uncertainty importance measure. *Risk Anal.* **34**(2), 235–251 (2014)
51. Zhang, L., Lu, Z., Cheng, L., Fan, C.: A new method for evaluating Borgonovo moment-independent importance measure with its application in an aircraft structure. *Reliab. Eng. Syst. Saf.* **132**, 163–175 (2014)

Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes

38

Loïc Le Gratiet, Stefano Marelli, and Bruno Sudret

Abstract

Global sensitivity analysis is now established as a powerful approach for determining the key random input parameters that drive the uncertainty of model output predictions. Yet the classical computation of the so-called Sobol' indices is based on Monte Carlo simulation, which is not affordable when computationally expensive models are used, as it is the case in most applications in engineering and applied sciences. In this respect metamodels such as polynomial chaos expansions (PCE) and Gaussian processes (GP) have received tremendous attention in the last few years, as they allow one to replace the original, taxing model by a surrogate which is built from an experimental design of limited size. Then the surrogate can be used to compute the sensitivity indices in negligible time. In this chapter an introduction to each technique is given, with an emphasis on their strengths and limitations in the context of global sensitivity analysis. In particular, Sobol' (resp. total Sobol') indices can be computed analytically from the PCE coefficients. In contrast, confidence intervals on sensitivity indices can be derived straightforwardly from the properties of GPs. The performance of the two techniques is finally compared on three well-known analytical benchmarks (Ishigami, G-Sobol', and Morris functions) as well as on a realistic engineering application (deflection of a truss structure).

Keywords

Polynomial chaos expansions • Gaussian process regression • Kriging • Error estimation • Sobol' indices • Model selection

L.L. Gratiet (✉)
EDF R&D, Chatou, France
e-mail: loic.legratiet@gmail.com

S. Marelli and B. Sudret
Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Zürich, Switzerland
e-mail: marelli@ibk.baug.ethz.ch; sudret@ibk.baug.ethz.ch

Contents

1	Introduction	1290
2	Polynomial Chaos Expansions	1292
2.1	Mathematical Setup	1292
2.2	Polynomial Chaos Basis	1292
2.3	Nonstandard Variables and Truncation Scheme	1294
2.4	Computation of the Coefficients and Error Estimation	1295
2.5	Error Estimation	1297
2.6	Post-processing for Sensitivity Analysis	1298
2.7	Summary	1301
3	Gaussian Process-Based Sensitivity Analysis	1302
3.1	A Short Introduction to Gaussian Processes	1302
3.2	Gaussian Process Regression Models	1302
3.3	Main Effects Visualization	1308
3.4	Variance of the Main Effects	1309
3.5	Numerical Estimates of Sobol' Indices by Gaussian Process Sampling	1310
3.6	Summary	1312
4	Applications	1312
4.1	Ishigami Function	1312
4.2	G-Sobol' Function	1314
4.3	Morris Function	1317
4.4	Maximum Deflection of a Truss Structure	1319
5	Conclusions	1321
	References	1322

1 Introduction

In modern engineering sciences computational models are used to simulate and predict the behavior of complex systems. The governing equations of the system are usually discretized so as to be solved by dedicated algorithms. In the end a computational model (a.k.a. simulator) is built up, which can be considered as a mapping from the space of input parameters to the space of quantities of interest that are computed by the model. However, in many situations the values of the parameters describing the properties of the system, its environment, and the various initial and boundary conditions are not perfectly well known. To account for such uncertainty, they are typically described by possible variation ranges or probability distribution functions.

In this context global sensitivity analysis aims at determining which input parameters of the model influence the most the predictions, i.e., how the variability of the model response is affected by the uncertainty of the various input parameters. A popular technique is based on the decomposition of the response variance as a sum of contributions that can be associated to each single input parameter or to combinations thereof, leading to the computation of the so-called Sobol' indices.

As presented earlier in this book (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”), the use of Monte Carlo simulation

to compute Sobol' indices requires a large number of samples (typically, thousands to hundreds of thousands), which may be an impossible requirement when the underlying computational model is expensive to evaluate. To bypass this difficulty, *surrogate models* may be built. Generally speaking, a surrogate model (a.k.a. metamodel or emulator) is an approximation of the original computational model:

$$\mathbf{x} \in \mathcal{D}_X \subset \mathbb{R}^d \mapsto y = G(\mathbf{x}) \quad (38.1)$$

which is constructed based on a limited number of runs of the true model, the so-called *experimental design*:

$$\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}. \quad (38.2)$$

Once a type of surrogate model is selected, the parameters have to be fitted based on the information contained in the experimental design \mathcal{X} and associated runs of the original computational model $\mathcal{Y} = \{y_i = G(\mathbf{x}^{(i)})\}, i = 1, \dots, n\}$. Then the accuracy of the surrogate shall be estimated by some kind of validation technique. For a general introduction to surrogate modeling, the reader is referred to [60] and to the recent review by Iooss and Lemaître [33].

In this chapter, we discuss two classes of surrogate models that are commonly used for sensitivity analysis, namely, *polynomial chaos expansions* (PCE) and *Gaussian processes* (GP). The reader is referred to the books of Santner et al. [52] and Rasmussen and Williams [49] for more detail about Gaussian process regression. The use of polynomial chaos expansions in the context of sensitivity analysis has been originally presented in Sudret [61, 63] using a nonintrusive least-square method. Other nonintrusive strategies for the calculation of PCE coefficients include spectral projection through sparse grids (e.g., Crestaux et al. [20], Buzzard and Xiu [17], Buzzard [16]) and sparse polynomial expansions (e.g., Blatman and Sudret [13]). In the last 5 years, numerous application examples have been developed using PCE for sensitivity analysis, e.g., Fajraoui et al. [28], Younes et al. [70], Brown et al. [15], and Sandoval et al. [51]. Recent extensions to problems with dependent input parameters can be found in Sudret and Caniou [65] and Munoz Zuniga et al. [46].

In parallel, Gaussian process modeling has been introduced in the context of sensitivity analysis by Welch et al. [68], Oakley and O'Hagan [48], and Marrel et al. [42, 43]. Recent developments in which metamodeling errors are taken into account in the analysis have been proposed by Le Gratiet et al. [38] and Chastaing and Le Gratiet [18].

The chapter first recalls the basics of the two approaches and details how they can be used to compute sensitivity indices. The two approaches are then compared on different benchmark examples as well as on an application in structural mechanics.

2 Polynomial Chaos Expansions

2.1 Mathematical Setup

Let us consider a computational model $G : \mathbf{x} \in \mathcal{D}_X \subset \mathbb{R}^d \mapsto y = G(\mathbf{x}) \in \mathbb{R}$. Suppose that the uncertainty in the input parameters is modeled by a random vector \mathbf{X} with prescribed joint probability density function (PDF) $f_X(\mathbf{x})$. The resulting (random) quantity of interest $Y = G(\mathbf{X})$ is obtained by propagating the uncertainty in \mathbf{X} through G . Assuming that Y has a finite variance (which is a physically meaningful assumption when dealing with physical systems), it belongs to the so-called Hilbert space of second-order random variables, which allows for the following spectral representation to hold [58]:

$$Y = \sum_{j=0}^{\infty} y_j Z_j. \quad (38.3)$$

The random variable Y is therefore cast as an infinite series, in which $\{Z_j\}_{j=0}^{\infty}$ is a numerable set of random variables (which form a basis of the Hilbert space), and $\{y_j\}_{j=0}^{\infty}$ are coefficients. The latter may be interpreted as the *coordinates* of Y in this basis. In the sequel we focus on *polynomial chaos expansions*, in which the basis terms $\{Z_j\}_{j=0}^{\infty}$ are multivariate orthonormal polynomials in the input vector \mathbf{X} , i.e., $Z_j = \Psi_j(\mathbf{X})$.

2.2 Polynomial Chaos Basis

In the sequel we assume that the input variables are statistically *independent*, so that the joint PDF is the product of the d marginal distributions: $f_X(\mathbf{x}) = \prod_{i=1}^d f_{X_i}(x_i)$, where the $f_{X_i}(x_i)$ are the marginal distributions of each variable $\{X_i, i = 1, \dots, d\}$ defined on \mathcal{D}_{X_i} . For each single variable X_i and any two functions $\phi_1, \phi_2 : x \in \mathcal{D}_{X_i} \mapsto \mathbb{R}$, we define the functional inner product by the following integral (provided it exists):

$$\langle \phi_1, \phi_2 \rangle_i \stackrel{\text{def}}{=} \mathbb{E} [\phi_1(X_i) \phi_2(X_i)] = \int_{\mathcal{D}_{X_i}} \phi_1(x) \phi_2(x) f_{X_i}(x) dx. \quad (38.4)$$

Using the above notation, classical algebra allows one to build a family of *orthogonal polynomials* $\{P_k^{(i)}, k \in \mathbb{N}\}$ satisfying:

$$\left\langle P_j^{(i)}, P_k^{(i)} \right\rangle_i \stackrel{\text{def}}{=} \mathbb{E} \left[P_j^{(i)}(X_i) P_k^{(i)}(X_i) \right] = \int_{\mathcal{D}_{X_i}} P_j^{(i)}(x) P_k^{(i)}(x) f_{X_i}(x) dx = a_j^{(i)} \delta_{jk}, \quad (38.5)$$

Table 38.1 Classical families of orthogonal polynomials (taken from Sudret [62])

Type of variable	Distribution	Orthogonal polynomials	Hilbertian basis $\psi_k(x)$
Uniform $\mathcal{U}(-1, 1)$	$\mathbf{1}_{[-1,1]}(x)/2$	Legendre $P_k(x)$	$P_k(x)/\sqrt{\frac{1}{2k+1}}$
Gaussian $\mathcal{N}(0, 1)$	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	Hermite $H_{ek}(x)$	$H_{ek}(x)/\sqrt{k!}$
Gamma $\Gamma(a, \lambda = 1)$	$x^a e^{-x} \mathbf{1}_{\mathbb{R}^+}(x)$	Laguerre $L_k^a(x)$	$L_k^a(x)/\sqrt{\frac{\Gamma(k+a+1)}{k!}}$
Beta $\mathcal{B}(a, b)$	$\mathbf{1}_{[-1,1]}(x) \frac{(1-x)^a(1+x)^b}{B(a)B(b)}$	Jacobi $J_k^{a,b}(x)$	$J_k^{a,b}(x)/\mathfrak{J}_{a,b,k}$
			$\mathfrak{J}_{a,b,k}^2 = \frac{2^{a+b+1}}{2k+a+b+1} \frac{\Gamma(k+a+1)\Gamma(k+b+1)}{\Gamma(k+a+b+1)\Gamma(k+1)}$

See, e.g., Abramowitz and Stegun [1]. In the above equation subscript k denotes the degree of the polynomial $P_k^{(i)}$, δ_{jk} is the Kronecker symbol equal to 1 when $j = k$ and 0 otherwise and $a_j^{(i)}$ corresponds to the squared norm of $P_j^{(i)}$:

$$a_j^{(i)} \stackrel{\text{def}}{=} \| P_j^{(i)} \|_i^2 = \left\langle P_j^{(i)}, P_j^{(i)} \right\rangle_i. \quad (38.6)$$

In general orthogonal bases may be obtained by applying the Gram-Schmidt orthogonalization procedure, e.g., to the canonical family of monomials $\{1, x, x^2, \dots\}$. For standard distributions, the associated families of orthogonal polynomials are well known [69]. For instance, if $X_i \sim \mathcal{U}(-1, 1)$ has a uniform distribution over $[-1, 1]$, the resulting family is that of the so-called *Legendre polynomials*. When $X_i \sim \mathcal{N}(0, 1)$ has a standard normal distribution with zero-mean value and unit standard deviation, the resulting family is that of the *Hermite polynomials*. The families associated to standard distributions are summarized in Table 38.1 (taken from Sudret [62]).

Note that the obtained family is usually not orthonormal. By enforcing normalization, an *orthonormal family* $\left\{ \psi_j^{(i)} \right\}_{j=0}^{\infty}$ is obtained from Eqs. (38.5), (38.6) as follows (see Table 38.1):

$$\psi_j^{(i)} = P_j^{(i)} / \sqrt{a_j^{(i)}} \quad i = 1, \dots, d, \quad j \in \mathbb{N}. \quad (38.7)$$

From the sets of univariate orthonormal polynomials, one can now build *multivariate* orthonormal polynomials with a *tensor product* construction. For this purpose let us define the multi-indices $\alpha \in \mathbb{N}^d$, which are ordered lists of integers:

$$\alpha = (\alpha_1, \dots, \alpha_d), \quad \alpha_i \in \mathbb{N}. \quad (38.8)$$

One can associate a multivariate polynomial Ψ_α to any multi-index α by:

$$\Psi_\alpha(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{i=1}^d \psi_{\alpha_i}^{(i)}(x_i), \quad (38.9)$$

where the univariate polynomials $\{\psi_k^{(i)}, k \in \mathbb{N}\}$ are defined above, see Eqs. (38.5), (38.7). By virtue of Eq.(38.5) and the above tensor product construction, the multivariate polynomials in the input vector X are also orthonormal, i.e.:

$$\mathbb{E}[\Psi_\alpha(X)\Psi_\beta(X)] \stackrel{\text{def}}{=} \int_{\mathcal{D}_X} \Psi_\alpha(x)\Psi_\beta(x) f_X(x) dx = \delta_{\alpha\beta} \quad \forall \alpha, \beta \in \mathbb{N}^d, \quad (38.10)$$

where $\delta_{\alpha\beta}$ is the Kronecker symbol which is equal to 1 if $\alpha = \beta$ and zero otherwise. With this notation, it can be proven that the set of all multivariate polynomials in the input random vector X forms a basis of the Hilbert space in which $Y = G(X)$ is to be represented [58]:

$$Y = \sum_{\alpha \in \mathbb{N}^d} y_\alpha \Psi_\alpha(X). \quad (38.11)$$

2.3 Nonstandard Variables and Truncation Scheme

In practical sensitivity analysis problems, the input variables may not necessarily have standardized distributions as the ones described in Table 38.1. Thus *reduced variables* U with standardized distributions are introduced first through an isoprobabilistic transform:

$$X = \mathcal{T}(U). \quad (38.12)$$

For instance, when dealing with independent uniform distributions with support $\mathcal{D}_{X_i} = [a_i, b_i]$, $i = 1, \dots, d$, the isoprobabilistic transform reads:

$$X_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} U_i \quad U_i \sim \mathcal{U}([-1, 1]). \quad (38.13)$$

In the case of Gaussian independent variables $\{X_i \sim \mathcal{N}(\mu_i, \sigma_i), i = 1, \dots, d\}$, the one-to-one mapping reads:

$$X_i = \mu_i + \sigma_i U_i, \quad U_i \sim \mathcal{N}(0, 1) \quad (38.14)$$

In the general case when the input variables are non-Gaussian (e.g., Gumbel distributions, see application in the last example of Sect. 4 of this chapter), the one-to-one mapping may be obtained as follows:

$$X_i = F_{X_i}^{-1}(\Phi(U_i)) \quad U_i \sim \mathcal{N}(0, 1) \quad (38.15)$$

where F_{X_i} (resp. Φ) is the cumulative distribution function (CDF) of variable X_i (resp. the standard normal CDF).

This isoprobabilistic transform approach also allows one to address problems with *dependent* variables. For instance, if the input vector X is defined by a set of marginal distributions and a Gaussian copula, it can be transformed into a set of independent standard normal variables using the Nataf transform [22, 40].

The representation of the random response in Eq. (38.11) is exact when the infinite series is considered. However, in practice, only a finite number of terms may be computed. For this purpose a *truncation scheme* has to be selected. Since the polynomial chaos basis consists of multivariate polynomials, it is natural to consider as a truncated series all the polynomials up to a given maximum degree. Let us define the *total degree* of a multivariate polynomial Ψ_α by:

$$|\alpha| \stackrel{\text{def}}{=} \sum_{i=1}^d \alpha_i. \quad (38.16)$$

The *standard truncation scheme* consists in selecting all polynomials such that $|\alpha|$ is smaller than or equal to a given p . This leads to a set of polynomials denoted by $\mathcal{A}^{d,p} = \{\alpha \in \mathbb{N}^d : |\alpha| \leq p\}$ of cardinality:

$$\text{card } \mathcal{A}^{d,p} = \binom{d+p}{p} = \frac{(d+p)!}{d! p!}. \quad (38.17)$$

The maximal polynomial degree p may typically be equal to $3 - 5$ in practical applications. Note that the cardinality of $\mathcal{A}^{d,p}$ increases exponentially with d and p . Thus the number of terms in the series, i.e., the number of coefficients to be computed, increases dramatically when d is large, say $d > 10$. This complexity is referred to as the *curse of dimensionality*. This issue may be solved using specific algorithms to compute sparse PCE, see, e.g., Blatman and Sudret [14], and Doostan and Owhadi [23].

2.4 Computation of the Coefficients and Error Estimation

The use of polynomial chaos expansions has emerged in the late 1980s in uncertainty quantification problems under the form of *stochastic finite element methods* [29]. In this setup the constitutive equations of the physical problem are discretized both in the physical space (using standard finite element techniques) and in the random space using polynomial chaos expansion. This results in coupled systems of equations which require ad hoc solvers, thus the term “intrusive approach.”

Nonintrusive techniques such as projection or stochastic collocation have emerged in the last decade as a means to compute the coefficients of PC expansions from repeated evaluations of the existing model G considered as a black-box function. In this section we focus on a particular nonintrusive approach based on least-square analysis.

Following Berveiller et al. [7, 8], the computation of the PCE coefficients may be cast as a least-square minimization problem (originally termed “regression” problem) as follows: once a truncation scheme $\mathcal{A} \subset \mathbb{N}^d$ is chosen (for instance, $\mathcal{A} = \mathcal{A}^{d,p}$), the infinite series is recast as the sum of the truncated series and a residual:

$$Y = G(X) = \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(X) + \varepsilon, \quad (38.18)$$

in which ε corresponds to all those PC polynomials whose index α is not in the truncation set \mathcal{A} . The least-square minimization approach consists in finding the set of coefficients $\mathbf{y} = \{y_\alpha, \alpha \in \mathcal{A}\}$ which minimizes the mean-square error:

$$\mathbb{E}[\varepsilon^2] \stackrel{\text{def}}{=} \mathbb{E} \left[\left(G(X) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(X) \right)^2 \right]. \quad (38.19)$$

The set of coefficients \mathbf{y} is computed at once by solving:

$$\mathbf{y} = \arg \min_{\mathbf{y} \in \mathbb{R}^{\text{card } \mathcal{A}}} \mathbb{E} \left[\left(G(X) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(X) \right)^2 \right]. \quad (38.20)$$

In practice the discretized version of the problem is obtained by replacing the expectation operator in Eq. (38.20) by the empirical mean over a sample set:

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y} \in \mathbb{R}^{\text{card } \mathcal{A}}} \frac{1}{N} \sum_{i=1}^N \left(G(\mathbf{x}^{(i)}) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{x}^{(i)}) \right)^2. \quad (38.21)$$

In this expression, $\mathcal{X} = \{\mathbf{x}^{(i)}, i = 1, \dots, n\}$ is a sample set of points called *experimental design* (ED) that is typically obtained by Monte Carlo simulation of the input random vector X . To solve the least-square minimization problem in Eq. (38.21), the computational model G is first run for each point in the ED, and the results are stored in a vector $\mathcal{Y} = \{y^{(1)} = G(\mathbf{x}^{(1)}), \dots, y^{(n)} = G(\mathbf{x}^{(n)})\}^\top$. Then the so-called *information matrix* is calculated from the evaluation of the basis polynomials onto each point in the ED:

$$\mathbf{A} = \left\{ \mathbf{A}_{ij} \stackrel{\text{def}}{=} \Psi_j(\mathbf{x}^{(i)}), i = 1, \dots, n, j = 1, \dots, \text{card } \mathcal{A} \right\}. \quad (38.22)$$

The solution of the least-square minimization problem finally reads:

$$\hat{\mathbf{y}} = (\mathbf{y} \mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathcal{Y}. \quad (38.23)$$

The points used in the experimental design may be obtained from crude Monte Carlo simulation. However other types of designs are of common use, especially Latin hypercube sampling (LHS), see McKay et al. [45], or quasi-random sequences such as the Sobol' or Halton sequence [47]. The size of the experimental design is of crucial importance: it must be larger than the number of unknowns $\text{card } \mathcal{A}$ for the problem to be well posed. In practice we use the thumb rule $n \approx 2\text{--}3 \text{ card } \mathcal{A}$ [10].

The simple least-square approach summarized above does not allow one to cope with the curse of dimensionality. Indeed the standard truncation scheme requires approximately $3 \cdot \binom{d+p}{p}$ runs of the original model $G(\mathbf{x})$, which is in the order of 10^4 when, e.g., $d \geq 15$, $p \geq 5$. However, in practice most of the problems lead eventually to *sparse expansions*, i.e., PCE in which most of the coefficients are zero or negligible. In order to find directly the significant polynomials and associated coefficients, sparse PCEs have been introduced recently by Blatman and Sudret [11, 12], and Bieri and Schwab [9]. The recent developments make use of specific selection algorithms which, by solving a penalized least-square problem, lead by construction to sparse expansions. Of interest in this chapter is the use of the *least-angle regression* algorithm (LAR, Efron et al. [27]), which was introduced in the field of uncertainty quantification by Blatman and Sudret [14]. Details can be found in Sudret [64]. Note that other techniques based on compressive sensing have also been developed recently, see, e.g., Doostan and Owhadi [23], Sargsyan et al. [53], and Jakeman et al. [34].

2.5 Error Estimation

The truncation of the polynomial chaos expansion introduces an approximation error which may be computed a posteriori. Based on the data contained in the experimental design, the *empirical error* may be computed from Eq. (38.21) once least-square minimization problem has been solved:

$$\varepsilon_{\text{emp}} = \frac{1}{N} \sum_{i=1}^N \left(G(\mathbf{x}^{(i)}) - \sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha \Psi_\alpha(\mathbf{x}^{(i)}) \right)^2. \quad (38.24)$$

However, this estimator usually underestimates severely the mean-square error in Eq. (38.19). In particular, if the size N of the experimental design is close to the number of unknown coefficients $\text{card } \mathcal{A}$, the empirical error tends to zero, whereas the true mean square error does not.

A more robust error estimator can be derived based on the *cross-validation* technique. The experimental design is split into a training set and a validation set: the coefficients of the expansion are computed using the training set (Eq. (38.21)), whereas the error is estimated using the validation set. The *leave-one-out* cross-validation is a particular case in which all points but one are used to compute the coefficients. Setting aside $\mathbf{x}^{(i)} \in \mathcal{X}$, a PCE denoted by $G^{\text{PC}\backslash i}(X)$ is built up using

the experimental design $\mathcal{X} \setminus \mathbf{x}^{(i)} \stackrel{\text{def}}{=} \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i-1)}, \mathbf{x}^{(i+1)}, \dots, \mathbf{x}^{(n)}\}$. Then the error is computed at point $\mathbf{x}^{(i)}$:

$$\Delta_i \stackrel{\text{def}}{=} G(\mathbf{x}^{(i)}) - G^{\text{PC}\setminus i}(\mathbf{x}^{(i)}). \quad (38.25)$$

The LOO error is defined by:

$$\varepsilon_{\text{LOO}} = \frac{1}{n} \sum_{i=1}^n \Delta_i^2 = \frac{1}{n} \sum_{i=1}^n \left(G(\mathbf{x}^{(i)}) - G^{\text{PC}\setminus i}(\mathbf{x}^{(i)}) \right)^2. \quad (38.26)$$

After some algebra this reduces to:

$$\varepsilon_{\text{LOO}} = \frac{1}{n} \sum_{i=1}^n \left(\frac{G(\mathbf{x}^{(i)}) - G^{\text{PC}}(\mathbf{x}^{(i)})}{1 - h_i} \right)^2, \quad (38.27)$$

where h_i is the i -th diagonal term of the projection matrix $\mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ (matrix \mathbf{A} is defined in Eq. (38.22)) and $G^{\text{PC}}(\cdot)$ is now the PC expansion built up from the *full* experimental design \mathcal{X} .

As a conclusion, when using a least-square minimization technique to compute the coefficients of a PC expansion, an *a posteriori* estimator of the mean-square error is readily available. This allows one to compare PCEs obtained from different truncation schemes and select the best one according to the leave-one-out error estimate.

2.6 Post-processing for Sensitivity Analysis

2.6.1 Statistical Moments

The truncated PC expansion $\hat{Y} = G^{\text{PC}}(\mathbf{X}) = \sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha \Psi_\alpha(\mathbf{X})$ contains all the information about the statistical properties of the random output $Y = G(\mathbf{X})$. Due to the orthogonality of the PC basis, mean and standard deviation of \hat{Y} may be computed directly from the coefficients \hat{y} . Indeed, since $\Psi_0 \equiv 1$, we get $\mathbb{E}[\Psi_\alpha(\mathbf{X})] = 0 \quad \forall \alpha \neq 0$. Thus the mean value of \hat{Y} is the first term of the series:

$$\mathbb{E}[\hat{Y}] = \mathbb{E} \left[\sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha \Psi_\alpha(\mathbf{X}) \right] = \hat{y}_0. \quad (38.28)$$

Similarly, due to Eq. (38.10) the variance of \hat{Y} may be cast as:

$$\sigma_{\hat{Y}}^2 \stackrel{\text{def}}{=} \text{Var}[\hat{Y}] = \mathbb{E} \left[(\hat{Y} - \hat{y}_0)^2 \right] = \sum_{\substack{\alpha \in \mathcal{A} \\ \alpha \neq 0}} \hat{y}_\alpha^2. \quad (38.29)$$

In other words the mean and variance of the random response may be obtained by a mere combination of the PCE coefficients once the latter have been computed.

2.6.2 Sobol' Decomposition and Indices

As previously discussed in ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms” global sensitivity analysis is based on Sobol' decomposition of the computational model G (a.k.a *generalized ANOVA decomposition*), which reads [55]:

$$G(\mathbf{x}) = G_0 + \sum_{i=1}^d G_i(x_i) + \sum_{1 \leq i < j \leq d} G_{ij}(x_i, x_j) + \cdots + G_{12\dots d}(\mathbf{x}), \quad (38.30)$$

that is, as a sum of a constant G_0 , univariate functions $\{G_i(x_i), 1 \leq i \leq d\}$, bivariate functions $\{G_{ij}(x_i, x_j), 1 \leq i < j \leq d\}$, etc. A recursive construction is obtained by the following recurrence relationship:

$$\begin{aligned} G_0 &= \mathbb{E}[G(\mathbf{X})] \\ G_i(x_i) &= \mathbb{E}[G(\mathbf{X})|X_i = x_i] - G_0 \\ G_{ij}(x_i, x_j) &= \mathbb{E}[G(\mathbf{X})|X_i, X_j = x_i, x_j] - G_i(x_i) - G_j(x_j) - G_0. \end{aligned} \quad (38.31)$$

Using the *set notation* for indices:

$$A \stackrel{\text{def}}{=} \{i_1, \dots, i_s\} \subset \{1, \dots, d\}, \quad (38.32)$$

the Sobol' decomposition in Eq. (38.30) reads:

$$G(\mathbf{x}) = G_0 + \sum_{\substack{A \subset \{1, \dots, d\} \\ A \neq \emptyset}} G_A(\mathbf{x}_A), \quad (38.33)$$

where \mathbf{x}_A is a subvector of \mathbf{x} which only contains the components that belong to the index set A . It can be proven that the summands are orthogonal with each other:

$$\mathbb{E}[G_A(\mathbf{x}_A) G_B(\mathbf{x}_B)] = 0 \quad \forall A, B \subset \{1, \dots, d\}, \quad A \neq B. \quad (38.34)$$

Using this orthogonality property, one can decompose the variance of the model output:

$$V \stackrel{\text{def}}{=} \text{Var}[Y] = \text{Var} \left[\sum_{\substack{A \subset \{1, \dots, d\} \\ A \neq \emptyset}} G_A(\mathbf{x}_A) \right] = \sum_{\substack{A \subset \{1, \dots, d\} \\ A \neq \emptyset}} \text{Var}[G_A(\mathbf{x}_A)] \quad (38.35)$$

as the sum of so-called *partial variances* defined by:

$$V_A \stackrel{\text{def}}{=} \text{Var}[G_A(X_A)] = \mathbb{E}[G_A^2(X_A)]. \quad (38.36)$$

The Sobol' index attached to each subset of variables $A \stackrel{\text{def}}{=} \{i_1, \dots, i_s\} \subset \{1, \dots, d\}$ is finally defined by:

$$S_A = \frac{V_A}{V} = \frac{\text{Var}[G_A(X_A)]}{\text{Var}[Y]}. \quad (38.37)$$

First-order Sobol' indices quantify the portion of the total variance V that can be apportioned to the sole input variable X_i :

$$S_i = \frac{V_i}{V} = \frac{\text{Var}[G_i(X_i)]}{\text{Var}[Y]}. \quad (38.38)$$

Second-order indices quantify the joint effect of variables (X_i, X_j) that cannot be explained by each single variable separately:

$$S_{ij} = \frac{V_{ij}}{V} = \frac{\text{Var}[G_{ij}(X_i, X_j)]}{\text{Var}[Y]}. \quad (38.39)$$

Finally, *total* Sobol' indices S_i^{tot} quantify the total impact of a given parameter X_i including all of its interactions with other variables. They may be computed by the sum of the Sobol' indices of any order that contain X_i :

$$S_i^{\text{tot}} = \sum_{A \ni i} S_A. \quad (38.40)$$

Among other methods, Monte Carlo estimators of the various indices are available in the literature and thoroughly discussed in (see ▶ Chap. 35, “[Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms](#)”). Their computation usually requires 10^{3-4} runs of the model G for each index, which leads to a global computational cost that is not affordable when G is expensive to evaluate.

2.6.3 Sobol' Indices and PC Expansions

As can be seen by comparing Eqs. (38.18) and (38.33), both polynomial chaos expansions and Sobol' decomposition are sums of orthogonal functions. Taking advantage of this property, it is possible to derive *analytic expressions* for Sobol' indices based on a PC expansion, as originally shown in Sudret [61, 63]. For this purpose let us consider the set of multivariate polynomials Ψ_α which depend *only* on a subset of variables $A = \{i_1, \dots, i_s\} \subset \{1, \dots, d\}$:

$$\mathcal{A}_A = \{\alpha \in \mathcal{A} : \alpha_k \neq 0 \text{ if and only if } k \in A\}. \quad (38.41)$$

The union of all these sets is by construction equal to \mathcal{A} . Thus we can reorder the terms of the truncated PC expansion so as to exhibit the Sobol' decomposition:

$$G^{\text{PC}}(\mathbf{x}) = y_0 + \sum_{\substack{A \subset \{1, \dots, d\} \\ A \neq \emptyset}} G_A^{\text{PC}}(\mathbf{x}_A) \quad \text{where} \quad G_A^{\text{PC}}(\mathbf{x}_A) \stackrel{\text{def}}{=} \sum_{\alpha \in \mathcal{A}_A} y_\alpha \Psi_\alpha(\mathbf{x}). \quad (38.42)$$

Consequently, due to the orthogonality of the PC basis, the partial variance V_A reduces to:

$$V_A = \text{Var}\left[G_A^{\text{PC}}(\mathbf{x}_A)\right] = \sum_{\alpha \in \mathcal{A}_A} y_\alpha^2. \quad (38.43)$$

In other words, from a given PC expansion, the Sobol' indices *at any order* may be obtained by a mere combination of the squares of the coefficients. More specifically, the PC-based estimator of the first-order Sobol' indices read:

$$\hat{S}_i = \frac{\sum_{\alpha \in \mathcal{A}_i} \hat{y}_\alpha^2}{\sum_{\alpha \in \mathcal{A}, \alpha \neq 0} \hat{y}_\alpha^2} \quad \text{where} \quad \mathcal{A}_i = \{\alpha \in \mathcal{A} : \alpha_i > 0, \alpha_j \neq i = 0\}. \quad (38.44)$$

and the total PC-based Sobol' indices read:

$$\hat{S}_i^{\text{tot}} = \frac{\sum_{\alpha \in \mathcal{A}_i^{\text{tot}}} \hat{y}_\alpha^2}{\sum_{\alpha \in \mathcal{A}, \alpha \neq 0} \hat{y}_\alpha^2} \quad \mathcal{A}_i^{\text{tot}} = \{\alpha \in \mathcal{A} : \alpha_i > 0\}. \quad (38.45)$$

2.7 Summary

Polynomial chaos expansions allow one to cast the random response $G(X)$ as a truncated series expansion. By selecting an orthonormal basis w.r.t. the input parameter distributions, the corresponding coefficients can be given a straightforward interpretation: the first coefficient y_0 is the mean value of the model output, whereas the variance is the sum of the squares of the remaining coefficients. Similarly, the Sobol' indices are obtained by summing up the squares of suitable coefficients. Note that in low dimension ($d < 10$), the coefficients can be computed by solving a mere ordinary least-square problem. In higher dimensions advanced techniques leading to sparse expansions must be used to keep the total computational cost (measured in terms of the size N of the experimental design) affordable. Yet the post-processing to get the Sobol' indices from the PCE coefficients is independent of the technique used.

3 Gaussian Process-Based Sensitivity Analysis

3.1 A Short Introduction to Gaussian Processes

Let us consider a probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$, a measurable space $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ and an arbitrary set T . A stochastic process $Z(\mathbf{x})$, $\mathbf{x} \in T$, is Gaussian if and only if, for any finite subset $C \subset T$, the collection of random variables $Z(C)$ has a Gaussian joint distribution. In our framework, T and S represent the input and the output spaces. Therefore, we have $T = \mathbb{R}^d$ and $S = \mathbb{R}$.

A Gaussian process is entirely specified by its mean $m(\mathbf{x}) = \mathbb{E}_Z[Z(\mathbf{x})]$ and covariance function $k(\mathbf{x}, \mathbf{x}') = \text{cov}_Z(Z(\mathbf{x}), Z(\mathbf{x}'))$ where \mathbb{E}_Z and cov_Z denote the expectation and the covariance with respect to $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$. The covariance function $k(\mathbf{x}, \mathbf{x}')$ is a positive definite kernel. It is often considered stationary, i.e., $k(\mathbf{x}, \mathbf{x}')$ is a function of $\mathbf{x} - \mathbf{x}'$. The covariance kernel is the most important term of a Gaussian process regression. Indeed, it controls the smoothness and the scale of the approximation. A popular choice for $k(\mathbf{x}, \mathbf{x}')$ is the stationary isotropic squared exponential kernel defined as :

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2\theta^2} \|\mathbf{x} - \mathbf{x}'\|^2\right).$$

It is parametrized by the parameter θ – also called characteristic length scale or correlation length – and the variance parameter σ^2 . We give in Fig. 38.1 examples of realizations of Gaussian processes with stationary isotropic squared exponential kernels.

We observe that $m(\mathbf{x})$ is the trend around which the realizations vary, σ^2 controls the range of their variation, and θ controls their oscillation frequencies. We highlight that Gaussian processes with squared exponential covariance kernels are infinitely differentiable almost surely. As mentioned in [59], this choice of kernel can be unrealistic due to its strong regularity.

3.2 Gaussian Process Regression Models

The principle of Gaussian process regression is to consider that the prior knowledge about the computational model $G(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$, can be *modeled* by a Gaussian process $Z(\mathbf{x})$ with a mean denoted by $m(\mathbf{x})$ and a covariance kernel denoted by $k(\mathbf{x}, \mathbf{x}')$. Roughly speaking, we consider that the true response is a realization of $Z(\mathbf{x})$. Usually, the mean and the covariance are parametrized as follows:

$$m(\mathbf{x}) = \mathbf{f}^\top(\mathbf{x})\boldsymbol{\beta}, \quad (38.46)$$

and

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 r(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}), \quad (38.47)$$

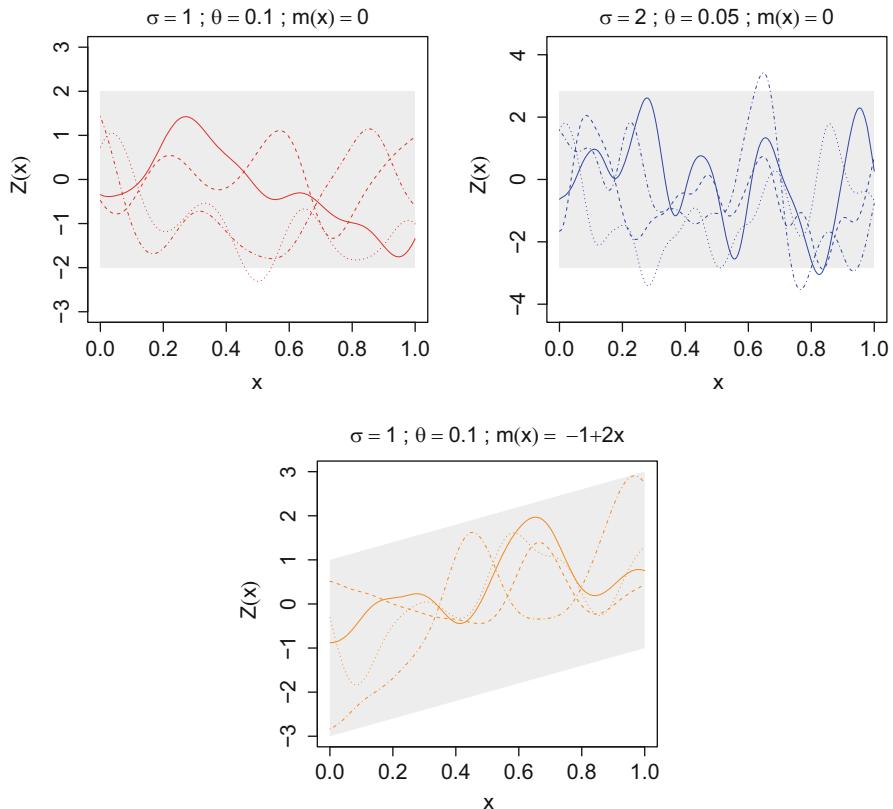


Fig. 38.1 Examples of Gaussian process realizations with squared exponential kernels and different means. The *shaded areas* represent the point-wise 95% confidence intervals

where $\mathbf{f}^T(\mathbf{x})$ is a vector of p prescribed functions, and $\boldsymbol{\beta}$, σ^2 , and $\boldsymbol{\theta}$ have to be estimated. The mean function $m(\mathbf{x})$ describes the trend and the covariance kernel $k(\mathbf{x}, \mathbf{x}')$ describes the regularity and characteristic length scale of the model.

3.2.1 Predictive Distribution

Consider an experimental design $\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$, $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and the corresponding model responses $\mathcal{Y} = G(\mathcal{X})$. The predictive distribution of $G(\mathbf{x})$ is given by:

$$[Z(\mathbf{x}) | Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \boldsymbol{\theta}] \sim \text{GP}(m_n(\mathbf{x}), k_n(\mathbf{x}, \mathbf{x}')), \quad (38.48)$$

where GP stands for “Gaussian Process,”

$$m_n(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\bar{\boldsymbol{\beta}} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathcal{Y} - \mathbf{F}\bar{\boldsymbol{\beta}}), \quad (38.49)$$

$$k_n(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 - (\mathbf{f}^\top(\mathbf{x}) \mathbf{r}^\top(\mathbf{x})) \begin{pmatrix} 0 & \mathbf{F}^\top \\ \mathbf{F} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \mathbf{f}(\mathbf{x}') \\ \mathbf{r}(\mathbf{x}') \end{pmatrix} \right). \quad (38.50)$$

In these expressions $\mathbf{R} = [r(\mathbf{x}^i, \mathbf{x}^j; \boldsymbol{\theta})]_{i,j=1,\dots,n}$, $\mathbf{r}(\mathbf{x}) = [r(\mathbf{x}, \mathbf{x}^{(i)}; \boldsymbol{\theta})]_{i=1,\dots,n}$, $\mathbf{F} = [\mathbf{f}^\top(\mathbf{x}^{(i)})]_{i=1,\dots,n}$ and

$$\bar{\boldsymbol{\beta}} = (\mathbf{F}^\top \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^\top \mathbf{R}^{-1} \mathcal{Y}. \quad (38.51)$$

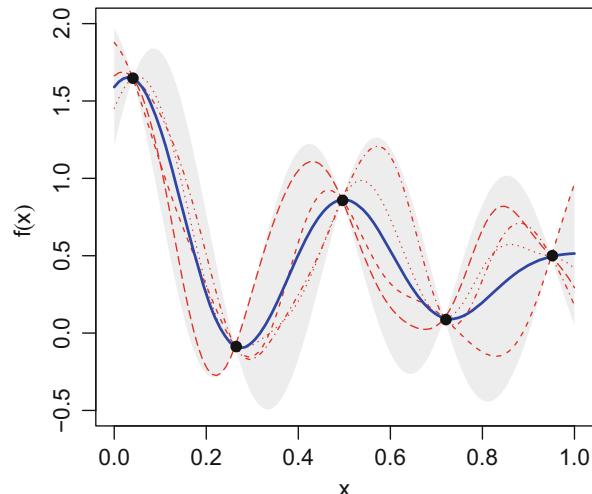
The term $\bar{\boldsymbol{\beta}}$ denotes the posterior distribution mode of $\boldsymbol{\beta}$ obtained from the improper non-informative prior distribution $\pi(\boldsymbol{\beta}) \propto 1$ [50].

Remark. The predictive distribution is given by the Gaussian process $Z(\mathbf{x})$ conditioned by the known observations \mathcal{Y} . The Gaussian process regression metamodel is given by the conditional expectation $m_n(x)$ and its mean-square error is given by the conditional variance $k_n(\mathbf{x}, \mathbf{x})$. An illustration of $m_n(x)$ and $k_n(\mathbf{x}, \mathbf{x})$ is given in Fig. 38.2.

The reader can note that the predictive distribution (38.48) integrates the posterior distribution of $\boldsymbol{\beta}$. However, the hyper-parameters σ^2 and $\boldsymbol{\theta}$ are not known in practice and shall be estimated with the maximum likelihood method [32, 52] or a cross-validation strategy [3]. Then, their estimates are plugged in the predictive distribution. The restricted maximum likelihood estimate of σ^2 is given by:

$$\hat{\sigma}^2 = \frac{(\mathcal{Y} - \mathbf{F}\bar{\boldsymbol{\beta}})^\top \mathbf{R}^{-1} (\mathcal{Y} - \mathbf{F}\bar{\boldsymbol{\beta}})}{n - p}. \quad (38.52)$$

Fig. 38.2 Examples of predictive distribution. The *solid line* represents the mean of the predictive distribution, the *nonsolid lines* represent some of its realizations, and the *shaded area* represents the 95% confidence intervals based on the variance of the predictive distribution



Unfortunately, such a closed form expression does not exist for θ and it has to be numerically estimated.

Remark. Gaussian process regression can easily be extended to the case of noisy observations. Let us suppose that \mathcal{Y} is tainted by a white Gaussian noise ε :

$$\mathcal{Y}_{\text{obs}} = \mathcal{Y} + \sigma_\varepsilon(\mathbf{X})\varepsilon.$$

The term $\sigma_\varepsilon(\mathbf{X})$ represents the standard deviation of the observation noise. The mean and the covariance of the predictive distribution $[Z(\mathbf{x})_{\text{obs}} | Z(\mathbf{X}) = \mathcal{Y}_{\text{obs}}, \sigma^2, \theta]$ is then obtained by replacing in Equations (38.49), (38.50), and (38.51) the correlation matrix \mathbf{R} by $\sigma^2 \mathbf{R} + \Delta_\varepsilon$ where Δ_ε is a diagonal matrix given by :

$$\Delta_\varepsilon = \begin{pmatrix} \sigma_\varepsilon(\mathbf{x}^{(1)}) & & & \\ & \sigma_\varepsilon(\mathbf{x}^{(2)}) & & \\ & & \ddots & \\ & & & \sigma_\varepsilon(\mathbf{x}^{(n)}) \end{pmatrix}.$$

We emphasize that the closed form expression for the restricted maximum likelihood estimate of σ^2 does not exist anymore. Therefore, this parameter has to be numerically estimated.

3.2.2 Sequential Design

To improve the global accuracy of the GP model, it is usual to augment the initial design set \mathbf{X} with new points. An important feature of Gaussian process regression is that it provides an estimate of the model mean-square error through the term $k_n(\mathbf{x}, \mathbf{x}')$ (38.50) which can be used to select these new points. The most common but not efficient sequential criterion consists in adding the point $\mathbf{x}^{(n+1)}$ where the mean-square error is the largest:

$$\mathbf{x}^{(n+1)} = \arg \max_{\mathbf{x}} k_n(\mathbf{x}, \mathbf{x}). \quad (38.53)$$

More efficient criteria can be found in Bates et al. [4], van Beers and Kleijnen [6], and Le Gratiet and Cannamela [37].

3.2.3 Model Selection

To build up a GP model, the user has to make several choices. Indeed, the vector of functions $\mathbf{f}(\mathbf{x})$ and the class of the correlation kernel $r(\mathbf{x}, \mathbf{x}'; \theta)$ need to be set (see Rasmussen and Williams [49] for different examples of correlation kernels). These choices and the relevance of the model are tested a posteriori with a validation procedure. If the number n of observations is large, an external validation may be performed on a test set. Otherwise, a cross-validation procedure may be used. An interesting property of GP models is that a closed form expression exists for the

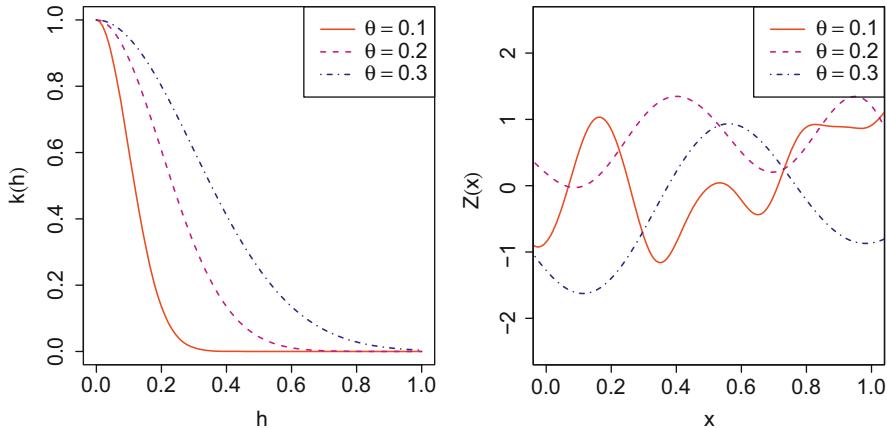


Fig. 38.3 The squared exponential kernel in function of $h = |x - x'|$ with different correlation lengths θ and examples of resulting Gaussian process realizations

cross-validation predictive distribution, see for instance Dubrule [25]. It allows for deriving efficient methods of parameter estimation [3] or sequential design [37].

Some usual stationary covariance kernel are listed below.

The squared exponential covariance function. The form of this kernel is given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2\theta^2} \|\mathbf{x} - \mathbf{x}'\|^2\right).$$

This covariance function corresponds to Gaussian processes which are infinitely differentiable in mean square and almost surely. We illustrate in Fig. 38.3 the one-dimensional squared exponential kernel with different correlation lengths and examples of resulting Gaussian process realizations.

The ν -Matérn covariance function. This covariance kernel is defined as follow (see [59]):

$$k_\nu(h) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2}||h||}{\theta} \right)^\nu K_\nu \left(\frac{\sqrt{2}\nu||h||}{\theta} \right),$$

where ν is the regularity parameter, K_ν is a modified Bessel function, and Γ is the Euler Gamma function. A Gaussian process with a ν -Matérn covariance kernel is ν -Hölder continuous in mean square and ν' -Hölder continuous almost

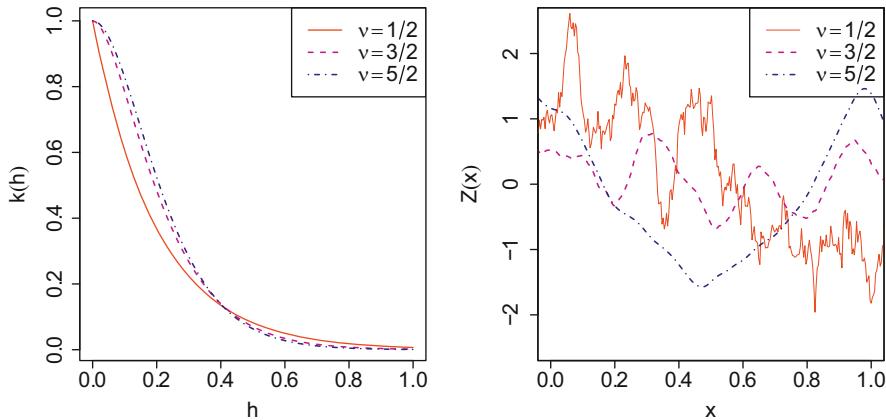


Fig. 38.4 The ν -Matérn kernel in function of $h = x - x'$ with different regularity parameters ν and examples of resulting Gaussian process realizations

surely with $\nu' < \nu$. Three popular choices of ν -Matérn covariance kernels are the ones for $\nu = 1/2$, $\nu = 3/2$, and $\nu = 5/2$:

$$k_{\nu=1/2}(h) = \exp\left(-\frac{\|h\|}{\theta}\right),$$

$$k_{\nu=3/2}(h) = \left(1 + \frac{\sqrt{3}\|h\|}{\theta}\right) \exp\left(-\frac{\sqrt{3}\|h\|}{\theta}\right),$$

and

$$k_{\nu=5/2}(h) = \left(1 + \frac{\sqrt{5}\|h\|}{\theta} + \frac{5\|h\|^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}\|h\|}{\theta}\right).$$

We illustrate in Fig. 38.4 the one-dimensional ν -Matérn kernel for different values of ν .

The γ -exponential covariance function. This kernel is defined as follow:

$$k_\gamma(h) = \exp\left(-\left(\frac{\|h\|}{\theta}\right)^\gamma\right).$$

For $\gamma < 2$ the corresponding Gaussian process are not differentiable in mean square, whereas for $\gamma = 2$ it is infinitely differentiable (it corresponds to the squared exponential kernel). We illustrate in Fig. 38.5 the one-dimensional γ -exponential kernel for different values of γ .

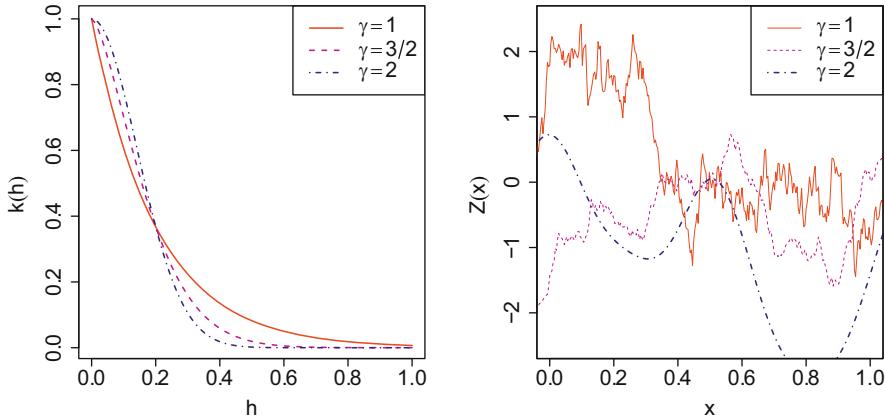


Fig. 38.5 The γ -exponential kernel in function of $h = \mathbf{x} - \mathbf{x}'$ with different regularity parameters γ and examples of resulting Gaussian process realizations

3.2.4 Sensitivity Analysis

To perform a sensitivity analysis from a GP model, two approaches are possible. The first one consists in substituting the true model $G(\mathbf{x})$ with the mean of the conditional Gaussian process $m_n(\mathbf{x})$ in (38.49). Then, a sensitivity analysis can be performed on the mean $m_n(\mathbf{x})$. This approach is used in Durrande et al. [26] which develops a class of covariance kernels dedicated to sensitivity analysis. However, it may provide biased sensitivity index estimates. Furthermore it does not allow one to quantify the error on the sensitivity indices due to the metamodel approximation. The second one consists in substituting $G(\mathbf{x})$ by a Gaussian process $Z_n(\mathbf{x})$ having the predictive distribution $[Z(\mathbf{x})|Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \theta]$ shown in (38.48) (see Oakley and O'Hagan [48], Marrel et al. [43], Le Gratiet et al. [38], Chastaing and Le Gratiet [18]). This approach makes it possible to quantify the uncertainty due to the metamodel approximation and allows for building unbiased index estimates.

Other interesting works on GP-based sensitivity analysis are the ones of Gramacy et al. [31], Gramacy and Taddy [30] to take into account non-stationarities, Storlie et al. [60] where a comparison between various metamodels with GP is performed, and Svenson et al. [67] to deal with massive data corrupted by a noise.

3.3 Main Effects Visualization

From now on, the input parameter $\mathbf{x} \in \mathbb{R}^d$ is considered as a random input vector $\mathbf{X} = (X_1, \dots, X_d)$ with independent components. Before focusing on variance-based sensitivity indices, the inference about the main effects is studied in this section. Main effects are a powerful tool to visualize the impact of each input variable on the model output (see, e.g., Oakley and O'Hagan [48]). The main effect of the group of input variables \mathbf{X}_A , $A \subset \{1, \dots, d\}$ is defined by $\mathbb{E}[G(\mathbf{X})|\mathbf{X}_A]$.

Since the original model G may be time-consuming to evaluate, it is substituted for by its approximation, i.e., $G(\mathbf{X}) \approx \mathbb{E}[Z_n(\mathbf{X})|\mathbf{X}_A]$, where $Z_n(\mathbf{x}) \sim [Z(\mathbf{x})|Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \theta]$. Since $\mathbb{E}[Z_n(\mathbf{X})|\mathbf{X}_A]$ is a linear transformation of the Gaussian process $Z_n(\mathbf{x})$, it is also a Gaussian process. The expectations, variances, and covariances with respect to the posterior distribution of $[Z(\mathbf{x})|Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \theta]$ are denoted by $\mathbb{E}_Z[\cdot]$, $\text{Var}_Z(\cdot)$ and $\text{Cov}_Z(\cdot, \cdot)$. Then, we have:

$$\mathbb{E}[Z_n(\mathbf{X})|\mathbf{X}_A] \sim \text{GP}(\mathbb{E}[m_n(\mathbf{X})|\mathbf{X}_A], \mathbb{E}[\mathbb{E}[k_n(\mathbf{X}, \mathbf{X}')|\mathbf{X}_A]|\mathbf{X}'_A]). \quad (38.54)$$

The term $\mathbb{E}[m_n(\mathbf{X})]$ represents the approximation of $\mathbb{E}[G(\mathbf{X})|\mathbf{X}_A]$, and $\mathbb{E}[\mathbb{E}[k_n(\mathbf{X}, \mathbf{X}')|\mathbf{X}_A]|\mathbf{X}'_A]$ is the mean-square error due to the metamodel approximation. Therefore, with this method, one can quantify the error on the main effects due to the metamodel approximation. For more detail about this approach, the reader is referred to Oakley and O'Hagan [48] and Marrel et al. [43].

3.4 Variance of the Main Effects

Although the main effect enables one to visualize the impact of a group of variables on the model output, it does not quantify it. To perform such an analysis, consider the variance of the main effect:

$$V_A = \text{Var}(\mathbb{E}[Z_n(\mathbf{X})|\mathbf{X}_A]), \quad (38.55)$$

or its normalized version which corresponds to the Sobol' index:

$$S_A = \frac{V_A}{V} = \frac{\text{Var}(\mathbb{E}[Z_n(\mathbf{X})|\mathbf{X}_A])}{\text{Var}(Z_n(\mathbf{X}))}. \quad (38.56)$$

Sobol' indices are the most popular measures to carry out a sensitivity analysis since their value can easily be interpreted as the part of the total variance due to a group of variables. However, in contrary to the partial variance V_A , it does not provide information about the order of magnitude of the contribution to the model output variance of variable group \mathbf{X}_A .

3.4.1 Analytic Formulae

The above indices are studied in Oakley and O'Hagan [48] where the estimation of V_A and V is performed separately. Indeed, computing the Sobol' index S_A requires considering the joint distribution of V_A and V , which makes it impossible to derive analytic formulae. According to Oakley and O'Hagan [48], closed form expressions in terms of integrals can be obtained for the two quantities $\mathbb{E}_Z[V_A]$ and $\text{Var}_Z(V_A)$. The quantity $\mathbb{E}_Z[V_A]$ is the sensitivity measure and $\text{Var}_Z(V_A)$ represents the error due to the metamodel approximation. Nevertheless, V_A is not a linear transform of $Z_n(\mathbf{X})$ and its full distribution cannot be established. We note that Marrel et al. [44] suggest a strategy to efficiently simulate V_A .

3.4.2 Variance Estimates with Monte Carlo Integration

To evaluate the Sobol' index S_A , it is possible to use the pick-freeze approaches presented previously in this chapter (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”) and in Sobol' [55], Sobol' et al. [57], and Janon et al. [35]. By considering the formula given in Sobol' [55], S_A can be approximated by:

$$S_{A,N} = \frac{\frac{1}{N} \sum_{i=1}^N Z_n(\mathbf{X}^{(i)}) Z_n(\mathbf{X}_A^{(i)}) - \left(\frac{1}{2N} \sum_{i=1}^N Z_n(\mathbf{X}^{(i)}) + Z_n(\mathbf{X}_A^{(i)}) \right)^2}{\frac{1}{N} \sum_{i=1}^N Z_n(\mathbf{X}^{(i)})^2 - \left(\frac{1}{2N} \sum_{i=1}^N Z_n(\mathbf{X}^{(i)}) + Z_n(\mathbf{X}_A^{(i)}) \right)^2}, \quad (38.57)$$

where $(\mathbf{X}^{(i)}, \mathbf{X}_A^{(i)})_{i=1,\dots,N}$ is a N -sample from the random variable $(\mathbf{X}, \mathbf{X}^A)$.

In particular, this approach avoids to compute the integrals presented in Oakley and O'Hagan [48] and thus simplify the estimation of V_A and V . Furthermore, it takes into account their joint distribution.

Remark. This result can easily be extended to the total Sobol' index $S_i^{\text{tot}} = \sum_{A \supset i} S_A$. The reader is referred to Sobol' et al. [57] and ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms” for examples of pick-freeze estimates of S_A^{tot} .

3.5 Numerical Estimates of Sobol' Indices by Gaussian Process Sampling

The sensitivity index $S_{A,N}$ (38.57) is obtained after substituting the Gaussian process $Z_n(\mathbf{x})$ for the original computational model $G(\mathbf{x})$. Therefore, it is a random variable defined on the same probability space as $Z_n(\mathbf{x})$. The aim of this section is to present a simple methodology to get a sample $S_{A,N}$ of S_A . From this sample, an estimate of S_A (38.56) and a quantification of its uncertainty can be deduced.

Sampling from the Gaussian Predictive Distribution

To obtain a realization of $S_{A,N}$, one has to obtain a sample of $Z_n(\mathbf{x})$ on $(\mathbf{X}^{(i)}, \mathbf{X}_A^{(i)})_{i=1,\dots,N}$ and then use Eq. (38.57). To deal with large N , an efficient strategy is to sample $Z_n(\mathbf{x})$ using the Kriging conditioning method, see, for example, Chilès and Delfiner [19]. Consider first the unconditioned, zero-mean Gaussian process:

$$\tilde{Z}(\mathbf{x}) = \text{GP}(0, k(\mathbf{x}, \mathbf{x}')). \quad (38.58)$$

Then, the Gaussian process:

$$\tilde{Z}_n(\mathbf{x}) = m_n(\mathbf{x}) - \tilde{m}_n(\mathbf{x}) + \tilde{Z}(\mathbf{x}), \quad (38.59)$$

where $\tilde{m}_n(\mathbf{x}) = \mathbf{f}^\top(\mathbf{x})\tilde{\boldsymbol{\beta}} + \mathbf{r}^\top(\mathbf{x})\mathbf{R}^{-1}\left(\tilde{Z}(\mathcal{X}) - \mathbf{F}\tilde{\boldsymbol{\beta}}\right)$ and $\tilde{\boldsymbol{\beta}} = (\mathbf{F}^\top\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{R}^{-1}$. $\tilde{Z}(\mathcal{X})$ has the same distribution as $Z_n(\mathbf{x})$. Therefore, one can compute realizations of $Z_n(\mathbf{x})$ from realizations of $\tilde{Z}(\mathbf{x})$. Since $\tilde{Z}(\mathbf{x})$ is not conditioned, the problem is numerically easier. Among the available Gaussian process sampling methods, several can be mentioned: Cholesky decomposition [49], Fourier spectral decomposition [59], Karhunen-Loeve spectral decomposition [49], and the propagative version of the Gibbs sampler [36].

Remark. Let suppose that a new point $\mathbf{x}^{(n+1)}$ is added to the experimental design \mathcal{X} . A classical result of conditional probability implies that the new predictive distribution $[Z(\mathbf{x})|Z(\mathcal{X}) = \mathcal{Y}, Z(\mathbf{x}^{(n+1)}) = G(\mathbf{x}^{(n+1)}), \sigma^2, \boldsymbol{\theta}]$ is identical to $[Z_n(\mathbf{x})|Z_n(\mathbf{x}^{(n+1)}) = G(\mathbf{x}^{(n+1)}), \sigma^2, \boldsymbol{\theta}]$. Therefore, $Z_n(\mathbf{x})$ can be viewed as an unconditioned Gaussian process and, using the Kriging conditioning method, realizations of $[Z(\mathbf{x})|Z(\mathcal{X}) = \mathcal{Y}, Z(\mathbf{x}^{(n+1)}) = G(\mathbf{x}^{(n+1)}), \sigma^2, \boldsymbol{\theta}]$ can be derived from realizations of $Z_n(\mathbf{x})$ using the following equation:

$$Z_{n+1}(\mathbf{x}) = \frac{k_n(\mathbf{x}^{(n+1)}, \mathbf{X})}{k_n(\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+1)})} (G(\mathbf{x}^{(n+1)}) - Z_n(\mathbf{x}^{(n+1)})) + Z_n(\mathbf{x}). \quad (38.60)$$

Therefore, it is easy to calculate a new sample of $S_{A,N}$ after adding a new point $\mathbf{x}^{(n+1)}$ to the experimental design set \mathcal{X} . This result is used in the function “sobolGP” of the R CRAN package “sensitivity” to perform sequential design for sensitivity analysis using a stepwise uncertainty reduction (SUR) strategy [5,38,39].

3.5.1 Metamodel and Monte Carlo Sampling Errors

Let us denote by $\{S_{A,i}^N, i = 1, \dots, m\}$ a sample set of $S_{A,N}$ (38.57) of size $m > 0$. From this sample set, the following unbiased estimate of S_A can be deduced:

$$\hat{S}_A = \frac{1}{m} \sum_{i=1}^m S_{A,i}^N. \quad (38.61)$$

with variance:

$$\hat{\sigma}_{\hat{S}_A}^2 = \frac{1}{m-1} \sum_{i=1}^m (S_{A,i}^N - \hat{S}_A)^2. \quad (38.62)$$

The term $\hat{\sigma}_{\hat{S}_A}^2$ represents the uncertainty on the estimate of S_A (38.56) due to the metamodel approximation. Therefore, with the presented strategy, one can obtain both an unbiased estimate of the sensitivity index S_A and a quantification of its uncertainty. The same procedure can be used to estimate the total Sobol’ indices.

Finally it may be of interest to evaluate the error due to the pick-freeze approximation and to compare it to the error due to the metamodel. To do so, one can use the central limit theorem [18,35] or a bootstrap procedure [38]. In particular,

a methodology to evaluate the uncertainty on the sensitivity index due to both the Gaussian process and to the pick-freeze approximations is presented in Le Gratiet et al. [38]. It makes it possible to determine the value of N such that the pick-freeze approximation error is negligible compared to that of the metamodel.

3.6 Summary

Gaussian process regression makes it possible to perform sensitivity analysis on complex computational models using a limited number of model evaluations. An important feature of this method is that one can propagate the Gaussian process approximation error to the sensitivity index estimates. This allows the construction of sequential design strategies optimized for sensitivity analysis. It also provides a powerful tool to visualize the main effect of a group of variables and the uncertainty of its estimate. Another advantage of this approach is that Gaussian process regression has been thoroughly investigated in the literature and can be used in various problems. For example, the method can be adapted for non-stationary numerical models by using a treed Gaussian process as in Gramacy and Taddy [30]. Furthermore, it can also be used for multifidelity computer codes, i.e., codes which can be run at multiple level of accuracy (see Le Gratiet et al. [38]).

4 Applications

In this section, metamodel-based sensitivity analysis is illustrated on several academic and engineering examples.

4.1 Ishigami Function

The Ishigami function is given by:

$$G(x_1, x_2, x_3) = \sin(x_1) + 7 \sin(x_2)^2 + 0.1x_3^4 \sin(x_1). \quad (38.63)$$

The input distributions of X_1 , X_2 , and X_3 are uniform over the interval $[-\pi, \pi]^3$. This is a classical academic benchmark for sensitivity analysis, with first-order Sobol' indices:

$$S_1 = 0.3138 \quad S_2 = 0.4424 \quad S_3 = 0. \quad (38.64)$$

To compare polynomial chaos expansions and Gaussian process modeling on this example, experimental designs of different sizes n are considered. For each size n , 100 Latin hypercube sampling (LHS) sets are computed so as to replicate the procedure and assess statistical uncertainty.

For the polynomial chaos approach, the coefficients are calculated based on a degree-adaptive LARS strategy (for details, see Blatman and Sudret [14]), resulting in a sparse basis set. The maximum polynomial degree is adaptively selected in the interval $3 \leq p \leq 15$ based on LOO cross-validation error estimates (see Eq. (38.27)).

For the Gaussian process approach, a tensorized Matérn-5/2 covariance kernel is chosen (see Rasmussen and Williams [49]) with trend functions given by:

$$\mathbf{f}^T(\mathbf{x}) = \{1 \ x_2 \ x_2^2 \ x_1^3 \ x_2^3 \ x_1^4 \ x_2^4\}. \quad (38.65)$$

To select $\mathbf{f}^T(\mathbf{x})$, we use a classical stepwise regression (i.e., the model errors are considered independent and identically distributed according to a normal distribution) with the Bayesian information criterion and a bidirectional selection. This allows to merely obtain a relevant linear trend for the Gaussian process regression.

The hyper-parameters $\boldsymbol{\theta}$ are estimated with a leave-one-out cross-validation procedure, while the parameters $\boldsymbol{\beta}$ and σ^2 are estimated with a restricted maximum likelihood method.

First we illustrate in Fig. 38.6 the accuracy of the models with respect to the sample size n . The Nash-Sutcliffe model efficiency coefficient (also called predictivity coefficient) is defined as follows:

$$Q^2 = 1 - \frac{\sum_{i=1}^{n_{\text{test}}} (G(\mathbf{x}^{(i)}) - \hat{G}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^{n_{\text{test}}} (G(\mathbf{x}^{(i)}) - \bar{G})^2}, \quad \bar{G} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} G(\mathbf{x}^{(i)}), \quad (38.66)$$

where $\hat{G}(\mathbf{x}^{(i)})$ is the prediction given by the polynomial chaos or the Gaussian process regression model on the i^{th} point of a test sample of size $n_{\text{test}} = 10,000$. This test sample set is randomly generated from a uniform distribution. The closer Q^2 is to 1, the more accurate the metamodel is.

We emphasize that checking the metamodel accuracy (see Fig. 38.6) is very important since a metamodel-based sensitivity analysis provides sensitivity indices for the metamodel and not for the true model $G(\mathbf{x})$. Therefore, the estimated indices are relevant only if the considered surrogate model is accurate.

Figure 38.7 shows the Sobol' index estimates with respect to the sample size n . For the Gaussian process regression approach, the convergence is reached for $n = 100$. It corresponds to a Q^2 coefficient greater than 90%. Convergence of the PCE approach is somewhat faster, with comparable accuracy achieved with $n = 60$ and almost perfect accuracy for $n = 100$. Therefore, the convergence of the estimators of the Sobol' indices in Eqs. (38.36), (38.37), (38.38) and (38.39) is expected to be comparable to that of Q^2 . Note that the PCE approach also provides second-order and total Sobol' indices for free, as shown in Sudret [64].

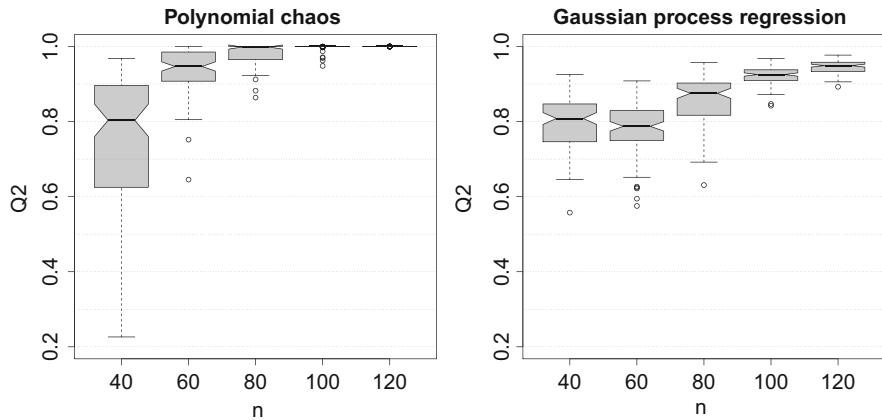


Fig. 38.6 Q^2 coefficient as a function of the sample size n for the Ishigami function. For each n , the box-plots represent the variations of Q^2 obtained over 100 LHS replications

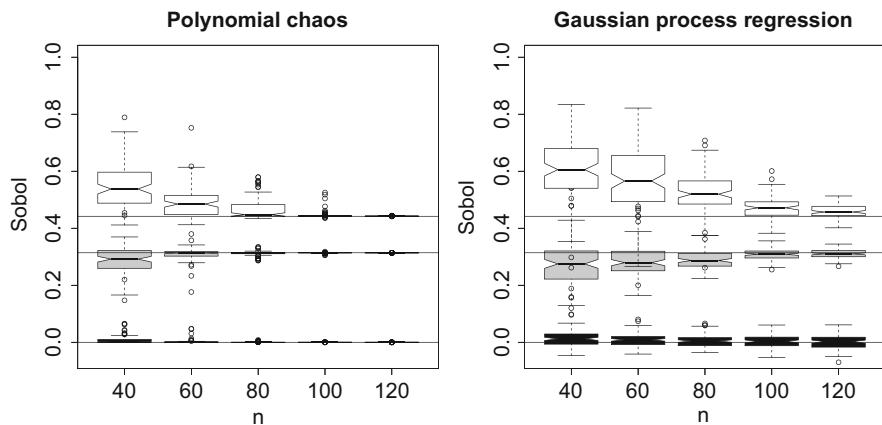


Fig. 38.7 First-order Sobol' index estimates as a function of the sample size n for the Ishigami function. The horizontal solid lines represent the exact values of S_1 , S_2 , and S_3 . For each n , the box-plot represents the variations obtained from 100 LHS replications. The validation set comprises $n_{\text{test}} = 10,000$ samples

4.2 G-Sobol' Function

The G-Sobol' function is given by :

$$G(\mathbf{x}) = \prod_{i=1}^d \frac{|4x_i - 2| + a_i}{1 + a_i}, \quad a_i \geq 0. \quad (38.67)$$

To benchmark the described metamodel-based sensitivity analysis methods in higher dimension, we select $d = 15$. The exact first-order Sobol' indices S_i are given by the following equations:

$$\begin{aligned} V_i &= \frac{1}{3(1+a_i)^2}, \quad i = 1, \dots, d, \\ V &= \prod_{i=1}^d (1+V_i) - 1, \\ S_i &= V_i/V. \end{aligned} \tag{38.68}$$

In this example, vector $\alpha = \{a_1, a_2, \dots, a_d\}$ is equal to:

$$\alpha = \{1, 2, 5, 10, 20, 50, 100, 500, 1000, 1000, 1000, 1000, 1000, 1000, 1000\}. \tag{38.69}$$

As in the previous section, different sample sizes n are considered and 100 LHS replications are computed for each n . Sparse polynomial chaos expansions are obtained with the same strategy as for the Ishigami function: adaptive polynomial degree selection with $3 \leq p \leq 15$ and LARS-based calculation of the coefficients. For the Gaussian process regression model, a tensorized Matérn-5/2 covariance kernel is considered with a constant trend function $f(x) = 1$. The hyper-parameter θ is estimated with a leave-one-out cross-validation procedure, and the parameters β and σ^2 are estimated with the maximum likelihood method.

The accuracy of the metamodels with respect to n is presented in Fig. 38.8. It is computed from a test sample set of size $n_{\text{test}} = 10,000$. The convergence of the estimates of the first four first-order Sobol' indices is represented in Fig. 38.9.

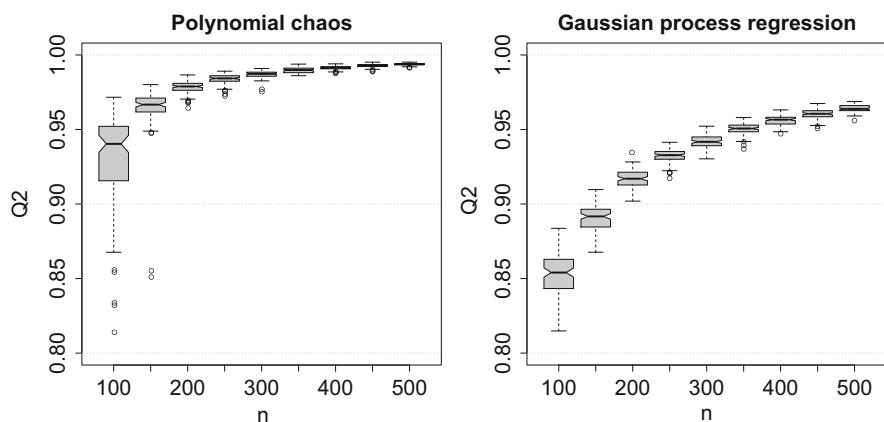


Fig. 38.8 Q^2 coefficient as a function of the sample size n for G-Sobol' academic example. For each n , the box-plot represents the variations of Q^2 obtained from 100 LHS

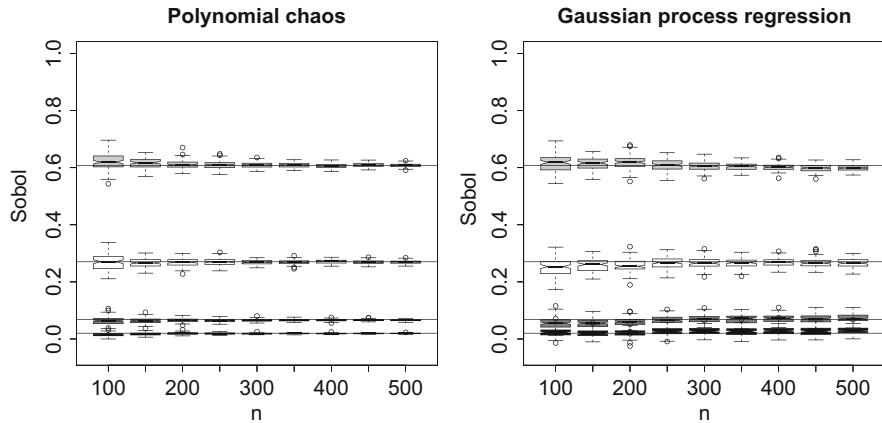


Fig. 38.9 Sobol' index estimates with respect to the sample size n for G-Sobol' function. The horizontal solid lines represent the true values of S_1 , S_2 , S_3 , and S_4 . For each n , the box-plot represents the variations obtained from 100 LHS

Table 38.2 Sobol' index estimates for the G-Sobol' function. The median and the root mean-square error (RMSE) of the estimates are given for $n = 100$ and $n = 500$

Index	Value	Polynomial chaos expansion				Gaussian process regression			
		Median	RMSE	Median	RMSE	Median	RMSE	Median	RMSE
100	500	100	500	100	500	100	500	100	500
S_1	0.604	0.619	0.607	0.034	0.007	0.618	0.599	0.035	0.012
S_2	0.268	0.270	0.269	0.027	0.005	0.233	0.245	0.046	0.026
S_3	0.067	0.063	0.065	0.014	0.003	0.045	0.070	0.029	0.016
S_4	0.020	0.014	0.019	0.008	0.001	0.008	0.023	0.018	0.013
S_5	0.005	0.002	0.005	0.003	0.001	8.6×10^{-4}	1.8×10^{-3}	0.014	0.013
S_6	0.001	0.000	7.2×10^{-4}	0.001	3.5×10^{-4}	6.4×10^{-4}	5.3×10^{-4}	0.013	0.013
S_7	0.000	0.000	1.1×10^{-4}	1.1×10^{-3}	1.4×10^{-4}	5.3×10^{-4}	3.0×10^{-4}	0.013	0.013
S_8	0.000	0.000	0.000	3.3×10^{-4}	1.7×10^{-5}	6.5×10^{-4}	7.1×10^{-4}	0.013	0.013
S_9	0.000	0.000	0.000	4.1×10^{-4}	1.1×10^{-5}	8.5×10^{-4}	4.4×10^{-4}	0.14	0.013
S_{10}	0.000	0.000	0.000	2.4×10^{-4}	1.1×10^{-5}	2.2×10^{-4}	1.7×10^{-4}	0.013	0.013
S_{11}	0.000	0.000	0.000	9.5×10^{-4}	1.2×10^{-5}	5.5×10^{-4}	-9.9×10^{-5}	0.013	0.013
S_{12}	0.000	0.000	0.000	5.2×10^{-4}	2.1×10^{-5}	2.6×10^{-4}	4.1×10^{-4}	0.013	0.013
S_{13}	0.000	0.000	0.000	5.1×10^{-4}	5.9×10^{-6}	9.8×10^{-4}	4.7×10^{-4}	0.013	0.013
S_{14}	0.000	0.000	0.000	8.8×10^{-4}	1.9×10^{-5}	1.8×10^{-4}	6.9×10^{-4}	0.013	0.013
S_{15}	0.000	0.000	0.000	8.6×10^{-4}	9.7×10^{-6}	7.2×10^{-4}	3.1×10^{-4}	0.013	0.013

Both metamodel-based estimations yield excellent results already with $n = 100$ samples in the experimental design. This is expected due to the good accuracy of both metamodels for all the n considered (see Fig. 38.8).

Finally, Table 38.2 provides the Sobol' index estimates median and root mean-square error for $n = 100$ and $n = 500$. As presented in Fig. 38.9, the estimates of the largest Sobol' indices are very accurate. Note that the remaining first-order

indices are insignificant. One can observe that the RMS error over the 100 LHS replications is slightly smaller when using PCE for both $n = 100$ and $n = 500$ ED points. Note that the second-order and total Sobol' indices are also available for free when using PCE.

4.3 Morris Function

The Morris function is given by:

$$G(\mathbf{x}) = \sum_{i=1}^{20} \beta_i w_i + \sum_{i < j}^{20} \beta_{ij} w_i w_j + \sum_{i < j < l}^{20} \beta_{ijl} w_i w_j w_l + 5w_1 w_2 w_3 w_4 \quad (38.70)$$

where $X_i \sim \mathcal{U}(0, 1)$, $i = 1, \dots, 20$, and $w_i = 2(x_i - 1/2)$ for all i except for $i = 3, 5, 7$ where $w_i = 2(1.1x_i/(x_i + 0.1) - 1/2)$. The coefficients are defined as $\beta_i = 20$, $i = 1, \dots, 10$; $\beta_{ij} = -15$, $i, j = 1, \dots, 6$; and $\beta_{ijl} = -10$, $i, j, l = 1, \dots, 5$. The remaining coefficients are set equal to $\beta_i = (-1)^i$ and $\beta_{ij} = (-1)^{i+j}$ and all the rest are zero. The reference values of the first-order Sobol' indices of the Morris function are calculated by a large Monte Carlo-based sensitivity analysis ($n = 10^6$).

As in the previous section, different sample sizes n are considered and 100 LHS replications are computed for each n . Sparse polynomial chaos expansions are obtained by adaptive polynomial degree selection $5 \leq p \leq 13$ and LARS-based calculation of the coefficients.

Furthermore, a tensorized Matérn-5/2 covariance kernel is used for the Gaussian process regression and the following trend has been selected with a stepwise regression procedure (as in Fig. 38.1):

$$\begin{aligned} \mathbf{f}^\top(\mathbf{x}) = \{ & 1 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{13} \ x_{14} \ x_{19} \ x_{1x2} \ x_{2x4} \ x_{1x4} \\ & x_{1x6} \ x_{2x6} \ x_{3x4} \ x_{4x6} \ x_{2x3} \ x_{1x5} \ x_{2x5} \ x_{3x6} \ x_{4x5} \ x_{1x3} \ x_{5x6} \ x_{3x5} \\ & x_{9x13} \ x_{7x8} \}. \end{aligned} \quad (38.71)$$

The accuracy of the metamodels with respect to n is presented in Fig. 38.10. It is computed from a test sample of size $n_{\text{test}} = 10,000$. As expected due to the complexity and dimensionality of the Morris function, both metamodels show a slower overall convergence rate with the number of samples with respect to the previous examples. Polynomial chaos expansions show in this case remarkably more scattering in their performance for smaller experimental designs with respect to Gaussian process regression. This is likely due to the comparatively large amount of prior information in the form of trend functions provided to the Gaussian process, not used for PCE.

The convergence of the estimates of three selected first-order Sobol' indices (the largest S_9 and two intermediate ones S_3 and S_8) is represented in Fig. 38.11. Both methods perform very well with as few as 250 samples. PCE, however,

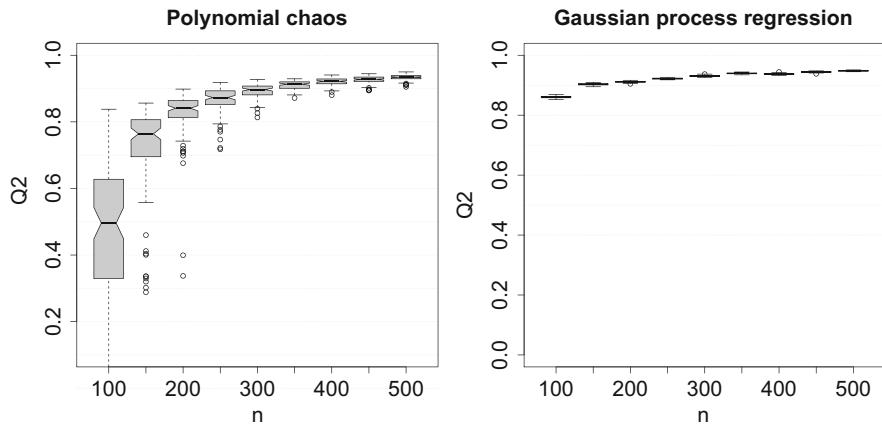


Fig. 38.10 Q^2 coefficient as a function of the sample size n for the Morris function example. For each n , the box-plot represents the variations of Q^2 obtained from 100 LHS

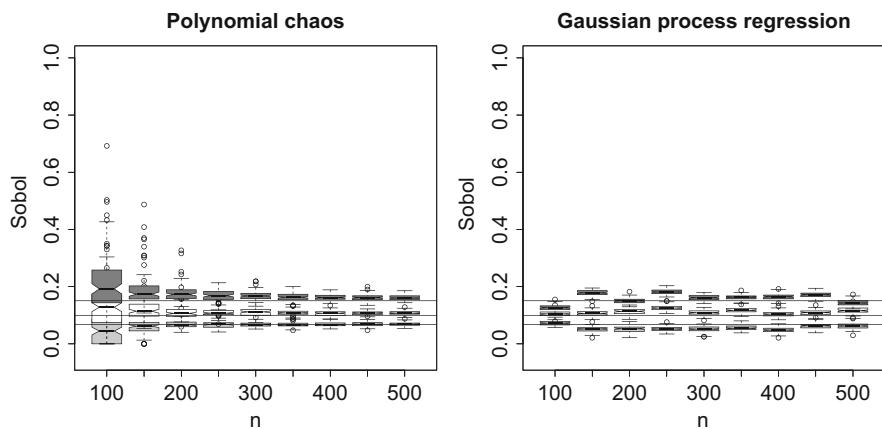


Fig. 38.11 First-order Sobol' index estimates as a function of the sample size n for the Morris function. The horizontal solid lines represent the exact values of S_3 , S_8 , and S_9 . For each n , the box-plot represents the variations obtained from 100 LHS replications

shows a more standard convergence behavior both in mean value in dispersion. Gaussian process regression retrieves the Sobol' estimates very accurately even with extremely small experimental designs, but no clear convergence pattern can be seen for larger datasets. This is due to the fact that the error on the Sobol' indices is essentially caused by the Monte Carlo errors. Therefore, increasing the accuracy of the Kriging model does not improve the accuracy of the index estimates.

Finally, Table 38.3 provides the a detailed breakdown of the Sobol' index estimates, including median and root mean-square error (RMSE), for $n = 100$ and $n = 500$.

Table 38.3 First-order Sobol' indices estimation for the Morris function. The median and the root mean-square error (RMSE) of the estimates are given for $n = 100$ and $n = 500$

		Polynomial chaos expansion				Gaussian process regression			
		Median		RMSE		Median		RMSE	
Index	Value	100	500	100	500	100	500	100	500
S_2	0.005	0.000	0.005	0.252	0.017	0.011	0.004	0.109	0.085
S_3	0.008	0.000	0.009	0.175	0.027	0.006	0.007	0.089	0.088
S_1	0.017	0.000	0.015	0.304	0.047	0.003	0.017	0.130	0.109
S_4	0.009	0.000	0.009	0.119	0.023	0.017	0.011	0.130	0.097
S_5	0.016	0.000	0.015	0.230	0.043	0.005	0.016	0.120	0.109
S_6	0.000	0.000	0.000	0.061	0.003	0.000	0.000	0.061	0.070
S_7	0.069	0.045	0.068	0.585	0.058	0.072	0.062	0.095	0.123
S_8	0.100	0.128	0.107	0.950	0.105	0.105	0.116	0.108	0.211
S_9	0.150	0.192	0.160	1.241	0.143	0.127	0.143	0.246	0.117
S_{10}	0.100	0.133	0.106	0.875	0.092	0.138	0.111	0.404	0.155
S_{11}	0.000	0.000	0.000	0.185	0.003	0.004	0.000	0.088	0.074
S_{12}	0.000	0.000	0.000	0.083	0.004	0.000	0.000	0.064	0.077
S_{13}	0.000	0.000	0.000	0.081	0.003	0.000	0.000	0.064	0.074
S_{14}	0.000	0.000	0.000	0.020	0.003	0.000	0.000	0.070	0.078
S_{15}	0.000	0.000	0.000	0.140	0.003	0.000	0.000	0.065	0.075
S_{16}	0.000	0.000	0.000	0.040	0.005	0.001	0.000	0.077	0.074
S_{17}	0.000	0.000	0.000	0.264	0.004	0.000	0.000	0.065	0.075
S_{18}	0.000	0.000	0.000	0.084	0.004	0.000	0.000	0.064	0.075
S_{19}	0.000	0.000	0.000	0.083	0.004	0.000	0.000	0.064	0.076
S_{20}	0.000	0.000	0.000	0.049	0.004	0.000	0.000	0.064	0.075

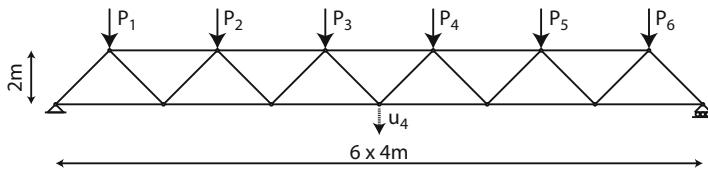


Fig. 38.12 Model of a truss structure with 23 members. The quantity of interest is the maximum displacement at mid-span u_4

4.4 Maximum Deflection of a Truss Structure

Sensitivity analysis is also of great interest for engineering models whose input parameters may have different distributions. As an example consider the elastic truss structure depicted in Fig. 38.12 (see, e.g., Blatman and Sudret [11]). This truss is made of two types of bars, namely, horizontal bars with cross-section A_1 and Young's modulus (stiffness) E_1 on the one hand and oblique bars with cross-section A_2 and Young's modulus (stiffness) E_2 on the other hand. The truss is loaded with

Table 38.4 Probabilistic input model of the truss structure

Variable	Distribution	Mean	Standard deviation
E_1, E_2 (Pa)	Lognormal	2.1×10^{11}	2.1×10^{10}
A_1 (m^2)	Lognormal	2.0×10^{-3}	2.0×10^{-4}
A_2 (m^2)	Lognormal	1.0×10^{-3}	1.0×10^{-4}
P_1-P_6 (N)	Gumbel	5.0×10^4	7.5×10^3

six vertical loads applied on the top chord. Of interest is the maximum vertical displacement (called deflection) at mid-span. This quantity is computed using a finite element model comprising elastic bar elements.

The various parameters describing the behavior of this truss structure are modeled by independent random variables that account for the uncertainty in both the physical properties of the structure and the applied loads. Their distributions are gathered in Table 38.4.

These input variables are collected in the random vector:

$$\mathbf{X} = \{E_1, E_2, A_1, A_2, P_1, \dots, P_6\}. \quad (38.72)$$

Using this notation, the maximal deflection of interest is cast as:

$$u_4 = G^{\text{FE}}(\mathbf{X}). \quad (38.73)$$

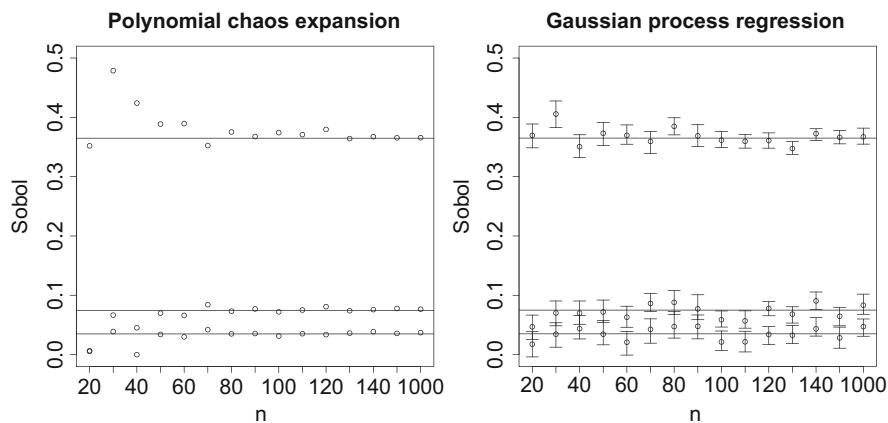
Different sparse polynomial chaos expansions are calculated assuming a maximal degree $3 < p < 10$ using LARS and the best expansion (in terms of smallest LOO error) is retained. For the Gaussian process regression model, a tensorized Matérn-5/2 covariance kernel is considered with a constant trend function $\mathbf{f}(\mathbf{x}) = 1$. The hyper-parameter θ is estimated with a leave-one-out cross-validation procedure, and the parameters β and σ^2 are estimated with the maximum likelihood method.

The first-order Sobol' indices obtained from PCE and GP metamodels are reported in Table 38.5 in the case when the experimental design is of size 100. In decreasing importance order, the important variables are the properties of the chords (horizontal bars), and then the loads close to mid-span, namely, P_3 and P_4 . The Sobol' indices of the latter are identical due to the symmetry of the model. Then come the loads P_2 and P_5 . The other variables (the loads P_1 and P_6 and the properties of the oblique bars) appear unimportant.

The estimates of the three largest first-order Sobol' indices which correspond to variables E_1 , P_3 , and P_5 obtained for various sizes n of the LHS experimental design are plotted in Fig. 38.13 as a function of n . The reference solution is obtained by Monte Carlo sampling with a sample set of size 6,000,000. Both PCE- and GP-based Sobol' indices converge to stable estimates as soon as $n \geq 60$. Furthermore, for the GP-based Sobol' index estimates, the 95% confidence intervals considering both the metamodel and the Monte Carlo errors are provided.

Table 38.5 Truss structure – First-order Sobol' indices

Variable	Reference	PCE	Gaussian Process
A_1	0.365	0.366	0.384
E_1	0.365	0.369	0.362
P_3	0.075	0.078	0.075
P_4	0.074	0.076	0.069
P_5	0.035	0.036	0.029
P_2	0.035	0.036	0.028
A_2	0.011	0.012	0.015
E_2	0.011	0.012	0.008
P_6	0.003	0.005	0.002
P_1	0.002	0.005	0.000

**Fig. 38.13** Truss structure – First-order Sobol' index estimates as a function of the sample size n for the truss model. The horizontal solid lines represent reference values of input variables E_1 , P_3 , and P_5 from a Monte Carlo estimate on 6,000,000 samples. For the Gaussian process regression, the segments represent the 95% confidence intervals taking both into account the metamodel and the Monte Carlo errors

5 Conclusions

Sobol' indices are recognized as good descriptors of the sensitivity of the output of a computational model to its various input parameters. Classical estimation methods based on Monte Carlo simulation are computationally expensive though. The required costs, in the order of $10^3 - 10^4$ model evaluations, are often not compatible with the advanced simulation models encountered in engineering applications.

For this reason, surrogate models may be first built up from a limited number of runs of the computational model (the so-called experimental design), and the sensitivity analysis is then carried out by substituting the surrogate model for the original one.

Polynomial chaos expansions and Gaussian processes are two popular methods that can be used for this purpose. The advantage of the PCE approach is that the Sobol' indices at any order may be computed *analytically* once the expansion is available. In this contribution, least-square minimization techniques are presented to compute the PCE coefficients, yet any intrusive or nonintrusive method could be used as an alternative.

In contrast Gaussian process surrogate models are used together with Monte Carlo simulation for estimating the Sobol' indices. The advantage of this approach is that the metamodel error can be included in the estimators. Note that bootstrap techniques can be used similarly to calculate and include metamodeling error also for PCE-based sensitivity analysis, as demonstrated by Dubreuil et al. [24].

As shown in the various comparisons, PCE and GP give similar accuracy (measured in terms of the Q^2 validation coefficient) for a given size of the experimental design in a broad range of applications. The replication of the analyses with different random designs of the same size show a smaller scatter using GP for extremely small designs, whereas PCE becomes more stable for medium-size designs. Selecting the best technique is in the end problem-dependent, and it is worth comparing the two approaches using the same experimental design, as it can be done in recent sensitivity analysis toolboxes such as OpenTURNS [2] and UQLab [41].

Finally it is worth mentioning that the so-called derivative-based global sensitivity measures (DGSM) originally introduced by Sobol' and Kucherenko [56] can also be computed using surrogate models. In particular, polynomial chaos expansions may be used to compute the DGSM analytically, as shown in Sudret and Mai [66]. Furthermore, Gaussian process regression can also be used as presented in De Lozzo and Marrel [21]. The recent combination of polynomial chaos expansions and Gaussian processes into *PC-Kriging* [54] also appears promising for estimating sensitivity indices from extremely small experimental designs.

References

1. Abramowitz, M., Stegun, I.: *Handbook of Mathematical Functions*. Dover Publications, New York (1970)
2. Andrianov, G., Burriel, S., Cambier, S., Dutfoy, A., Dutka-Malen, I., de Rocquigny, E., Sudret, B., Benjamin, P., Lebrun, R., Mangeant, F., Pendola, M.: Open TURNS, an open source initiative to Treat Uncertainties, Risks'N Statistics in a structured industrial approach. In: Proceedings of the ESREL'2007 Safety and Reliability Conference, Stavenger (2007)
3. Bachoc, F.: Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Comput. Stat. Data Anal.* **66**, 55–69 (2013)
4. Bates, R.A., Buck, R., Riccomagno, E., Wynn, H.: Experimental design and observation for large systems. *J. R. Stat. Soc. Ser. B* **58**(1), 77–94 (1996)
5. Bect, J., Ginsbourger, D., Li, L., Picheny, V., Vazquez, E.: Sequential design of computer experiments for the estimation of a probability of failure. *Stat. Comput.* **22**, 773–793 (2012)
6. van Beers, W., Kleijnen, J.: Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *Eur. J. Oper. Res.* **186**, 1099–1113 (2008)

7. Berveiller, M., Sudret, B., Lemaire, M.: Presentation of two methods for computing the response coefficients in stochastic finite element analysis. In: Proceedings of the 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque (2004)
8. Berveiller, M., Sudret, B., Lemaire, M.: Stochastic finite elements: a non intrusive approach by regression. *Eur. J. Comput. Mech.* **15**(1–3), 81–92 (2006)
9. Bieri, M., Schwab, C.: Sparse high order FEM for elliptic sPDEs. *Comput. Methods Appl. Mech. Eng.* **198**, 1149–1170 (2009)
10. Blatman, G.: Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis. PhD thesis, Université Blaise Pascal, Clermont-Ferrand (2009)
11. Blatman, G., Sudret, B.: Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *C. R. Mécanique* **336**(6), 518–523 (2008)
12. Blatman, G., Sudret, B.: An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Prob. Eng. Mech.* **25**, 183–197 (2010)
13. Blatman, G., Sudret, B.: Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **95**, 1216–1229 (2010)
14. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys.* **230**, 2345–2367 (2011)
15. Brown, S., Beck, J., Mahgerefteh, H., Fraga, E.: Global sensitivity analysis of the impact of impurities on CO₂ pipeline failure. *Reliab. Eng. Syst. Saf.* **115**, 43–54 (2013)
16. Buzzard, G.: Global sensitivity analysis using sparse grid interpolation and polynomial chaos. *Reliab. Eng. Syst. Saf.* **107**, 82–89 (2012)
17. Buzzard, G., Xiu, D.: Variance-based global sensitivity analysis via sparse-grid interpolation and cubature. *Commun. Comput. Phys.* **9**(3), 542–567 (2011)
18. Chastaing, G., Le Gratiet, L.: Anova decomposition of conditional Gaussian processes for sensitivity analysis with dependent inputs. *J. Stat. Comput. Simul.* **85**(11), 2164–2186 (2015)
19. Chilès, J., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. Wiley Series in Probability and Statistics (Applied Probability and Statistics Section). Wiley, New York (1999)
20. Crestaux, T., Le Maître, O., Martinez, J.M.: Polynomial chaos expansion for sensitivity analysis. *Reliab. Eng. Syst. Saf.* **94**(7), 1161–1172 (2009)
21. De Lozzo, M., Marrel, A.: Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel (2015, submitted)
22. Ditlevsen, O., Madsen, H.: Structural reliability methods. Wiley, Chichester (1996)
23. Doostan, A., Owhadi, H.: A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.* **230**(8), 3015–3034 (2011)
24. Dubreuil, S., Berveiller, M., Petitjean, F., Salatiñ, M.: Construction of bootstrap confidence intervals on sensitivity indices computed by polynomial chaos expansion. *Reliab. Eng. Syst. Saf.* **121**, 263–275 (2014)
25. Dubrule, O.: Cross validation of Kriging in a unique neighborhood. *Math. Geol.* **15**, 687–699 (1983)
26. Durrande, N., Ginsbourger, D., Roustant, O., Laurent, C.: Anova kernels and RKHS of zero mean functions for model-based sensitivity analysis. *J. Multivar. Anal.* **115**, 57–67 (2013)
27. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. *Ann. Stat.* **32**, 407–499 (2004)
28. Fajraoui, N., Ramasomanana, F., Younes, A., Mara, T., Ackerer, P., Guadagnini, A.: Use of global sensitivity analysis and polynomial chaos expansion for interpretation of nonreactive transport experiments in laboratory-scale porous media. *Water Resour. Res.* **47**(2) (2011)
29. Ghanem, R., Spanos, P.: Stochastic Finite Elements – A Spectral Approach. Springer, New York (1991). (Reedited by Dover Publications, Mineola, 2003)
30. Gramacy, R., Taddy, M.: Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models. *J. Stat. Softw.* **33**, 1–48 (2010)

31. Gramacy, R.B., Taddy, M., Wild, S.M.: Variable selection and sensitivity analysis using dynamic trees, with an application to computer performance tuning. *Ann. Appl. Stat.* **7**(1), 51–80 (2013)
32. Harville, D.: Maximum likelihood approaches to variance component estimation and to related problems. *J. Am. Stat. Assoc.* **72**(358), 320–338 (1977)
33. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Meloni, C., Dellino, G. (eds.) *Uncertainty management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Springer, New York (2015)
34. Jakeman, J., Eldred, M., Sargsyan, K.: Enhancing ℓ_1 -minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.* **289**, 18–34 (2015)
35. Janon, A., Klein, T., Lagnoux, A., Nodet, M., Prieur, C.: Asymptotic normality and efficiency of two Sobol' index estimators. *ESAIM: Prob. Stat.* **18**, 342–364 (2014)
36. Lantuéjoul, C., Desassis, N.: Simulation of a Gaussian random vector: a propagative version of the Gibbs sampler. In: The 9th International Geostatistics Congress, Oslo, p. 1747181, <https://hal-mines-paristech.archives-ouvertes.fr/hal-00709250> (2012)
37. Le Gratiet, L., Cannamela, C.: Cokriging-based sequential design strategies using fast cross-validation techniques for multifidelity computer codes. *Technometrics* **57**(3), 418–427 (2015)
38. Le Gratiet, L., Cannamela, C., Iooss, B.: A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA J. Uncertain. Quantif.* **2**(1), 336–363 (2014)
39. Le Gratiet L, Couplet, M., Iooss, B., Pronzato, L.: Planification d'expériences séquentielle pour l'analyse de sensibilité. In: 46èmes Journées de Statistique de la SFdS, Rennes (2014)
40. Lebrun, R., Dutfoy, A.: An innovating analysis of the Nataf transformation from the copula viewpoint. *Prob. Eng. Mech.* **24**(3), 312–320 (2009)
41. Marelli, S., Sudret, B.: UQLab: a framework for uncertainty quantification in Matlab. In: *Vulnerability, Uncertainty, and Risk (Proceedings of the 2nd International Conference on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool)*, pp. 2554–2563, doi:10.1061/9780784413609.257, <http://ascelibrary.org/doi/abs/10.1061/9780784413609.257>, <http://ascelibrary.org/doi/pdf/10.1061/9780784413609.257> (2014)
42. Marrel, A., Iooss, B., Van Dorpe, F., Volkova, E.: An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput. Stat. Data Anal.* **52**(10), 4731–4744 (2008)
43. Marrel, A., Iooss, B., Laurent, B., Roustant, O.: Calculations of Sobol indices for the Gaussian process metamodel. *Reliab. Eng. Syst. Saf.* **94**, 742–751 (2009)
44. Marrel, A., Iooss, B., Da Veiga, S., Ribatet, M.: Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput.* **22**(3), 833–847 (2010)
45. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **2**, 239–245 (1979)
46. Munoz Zuniga M, Kucherenko, S., Shah, N.: Metamodelling with independent and dependent inputs. *Comput. Phys. Commun.* **184**, 1570–1580 (2013)
47. Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia (1992)
48. Oakley, J., O'Hagan, A.: Probabilistic sensitivity analysis of complex models a Bayesian approach. *J. R. Stat. Soc. Ser. B* **66**(part 3), 751–769 (2004)
49. Rasmussen, C., Williams, C.: *Gaussian Processes for Machine Learning*. MIT, Cambridge (2006)
50. Robert, C.: *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, New York (2007)
51. Sandoval, E.H., Anstett-Collin, F., Basset, M.: Sensitivity study of dynamic systems using polynomial chaos. *Reliab. Eng. Syst. Saf.* **104**, 15–26 (2012)
52. Santner, T., Williams, B., Notz, W.: *The Design and Analysis of Computer Experiments*. Springer, New York (2003)

53. Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., Thornton, P.: Dimensionality reduction for complex models via Bayesian compressive sensing. *Int. J. Uncertain. Quantif.* **4**(1), 63–93 (2014)
54. Schöbi, R., Sudret, B., Wiart, J.: Polynomial-chaos-based Kriging. *Int. J. Uncertain. Quantif.* **5**(2), 171–193 (2015)
55. Sobol', I.: Sensitivity estimates for non linear mathematical models. *Math. Model. Comput. Exp.* **1**, 407–414 (1993)
56. Sobol', I., Kucherenko, S.: Derivative based global sensitivity measures and their link with global sensitivity indices. *Math. Comput. Simul.* **79**(10), 3009–3017 (2009)
57. Sobol', I., Tarantola, S., Gatelli, D., Kucherenko, S., Mauntz, W.: Estimating the approximation error when fixing unessential factors in global sensitivity analysis. *Reliab. Eng. Syst. Saf.* **92**(7), 957–960 (2007)
58. Soize, C., Ghanem, R.: Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM J. Sci. Comput.* **26**(2), 395–410 (2004)
59. Stein, M.: Interpolation of Spatial Data. Springer Series in Statistics. Springer, New York (1999)
60. Storlie, C., Swiler, L., Helton, J., Sallaberry, C.: Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliab. Eng. Syst. Saf.* **94**(11), 1735–1763 (2009)
61. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. In: Spanos, P., Deodatis, G. (eds.) In: Proceeding of the 5th International Conference on Computational Stochastic Mechanics (CSM5), Rhodos (2006)
62. Sudret, B.: Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods. Tech. rep., Université Blaise Pascal, Clermont-Ferrand, France, habilitation à diriger des recherches (229 pages) (2007)
63. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **93**, 964–979 (2008)
64. Sudret, B.: Polynomial chaos expansions and stochastic finite element methods. In: Phoon, K.K., Ching, J. (eds.) Risk and Reliability in Geotechnical Engineering. Taylor and Francis, Boca Raton (2015)
65. Sudret, B., Caniou, Y.: Analysis of covariance (ANCOVA) using polynomial chaos expansions. In: Deodatis, G. (ed.) Proceeding of the 11th International Conference on Structural Safety and Reliability (ICOSSAR'2013), New York (2013)
66. Sudret, B., Mai, C.V.: Computing derivative-based global sensitivity measures using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **134**, 241–250 (2015)
67. Svenson, J., Santner, T., Dean, A., Hyejung, M.: Estimating sensitivity indices based on Gaussian process metamodels with compactly supported correlation functions. *J. Stat. Plan. Inference* **114**, 160–172 (2014)
68. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D.: Screening, predicting, and computer experiments. *Technometrics* **34**(1), 15–25 (1992)
69. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
70. Younes, A., Mara, T., Fajraoui, N., Lehmann, F., Belfort, B., Beydoun, H.: Use of global sensitivity analysis to help assess unsaturated soil hydraulic parameters. *Vadose Zone J.* **12**(1) (2013)

Amandine Marrel, Nathalie Saint-Geours, and Matthias De Lozzo

Abstract

This section presents several sensitivity analysis methods to deal with spatial and/or temporal models. Focusing on the variance-based approach, solutions are proposed to perform global sensitivity analysis with functional inputs and outputs. Some of these solutions are illustrated on two industrial case studies: an environmental model for flood risk assessment and an atmospheric dispersion model for radionuclide release. These test cases are fully described at the beginning of the paper. Then a section is dedicated to spatiotemporal inputs and proposes several sensitivity analysis methods. The use of metamodels is also addressed. Pros and cons of the various methods are then discussed. In a subsequent section, solutions to deal with spatiotemporal outputs are proposed: aggregated, site, and block indices are described. The use of functional metamodels for sensitivity analysis purpose is also discussed.

Keywords

Spatiotemporal inputs • Spatiotemporal outputs • Metamodel • Macro-parameter • Joint metamodeling • Dimension reduction • Distance-based dissimilarity measure • Trigger input • Map labeling • Aggregated indices • Site indices • Block indices • Change of support

A. Marrel (✉) • M. De Lozzo

CEA, DEN, DER, Saint-Paul-lez-Durance, France

e-mail: amandine.marrel@cea.fr; matthias.delozzo@gmail.com

N. Saint-Geours

ITK - Predict & Decide, Clapiers, France

e-mail: nathalie.saint-geours@itk.fr

Contents

1	Introduction	1328
2	Environmental and Industrial Applications	1329
2.1	NOE Environmental Model for Flood Risk Assessment	1329
2.2	Ceres-Mithra Model for Radionuclide Atmospheric Dispersion	1330
3	Methods for Spatiotemporal Inputs	1331
3.1	Variance-Based Sensitivity Indices for Spatiotemporal Inputs (Without Metamodeling)	1334
3.2	Variance-Based Sensitivity Analysis with Metamodels	1337
3.3	Pros and Cons of Available Methods	1340
3.4	Application on NOE Test Case	1342
4	Methods for Spatiotemporal Outputs	1344
4.1	Sensitivity Index Maps	1344
4.2	Block Sensitivity Indices	1345
4.3	Aggregated Sensitivity Indices	1345
4.4	Use of Metamodels	1348
4.5	Application on NOE Test Case	1349
4.6	Application on Ceres-Mithra Test Case	1351
5	Conclusions	1355
	References	1355

1 Introduction

Most of the methods of sensitivity analysis (SA) tools presented in the previous papers are initially defined for scalar variables, while industrial and environmental applications often deal with more complex input or output parameters. Indeed, these parameters can be spatial, temporal, spatiotemporal, or functional in a more general way. Classical SA methods must be adapted to address the problems posed by this kind of data as input of numerical models but also as output. Among the factors hindering the use of SA methods in the domain of spatiotemporal data are the explosion of dimensionality problems and the lack of maturity of certain models. We will examine in particular several other limits: (i) spatial model input data generally exhibit some autocorrelation, yet conventional sensitivity analysis methods only consider independent scalar variables; (ii) notions of scale, support, and resolution, which play a prominent role in spatial modeling, are ignored in the formal frameworks of conventional sensitivity analysis methods; and (iii) the problem of the interpretation of SA in the case of functional output. Hence, there appears to be a great need to adapt sensitivity analysis methods to the specific context of spatially distributed modeling. Some ideas have already been provided in the literature to address this issue. First, in existing research related to sensitivity analysis, one may find some publications that deal with correlated input variables, but these studies rarely examine the particular case of spatial dependence. In addition, spatial statistics, and more specifically geostatistics, offer theoretical frameworks to describe the uncertainty weighting on spatially distributed data and to grasp the notions of spatial scale, support, or resolution. Geostatistics also provide tools to simulate these spatial uncertainties. However, this theoretical corpus has never been linked to that of sensitivity analysis of numerical models.

This paper aims at proposing solutions to perform SA with functional inputs and outputs, focusing on the variance-based global sensitivity analysis (VB-GSA). First of all, some industrial and environmental applications which involve spatial and/or temporal uncertain inputs and outputs are presented. Then, solutions recently proposed to deal with functional inputs and outputs are detailed, and in parallel, some of them are applied on the industrial test cases in order to illustrate their implementation and efficiency.

2 Environmental and Industrial Applications

2.1 NOE Environmental Model for Flood Risk Assessment

Flood risk research makes intensive use of numerical modeling to forecast flooding events, assess the impacts of potential flooding events in terms of human well-being and economic and social development, and design efficient flood management policies. These models simulate hydrological, hydraulic, and economic processes over a given study area. They are usually spatially distributed and often make use of geographic information systems (GIS) tools.

The NOE model is a spatially distributed code which is used to assess the economic impact of flood risk [11]. It calculates the so-called EAD risk indicator (expected annual damage [€/year]), which is defined as the expected economic losses due to flooding events over one year on a given study area. The EAD indicator is spatially distributed and mapped over the floodplain. Its estimation requires to compute flood damages and flood return intervals for a range of flood scenarios of various magnitudes (e.g., a 10-year flood, a 30-year flood, etc.). The return intervals of these scenarios are computed from a series of annual maximum flows at a gauging station. The estimation of economic damages caused by a given flood scenario (Fig. 39.1) is based on the combination of various input data: (i) a *hazard map* which gives maximum water depths, water velocity, and flood duration over the study area for this flood scenario; (ii) a *land use map* which identifies and locates the various assets at stake over the study area (e.g., buildings, agricultural land, shops, factories, etc.); and (iii) a set of so-called damage functions that describe the vulnerability of flood-exposed assets.

As a study area, we selected the Lower Orb River fluvial plain, located in the south of France (Hérault). This 63 sq. km area has a Mediterranean subhumid regime and suffers from frequent flooding events that may result in large economic losses. For example, the flood of Dec. 1995–Jan. 1996, with a peak discharge of 1,700 m³/s, affected approximately 15,000 people and caused a total amount of damage of 53 M€ [42].

The NOE model has both spatially distributed inputs and a spatial output. It also has two important characteristics that need to be stressed out. First, the model end user (e.g., a water manager) is interested in the whole output map of the EAD risk indicator, but he/she is also interested in knowing the aggregated value of the EAD risk indicator over one district, one city, or the entire floodplain. We

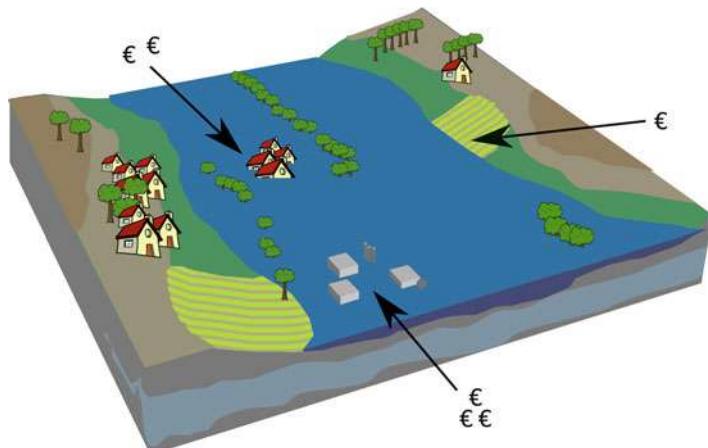


Fig. 39.1 NOE model: estimation of economic losses for a given flood scenario

thus say that the NOE model is *spatially additive*, which means that the output of interest over a given spatial support $v \subset \mathbb{R}^2$ is the sum $Y_v = \int_{z \in v} Y(z) dz$. Many other environmental models are also spatially additive (e.g., models that describe the rainfall over a study area), but some are not (e.g., a model that predicts the maximum pollutant concentration $\max \{Y(z)|z \in v\}$ over a study area). Second, the NOE model is also a *point-based* model, meaning that the value of model output at a given location $z \in \mathbb{R}^2$ only depends on the set of scalar inputs and on the values $W_i(z)$ of spatially distributed inputs *at that same location only* [21]. Such point-based models are encountered in various fields of environmental and earth sciences, whenever spatial interactions in the physical processes under study can be neglected in a first approximation.

2.2 Ceres-Mithra Model for Radionuclide Atmospheric Dispersion

In case of radioactive material release in the environment as a consequence of nuclear power plant accident or accidental releases due to other events, the first concern relates to the dispersion of these radioactive substances in the air. Atmospheric dispersion process is very efficient to characterize the transport of the released substances to long distance and define the geographical areas that are likely to be contaminated. The prediction of radionuclide dispersion in the atmosphere is an important element for the emergency response procedures and risk assessment. Numerical modeling is an essential tool for an accurate prediction of the plume contamination spread and an assessment of environmental risks associated with the site. The CEA has developed the Ceres-Mithra (C-M) application to model the radionuclide atmospheric dispersion and evaluate the consequences

on human health of radioisotope releases in the environment. This application is used either for crisis management or to perform assessment calculations for regulatory safety documents relative to nuclear facilities. Following an accidental release of radionuclides in the atmosphere, C-M evaluates instantaneous and time-integrated activity concentrations (resp. Bq.m^{-3} and Bq.s.m^{-3}) for different points and moments. To do this, several phenomena are simulated: transport, diffusion, impaction, and sedimentation. Atmospheric transport modeling is carried out with the Gaussian puff model [32]. This model assumes that a sequence of individual puffs of pollutant is released from the source. Different standard deviation equations can be used; Doury's formulas [10] which are function of travel time are the default option used in this study. Instantaneous and time-integrated volume activity concentrations are predicted.

However, C-M code depends on many uncertain input parameters related to the various phenomena involved: radionuclide parameters (e.g., deposition velocity), release parameters (height of release, source term activity), and meteorological parameters (wind speed and direction, atmospheric stability, and rain). Given the uncertain parameters, C-M code provides spatial maps of radionuclide volume and surface concentration for various moments defined by the scenario. These data can be instantaneous or integrated over time. C-M outputs can be viewed as functional outputs (spatiotemporal).

In this paper, we consider the scenario studied by Marrel et al. [30]. It is made up of a *cesium 137* (noted ^{137}Cs) simultaneous release occurring in two CEA installations, with distinct locations and heights of release, during one hour with a constant ^{137}Cs activity. The radionuclide atmospheric dispersion is studied during two hours from the beginning of the accidental releases, with time steps of 20 min. This dispersion process is particularly a function of the radionuclide deposition velocity and of the released radionuclide quantity. With respect to the weather conditions, there is no rain, and the temperature, the hydrometry, and the atmospheric stability are constant. Moreover, we assumed that the wind is blowing from the West, with an origin direction lying in $[249^\circ; 333^\circ]$; this situation represents a dry weather class which is the most likely one. Moreover, it is important to take into account the time variation of the wind speed and direction because of its role in the radioactive puff spread. Figure 39.2 represents the flowchart of this scenario. The objective is to assess how the uncertainties on the inputs can affect the C-M forecasts and more precisely to quantify the influence of each input variable (in their whole range of variations) on model responses. The challenge here is to extend the sensitivity analysis tools to an output which varies in space and time.

3 Methods for Spatiotemporal Inputs

Most SA techniques were initially designed for models with independent and scalar random inputs only: this is a first obstacle that limits its extension to spatial models. Indeed, in a spatial model, some model inputs are not scalar values but 2D raster

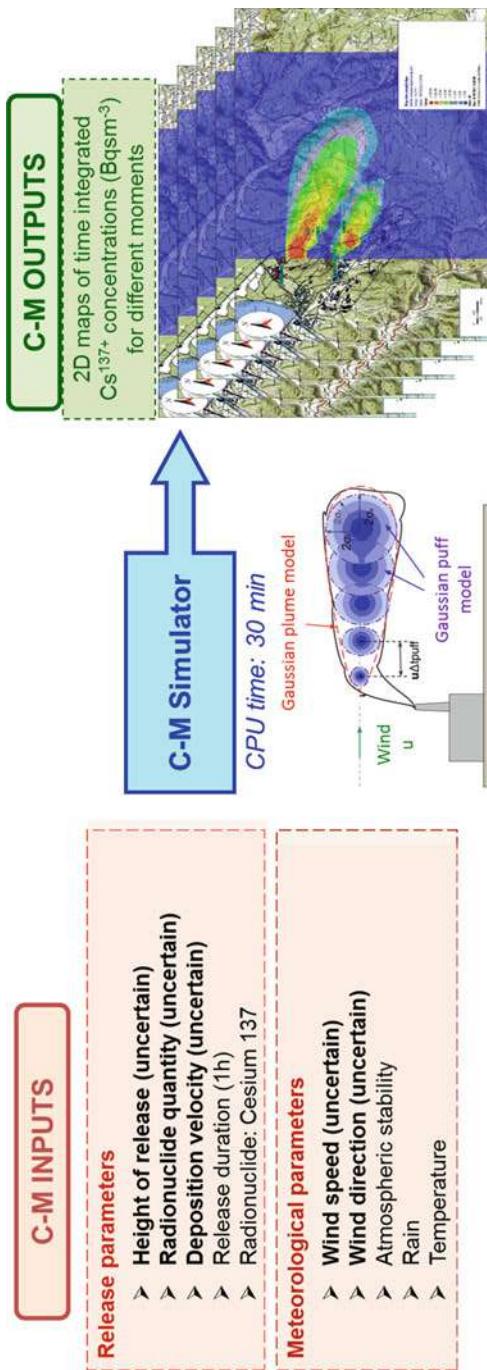


Fig. 39.2 Flowchart of the C-M model, with the uncertain inputs in bold and the spatial outputs of the simulator

or vector data and may exhibit some sort of spatial autocorrelation; the original framework of VB-GSA does not fit any longer in this case. In this paper, we explore this issue and investigate how variance-based sensitivity indices can be calculated in numerical models with spatially distributed inputs. Only models with scalar outputs are considered here (spatially distributed outputs will be studied in the second part of the paper). We consider the model f with d scalar inputs $\mathbf{X} = (X_1, \dots, X_d)$ and a functional input $\{W_z\}_{z \in \mathbb{D}_z}$ indexed by the continuous variable z defined on the domain \mathbb{D}_z , where \mathbb{D}_z can be a time interval or a spatial domain, for example. The model is defined by:

$$\begin{aligned} f : \mathbb{R}^d \times \mathbb{F} &\rightarrow \mathbb{R} \\ (\mathbf{X}, W_z) &\mapsto Y = f(\mathbf{X}, W_z) \end{aligned} \quad (39.1)$$

where \mathbb{F} is functional space composed of functions mapping from \mathbb{D}_z to \mathbb{R} . The inputs \mathbf{X} are modeled by random variables which are characterized by their probability density function $\mu_{\mathbf{X}}$, and $\{W_z\}_{z \in \mathbb{D}_z}$ is a stochastic process. $\{W_z\}_{z \in \mathbb{D}_z}$ can also be denoted $\{W(z) : z \in \mathbb{D}_z\}$. In the case of a spatiotemporal input, z can be time or spatial coordinates. One realization of $\{W_z\}_{z \in \mathbb{D}_z}$ is a function mapping from \mathbb{D}_z to \mathbb{R} , where \mathbb{D}_z can be a time interval or a spatial domain, for example.

In its initial form summarized in Paper 5 (see ►Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”), VB-GSA is designed to deal with scalar inputs X_i only, which are modeled as independent random variables. However, the definition of variance-based sensitivity indices can be extended without modification to vector or functional inputs, as long as the independence assumption between inputs is verified. Hence, the issue of computing variance-based sensitivity indices for spatiotemporal inputs is not a problem of *defining* such indices.

A first issue related to spatiotemporal inputs is *simulating random samples* for such inputs. This topic is extensively covered in the spatial statistics literature. Several approaches are available to model and simulate uncertainty on spatiotemporal data: stationary random fields, Markov random fields, random point processes, spectral decomposition, to name a few. This subject will not be further elaborated in this chapter. A second issue related to spatiotemporal inputs is purely technical: how to handle these inputs in the available open-source libraries dedicated to sensitivity analysis? How to store them? How to pass them to standardized routines that compute sensitivity indices? A third issue related to spatiotemporal inputs is how to handle these inputs in metamodeling.

A number of authors have recently introduced methods to tackle these issues. Iooss and Ribatet [24], Lilburne and Tarantola [27], and Iooss [23] make a partial state of the art of these approaches. In this section, we offer to make an updated state of the art of these methods, compare these methods on some analytical test cases, discuss their pros and cons, and give practice-oriented criteria to choose the appropriate method for a given problem.

3.1 Variance-Based Sensitivity Indices for Spatiotemporal Inputs (Without Metamodeling)

3.1.1 Macro-parameter

A first approach is to consider the spatiotemporal input $\{W(z) : z \in \mathbb{D}_z\}$, or more precisely its numerical representation, as a vector of m scalar inputs $\mathbf{W} = (W_1 \dots W_m)$. Following [24], we will use the term *macro-parameter* to refer to this method. Most often, the temporal or spatial domain \mathbb{D}_z is discretized as a regular grid with m elements:

$$\mathbb{D}_z \xrightarrow{\text{discretization}} (z^{(1)}, \dots, z^{(m)}) \in \mathbb{D}_z^m.$$

In this case, scalar inputs $\mathbf{W} = (W_1, \dots, W_m)$ are simply the values $(W(z^1), \dots, W(z^m))$ at the grid points. Spatiotemporal inputs may also be stored in a more complex format, for example, as a vector layer in a geographic information system, comprising numerous spatial objects with various alpha-numerical attributes. In such case, scalar inputs $\mathbf{W} = (W_1, \dots, W_m)$ are the values of the attributes of each spatial object in the layer, and m is a multiple of the number of objects. In both cases, the model (39.1) is replaced by its discretized version:

$$\begin{aligned} f^* : \mathbb{R}^d \times \mathbb{R}^m &\rightarrow \mathbb{R} \\ (\mathbf{X}, \mathbf{W}) &\mapsto Y = f^*(\mathbf{X}, \mathbf{W}) \end{aligned} \tag{39.2}$$

The scalar inputs W_i are modeled by random variables and characterized by their joint probability density function $\mu_{\mathbf{W}}$. In some cases, it may be possible to consider that the W_i are independent random variables: spatiotemporal correlation of uncertainty in $\{W(z) : z \in \mathbb{D}_z\}$ is then neglected. In other cases, random variables (W_1, \dots, W_m) are not independent, and their joint probability $\mu_{\mathbf{W}}$ must be fully characterized. The first- and total-order variance-based sensitivity indices of spatiotemporal input $\{W(z) : z \in \mathbb{D}_z\}$ are then computed as the variance-based sensitivity indices of the group of scalar inputs (W_1, \dots, W_m) .

The macro-parameter approach was used by [22] to carry out a sensitivity analysis of the GeoPEARL pesticide leaching model. In this work, each spatially distributed soil property (soil horizon thickness, clay or organic matter content, etc.) was represented on a spatial grid by a set of 258 scalar parameters W_i that were assumed to be statistically independent. Hence, spatial autocorrelation of uncertainty was neglected. For each soil property, sensitivity indices were estimated for the whole group (W_1, \dots, W_m) by a pseudo Monte Carlo procedure.

The main drawback of the macro-parameter approach is its computational burden. The number of scalar inputs that must be sampled to estimate sensitivity indices is equal to $d + m$ where d is the number of model inputs X_1, \dots, X_d and m is the dimension of spatiotemporal input $\{W(z) : z \in \mathbb{D}_z\}$ (e.g., the number of cells in a grid or number of spatial objects in a GIS vector layer). As soon as the

dimension m gets too large ($m \gg 100$), the large samples required to estimate sensitivity indices may be difficult to handle.

3.1.2 Dimension Reduction

The dimension reduction approach is similar to the macro-parameter method, except that it results in a partial loss of the information contained in the initial spatiotemporal input $\{W(z) : z \in \mathbb{D}_z\}$. The idea is to find a way to approximate $\{W(z) : z \in \mathbb{D}_z\}$ by a deterministic function ϕ of a small number of scalar inputs W_1, \dots, W_m , with m small—typically $m \leq 100$, much less than the dimension of the initial spatiotemporal data:

$$\forall z \in \mathbb{D}_z, \quad W(z) \approx \phi(W_1, \dots, W_m, z) \quad (39.3)$$

Scalar inputs $\mathbf{W} = (W_1, \dots, W_m)$ are modeled by random variables and characterized by their probability density function $\mu_{\mathbf{W}}$. Model (39.1) is then replaced by its discretized version (39.2). Variance-based sensitivity indices are computed either for each scalar parameter W_j if they are assumed independent or for the whole input group (W_1, \dots, W_m) .

Here are some examples of works based on this approach. Volkova et al. [46] carried out sensitivity analysis of a model of groundwater transport for radionuclide migration on a radwaste disposal site. To represent spatially distributed soil properties such as the hydraulic conductivity or infiltration, they divided up the spatial domain \mathbb{D}_z into four zones v_1 to v_4 ($m = 4$); each soil property was then described by a single scalar random variable W_i on each zone v_i , which was assumed to represent the average value of the soil property over the zone. All these random variables were assumed independent, and first- and total-order sensitivity indices were estimated for each of them. Romary [34] also uses a *dimension reduction* approach to perform the sensitivity analysis of an oil reservoir model in which one of the model inputs is a spatially distributed permeability field $\{W(z) : z \in \mathbb{D}_z\}$. This input field is first expanded on an orthogonal basis (Karhunen-Loëve expansion); the main coefficients W_i of this expansion are then considered as new, uncorrelated inputs, and sensitivity indices are estimated for each of them. A last example of the *dimension reduction* approach is given by [15], who performed the variance-based sensitivity analysis of the SWAT computer model (Soil and Water Assessment Tool). In the original SWAT model, the study area \mathbb{D}_z was divided into 11 topographical sub-basins and further into 156 hydrological response units, and 15 soil properties were given for each of these units. For sensitivity analysis, the spatial variability of soil properties was reduced, and each spatially distributed input variable was considered homogeneous over supports of larger size (either sub-basins or region composed of many hydrological response units grouped by soil types, land use types, or growing crops). This dimension reduction resulted in a downsizing of the total number of input variables, from more than 1000 in the initial model to only $m = 82$ in the reduced setting.

3.1.3 Switch or Trigger Input

The *switch input* or *trigger input* approach was first suggested by [7]. It is not specifically dedicated to spatiotemporal inputs and can also be used for scalar inputs. As we will explain it later, it is different in its principle compared to the macro-parameter and dimension reduction methods, and the practitioner should be careful in the interpretation of the resulting sensitivity indices. However, because this approach was used in the literature to deal with spatially distributed inputs, we find it useful to mention it here.

The core idea of the method is to modify model (39.1) by introducing a boolean random variable ξ such that $\mathbb{P}(\xi = 0) = \mathbb{P}(\xi = 1) = 1/2$:

$$\begin{aligned} f^* : \mathbb{R}^d \times \{0, 1\} &\rightarrow \mathbb{R} \\ (\mathbf{X}, \xi) &\mapsto Y = f^*(\mathbf{X}, \xi). \end{aligned} \quad (39.4)$$

ξ is named the *trigger* input. It is a switch between two situations: when $\xi = 0$, model output Y is evaluated using a fixed nominal value W_z^0 of input W_z , and when $\xi = 1$, model output Y is evaluated using a random realization of input W_z . Using this stratagem, the variance-based sensitivity indices S_ξ and S_ξ^{tot} of trigger input ξ are taken as a measure of the influence of spatiotemporal input W_z on the variance of model output Y .

Crosetto and Tarantola [7] applied this method on a GIS-based hydrologic model that simulate flood discharges from forecast rainfall on a given area. Five spatially distributed inputs $W_i(z)$ were considered: rainfall intensity maps, vector layer on flood-exposed assets, porosity maps, interception map, and soil moisture map. These inputs include 2D and 3D data, quantitative raster data, categorical raster data, and vector data. A stochastic process was defined for each of them to generate random error maps to be added to the initial maps. For one of the inputs (the rainfall intensity map), the stochastic process accounted for spatial autocorrelation in $W_i(z)$: random realizations of an error Gaussian random field were generated using Cholesky decomposition technique. For the other inputs, stochastic processes did not account for the spatial autocorrelation of uncertainty. Five trigger variables ξ_1 to ξ_5 were included in the sensitivity analysis, and first- and total-order sensitivity indices S_ξ and S_ξ^{tot} were estimated for each of them using the E-FAST procedure.

One must note that the sensitivity indices obtained using the trigger input approach do not have the same meaning than the sensitivity indices obtained with macro-parameter or dimension reduction methods. This point is discussed in further section (pros and cons of available methods).

3.1.4 Map Labeling

Lilburne and Tarantola [27] suggest to represent the uncertainty of spatiotemporal input W_z using a restricted set of n_L randomly generated realizations (n_L possibly large). Each realization is *labeled* by a single integer ranging from 1 to n_L and stored in some permanent memory space. The random realization associated with label l is denoted by $W_z^{(l)}$. These n_L realizations are considered as equiprobable.

Model (39.1) is then modified by introducing a random label L which follows a discrete uniform pdf in $\llbracket 1; n_L \rrbracket$:

$$\begin{aligned} f^* : \mathbb{R}^d \times \llbracket 1; n \rrbracket &\rightarrow \mathbb{R} \\ (\mathbf{X}, L) &\mapsto Y = f(\mathbf{X}, W_z^{(L)}). \end{aligned} \quad (39.5)$$

Hence, spatiotemporal input W_z is replaced by a reference to the random label L : for each model run, the sampled value $l^{(i)}$ of the label indicates that random realization $W_z^{(l^{(i)})}$ of the spatiotemporal input must be considered to evaluate model f for this run. First- and total-order sensitivity indices S_L and S_L^{tot} of random label L are then computed: these indices are taken as a measure of the influence of W_z on the variance of model output Y .

Lilburne and Tarantola [27] applied the *map labeling* approach to a spatial model for simulating nitrate transport from paddock to groundwater (AquiferSim). Four spatially distributed inputs were considered (soil map, land use map, river recharge map, and aquifer transmissivity map), and a small set of up to $n_L = 4$ random realizations was generated for each of them. The *map labeling* approach was also used by [35] to perform sensitivity analysis of a model for oil reservoir production forecasting. Two spatially distributed inputs were considered (basin geometry and heat flow map), and a small set of $n_L = 8$ (basin geometry) or $n_L = 4$ (heat flow) random realizations was generated for each of them.

3.2 Variance-Based Sensitivity Analysis with Metamodels

When the computer model is too CPU time-consuming to allow a sufficient number of model calculations, it can be approximated by a metamodel, estimated on a few simulations of the model (learning sample), as described in Paper 8 (see ▶ Chap. 38, “Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes”). Then, the sensitivity indices are directly estimated on the metamodel. In the case of functional inputs, the application of such approach raises the problem of taking into account a functional input in a metamodel. In what follows, several strategies are proposed to address this problem.

3.2.1 Metamodel with Dimension Reduction

At first, a reduction method such as projection on a functional basis can be used. The functions of the decomposition can be *known* (polynomial, Fourier, or wavelet basis) or estimated (e.g., functional proper component analysis). The main coefficients of the decomposition are selected (reduction model step), and a metamodel is estimated taking as inputs these coefficients. Then, using the obtained metamodel, VB-GSA methods defined for group of variables such as Sobol indices [25] can be applied to compute the global influence of this group of coefficients and, consequently, estimate the influence of the functional input.

3.2.2 Joint Metamodeling

As proposed by [24], another method could be to use a joint metamodel approach. This technique was initially introduced to approximate stochastic computer codes; [48] proposed to model the mean and dispersion of computer code outputs by two interlinked generalized linear models. This approach, called joint modeling, was previously studied in the context of experimental data modeling [31, 43]. Iooss and Ribatet [24], Gijbels et al. [17], and Marrel et al. [29] proposed to use more complex metamodels such as nonparametric models (generalized additive models, extended generalized linear models, and Gaussian processes, respectively) and to extend this method to functional input, considering this input as an uncontrollable parameter (i.e., governed by a seed variable). In this framework, the functional input is represented by a seed variable denoted X_ε lying in the set of all natural numbers \mathbb{N} , the other scalar inputs remain $\mathbf{X} = (X_1, \dots, X_d)$, and the model is defined by:

$$\begin{aligned} f : \mathbb{R}^d \times \mathbb{N} &\rightarrow \mathbb{R} \\ (\mathbf{X}, X_\varepsilon) &\mapsto Y = f(\mathbf{X}, X_\varepsilon) . \end{aligned} \quad (39.6)$$

Joint metamodels yield two metamodels, one for the mean and another for the dispersion component:

$$Y_m(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}) \quad (39.7)$$

$$Y_d(\mathbf{X}) = \text{Var}(Y|\mathbf{X}) = \mathbb{E}[(Y - Y_m(\mathbf{X}))^2|\mathbf{X}] . \quad (39.8)$$

Referring to the total variance formula, the variance of the output variable Y can be rewritten and deduced from the two metamodels:

$$\begin{aligned} \text{Var}(Y) &= \text{Var}[\mathbb{E}(Y|\mathbf{X})] + \mathbb{E}[\text{Var}(Y|\mathbf{X})] \\ &= \text{Var}[Y_m(\mathbf{X})] + \mathbb{E}[Y_d(\mathbf{X})] . \end{aligned} \quad (39.9)$$

Furthermore, the variance of Y is the sum of the contributions of all the input variables $\mathbf{X} = (X_1, \dots, X_d)$ and X_ε :

$$\text{Var}(Y) = V_\varepsilon(Y) + \sum_{i=1}^d \sum_{|J|=i} [V_J(Y) + V_{J\varepsilon}(Y)] \quad (39.10)$$

where $V_\varepsilon(Y) = \text{Var}[\mathbb{E}(Y|X_\varepsilon)]$, $V_i(Y) = \text{Var}[\mathbb{E}(Y|X_i)]$, $V_{i\varepsilon}(Y) = \text{Var}[\mathbb{E}(Y|X_i X_\varepsilon)] - V_i(Y) - V_\varepsilon(Y)$, $V_{ij}(Y) = \text{Var}[\mathbb{E}(Y|X_i X_j)] - V_i(Y) - V_j(Y)$, ... Variance of the mean component $Y_m(\mathbf{X})$ denoted hereafter Y_m can be also decomposed:

$$\text{Var}(Y_m) = \sum_{i=1}^d \sum_{|J|=i} V_J(Y_m) . \quad (39.11)$$

Note that

$$\begin{aligned} V_i(Y_m) &= \text{Var}[\mathbb{E}(Y_m|X_i)] \\ &= \text{Var}\{\mathbb{E}[\mathbb{E}(Y|\mathbf{X})|X_i]\} \\ &= \text{Var}[\mathbb{E}(Y|X_i)] = V_i(Y). \end{aligned} \quad (39.12)$$

Moreover, Sobol indices for variable Y according to input variables $\mathbf{X} = (X_i)_{i=1..d}$ can be derived from their usual definition given in Paper 5 (see ► Chap. 35, “Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms”):

$$S_J = \frac{V_J(Y_m)}{\text{Var}(Y)}. \quad (39.13)$$

These Sobol indices can be computed using the same classical Monte Carlo techniques: these algorithms are applied to the metamodel defined by the mean component Y_m of the joint model.

Thus, all terms contained in $\text{Var}[Y_m(\mathbf{X})]$ of Eq. (39.9) have been considered. Then, $\mathbb{E}[Y_d(\mathbf{X})]$ can be estimated by a simple numerical integration of $Y_d(\mathbf{X})$ following the distribution of \mathbf{X} . $Y_d(\mathbf{X})$ is evaluated with a metamodel, for example, the dispersion component of the joint model. Therefore, the total sensitivity index of X_ε is given by:

$$S_\varepsilon^{\text{tot}} = \frac{V_\varepsilon(Y) + \sum_{i=1}^d \sum_{|J|=i} V_{J\varepsilon}(Y)}{\text{Var}(Y)} = \frac{\mathbb{E}[Y_d(\mathbf{X})]}{\text{Var}(Y)}. \quad (39.14)$$

As $Y_d(\mathbf{X})$ is a positive random variable, positivity of $S_\varepsilon^{\text{tot}}$ is guaranteed. In practice, [29] propose to estimate $\text{Var}(Y)$ from the data or from simulations of the fitted joint model, using Eq. (39.9). If $\text{Var}(Y)$ is computed from the data, the authors recommend to estimate $\mathbb{E}[Y_d(\mathbf{X})]$ with $\text{Var}(Y) - \text{Var}[Y_m(\mathbf{X})]$ to satisfy Eq. (39.9).

If the seed variable X_ε manages one (or several) stochastic process (which is here the functional input), $S_\varepsilon^{\text{tot}}$ is interpreted as the total sensitivity index of this stochastic process. Consequently, $S_\varepsilon^{\text{tot}}$ could reveal if the functional input is nonsignificant. This approach can also be used for scalar inputs: X_ε can manage one of a group of scalar inputs.

The limitation of this approach is that only the total part of uncertainty related to W_ε is estimated; its individual effect is not distinguished from its interaction with the other parameters. However, these potential interactions could be pointed out, considering all the primary and total effects of all the other parameters. This sensitivity analysis could constitute, for example, a preliminary step before a VB-GSA taking into account more explicitly the functional input.

3.2.3 Distance-Based Metamodel

In order to take into account more explicitly the functional input in a metamodel, the uncertainty related to $\{W_t\}_{t \in T}$ can be represented by a distance function measuring the *dissimilarity* between any two realizations of this stochastic process. This approach is based on the concept of distance between parameter fields [3, 40, 44].

Several types of distances can be defined, but the important point is that the distance should be chosen such that it can be computed rapidly and help predicting if two parameter fields will lead to similar or different model outputs Y . For example, [45] propose to use the Hausdorff distance to quantify the differences in the geometry of complex 3D models (having different fault systems, horizon geometries, etc.). In their application of oil recovery from a production well, [40] define the square distance between two parameter fields as the integrated square difference between the responses computed for the two parameter fields with a fast streamline solver. The distance between every pair of parameter fields is computed and used as a base for mapping all the parameter fields in an abstract metric space. Going a step further, [4] use the same framework to formulate the inverse problem. In a similar way, several authors recently proposed to handle functional inputs in a Gaussian process metamodel using covariance functions adapted to the study of functional variables [12, 18], such as a covariance function depending on functional distances as those previously mentioned. Still in the framework of Gaussian process metamodel, [33] proposes, for time-varying inputs and outputs, a correlation function based on a weighted distance between input functions, suggesting a belief that at any time, the output is most sensitive to *recent* values of the input function.

Whatever the kind of distance chosen to take into account the functional inputs in the distance-based metamodel, the latter, once built, is then directly used to compute the VB-GSA indices with classical intensive Monte Carlo methods.

3.3 Pros and Cons of Available Methods

We discuss here the pros and cons of the various methods displayed in the previous section. All these methods intend to compute variance-based sensitivity indices that could measure the influence of an uncertain spatiotemporal input $\{W(z) : z \in \mathbb{D}_z\}$ on the variance of model output Y . However, they do not produce exactly the same information.

3.3.1 Methods Do Not Produce the Same Information

First, the information brought by the *macro-parameter* and *dimension reduction* methods is slightly different depending on whether sensitivity indices S_W , S_W^{tot} are computed for the whole group of inputs $\mathbf{W} = (W_1, \dots, W_m)$ or if separate sensitivity indices S_{W_j} , $S_{W_j}^{\text{tot}}$ are computed for each parameter W_j , $j \in \{1, \dots, m\}$.

Indeed, the set of first-order indices $(S_{W_j})_{j \in \llbracket 1; m \rrbracket}$ does not bring the same information as the first-order sensitivity index S_W of the whole spatiotemporal input W_z , because S_W accounts for the role of the interactions between $(W_j)_{j \in \llbracket 1; m \rrbracket}$, while first-order indices S_{W_j} do not. In the same way, the set of total-order indices

$(S_{W_j}^{\text{tot}})_{j \in \llbracket 1; m \rrbracket}$ does not bring the same information as the total-order sensitivity index S_W^{tot} , because interactions between scalar parameters W_j are counted multiple times in the set of indices $(S_{W_j}^{\text{tot}})_{j \in \llbracket 1; m \rrbracket}$, while it is counted just once in S_W^{tot} . Hence,

the set of sensitivity indices $(S_{W_j}, S_{W_j}^{\text{tot}})_{j \in [1; m]}$ does not yield a good measure of the main and total contributions of W_z to the variance of model output Y . One must also note that separate sensitivity indices S_{W_j} , $S_{W_j}^{\text{tot}}$ can be computed for each parameter W_j , $j \in \{1, \dots, m\}$ if and only if those parameters W_j are assumed to be independent. Otherwise, only group sensitivity indices should be computed, because usual variance-based sensitivity indices are not defined for dependent variables. Besides, the *dimension reduction with grouping* method only yields an approximation of sensitivity indices of W_z , as the total information initially contained in W_z is reduced to a small set of scalar parameters.

It should also be noted that in the *joint metamodeling* method, only total-order sensitivity index S_W^{tot} can be estimated, but first-order index S_W remains unknown.

Results of a numerical study [36] also clearly suggest that the *trigger* method does not produce the same information as other methods. Sensitivity index of the *trigger* parameter ξ proved to be always smaller than the measure S_W brought by the *macro-parameter* or *map labeling* methods. One possible explanation is that, in the *trigger* method, uncertainty in W_z is taken into account only when the sampled value of the trigger input is equal to $\xi = 1$, that is, one out of every two model runs in average ($\mathbb{P}(\xi = 0) = \mathbb{P}(\xi = 1) = 1/2$). Hence, the effect of uncertain model input W_z on the variance of model output Y is systematically underestimated. This result leads us to believe that the *trigger* method is not appropriate to deal with spatially distributed inputs in variance-based global sensitivity analysis.

3.3.2 Computational Cost

The methods displayed in the previous section also differ in their computational cost. The cost of a method depends on (i) the number of model simulations it requires and (ii) the number of random realizations of spatiotemporal input W_z needed. We compare these costs with the hypothesis that the same sampling procedure is used for all methods in which sensitivity indices are estimated from a pseudo Monte Carlo sample, with a base sample size N and a total sample size $N \cdot (d + 2)$ where d is the number of model inputs or groups of model inputs. Computational costs associated with each method are given in Table 39.1. It appears that the least CPU-intensive approaches are the *map labeling* and the *joint metamodeling* methods.

3.3.3 Spatiotemporal Data Handling

In many environmental case studies, spatiotemporal model inputs are stored in user-specific formats, for example, .tiff images or shapefiles to be read by geographic information systems. Handling such files may raise some technical issues when performing sensitivity analysis. Some methods displayed in the previous section allow to easily handle spatiotemporal inputs, no matter how they are stored on the computer. In the *map labeling*, *trigger*, and *joint metamodeling* methods, the random realizations of W_z can be generated using any stochastic process \mathcal{P} , based on any algorithm and any software. On the contrary, in the *macro-parameter* or *dimension reduction* methods, the spatiotemporal inputs must be represented by a vector of

Table 39.1 Methods for VB-GSA indices with spatiotemporal inputs. Column (a): number of map realizations needed. Column (b): coupling with a metamodel is possible. Column (c): possibility to cope with several spatial inputs. Column (d): Possibility to model spatiotemporal autocorrelation in $\{W(z) : z \in \mathbb{D}_z\}$. Column (e): sensitivity indices associated with W_z

Method	(a)	(b)	(c)	(d)	(e)	Comments
Macro-parameter	N	No	Yes	No	Indices of the group of inputs $(W_j)_{j \in \{1, \dots, m\}}$	Possibly intractable for large data
Dimension reduction	N	No	Yes	No	Indices of the group of inputs $(W_j)_{j \in \{1, \dots, m\}}$	Simplification of the input data
Trigger	$\sim N/2$	No	Yes	Yes	Indices of the trigger input ξ	A measure of sensitivity is obtained, which is not equal to variance-based sensitivity indices of $W(z)$.
Map labeling	$n_L < N$	No	Yes	Yes	Indices of the label input L	Spatially distributed input $W(z)$ is under-sampled compared to other inputs
Joint GAM metamodel	$\ll N$	Yes	No	Yes	Total-order index S_Z^{tot} only	Use of a metamodel

numbers or a matrix to be used in sensitivity analysis. It is then more difficult with this method to deal with spatiotemporal data stored in user-specific file formats.

Before illustrating some of the previously presented methods on the NOE case study, we can mention the recent work of [14] who have proposed a VB-GSA method in the specific case where the computer code is a functional linear model and the functional inputs are known stochastic processes. Nevertheless, as these two requirements are rarely met in practice, we will not detail this method here.

3.4 Application on NOE Test Case

Saint-Geours et al. [38] applied a VB-GSA to the NOE model. Here we briefly summarize their main results, considering a single scalar output Y , which is the total expected annual damage [€] over the entire Orb Delta.

3.4.1 Modeling Uncertainty on Spatial Inputs

The hazard map $(W_{z,1})$ is a GIS raster map (regular grid) that gives water depths across the study area. Uncertainty in $W_{z,1}$ derives from the combination of various input uncertainties in the inundation mapping process; however, for the sake of

simplicity, we considered the error on the high-resolution digital terrain model (DTM) as the single uncertainty source in the hazard map $W_{z,1}$. The DTM—a raster surface of 5 m cell size—was initially built by stereophotogrammetry. Both measurement errors and interpolation errors affect the quality of this input data [47]. These errors are spatially auto-correlated and are modeled by a zero-mean Gaussian random field with an exponential variogram model $\gamma(\cdot)$, whose characteristics were determined from a set of 500 control field points (sill = 17 cm, range = 500 m, nugget = 0.02).

The land use map ($W_{z,2}$) is a GIS vector map, which exhibits at least two types of uncertainty: (i) semantic errors (misclassification of polygonal features representing assets) and (ii) geometric errors [2]. Semantic errors are described by a confusion matrix that gives a confusion probability $p(a, b)$ for each pair of possible land use types (a, b) [13]. Geometric errors are accounted for by randomizing the surface area of each polygonal feature, using a corrective random coefficient $\alpha \sim U[0.75; 0.85]$ —which corresponds to a digitalizing error of 0.3 mm at the map scale [20].

From this description of uncertainty, two stochastic processes \mathcal{P}_1 and \mathcal{P}_2 were designed to generate any number of random realizations of $W_{z,1}$ and $W_{z,2}$.

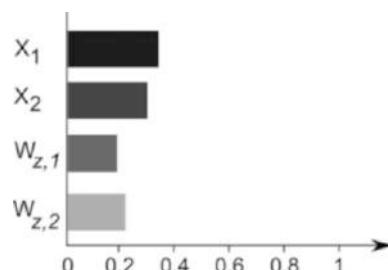
3.4.2 Propagating Uncertainty: The Map Labeling Method

To estimate variance-based sensitivity indices associated with spatial inputs $W_{z,1}$ and $W_{z,2}$, we use the *map labeling* method of [27]. The uncertainty of W_z is represented using a set of n_L randomly generated realizations (n_L possibly large). Each realization is then *labeled* by a single integer. Sample sizes $n_{L,1} = 1000$ and $n_{L,2} = 100$ were used for the two spatial inputs of the NOE model (hazard map $W_{z,1}$ and land use map $W_{z,2}$).

3.4.3 Results

Figure 39.3 displays the total-order sensitivity indices S_i^{tot} computed for each uncertain model input with respect to the aggregated model output $Y_v = \int_{z \in v} Y(z) dz$ over the entire Orb floodplain. The nonspatially distributed inputs (flood return intervals X_1 and damage functions X_2) prove to be the most important sources of uncertainty when aggregating the EAD risk indicator over the total floodplain. On the contrary, the contribution of the two spatially distributed inputs, that is, the hazard map $W_{z,1}$ and the land use map $W_{z,2}$, is small.

Fig. 39.3 Total-order sensitivity indices S_i^{tot} of model inputs X_1 (flood return intervals), X_2 (depth-damage functions), $W_{z,1}$ (hazard map), and $W_{z,2}$ (land use map) with respect to the aggregated output (EAD flood damage indicator) over the entire Orb floodplain



4 Methods for Spatiotemporal Outputs

We consider the model f with d real scalar inputs $\mathbf{X} = (X_1, \dots, X_d)$ and a functional output $Y \in \mathbb{F}$. The functional space \mathbb{F} is a subset of functions mapping from \mathbb{D}_z to \mathbb{R} , where \mathbb{D}_z can be a time interval or a spatial domain, for example. The model is defined by:

$$\begin{aligned} f : \mathbb{R}^d &\rightarrow \mathbb{F} \\ \mathbf{X} &\mapsto Y(z) = f(\mathbf{X}; z) \end{aligned} \quad (39.15)$$

for any $z \in \mathbb{D}_z$. The inputs \mathbf{X} are modeled by random variables and characterized by their probability density function $\mu_{\mathbf{X}}$.

We also consider a discretization of \mathbb{D}_z with p elements:

$$\mathbb{D}_z \xrightarrow{\text{discretization}} \mathcal{Z} = (z^{(1)}, \dots, z^{(p)}) \in \mathbb{D}_z^p.$$

In this case, the model (39.15) is replaced by its discretized version:

$$\begin{aligned} \mathbf{f} : \mathbb{R}^d &\rightarrow \mathbb{R}^p \\ \mathbf{X} &\mapsto \mathbf{Y} = \mathbf{f}(\mathbf{X}) = (f(\mathbf{X}; z^{(1)}), \dots, f(\mathbf{X}; z^{(p)}))^T. \end{aligned} \quad (39.16)$$

where the output \mathbf{Y} belongs to \mathbb{R}^p .

4.1 Sensitivity Index Maps

For any sensitivity measure S , a common approach to deal with multidimensional output consists in running a separate sensitivity analysis for each component of the vectorial output in Eq. (39.16). This leads to d vectors of sensitivity indices $(\mathbf{S}_i)_{1 \leq i \leq d}$, with $\mathbf{S}_i = (S_i^{(1)}, \dots, S_i^{(p)})$ where $S_i^{(j)}$ is the sensitivity index associated to the input X_i with respect to the j th output component. When \mathbf{Y} is a spatial field, as in the Ceres-Mithra and the NOE test cases, for example, \mathbf{S}_i can be plotted over the grid \mathcal{Z} and provide a sensitivity index map for any input parameter (see, e.g., Marrel et al. [30] and Ceres-Mithra results on page 1351 or NOE results on page 1349). Such spatial measures can be called “site sensitivity indices” and make sense.

This approach has some severe limits. Firstly, computing sensitivity index maps is computationally expensive : $d \times p$ sensitivity indices have to be estimated, instead of only d sensitivity indices for a scalar output. This is especially true when \mathcal{Z} is a fine spatial grid with hundreds, even thousands, of nodes. Secondly, the interpretation of sensitivity index maps is not straightforward: conclusions have to be drawn based on the comparison of vectorial sensitivity indices, not scalar ones; it is usually impossible to find a unique ranking of model inputs from these sensitivity index maps (see Ceres-Mithra results on page 1351 and NOE results on page 1349).

4.2 Block Sensitivity Indices

Another common approach to deal with a multidimensional output \mathbf{Y} is to simply extract a scalar quantity of interest from \mathbf{Y} and to compute sensitivity indices with respect to it. When a model has a spatiotemporal output $Y(z)$, modelers are most often interested in the aggregated value of model output over a given spatiotemporal subset $v \subset \mathbb{D}_z$ of the spatial domain or time interval. Hence, a common quantity of interest is the average value $Y_v = \int_{z \in v} Y(z) dz$, or $Y_v = \frac{1}{q} \sum_{j=1}^q f(\mathbf{X}; z^{(j)})$ using a discretization of v . We call *block sensitivity indices*, denoted by $S_{\{i\}}^{\text{tot}}(v)$, the total-order indices computed with respect to the scalar output Y_v .

These *block sensitivity indices* $S_{\{i\}}^{\text{tot}}(v)$ can be computed over subsets $v \subset \mathbb{D}_z$ of varying sizes and locations. They can bring valuable insights on the model behavior at different spatiotemporal scales (see NOE results on page 1349). However, they do not give a full functional restitution of sensitivity analysis and do not provide complete information on local model behavior in the functional domain \mathbb{D}_z .

4.3 Aggregated Sensitivity Indices

A full functional sensitivity analysis for spatiotemporal outputs, which would provide both local and global information on model behavior in the spatiotemporal domain \mathbb{D}_z , remains a challenge. In the following paragraphs, we will attempt to give some solutions to address the shortcomings of both the sensitivity index maps and the block sensitivity indices.

Firstly, a few authors suggested to use dimension reduction to deal with spatiotemporal outputs in sensitivity analysis. They expanded multidimensional output \mathbf{Y} on some appropriate basis and then separately computed sensitivity measures with respect to the more informative components of this expansion. Particularly, [5] used an empirical orthogonal decomposition of the multidimensional output (Legendre polynomial, principal component analysis, or partial least square) and computed usual variance-based sensitivity indices with respect to the coefficients of this expansion, keeping only the main components. However, this method limits the analysis to the principal characteristics rather than the whole behavior of the model output. Furthermore, the interpretation of sensitivity indices computed with respect to the coefficient of a functional decomposition is tricky.

More recently, [16] defined *aggregated Sobol indices* for any input parameter X_i , $i \in \{1, \dots, d\}$, based on the decomposition of the multidimensional output variance. To achieve this, they consider the multidimensional model $\mathbf{Y} = \mathbf{f}(\mathbf{Y}) \in \mathbb{R}^p$ satisfying $\mathbb{E}[\|\mathbf{f}(\mathbf{X})\|^2] < \infty$ and apply the Hoeffding decomposition to the function $\mathbf{f}, \mathbb{R}^d \mapsto \mathbb{R}^p$:

$$\mathbf{f}(\mathbf{X}) = \mathbf{f}_0 + \mathbf{f}_i(X_i) + \mathbf{f}_{\sim i}(\mathbf{X}_{\sim i}) + \mathbf{f}_{i,\sim i}(X_i, \mathbf{X}_{\sim i})$$

where $\mathbf{f}_0 = \mathbb{E}[\mathbf{Y}]$, $\mathbf{f}_i = \mathbb{E}[\mathbf{Y}|X_i] - \mathbf{f}_0$, $\mathbf{f}_{\sim i} = \mathbb{E}[\mathbf{Y}|\mathbf{X}_{\sim i}] - \mathbf{f}_0$ and $\mathbf{f}_{i,\sim i} = \mathbf{Y} - \mathbf{f}_{\sim i} - \mathbf{f}_i - \mathbf{f}_0$, with $\mathbf{X}_{\sim i} = (X_j)_{\substack{1 \leq j \leq d \\ j \neq i}}$. Then, they compute the covariance matrix of both sides of

this equality and apply the trace operator in order to turn the matrix format into a scalar one:

$$\text{trace}(\Sigma) = \text{trace}(C_i) + \text{trace}(C_{\sim i}) + \text{trace}(C_{i,\sim i})$$

where $\Sigma = \mathbb{V}[\mathbf{f}(X)]$, $C_i = \mathbb{V}[\mathbf{f}_i(X)]$, and so on. Contrarily to the determinant operator, the trace one allows an additive decomposition of the output to the Hoeffding expansion. Finally, dividing both sides by $\text{trace}(\Sigma)$ leads notably to the first-order index:

$$S_i = \frac{\text{trace}(C_i)}{\text{trace}(\Sigma)} = \frac{\sum_{l=1}^p \mathbb{V}[Y_l] S_i^{(l)}}{\sum_{l=1}^p \mathbb{V}[Y_l]} \quad (39.17)$$

where $\mathbb{V}[Y_l]$ is the variance associated to the l th scalar output and $S_i^{(l)}$ is the first-order Sobol index associated to the l th output for the i th input parameter.

Gamboa et al. [16] estimate this index using the pick-and-freeze estimator

$$S_{i,N} = \frac{\sum_{l=1}^p \left(\frac{1}{N} \sum_{j=1}^N Y_l^{(j)} Y_l^{[i],(j)} - \left(\frac{1}{N} \sum_{j=1}^N \frac{Y_l^{(j)} + Y_l^{[i],(j)}}{2} \right)^2 \right)}{\sum_{l=1}^p \left(\frac{1}{N} \sum_{j=1}^N \frac{(Y_l^{(j)})^2 + (Y_l^{[i],(j)})^2}{2} - \left(\frac{1}{N} \sum_{j=1}^N \frac{Y_l^{(j)} + Y_l^{[i],(j)}}{2} \right)^2 \right)} \quad (39.18)$$

where $(X^{(j)}, Y^{(j)})_{1 \leq j \leq N}$ is a sample of N independent realizations of (\mathbf{X}, \mathbf{Y}) and where $\mathbf{Y}^{[i]} = \mathbf{f}(X_i, \mathbf{X}'_{\sim i})$ with $\mathbf{X}'_{\sim i}$ being an independent copy of $\mathbf{X}_{\sim i}$. Total indices can be computed in the same way thanks to the relation $S_i^{\text{tot}} = S_i + S_{i,\sim i} = 1 - S_{\sim i}$ and formulas (39.17)–(39.18).

These aggregated Sobol indices are equal to the sensitivity measures defined by Lamboni et al. [26], when all the terms of the output Proper Orthonormal Decomposition Chatterjee [6] used in their work are kept in the global sensitivity index computation. More precisely, these authors consider the following expansion of the model output:

$$\mathbf{Y} = \boldsymbol{\mu} + \mathcal{V}\mathbf{H}$$

where $\boldsymbol{\mu} = \mathbb{E}[\mathbf{Y}]$, \mathcal{V} is the matrix whose columns $\mathbf{v}_1, \dots, \mathbf{v}_p$ are the eigenvectors of Σ , also called components, and $\mathbf{H} = \mathcal{V}^T (\mathbf{Y} - \boldsymbol{\mu})$ is a random vector. Note that for any $k, l \in \{1, \dots, p\}$, $\mathbb{E}[H_k] = 0$, $\mathbb{E}[H_k H_l] = \lambda_k \delta_{kl}$ and $\mathbb{V}[H_k] = \lambda_k$, λ_k being the k th eigenvalue of Σ associated to the eigenvector \mathbf{v}_k , also called principal component.

Then, for any $w \subset \{1, \dots, d\} \setminus \emptyset$, the corresponding sensitivity index for the k th principal component \mathbf{v}_k is defined by

$$S_{w,k} = \frac{\mathbb{V}_{w,k}}{\lambda_k}$$

where $\mathbb{V}_{w,k} = \mathbb{V}[\mathbb{E}[H_k | \mathbf{X}_w]] - \sum_{u,u \subset w} \mathbb{V}_{u,k}$ and $\mathbb{V}_{\emptyset,k} = 0$, with $\mathbf{X}_w = (X_i)_{i \in w}$. Particularly, for any $i \in \{1, \dots, d\}$, the first-order and total variance-based sensitivity indices \mathbf{v}_k are

$$S_{i,k} = \frac{\mathbb{V}[\mathbb{E}[H_k | X_i]]}{\lambda_k} \text{ and } S_{i,k}^{\text{tot}} = \sum_{\substack{w \subset \{1, \dots, d\} \\ w \ni i}} S_{w,k}.$$

Using these component-based sensitivity indices, Lamboni et al. [26] suggested to define corresponding first-order and total-aggregated sensitivity indices:

$$S_i = \frac{\sum_{k=1}^p \lambda_k S_{i,k}}{\text{trace}(\Sigma)} = \frac{\sum_{k=1}^p \lambda_k S_{i,k}}{\sum_{k=1}^p \lambda_k} \text{ and } S_i^{\text{tot}} = \sum_{\substack{w \subset \{1, \dots, d\} \\ w \ni i}} S_i. \quad (39.19)$$

This first-order Sobol index S_i has an expression similar to the one of [16] presented in Eq. (39.17). Moreover, it can easily be proven that these terms are equal. For the estimation step, we consider an N -sample $(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})_{1 \leq i \leq N}$ associated to a particular Sobol index computation method and replace the covariance matrix Σ by the empirical one

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N (\mathbf{Y}^{(i)} - \bar{\mathbf{Y}})(\mathbf{Y}^{(i)} - \bar{\mathbf{Y}})^T \quad (39.20)$$

where $\bar{\mathbf{Y}} = \frac{1}{N} \sum_{i=1}^N \mathbf{Y}^{(i)}$. Similarly, we note $\hat{\lambda}_k$ and $\hat{\mathbf{v}}_k$, the k th eigenvalue and eigenvector of $\hat{\Sigma}$, respectively, and we consider the score matrix $\hat{H} = (\hat{H}_{ik})_{\substack{1 \leq i \leq N \\ 1 \leq k \leq p}}$

where $\hat{H}_{ik} = \langle \mathbf{Y}^{(i)} - \bar{\mathbf{Y}}, \hat{\mathbf{v}}_k \rangle_2$, $\langle \cdot, \cdot \rangle_2$ being the scalar product on \mathbb{R}^p . Finally, for any $k \in \{1, \dots, p\}$ and any $i \in \{1, \dots, d\}$, $S_{i,k}$ is estimated by $\hat{S}_{i,k}$ computed from the chosen Sobol index computation method and the N -sample $(\mathbf{X}^{(j)}, H_{jk})_{1 \leq j \leq N}$. Moreover, we have

$$\hat{S}_i = \frac{\sum_{k=1}^p \hat{\lambda}_k \hat{S}_{i,k}}{\text{trace}(\hat{\Sigma})}. \quad (39.21)$$

To conclude, the variance-based global sensitivity analysis method suggested by Gamboa et al. [16] gives aggregated sensitivity indices which summarize the influence of each model input X_i on the variance of multidimensional output \mathbf{Y} . The approach of Lamboni et al. [26] also leads to the computation of similar aggregated sensitivity indices: when all the terms are kept in the expansion of model output \mathbf{Y} , both aggregated sensitivity indices of Gamboa et al. [16] and Lamboni et al. [26] are

equal. These aggregated sensitivity indices are global measures that summarize the local model behavior over spatiotemporal domain \mathbb{D}_z . A unique ranking of model inputs can be derived from these aggregated sensitivity indices.

Remark 1. Previous sensitivity index maps and block sensitivity indices and their estimation often require thousands of model evaluations. Recently, Da Veiga [8] introduced the use of dependence measures in global sensitivity analysis, and De Lozzo and Marrel [9] developed it to the specific case of screening. One of these new indices is based on the Hilbert-Schmidt independence criterion (HSIC), which is defined, for any pair of measurable random variables (X, Y) , as the Hilbert-Schmidt norm of the cross-covariance operator between some function transformations of X and Y in reproducing kernel Hilbert spaces [19]. Da Veiga [8] noted that SA with HSIC-based sensitivity measure requires fewer model evaluations compared to variance-based ones. Moreover, kernels dedicated to functional data allow to deal easily with spatiotemporal inputs and outputs. These recent tools have to be confirmed by industrial applications but already represent an alternative for sensitivity analysis with spatial output.

4.4 Use of Metamodels

All the methods discussed above (sensitivity index maps, block sensitivity indices, aggregated sensitivity indices) are based on intensive Monte Carlo methods and require lots of simulations to be estimated. Some spatiotemporal models can be too time-consuming to be directly used to conduct such variance-based methods. To avoid the problem of huge calculation time, it can be useful to replace the complex computer code by a metamodel as introduced in the first section. The problem of building a metamodel for a functional output has recently been addressed by Shi et al. [41] and Bayarri et al. [1] who proposed an approach based upon functional decomposition, such as wavelets. Recently, Marrel et al. [28] proposed a complete methodology to perform sensitivity analysis of two-dimensional output: their approach consists in building a functional metamodel and computing variance-based sensitivity indices at each location of the spatial output map. The maps of Sobol indices thus obtained yield the identification of global and local influence of each input variable and the detection of areas with interactions. The functional metamodel is based on a decomposition on a basis of orthogonal functions followed by a metamodeling of the coefficients related to the main decomposition components. This method can be applied with any kind of orthogonal functions such as wavelets, Fourier functions, principal component analysis, polynomial chaos, etc. For example, in Marrel et al. [28], the authors use a wavelet decomposition and Gaussian process metamodels. As this decomposition seems to require a large number of metamodels to be built, Marrel et al. [30] then propose to apply this method but using a Proper Orthogonal Decomposition Chatterjee [6] instead of the wavelet decomposition in order to reduce the number of metamodels.

4.5 Application on NOE Test Case

Saint-Geours et al. [38] performed a variance-based global sensitivity analysis on the NOE model, considering as a spatial output the map $Y(z)$ of the EAD risk indicator. They first computed sensitivity index maps and then block sensitivity indices over spatial subsets $v \subset \mathbb{D}_z$. They investigated how the value of block sensitivity indices may vary depending on the spatial support chosen for averaging model output. We briefly summarize their findings below.

4.5.1 Sensitivity Index Maps

As explained on page 1344, sensitivity index maps can be computed by estimating $\mathbf{S}_i = (S_i^{(1)}, \dots, S_i^{(p)})$ where $S_i^{(j)}$ is the sensitivity index associated to the input X_i with respect to the value of output map at a specific location $z^{(j)} \in \mathbb{D}_z$. \mathbf{S}_i is then plotted over the grid \mathcal{Z} and provides a sensitivity index map for any input parameter X_i . Figure 39.4 displays the maps \mathbf{S}_i for all uncertain inputs of the NOE model. Spatial distribution of sensitivity indices proves to be heterogeneous. For example, by comparing these sensitivity maps with a map of the study area, we can find that the hazard map $W_{z,1}$ and the land use map $W_{z,2}$ display smaller sensitivity indices on the urban areas than on the areas covered with agricultural land. The interpretation of this finding will not be discussed here, but it brought a better understanding of the NOE model behavior.

4.5.2 Block Sensitivity Indices and Change of Support Effect

On the NOE model, three individual spatial supports v_1 , v_2 , and v_3 of increasing surface area were considered. These three supports were selected at random, and their study is only meant to be illustrative: v_1 is a single house located on the western bank of the Orb Delta ($|v_1| = 1$ ha), second support v_2 is the administrative district of Sauvian ($|v_2| = 13$ sq. km), and v_3 is the entire floodplain ($|v_3| = 63$ sq. km). Figure 39.5 displays the block sensitivity indices $S_{\{i\}}^{\text{tot}}(v)$ (as explained on page 1345) computed for each uncertain model input with respect to the aggregated model output $Y_v = \int_{z \in v} Y(z) dz$ over spatial supports v_1 (house), v_2 (district), and v_3 (floodplain). It clearly suggests that the ranking of uncertainty sources depends on the surface area of the spatial support v (*change of support effect*). The variance of the aggregated model output on v_1 (smallest support) appears to be mainly explained by the uncertainty on the two spatially distributed inputs, that is, the hazard map $W_{z,1}$ and the land use map $W_{z,2}$ (sensitivity indices, 0.65 and 0.8, respectively). On the contrary, the nonspatially distributed inputs (flood return intervals X_1 and damage functions X_2) prove to be the most important sources of uncertainty when aggregating the EAD risk indicator over the total floodplain v_3 (largest support).

This *change of support* effect can be explained theoretically. Let us consider from now on a model $Y = f(\mathbf{X}, W_z)$ with a *single* spatial input $W_z = \{W(z) : z \in \mathbb{D}_z\}$ which is assumed to be a second-order stationary random field (SRF) with an isotropic covariance function $C(\cdot)$. Let us also assume that model f is point based, that is, there exists a mapping $f_{loc.} : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ such that $\forall z \in \mathbb{D}_z$,

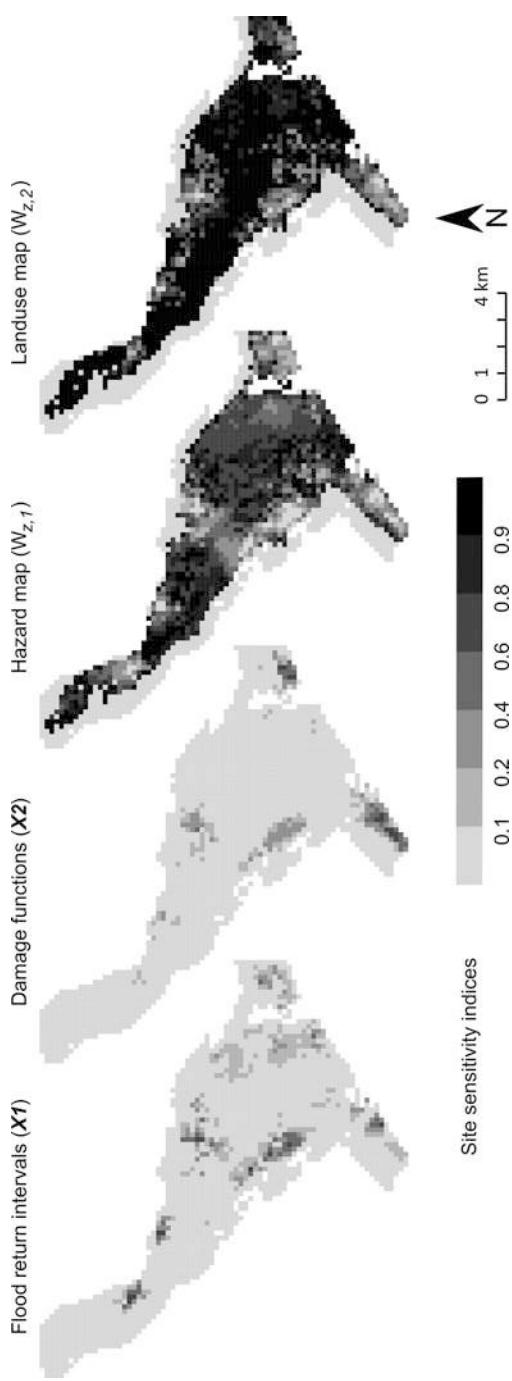


Fig. 39.4 Maps S_{ti} of site sensitivity indices for each model input

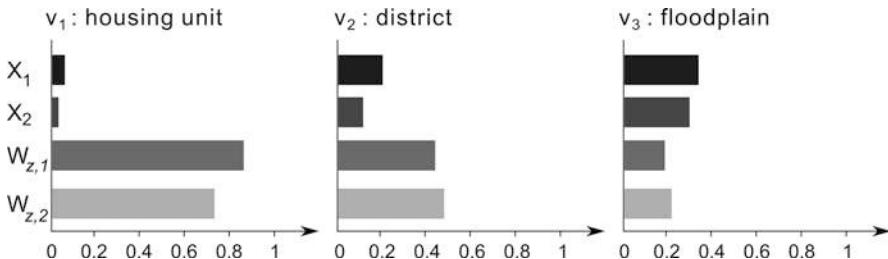


Fig. 39.5 Block sensitivity indices $S_{i,i}^{tot}(v)$ on spatial supports v_1 to v_3

$Y(z) = f_{loc}(\mathbf{X}, W(z))$. Under the stationary hypothesis on W_z , site sensitivity indices $S_{\{i\}}^{tot}(z)$ do not depend on site $z \in \mathbb{D}_z$ and can simply be denoted by $S_{\{i\}}^{tot}$. For any support $v \subset \mathbb{D}_z$, the block sensitivity indices of \mathbf{X} and W_z over support v verify [37, for a detailed proof]:

$$\frac{S_{W_z}^{tot}(v)}{S_{\mathbf{X}}^{tot}(v)} \simeq \frac{v_c}{|v|} \quad (39.22)$$

where v_c depends on the covariance function $C(\cdot)$. Equation (39.22) shows that the ratio $v_c/|v|$ determines the relative contribution of the model inputs W_z and \mathbf{X} to the output variance $\text{var}(Y_v)$. For a small ratio (i.e., when the area of the support v is large compared with the critical size v_c), the variability of spatial input W_z is mainly *local*, and the spatial correlation of W_z over v is weak. This local variability averages over the support v when the aggregated model output Y_v is computed; hence, spatial input W_z explains a small fraction of the output variance $\text{var}(Y_v)$. This *change of support effect* is of great importance to better understand how the result of a sensitivity analysis may change under model upscaling or downscaling.

4.6 Application on Ceres-Mithra Test Case

The model output of the Ceres-Mithra test case is a discretized spatial map of time-integrated surface activity concentration of *cesium 137* (Bq.s.m^{-3}), at moments $t_0 + 20'$, $t_0 + 40'$, ..., $t_0 + 2h$ where t_0 is the beginning of releases. This output is function of six uncertain inputs: the heights of release (HR1 and HR2), the deposition velocity (DV), the source term activity (STA), and the wind direction and speed (WD and WS). The associated two-dimensional spatial grid is made up of $60 \times 45 = 2700$ points and is the same for all the Ceres-Mithra simulations.

In the following, a global sensitivity analysis is performed on the Ceres-Mithra test case, using first the sensitivity index maps described on page 1344 to get some insight on the local model behavior and then using the aggregated sensitivity indices defined on page 1345 to get a global ranking of model inputs. With these sensitivity measures requiring a high number of evaluations, the surrogate model is necessary because of the important CPU time cost of a Ceres-Mithra run. This

metamodel is parametrized using an n -sample based on a *maximin* Latin hypercube sampling design, with $n = 200$, and validated by cross-validation (see [30] for details). Then, the sensitivity indices are estimated from ten thousand metamodel predictions and using Monte Carlo-based formulas of [39] and pick-and-freeze ones (see Eq. (39.18)) for the sensitivity index maps and aggregated Sobol indices, respectively.

4.6.1 Sensitivity Index Maps

Firstly, Figs. 39.6 and 39.7 represent the time evolution of the primary and total Sobol indices for each pixel of the maps. For the sake of brevity, only the variables with significant Sobol indices are shown and only at three different times. The local analysis of Sobol indices reveals that the source term activity influence is predominant in the central part of the plume, while the wind direction is less influential in this area. On the flip side, the situation is reversed in the limit of the plume: logically, the wind conditions have a strong impact on the plume trajectory.

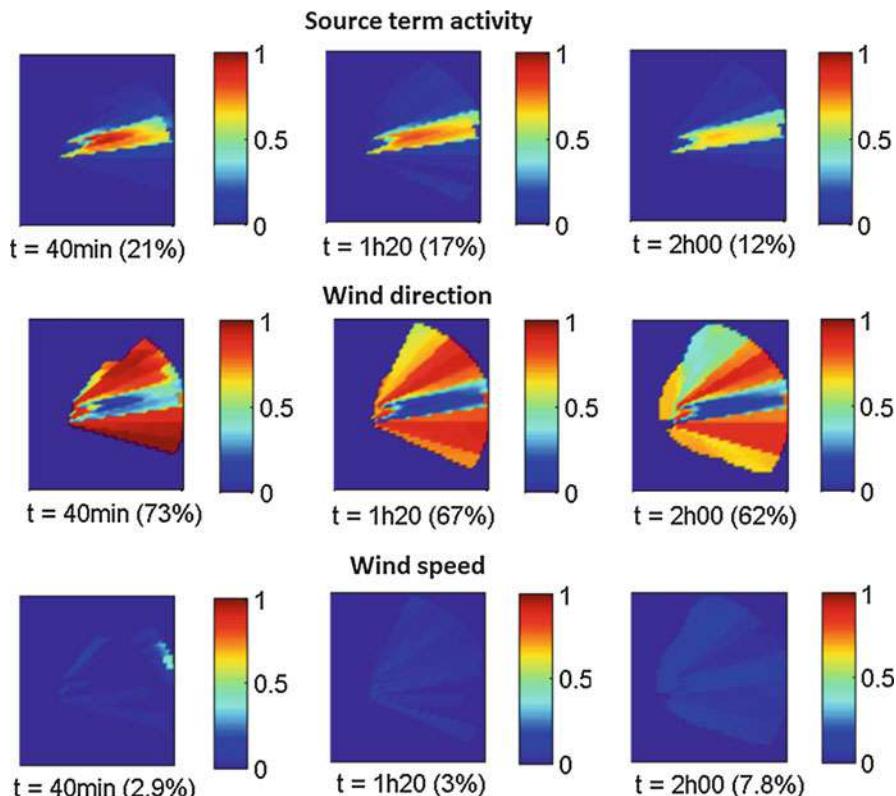


Fig. 39.6 Temporal evolution of the site 1st order Sobol indices for the ^{137}Cs integrated surface activity

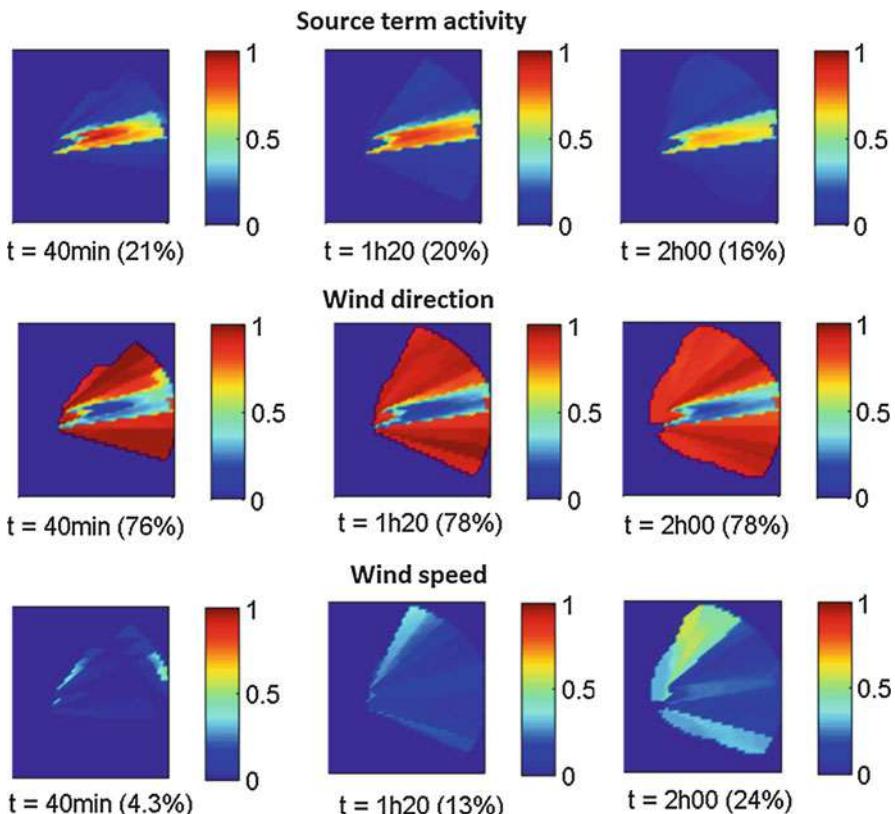


Fig. 39.7 Temporal evolution of the site total Sobol indices for the ^{137}Cs integrated surface activity

From a global point of view, the wind direction is the most influential variable on the surface activity forecast: it explains more than half of the activity variance. The source term activity is the second most influential variable, with on average about 15% of explained variance. The uncertainty on wind speed is less influential, while the deposition velocity and heights of release do not affect the forecast. The sum of the primary effects (i.e., without considering any interactions between variables) explains between 80% and 100% of the variance. Thus, the interactions account for less than 10% of the variance, and the only significant interaction is the one between the wind variables. The estimation of second-order indices shows that this interaction is located at the limit of the plume and increases slightly over time, while the direction wind decreases; see Fig. 39.8.

4.6.2 Aggregated Sensitivity Indices

Secondly, we turn to the aggregated total Sobol indices proposed by [16] in order to summarize previous site information. Figure 39.9 represents the evolution of

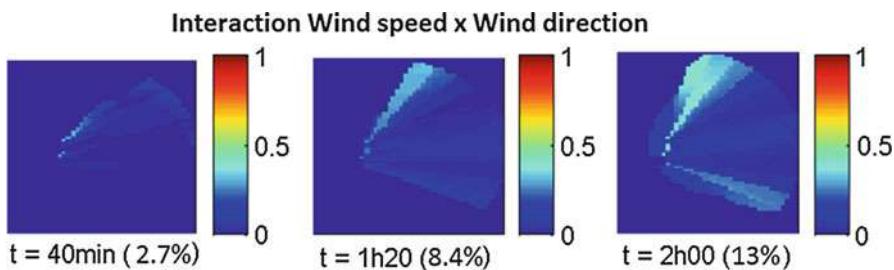


Fig. 39.8 Temporal evolution of the site 2nd order Sobol indices between wind components, for the ^{137}Cs integrated surface activity

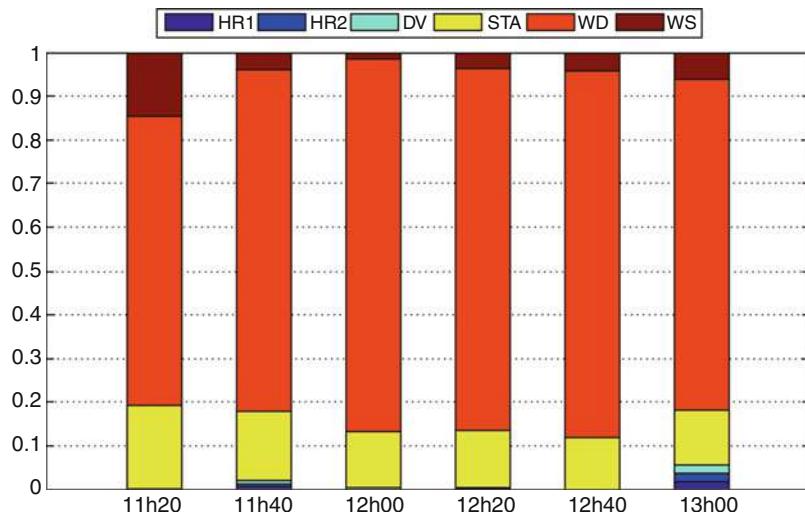


Fig. 39.9 Temporal evolution of the normalized block total Sobol indices for the ^{137}Cs integrated surface activity

these sensitivity indices after a normalization step, in order to sum to one. Clearly, the wind direction is the predominant input, explaining between 66% and 76% of the global output variance, according to the considered moment. Then, the source term activity represents between 12% and 19% of this variability. The wind speed uncertainty is less influential, with a contribution between 6% and 15%.

4.6.3 Conclusion on Ceres-Mithra Test Case

To conclude, the main benefit of the aggregated sensitivity indices is to give a single sensitivity measure for each model input, thus allowing the inputs to be ranked according to their global influence on model output; this leads clearly to easier conclusions in order to reduce the output variability by means of efforts deployed to the control of the most influential inputs. Besides, sensitivity index maps bring complementary information about the local contribution of the inputs to the output

variability, which can be helpful from a physical point of view, to better understand the local model behavior. Consequently, it is recommended to perform both types of sensitivity analyses in parallel, in order to deal with a spatial model output.

5 Conclusions

We discussed in this section how sensitivity analysis can be applied to computer codes with spatiotemporal inputs and/or outputs. The methods presented to deal with spatiotemporal inputs are practice oriented and have already been used in several industrial and environmental case studies. They all focus on the computation of variance-based sensitivity indices, but some of them can easily be adapted to other important measures such as elementary effects (Morris method). However, a homogeneous theoretical framework is still missing to properly define importance measures with respect to spatiotemporal inputs and more generally functional inputs.

Regarding spatiotemporal outputs, we distinguished the use of (i) sensitivity index maps, which scrutinize model behavior at each point of a spatiotemporal output domain; (ii) block sensitivity indices, which analyze the influence of model inputs on the average value of model output over a given subset of the spatiotemporal domain; and (iii) aggregated sensitivity indices, which summarize the influence of a model input over the entire spatiotemporal output domain. All types of indices give valuable insight on model behavior, and we strongly suggest to carry out different analyses whenever possible.

Finally, we tried to emphasize how sensitivity analysis can shed a new light on scale-related questions in spatiotemporal modeling, by analyzing how sensitivity indices and the ranking of uncertain inputs depend on the spatial/temporal support of model output. In spite of these advances, a general framework for sensitivity analysis of spatiotemporal models is still to be developed, and further research in this field is strongly needed.

References

1. Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., Walsh, D.: Computer model validation with functional output. *Ann. Stat.* **35**, 1874–1906 (2007)
2. Bonin, O.: Sensitivity analysis and uncertainty analysis for vector geographical applications. In: 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Lisbon, 5–7 July 2006
3. Caers, J.: Modeling Uncertainty in the Earth Sciences. Wiley-Blackwell, Hoboken (2011)
4. Caers, J., Park, K., Scheidt, C.: Modeling uncertainty of complex earth systems in metric space. In: Handbook of Geomathematics, pp. 865–889. Springer, Berlin/New York (2010)
5. Campbell, K., McKay, M.D., Williams, B.J.: Sensitivity analysis when model outputs are functions. *Reliab. Eng. Syst. Saf.* **91**(10), 1468–1472 (2006)
6. Chatterjee, A.: An introduction to the proper orthogonal decomposition. *Curr. Sci.* **78**(7), 808–817 (2000)

7. Crosetto, M., Tarantola, S.: Uncertainty and sensitivity analysis: tools for GIS-based model implementation. *Int. J. Geogr. Inf. Sci.* **15**, 415–437 (2001)
8. Da Veiga, S.: Global sensitivity analysis with dependence measures. *J. Stat. Comput. Simul.* **85**(7), 1283–1305 (2014)
9. De Lozzo, M., Marrel, A.: New improvements in the use of dependence measures for sensitivity analysis and screening. *J. Stat. Comput. Simul.* (2015, submitted) doi:<https://hal.archives-ouvertes.fr/hal-01090475>
10. Doury, A.: Pratiques françaises en matière de prévision quantitative de la pollution atmosphérique potentielle liée aux activités nucléaires. In: Seminar on radioactive releases and their dispersion in the atmosphere following a hypothetical reactor accident, vol. I, pp. 403–448 (1980)
11. Erdlenbruch, K., Gilbert, E., Grelot, F., Lescouliers, C.: Une analyse coût-bénéfice spatialisée de la protection contre les inondations : application de la méthode des dommages évités à la basse vallée de l'Orb. *Ingénieries Eau-Agriculture-Territoires* **53**, 3–20 (2008)
12. Ferraty, F., Vieu, P.: Nonparametric Functional Data Analysis. Springer Series in Statistics, Springer, New York (2006)
13. Fisher, P.F.: Modelling soil map-unit inclusions by Monte Carlo simulation. *Int. J. Geogr. Inf. Sci.* **5**(2), 193–208 (1991)
14. Fort, J.C., Klein, T., Lagnoux, A., Laurent, B.: Estimation of the Sobol indices in a linear functional multidimensional model. *J. Stat. Plan. Inference* **143**, 1590–1605 (2013)
15. Francos, A., Elorza, F.J., Bouraoui, F., Bidoglio, G., Galbiati, L.: Sensitivity analysis of distributed environmental simulation models: understanding the model behaviour in hydrological studies at the catchment scale. *Reliab. Eng. Syst. Saf.* **79**(2), 205–218 (2003)
16. Gamboa, F., Janon, A., Klein, T., Lagnoux, A., et al.: (2014) Sensitivity analysis for multidimensional and functional outputs. *Electron. J. Stat.* **8**, 575–603
17. Gijbels, I., Prosdocimi, I., Claeskens, G.: Nonparametric estimation of mean and dispersion functions in extended generalized linear models. *Test* **19**(3), 580–608 (2010)
18. Ginsbourger, D., Rossopopoff, B., Pirot, G., Durrande, N., Renard, P.: Distance-based kriging relying on proxy simulations for inverse conditioning. *Adv. Water Resour.* **52**, 275–291 (2013)
19. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Algorithmic Learning Theory, vol. 3734, pp. 63–77. Springer, Berlin/Heidelberg (2005)
20. Hengl, T.: Finding the right pixel size. *Comput. Geosci.* **32**, 1283–1298 (2006)
21. Heuvelink, G.B.M., Brus, D.J., Reinds, G.: Accounting for spatial sampling effects in regional uncertainty propagation analysis. In: Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Leicester, pp. 85–88, 20–23 July 2010
22. Heuvelink, G.B.M., Burgers SLGE, Tiktak, A., Den Berg, F.V.: Uncertainty and stochastic sensitivity analysis of the GeoPEARL pesticide leaching model. *Geoderma* **155**(3–4), 186–192 (2010)
23. Iooss, B.: Treatment of spatially dependent input variables in sensitivity analysis of model output methods. Technical report, European project PAMINA (Performance assessment methodologies in application to guide the development of the safety case) (2008)
24. Iooss, B., Ribatet, M.: Global sensitivity analysis of computer models with functional inputs. *Reliab. Eng. Syst. Saf.* **94**, 1194–1204 (2009)
25. Jacques, J., Lavergne, C., Devictor, N.: Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1126–1134 (2006)
26. Lamboni, M., Monod, H., Makowski, D.: Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliab. Eng. Syst. Saf.* **96**(4), 450–459 (2011)
27. Lilburne, L., Tarantola, S.: Sensitivity analysis of spatial models. *Int. J. Geogr. Inf. Sci.* **23**(2), 151–168 (2009)
28. Marrel, A., Iooss, B., Jullien, M., Laurent, B., Volkova, E.: Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics* **22**(3), 383–397 (2011)

29. Marrel, A., Iooss, B., Da Veiga, S., Ribatet, M.: Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput.* **22**(3), 833–847 (2012)
30. Marrel, A., Perot, N., Mottet, C.: Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators. *Stoch. Environ. Res. Risk Assess.* **29**(3), 959–974 (2015)
31. McCullagh, P., Nelder, J.A., McCullagh, P.: Generalized Linear Models, vol. 2. Chapman and Hall, London (1989)
32. Monfort, M., Patryl, L., Armand, P.: Presentation of the ceres platform used to evaluate the consequences of the emissions of radionuclides in the environment. *HARMO* **13**, 1–4 (2010)
33. Morris, M.D.: Gaussian surrogates for computer models with time-varying inputs and outputs. *Technometrics* **54**(1), 42–50 (2012)
34. Romary, T.: Heterogeneous media stochastic model inversion. PhD thesis, Université Pierre et Marie Curie – Paris VI (2008)
35. Ruffo, P., Bazzana, L., Consonni, A., Corradi, A., Saltelli, A., Tarantola, S.: Hydrocarbon exploration risk evaluation through uncertainty and sensitivity analyses techniques. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1155–1162 (2006)
36. Saint-Geours, N.: Sensitivity analysis of spatial models: application to cost-benefit analysis of flood risk management plans. PhD thesis, Université Montpellier 2, <http://tel.archives-ouvertes.fr/tel-00761032> (2012)
37. Saint-Geours, N., Lavergne, C., Bailly, J.S., Grelot, F.: Change of support in variance-based spatial sensitivity analysis. *Math. Geosci.* **44**(8), 945–958 (2012)
38. Saint-Geours, N., Bailly, J.S., Grelot, F., Lavergne, C.: Multi-scale spatial sensitivity analysis of a flood damage assessment model. *Environ. Model. Softw.* **60**, 153–166 (2014). <http://dx.doi.org/10.1016/j.envsoft.2014.06.012>
39. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**(2), 259–270 (2010)
40. Scheidt, C., Caers, J.: Representing spatial uncertainty using distances and kernels. *Math. Geosci.* **41**(4), 397–419 (2009)
41. Shi, J.Q., Wang, B., Murray-Smith, R., Titterington, D.M.: Gaussian process functional regression modeling for batch data. *Biometrics* **63**(3), 714–723 (2007)
42. SMVOL: Programme d’actions de prévention des inondations sur les bassins de l’Orb et du Libron (34) pour les années 2011–2015. Note de présentation, Syndicat Mixte des Vallées de l’Orb et du Libron (2011)
43. Smyth, G.K.: Generalized linear models with varying dispersion. *J. R. Stat. Soc. Ser. B (Methodol.)* **51**, 47–60 (1989)
44. Suzuki, S., Caers, J.: A distance-based prior model parameterization for constraining solutions of spatial inverse problems. *Math. Geosci.* **40**(4), 445–469 (2008)
45. Suzuki, S., Caumon, G., Caers, J.: Dynamic data integration for structural modeling: model screening approach using a distance-based model parameterization. *Comput. Geosci.* **12**(1), 105–119 (2008)
46. Volkova, E., Iooss, B., Van Dorpe, F.: Global sensitivity analysis for a numerical model of radionuclide migration from the RRC Kurchatov Institute radwaste disposal site. *Stoch. Environ. Res. Risk A* **22**(1), 17–31 (2008)
47. Wechsler, S.: Uncertainties associated with digital elevation models for hydrologic applications: a review. *Hydrol. Earth Syst. Sci.* **11**(4), 1481–1500 (2007)
48. Zabalza, I., Dejean, J.P., Collombier, D.: Prediction and density estimation of a horizontal well productivity index using generalized linear models. In: 6th European Conference on the Mathematics of Oil Recovery. Peebles, Scotland (1998)

Part V
Risk

Decision Analytic and Bayesian Uncertainty Quantification for Decision Support

40

D. Warner North

Abstract

This essay introduces probability in support of decision-making, as a state of mind and not of things. Probability provides a framework for reasoning coherently about uncertainty. According to Cox's theorem, it is the *only* way to reason coherently about uncertainty. Probability summarizes states of information. A basic desideratum is that states of information judged equivalent should lead to the same probability distributions. Some widely used probabilistic models have both sufficient statistics and maximum entropy characterizations. In practice, outside of certain physics and engineering applications, human judgment is usually needed to quantify uncertainty as probabilities. Reality is highly complex, but one can test judgments against what is known, drawing upon many specialized areas of human knowledge. Sensitivity analysis can evaluate the importance of multiple uncertainties in a decision context. When the choice between alternatives is close, what is the value of further information (VOI)? Important concepts discussed in this chapter are the expected value of perfect information and the expected value of further experimental testing.

Practical application is illustrated in two case studies. The first involves assessing the probability for replication of a terrestrial microbe on Mars, in the decision context of a constraint on planetary exploration. The second involves weather modification of hurricanes/typhoons: whether to deploy this technology to reduce damage from hurricanes impacting US coastal areas and valuing experimental testing offshore. The decision context is that of a natural disaster turned political by deployment of new technology where such deployment might be followed by a reduction, or by an increase, in adverse impacts, compared to not deploying the technology.

Decision support should be viewed as an iterative process of working with those with decision responsibility and with experts who are the best available

D.W. North (✉)
NorthWorks, San Francisco, CA, USA
e-mail: northworks@mindspring.com

sources of information. The decision support process includes characterizing uncertainties and values for outcomes in the context of a choice among decision alternatives. Sensitivity analysis and VOI can give insight on whether to act now or to seek more information and more refined analysis. In a public policy decision context, the process can facilitate stakeholders sharing and critiquing information as the basis for characterizing uncertainties and learning which uncertainties and value judgments are most important for decisions.

Keywords

Bayes' theorem • Decision analysis • Value of information

Contents

1	A Viewpoint on Uncertainty Quantification.....	1362
1.1	Many Problems with Using Probability Theory Remain.....	1367
1.2	Bayes' Theorem – A Consistency Condition Among Probabilities.....	1372
1.3	A Basic Desideratum: Two States of Information Perceived to be Equivalent Should Lead to the Same Probability Assignments.....	1373
1.4	Decision Analysis.....	1377
2	Two Case Studies.....	1379
2.1	Case Study #1: Terrestrial Microbes on Mars.....	1379
2.2	Case Study #2: Decision Analysis of Hurricane Modification.....	1384
3	Conclusions.....	1395
	References.....	1396

1 A Viewpoint on Uncertainty Quantification

Introduction and purpose: This chapter of the handbook is a guide to using decision analysis with emphasis on uncertainty quantification. It assumes that the reader is familiar with probability theory as a mathematical structure based on the Kolmogorov axioms, as described in numerous modern textbooks. The emphasis in this chapter is on interpreting and using probability theory for uncertainty quantification as a part of applying decision analysis in a decision support context.

Normative context and the axiomatic basis for probability. Modern decision theory was developed by Abraham Wald [61] and Leonard J. Savage [48] in the mid-twentieth century, building on foundations laid by Ramsey [47] and von Neumann and Morgenstern [59]. Decision analysis, as the application of decision theory to complex practical problems, came about in the 1960s largely through the research and writings of Ward Edwards, Ronald Howard, and Howard Raiffa and their colleagues [2, 3, 15–17, 32, 44–46, 60].

Many people trained in decision analysis view its theoretical basis as the axiomization by Leonard J. Savage in his 1954 book, *The Foundations of Statistics*

[48]. Rational decision-making is based on a combination of the Kolmogorov axioms for probability theory and representation of subjective beliefs by probabilities, plus axioms for utility theory from John von Neumann and Oskar Morgenstern in their book, *Theory of Games and Economic Behavior*, 2nd Edition [59].

In this decision theory framework, probabilities for outcomes, which may depend on a state of nature or may result from a choice among decision alternatives, are described as “subjective.” (For example, note the title *Studies in Subjective Probability*, a book compiling key papers, edited by Kyburg and Smokler [29], including Frank P. Ramsey’s seminal 1926 essay, “Truth and Probability” [47]). A frequent criticism of subjective probabilities has been that they are not based on any type of objective or scientific measurement process. The assessment of subjective probability is carried out by asking a person – someone who has a decision to make or who is to provide judgment as an input into the decision-making process – for a comparison of uncertain situations involving monetary rewards: “Would you be willing to take a particular bet?” Probability assessment methods are explained later. Bayesian probability in decision theory has been pejoratively linked to this subjective aspect, that it involves assessing subjective judgment and personal beliefs, rather than objective measurements, as is the usual practice in science.

The wiki “Probability interpretations” [69] notes that “On most accounts, evidential probabilities are considered to be degrees of belief, defined in terms of dispositions to gamble at certain odds.” This may be an accurate portrayal of views still widely held. The wiki gives historical detail on contributors as the various interpretations of probability were developed. The reader may wish to consult this history, some of which is described later in this chapter, but two key authors, Richard T. Cox and E. T. Jaynes, are not mentioned.

Jaynes’ “logic of science” view seems preferable to the subjective and sometimes pejorative “betting odds” viewpoint on subjective probability. Gamblers are viewed by many scientists as irrational. As illustrated in the case studies later, one should build probability assessments based on data, models, and expert judgment and should use such probability assessments to support decision-making. Review by the experts who have been assessed and peer review by other experts are useful as part of the process. The information base supporting the probability assessment should be made evident.

Many scientists trained in statistics have been inclined to reject the use of subjective probability. If the numerical quantity is not a frequency in experimental data, then it should not be regarded as a legitimate use of the probability concept. An alternative interpretation is due to E.T. Jaynes and his immediate intellectual predecessors, geophysicist Harold Jeffreys [22] and physicist Richard T. Cox [9]: Probability as used in science and in decision theory should not be viewed as the output of an intrinsically personal process of judgment, but rather as a structure for scientific inference about uncertainty.

The viewpoint acquired from Jaynes and his intellectual predecessors (to Harold Jeffreys, add economist John Maynard Keynes and their Cambridge mentor, Professor William E. Johnson) is that probability theory provides the logic for

quantitative reasoning about uncertainty. Jaynes' posthumous text [21] carries the title *Probability Theory: The Logic of Science*. Jaynes' interpretation has other enthusiastic fans, such as Nassim Nicholas Taleb, author of *The Black Swan* ([52], see p. 319, notes on chapter 9), *Fooled by Randomness* ([53], see p. 289, notes on chapter 13), and *Antifragile* [54]. Many practitioners of risk and decision analysis are not familiar with the Jaynes and Jeffreys interpretation, sometimes called the “necessarist” school of probability. A central idea of from Jaynes and his predecessors is that the assessment of a probability reflects a state of information. All probability assessments should be regarded as conditional on this state of information. The state of information is not a physically measurable entity but a mental construct: what an individual has as knowledge, which may include a great deal of observations plus models and mental constructs used to organize this information. While in many applications information is shared, in a specific instance, different individuals may have different information about the same physical event.

As a very simple example, consider a coin toss. The coin is assumed to be “fair” in having a “Head” on one side, a “Tail” on the other, and tossed in such an imprecise way that these two outcomes become equally likely, and one of the two is certain to occur. Individual one (“Tom”) watches individual two (“Bill”) toss the coin high in the air, catch it in the palm of the hand, and then peek at the landed coin, so that Bill knows which side is up. Tom should assign a probability of 0.5 to the outcome of a Head. Individual two, Bill, has seen the result of the toss. (We assume good vision, so Bill will assign a probability of either 0 or 1 to the outcome of a Head.) The event is the same, the time of the probability assessment is the same, and the two individuals may generally agree on most items of information. But one of them has seen the outcome of the coin toss and the other has not. This example illustrates that all probabilities are conditional: Uncertainty can be a matter of who has what knowledge from one or more observations, rather than about the physics of the coin and the coin tossing system. Probability reflects the state of mind, rather than being a measurement of the state of a physical system, such as temperature as a measure of the energy in a material object.

The concepts from philosophy underlying this idea are that probability is the logic for reasoning, and the numerical probabilities we use reflect states of knowledge, not physical states of a system. The ideas come down from Plato through Immanuel Kant to his disciple, Hans Vaihinger, author of “The Philosophy of As-If,” written in 1877, published in German in 1911 and in English in 1924 [57]. We humans think using numbers and models, and we should not confuse these models with reality. They are useful “fictions” in Vaihinger’s writings. Human brains connect to underlying physical reality only through sensory data – observations, which may be numerical measurements. How well can we predict what we do not yet know? The answer lies in the realm of mental processing – in the brain(s) of the predictor(s), who use their state of knowledge to make the prediction. Uncertainty is a lack of knowledge – in the human brain, and not as some sort of objective reality. Probability, as a measure of uncertainty, reflects one’s state of mind, and not a state of things.

Is there another way of reasoning logically about uncertainty other than probability theory? Work by Richard T. Cox [8, 9], now called “Cox’s theorem,” shows that any logically consistent system for quantifying uncertainty based on propositional logic – that is, based on logical combinations of statements each of which is either true or false – must be equivalent to probability theory based on the Kolmogorov axioms, up to the choice of a constant, the number attached to the certain event. The convention is that the certain event has probability one. So we do not have to hunt for a new method for characterizing particularly deep or challenging uncertainties, if we are content to deal with propositional logic augmented by numerical expressions of uncertainty. Standard probability theory based on the Kolmogorov axioms should suffice.

Below is a simplified sketch of the proof of Cox’s theorem, as an application of logic in the form of Boolean algebra.

Let us consider a system for reasoning about uncertainty. We will assume that our measure, called “plausibility,” P , is a *real number*, and that degrees of plausibility are associated with real numbers within some interval range, which by convention we have set to be the interval from 0 to 1, with greater plausibility associated with larger numbers.

We now apply our plausibility system to a set of outcomes/events/propositions (denoted by capital letters) that obey the rules of propositional logic, in particular, Boolean algebra. We want each of the entities, which following Jaynes and others we will call propositions, to have *an unambiguous meaning with two-valued logic*: A proposition can be true or it can be false, but it cannot be inherently ambiguous. For example, consider the cat in Schrödinger’s famous thought experiment. Propositional logic requires that a cat be either alive or dead, so both the Proposition A (the cat is alive) and the denial of A, sometimes denoted A' , are well defined. Because the logic is two valued, the denial of the denial brings us back to the original proposition: If the denial of A' (the cat is not alive, but dead), the denial of the denial, A'' , is that the cat is not dead, but alive. Propositions may depend on information in the form of other propositions, but not on information in the form of “modal” operators of knowledge and belief. (For a discussion on modal operators and modal logic, see Stanford Encyclopedia of Philosophy, Modal Logic [51].) Examples of modal operators are “It is known that p ”; “It is necessarily the case that p ”; “It is possible that p ”; “It is not necessarily the case that not- p ”; “It is not known that not- p ”; “It is more likely than not that p ”; and so forth, where p is a proposition. Modal and other logics for reasoning under uncertainty have been developed, but they will not be further discussed here.

We call a collection of propositions established to be true to be a state of information. The notation $A|B$ denotes the event of A being true given that B is true. $A|BC$ is the event that A is true given that B and C are both true. And when we use the “and” to connect two propositions (denoted AB), we assume that they are not mutually contradictory, so that the conjunction (AB : logical AND) and union (non-exclusive logical OR: $A+B$) of the propositions make sense.

$$(A + B)|CD$$

is the event that at least one of the propositions A and B is true given that C and D are (both) true.

Greater plausibility is represented by a greater number, so we can write

$$P(A|B) > P(C|B)$$

to mean that, given B, A is more plausible than C.

If new information is obtained so that B becomes B' , and this change makes the plausibility of A increase, then we can write

$$P(A|B') > P(A|B)$$

But for another proposition C, the change from B to B' does not change the plausibility, then

$$P(C|AB') = P(C|AB)$$

and if we consider the plausibility of both A and C, it must increase because A is more plausible with B' than B, and C's plausibility has not changed:

$$P(AC|B') > P(AC|B)$$

The plausibility of the combined proposition AB, that A and B are both true, depends on the plausibility of B and of A given B, $A|B$. So the functional form of plausibility for AB must be a function of the plausibilities B and $A|B$. The Cox proof shows that conjunction of plausibilities must be associative, and since all strictly increasing associative binary operations on real numbers are isomorphic to multiplication (Cox [9], pp. 12–16; Jaynes [21], Chapter 2; Wiki, Cox's theorem [64]), then, our plausibility model might as well be ordinary multiplication:

$$P(AB) = P(A|B)P(B)$$

The final stage in the proof, that the functional relationship must be multiplication, is required by *consistency*: If the plausibility of a proposition can be derived in multiple (valid) ways, the numerical results must be equal. Suppose that $A+B$ is equivalent in plausibility to $C+D$. Then we acquire new information on A and then new information on B; the numerical results of conditioning on the new information must be the same as if we have acquired new information on C and then new information on D.

Jaynes [21] devotes most of two chapters to building up to Cox's theorem. The derivations of Cox and successors get into subtleties on the domain of what we have called propositions and descriptions of these propositions using set theory. The wiki on “Cox's theorem” describes relevant literature. The remainder of this chapter assumes that probability theory is the appropriate logic to use in dealing with uncertainty, assuming that two-valued logic applies and that ambiguities have been removed through careful definition. If propositions or outcomes of concern in

a decision context can be described as a set of mutually exclusive and collectively exhaustive propositions with associated numbers, then such ambiguities should be removed. (Difficulties arise when this set is not continuous, or not differentiable. Such difficulties are discussed in advanced probability texts such as Loève [31] and to recent papers such as by van Horn [58] and other scholars concerned with these difficulties in the context of Cox’s theorem. As a practitioner of probabilistic analysis, this author has, as yet, no cause to believe these difficulties pose serious practical impediments to using standard probability theory to characterize uncertainties.)

Other approaches. Many alternatives have been proposed, as opposed to using the standard system of probability to associate one number with an uncertain event. Mostly these alternatives have a behavioral motivation – people have difficulty or refuse to summarize their state of information on an uncertain quantity into a single number. One can use more numbers – a range, perhaps. More complex structures than traditional probability theory have their advocates. This discussion advocates sticking with traditional probability theory and an understanding that probabilities will change with new information: Bayes’ theorem requires consistency between probabilities assigned prior and posterior to the new information. Most people cannot reason consistently about uncertainty without extensive training in probability theory, especially if not trained to apply Bayes’ theorem. That is motivation for introducing some of the examples given below. Many people trained in probability in the tradition that probabilities represent frequencies in data cannot calculate the correct answer, because they were not taught to think about revising probabilities, but only of estimating probabilities from a stream of data from presumably independent, identical experimental trials. When the situation does not involve presumably independent, identical experimental trials, but rather complex influences on an event that is unique, traditional methods taught in statistics courses may seem inadequate. But probability theory does apply.

1.1 Many Problems with Using Probability Theory Remain

Ambiguity. Let us assume that we can apply probability theory to propositions provided they meet a test that the proposition is described at a level of precision so that a finding of true or false is unambiguous. If it is a hypothesis or theory, it can be confirmed or disproved; if it is an event or a numerical outcome such as a measurement, we can say at least conceptually that the result was in an interval (an estimate, plus or minus some amount to account for measurement precision), or it was not. We might invoke a “clairvoyant” test: A description of the event, outcome, or proposition must be sufficiently specific so that if a clairvoyant were to be asked if this event will occur, no further specification is needed: The clairvoyant is able to answer true or false to the question without any unresolved ambiguity.

Poorly supported inputs. If one uses probability theory in the form of a model to calculate probabilities of outcomes from probabilities placed on inputs, including

model structure and assumptions, the resulting probabilities on outcomes are no better in quality than these inputs. An applicable phrase is “garbage in, garbage out,” abbreviated GIGO. Logic in making the probability calculations does not compensate for weaknesses in the inputs. The probability numbers are simply a summary of a state of information about an uncertain event or quantity. They do not have an objective life of their own. In some situations these numbers can be based upon data: a frequency in repetitive situations. These situations are discussed in more detail as the special case of aleatory uncertainty. In other situations, the basis for the probability number may be the judgment of one or more experts. Motivational and cognitive biases and differing states of information among experts, whose judgment is considered appropriate as the best source for a needed input, can make the problem of getting high-quality input data formidably hard. Sensitivity analysis is a key step in achieving insight on how precision (or lack thereof) in the inputs affects the calculated output.

These critical matters of assessing expert judgment about uncertainty are a main focus in the two case studies that form the second half of this essay. The task is done when the experts understand the elicitation process and support the probability numbers. That happened in both case studies, and the process to complete this task took weeks to months.

Relying on expert judgment. When a probability number comes from expert judgment, it is a reflection of the information that this expert has. It is an output from a reasoning process, which may be fast and intuitive, slow with laborious mental processing of available information (e.g., pro and con entries on many pages of a yellow pad of paper), or done with the aid of a computer-implemented mathematical model that performs a large set of detailed calculations. The resulting probability number may, or may not, be a good basis for making a decision. Is this expert well qualified by virtue of training and experience? Is there motivational bias – the expert attempting to influence the output of the analysis in a way that might benefit the expert? More generally, is this expert trustworthy?

Most people, even those trained in probability and statics, are not intuitively good at reasoning about situations that are both complex and uncertain. People are much better at recognizing patterns – situations that offer opportunities or situations that pose danger. There is a large relevant literature in the social sciences on human judgment about uncertainty. An excellent introductory summary is Daniel Kahneman’s 2011 book [26], *Thinking, Fast and Slow*, which is still on the *New York Times* Science Best Seller List as of January 2016. An earlier 1974 paper in *Science* by Kahneman and his colleague Amos Tversky summarizes their early experimental data. This paper was later included in a 1982 book [27] with the same title, *Judgment under Uncertainty: Heuristics and Biases*. Carl Spetzler and Carl Axel Stael von Holstein published a paper based on Kahneman and Tversky’s research on assessing expert judgment about uncertainty [50].

The approach is as follows: Ask the expert to explain in detail on how the outcomes might occur. Make the reasoning, including assumptions/models (and mental models), explicit. When in doubt, construct models and use them as basis

for calculating probabilities on outputs from probability judgments on inputs. The second case study below illustrates how statistical data may be included.

Science and engineering are the “slow” thinking process in the title of Kahneman’s book [26]: Observe and record the data, hypothesize the relationships of how nature works, and check the accuracy of predictions based on these relationships with further experiments. Quantitative analysis is an intrinsic part of the process. As we come to understand cause-and-effect relationships for how nature works, we are enabled to make more accurate predictions. Many authors will refer to this process as the scientific method. Its contributions to improving the state of human knowledge and to the quality of human life have been enormous. But many people resist applying this same process to their own decision-making, in the ways described in Kahneman [26].

This is not to say that intuition based on experience has little value. On the contrary, human intuition and ability to recognize patterns can be of enormous value. But intuition often conflicts with what we can learn through a systematic process of collecting and analyzing the available information. People who have been successful in past decision-making may resist advice from people who have differing experience and backgrounds. People often make errors whose patterns are predictable by those who study human reasoning [26, 27]. Knowledge of these patterns is fundamental in assessment of judgment about uncertainty when this must be done by asking experts to do a mental integration of what they know and provide probability numbers. A good analyst will try to document what the experts know, achieve an understanding of their reasoning process, and check their judgment against that from other experts. This can be a lengthy and expensive process. Those who are used to the scientific method learn the high value of peer review, including careful checking of data quality and analytical methods. The use of probabilistic modeling has, as a major advantage, that the logic of probability theory is well established. There should not be surprises or errors because the logic is flawed. But for many decision situations, analysis must rely on human judgment, because the important uncertainties for the decision have not been resolved by a process of repeated experimental measurements providing frequency data on which to base the probabilities.

Combining the information from diverse sources – experts and data. Characterizing uncertainty in a complex decision context should involve adequate time learning about the uncertainty and how it arises. There may be knowledge about cause and effect. Various areas of science and engineering may apply. An analyst may need to learn new concepts and specialized terminology before engaging in dialogue with experts in the subject area, so as to understand their reasoning and assess their judgment in the form of probabilities. Different people may have knowledge of different aspects or of different events in a causal chain. An analyst may need to invest considerable effort in building a model of “how things work” before beginning to assess the uncertainties.

There are methods for combining the judgment of multiple experts by weighting their probabilities with calibration based on test questions [7]. Such methods can

be useful when a quick assessment is needed on a well-defined uncertainty, with a relatively low budget. But many experienced analysts [33, 34, 72] believe that there is no substitute for a process of in-depth dialogue with the experts – investing the time to learn the experts’ reasoning and, often, to build this reasoning into a model. An excellent process for assessing a complex uncertainty is that developed by Budnitz et al. for assessing probabilities from a set of experts on seismic risk at nuclear power plants [5, 6]. If the analysis budget does not permit such a costly process, information sharing, review by other experts, and documentation of the experts’ reasoning process can supplement assessment of uncertainties by eliciting probabilities from experts.

The best (but often, the slowest) process for reducing uncertainty may be the scientific method. Experts propose a hypothesis and refine it as further information is obtained. Many decisions won’t wait until uncertainty is resolved, through a process of hypothesis generation and then experiments to determine if the hypothesis predicts well, so the hypothesis can be upgraded to confirmed theory. Action may be needed more quickly. Those with decision responsibility may have to choose between taking action now in the face of large uncertainties, or deferring the action, accepting that the action may become more costly and less effective because of waiting until the uncertainties are reduced.

Epistemic versus aleatory uncertainties. Many processes involve variation in repeated occurrences. Statistical knowledge is accumulated in these situations, but not the ability to predict what will happen in a particular occurrence. Coin tossing and rolling of dice were examples used as probability theory was developed historically, as a way of understanding games of chance. These are special cases where, given that the underlying assumptions are valid, probabilities on complex events may be computed with high precision.

There is an important implicit idea here that the input probabilities can be established as true with high precision. This may be a matter of design and reasoning, or of having massive experience so as to test the probability number against the frequency observed in a large number of repeated experiments, or a combination of both.

For coin tossing, games of chance based on rolling dice, roulette wheels or wheels of fortune, and decks of playing cards, one can reason based on symmetry that there is equivalence in the probability among the set of possible outcomes. The coin has two sides, each die has six faces, the wheel is divided into a set of arcs of equal length, and for a very well-shuffled deck, the chance of drawing any particular card is one over the number of cards in the deck. Furthermore, the process either has no dependence on the past, as the coin toss, roll of dice, or spin of the wheel are done with sufficient imprecision in the initial conditions for the physics of the applied forces, so that these situations can be considered as identical, independent experimental trials with no dependence on the results from past trials. Such input probabilities are considered to be known, such as each of the six faces of a die having a probability of 1/6 of being on top when the die comes to rest after rolling. Given the input of 1/6 from the assumption of symmetry and the rolling process being

imprecise by design (e.g., a long toss with a bounce off the side onto the table), one can calculate quite precisely the frequency of winning in a large numbers of repetitions of a dice game such as Craps or a card game such as 21. (One of the founders of decision analysis, Ward Edwards, did such calculations, impressing a gambler sponsor who subsequently funded a series of psychology experiments as games of chance in the Four Queens Casino in Las Vegas [42].)

The same idea can be applied in other contexts. Life insurance is another example, and it illustrates how the process becomes more complicated. The length of a person's life is an uncertain quantity. People of the same age and sex without known health impairments or disabilities may be regarded as having equivalent probabilities of dying after n years. The insurance industry has used mortality data to develop life tables. The current versions are quite sophisticated. (Example: <https://www.johnhancockinsurance.com/life/life-expectancy-tool.aspx>.) Put in information on known health impairments, exercise pattern, alcoholic beverage consumption, driving record, etc., and observe how the expected remaining lifetime changes with these factors. There is an important message here. The probability distribution on a person's lifetime as determined from mortality statistics is dependent on a list of factors influencing life span. As more factors are added in the evaluation, probabilities that were initially equal can become unequal, because the more detailed state of information provides a basis for distinguishing among individuals. That same message applies in many other contexts. It is therefore helpful to consider that ALL probabilities are conditional on a state of information. Outside of games of chance, there are few situations where conditioning can be ignored. Ignoring possible conditioning factors is sensible if one can reasonably assume that, based on symmetry or experience, the probabilities of the uncertain outcomes are known to high precision, so knowledge of these additional factors is not significant for the assessments. But even in the context of games of chance, the simplifying assumptions underlying such "assumed-to-be-precise" probabilities may be absent. Consider flipping bent coins or thumbtacks, rolling loaded dice, or the spin of an unbalanced roulette wheel or wheel of fortune: The outcomes are now not equally likely.

Italian scholar Bruno de Finetti developed the concept of **exchangeable sequences** as the appropriate foundation for uncertain situations in which statistical methods apply. Such situations are often described as **aleatory** uncertainties. The basic idea is that a sequence of numbers X_1, X_2, X_3, \dots , representing experimental data, observations, or outcomes, can be considered as having a joint probability distribution that is invariant to relabeling the indices. In other words, the order of the data elements does not matter in making the probability assignments. Di Finetti's theorem states that the elements of such a sequence are conditionally independent of each other, given one or more latent underlying variables that are uncertain (i.e., described by a probabilistic model). For these variables the term **epistemic** uncertainty applies. Inference or learning about these variables can occur by observing elements of the sequence [10, 65].

Now we have a theory into which probabilistic models such as the Bernoulli process apply. For flipping thumbtacks instead of tossing coins, the probability of

a “Head” (point down, head of tack up) is unknown, but as one flips a thumbtack a large number of times, the observed frequency of heads compared to the number of flips converges to a number which is the probability of heads on a single flip. This is the process of statistical inference described in many texts on probability and statistics.

1.2 Bayes’ Theorem – A Consistency Condition Among Probabilities

How does one revise a probability distribution based on new information, such as the number of heads in additional tosses of a thumbtack? The process is a simple application of the logical consistency requirement in probability theory. Let A and B be two uncertain quantities, and consider how one can write the probability of the joint event A and B:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A).$$

Then, dividing, with an assumption that $P(A)$ is not zero (which would mean A is not possible),

$$P(B|A) = P(A|B)P(B)/P(A) \quad (40.1)$$

We have a new probability for B given that A is known to have occurred. $P(B|A)$ is often called the posterior probability, and $P(B)$ the prior probability, referring to time periods before and after the resolution of uncertainty on A, which revised the state of information. The expression $P(A|B)/P(A)$ is often called the likelihood function. The relationship works with A and B as random variables as this term is used in probability and statistics texts.

This relationship (40.1) is known as Bayes’ theorem, after the Reverend Thomas Bayes, an English minister, logician, and Fellow of member of Britain’s Royal Society. The relationship was published by a friend in 1763, 2 years after Bayes’ death.

With the combination of exchangeable sequences and Bayes’ theorem, much of traditional statistics can be reinterpreted as using new information to revise probability distributions.

Here is a simple example. A fair coin is flipped four times. You are told that the outcomes included at least one head. What is the probability of all heads?

Many people who have taken classes in probability and statistics fail to get the correct answer. Finding the correct answer is simple for those not trapped in the thought pattern of traditional statistics, where the usual assumption is that probabilities do not change. Draw a “tree” diagram showing the sequence of possible head-tail sequences of four flips. There are 16 different sequences, ranging from HHHH to TTTT. Only the last is eliminated by the information that there was at least one head, and the relative probability of the other 15 outcomes remains

the same. The answer is 1/15. The reasoning is a very simple example of Bayes' theorem.

A more complex application is the “Monty Hall” [68] problem arising from the American television show, “Let’s Make a Deal,” and publicized in Marilyn Savant’s widely read column “Ask Marilyn” in *Parade* magazine.

Suppose you’re on a game show, and you’re given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what’s behind the doors, opens another door, say No. 3, which has a goat. He then says to you, “Do you want to pick door No. 2?” Is it to your advantage to switch your choice?

The correct answer is to switch the choice of the door, which increases the probability of winning from 1/3 to 2/3. Many readers of the magazine claimed this answer was wrong. It is counterintuitive, and as such, an excellent example of why logical reasoning is needed in complex uncertain decision situations. Apply Bayes’ theorem: The probability of the car being behind door 1 is 1/3. The host now reveals that the car is not behind door 3. So it must be behind either door 1 or door 2. Has the probability on door 1 changed? No, the information from what the host has done should only change the probability on the other two doors. The host is not going to open the door with the car or the door that you chose. He opens the third (No. 3), one with a goat. The other door, No. 2, now has the probability that you would have earlier placed on door 3, plus its earlier probability: $1/3 + 1/3 = 2/3$.

At this point readers who may be unfamiliar or only partially knowledgeable have been introduced to Bayes’ theorem and exchangeable sequences, two key ideas for how uncertainty can be characterized in quantitative form using probability theory. Probabilities reflect a state of information that changes as new information is obtained. Probability illustrated through games of chance is unusual, because in this context people believe they know the input probabilities and can compute with confidence probabilities of complex sequences of events, such as winning at the dice game Craps, drawing a Royal Flush hand in poker, etc.

1.3 A Basic Desideratum: Two States of Information Perceived to be Equivalent Should Lead to the Same Probability Assignments

The basic desideratum is a powerful fundamental principle for testing whether probability assignments are properly done. This principle is quite different than the subjectivist test of whether the person being assessed would be willing to bet using these probabilities. The idea comes from Laplace’s writings from two centuries ago [30], and it has been developed by E.T. Jaynes and others who have followed the idea that probability is a logic that can be applied broadly to uncertain situations and not restricted to repetitive situations that meet the test of exchangeable sequences.

Two applications of this basic desideratum have already been discussed: the assignment of probabilities to symmetrical objects used in games of chance, the fair coin, the die, and equal arcs on the roulette wheel or wheel of fortune. By Laplace’s Principle of Insufficient Reason, if the sides of the coin, the six faces of the die,

or the equal arcs around the wheel are judged equal in the sense that relabeling leads to an equivalent state of information, then the probability assignments should be the same: $1/2$ for each side of the fair coin, $1/6$ for each face of a fair die, and $1/360$ th times the degree of arc for the wheel, (with correction for space taken up by dividers).

The second application is exchangeable sequences. It's the same idea – the probabilities are invariant under rearrangements of the order of the elements in the sequence. Using Bayes' theorem enables observations of sequence elements to revise probability distributions on parameters, which turn out to have meaning as frequencies in the limit when the sequence has a large number of repetitions. The most well-known models from probability theory fit this description: the Bernoulli process, a generalization of the coin-tossing game to binary events where the probability of heads versus tails (or success versus failure) is unknown; the Poisson process where nonnegative numbers come from events such as the time of the next arrival for each of a succession, such as customers at a service center or decays of radioisotopes; and the Gaussian or normal distribution composed of the sum of lots of small independent increments – random walk, velocity of molecules in an ideal gas, or distribution of a characteristic such as height in a male or female human population. Readers may have learned about such processes from a standard text on probability and statistics. But what they may not have learned is that these probability distributions are special cases in which Bayes' theorem can be done by simple arithmetic on parameters, because observations can be characterized by sufficient statistics.

E.T. Jaynes' writings including his 2003 textbook assert maximum entropy as a principle for choosing probability distributions. Is maximum entropy an appropriate principle for this purpose? An answer is provided in North [38]. Maximum entropy turns out to be an application of the Basic Desideratum.

Howard Raiffa, his Harvard colleague Robert Schlaifer, and others noted the special character of probability distributions that have a “conjugate prior,” such that revision from prior to posterior for these distributions by Bayes' theorem can be done within the same family: Bernoulli, Poisson, normal, gamma, and a few others [46]. The relation to sufficient statistics is clear. For example, in the Bernoulli process a prior distribution is used for the parameter ρ , which corresponds to the long-run frequency of heads and the probability of a head on the next flip with current information. The beta prior probability distribution is defined over the interval from 0 to 1, with two parameters u and v :

$$\text{Beta}(u,v) = P(\rho|u,v) = B(u,v)\rho^{(u-1)}(1-\rho)^{(v-1)}$$

where $B(u,v)$ is a normalizing constant (made up of gamma functions) such that the probability density function integrated over $[0,1]$ sums to one. If r heads are observed in n trials, the posterior distribution is beta with $u+r$ and $v+n-r$ as the parameters.

(Detailed discussion can be found in many texts and papers, such as Navarro and Perfors [37]).

Similar discussions are readily found for the Poisson process and exponential distribution with unknown mean inter-arrival time and the Gaussian or normal distribution with unknown mean and variance.

The mathematical proof showing the equivalence of the exponential family of probability distributions with sufficient statistics with the solutions of finding the probability distribution that maximizes entropy subject to constraints that correspond to knowing the mean and/or higher moments of the probability distribution is found in North [38]. Readers interested in these details can find them there. The following is a sketch of the argument, for discrete outcomes (intervals) which become in the usual take-the-limit argument in calculus arbitrarily small with a correspondingly large number of them.

Consider the data from a series of observations from an exchangeable sequence. A histogram of the results summarizes these data. Relabeling the observations does not change the histogram. Count all the sequences that lead to a given histogram, and these sequences are judged to have equivalent probabilities.

Let n be the number of trials and n_k the number in the k th interval, where k runs from 1 to N and N (as well as n) will be large numbers, with $n \gg N$, and for all k , $n_k \gg N$. Let $f_k = n_k/n$, the frequency of the k th interval in the n trials.

The number of ways W to arrange n objects in N categories is the multinomial expression:

$$W = n! / n_1! n_2! \dots n_N!$$

Sterling's approximation gives $n! = e^{-n} n^n \times \sqrt{2\pi n}$

Therefore,

$$\log n! = n \log n - n + O(\log n)$$

where $O(\log n)$ are terms that increase no faster than as $\log n$ as n gets large.

Then

$$\log W = n \log n - n - \sum_{k=1}^N n_k \log n_k + \sum_{k=1}^N n_k + O(\log n)$$

Dividing by n , the number of trials,

$$1/n \times \log W = - \sum_{k=1}^N f_k \log f_k + O((\log n)/n)$$

As we go to the limit of very large n , this goes to the entropy function:

$$H(p_1, \dots, p_N) = - \sum_{k=1}^N p_k \log p_k$$

Essentially, entropy is counting the most likely histograms as the numbers n and N become large. One might think of this as follows: The exchangeable sequence process becomes random, subject to what is known on expected value of functions of the probability distribution, such as its mean, variance, and possibly higher moments. The usual proof of the Central Limit Theorem follows the essentially the same logic.

In case of conjugate prior-posterior families, the insight is that all one needs to do for Bayesian updating is to keep track of statistics such as the sample mean and sample variance, and other aspects of the observational history average out in a randomization process. There are a number of important applications in physics to this line of reasoning. It means one can avoid the need for an ergodic hypothesis, an assumption that time averages equal ensemble averages. Rather, for a system in equilibrium, in assigning probabilities on the state of the system, it should not matter when the experimental observation occurs. So, one can randomize the time of the experiment, and the probabilities on the state of the system should remain the same. Invariance to such randomization leads directly to the probability distributions developed by J. Willard Gibbs for statistical mechanics, without requiring ensemble as an intermediate concept [38].

Here is a summary of applications of the basic desideratum:

- (1) relabel outcomes: Laplace's Principle of Insufficient Reason;
- (2) Rearrange order of experimental results: de Finetti's concept of exchangeability.
- (3) Invariance to transformation – transformation groups. (Not covered here: see Jaynes [20, 21] and North [38])
- (4) Invariance to the time of an experiment to measure a dynamic system: the concept of equilibrium in statistical mechanics.

This brief digression into conjugate prior/posterior families of probability distributions explains why these families of probability distributions have both sufficient statistics and a maximum entropy characterization. These families are special cases where uncertainty can be described as random or having no memory, with relevant information for Bayesian updating very simply characterized, as by a sample mean and variance. With lots of small fluctuations or influencing forces, the Central Limit Theorem applies, and the composite distribution describing the system's behavior as a sum of these fluctuations ("random errors" or "random walk") becomes the normal distribution. Exponential inter-arrival times describe a process in which succeeding events do not depend on the timing of preceding events, but rather a process where the probability of an event in each increment of time is the same.

It seems like excellent intellectual background to think about complicated systems as having "state variables" that describe what information is needed to make good predictions on what will happen to these systems. In physics such state variables often work well to enable predictions: Energy is conserved and evenly distributed in a system at equilibrium. Therefore, knowing the temperature of a volume of gas in a box allows accurate prediction of the distribution of velocities of the gas molecules that compose the volume of gas: The result, due to Maxwell,

is that the velocities follow a normal distribution, and for gases where molecules do not interact (ideal gas assumption), this distribution is experimentally verified as correct [67].

In most applications in risk and decision analysis, we are not dealing with the orderly randomness of the normal or exponential probability distributions. But thinking about what state variables influence the behavior of a complex system will help to identify what variables are important to include in a description of the state of information. Conditioning on these variables may enable describing how one uncertain quantity is linked to another. No linkage, or independence, of one uncertain quantity from another, may be motivated by a lack of any known causal relationship between them. But in a physical, economic, political, or interpersonal situation, it is often the case that there are linkages between uncertain events. Then conditional probabilities should be used to characterize these linkages where the occurrence of predecessor events can make a successor event more, or less, probable. And there may be changes in the probabilities with time – a nonstationary process. An analyst wishing to characterize an uncertain event in terms of a probability should consider carefully how the event probability might change over time or with further information about other, possibly related, events.

It is therefore advisable to consider all probabilities as conditional on one's state of knowledge. Markov models in probability theory consider transitions among states, and such models may be useful. The second case study below included a Markov model, but when probability numbers became established, the complexity of considering multiple transitions among states was judged unnecessary.

Networks of events linked through having the probability of successor events depend on the outcome of predecessor events that have become widely used both in decision analysis as influence diagrams [18] and in computer science as Bayesian networks or Bayesian belief networks [63, 66].

1.4 Decision Analysis

What is decision analysis? This term comes both from Howard Raiffa [45] and Ronald Howard [15]. Decision analysis is as a process, rather than a product. The case studies in the latter part of this chapter can be viewed as products, descriptions of what came out of a process. The National Research Council reports of [35] and [36] emphasize analysis as a process. See also Fischhoff [11].

Decision analysis as a process supports decision-making by those with decision responsibility. It involves characterizing uncertainties, which are usually what make decision-making difficult. Putting the task in the second person with the reader as the decision-maker, the process involves structuring through quantitative analysis your *alternatives* (what you can do); your *information* (what you know, including information relevant to uncertainties), and your *preferences* (what you want) to identify the alternative(s) that are *best* in terms of maximizing expected utility. A good decision is one that is logically consistent with the alternatives, information,

and preferences. It is to be distinguished from a good outcome – ultimately, what you want. In the presence of uncertainty and the absence of clairvoyance or wizardry, is there a better way to get good outcomes than to make good decisions? Such a better way has not yet been identified. Over many decision situations the best way to get good outcomes is to make good decisions. Now make a transition from a personal orientation from you as the decision-maker to providing decision support for other people – an individual, a group, or a government agency trying to act in accordance with law – who have decision responsibility: The “you” becomes “they” – and “they” may not agree among themselves.

The existential reality is that humans, as individuals and in groups, must make choices in the face of uncertainty. This is not usually done well in the sense of logical consistency with what they know, what they can do, and what they want. Is there a better way to get good outcomes than by carefully assessing situations, planning ahead, and otherwise trying to make good decisions in managing resources and opportunities?

As available information changes, one can use an informal analogue to the scientific method: Learn from your mistakes, and use what you learn to improve your judgment and your decision-making. Tetlock and Gardner [55] describe how well this can work for forecasting events of the kind that are of interest to the intelligence community. Learned skill in dealing with uncertain situations enables improved decisions for oneself and support for improved decision-making by others.

Probability theory is not limited to frequencies in repeated measurements or subjective judgments made in the form of choices on whether to accept a bet. Probability theory is a branch of mathematics that facilitates logical reasoning about uncertainty, by associating a numerical measure, on a scale of zero to one. Very small numbers approaching zero imply that an outcome is unlikely to impossible, while numbers approaching one imply the event is highly likely to certain. Especially in recent decades, many people have come to use in common parlance the concept of probability as quantified uncertainty. This handbook chapter discusses how the process of using probability as quantified uncertainty can be done well, as opposed to not so well.

Probability theory facilitates reasoning about uncertainty in a way that is unambiguous and logically consistent. It is the appropriate logic for scientific inference. The “scientific method” is applied to the available information, which implies that analysis should keep track of significant changes in the information.

Preferences. “What you want” is an area of similar complexity to dealing with information and uncertainty. Valuation in the context of making choices is the subject of economics and, more broadly, social sciences – the study of how human beings make choices. What do people want, as individuals and as a group of individuals who share in outcomes? Most individuals will wish to avoid being subject to “Dutch Book” – becoming a “money pump” by a sequence of trades violating transitivity, such that a person gets back to his or her starting point but with less money. It may not be possible to construct transitive preference orderings

for a group from the preference orderings of the individuals in the group, even if individual preference orderings are transitive. Majority voting may not work: The Condorcet paradox on majority voting [71] and the Arrow Impossibility Theorem [1, 62] show that construction of preferences for the group obeying transitivity may not be possible. Cultural change, ethics, and morality may come into play in helping members of a group agree on preferences. To the extent a group or society can agree in the sense of having transitive group preferences; preference measures can be constructed to evaluate outcomes. But obtaining agreement on the ordering of good outcomes in the context of a public policy decision may involve considerable difficulty [14].

Decision analysis involves defining outcome events related to the decisions so that both the answers to questions “What will happen?” and “What do those responsible want?” can have answers that are well defined and not ambiguous. And the process continues, to obtain for each outcome or event a numerical measure of uncertainty (a probability) and a numerical measure of value. (This “numeraire” may be a utility, or in the absence of a nonlinear measure of risk preference, it might be a value assessed with dollars or another currency.) Cost-benefit analysis involves assessing costs and benefits from the outcomes from the choice among the set of decision alternatives. More sophisticated versions examine the distribution of costs and benefits and not just the aggregate amounts of cost and benefit across the population that experiences the outcomes.

Discussion of multi-attribute utility. If the von-Neumann-Morgenstern axioms hold, then one can construct a multi-attribute utility over a set of outcome measures in two stages [12, 13]. (1) Deterministic trade-off valuation: Use one scale as the numeraire. (It is often convenient to do this with a currency – money, such as US dollars.) One can add or subtract monetary value for distributional impacts. A Pareto criterion avoids this judgment, but it may not allow some alternative policies to be distinguished from each other. This is consistent with usual practice for cost-benefit analysis in economics. (2) Use a risk preference measure, constructed as utility function on money. All ways of constructing multi-attribute utility function by trade-offs with risk should give the same result. Often risk preference is not included on public sector decisions: Risks are judged small relative to resources of government, so an expected monetary value criterion is argued to be adequate. Exponential utility with one parameter – constant risk aversion – allows sensitivity analysis for the impact on the decision of choice of risk attitude [43].

2 Two Case Studies

2.1 Case Study #1: Terrestrial Microbes on Mars

In 1972 the author and his colleagues at SRI International received a challenging assignment: to assist the US National Aeronautics and Space Administration (NASA) by making an assessment of the probability that the planned 1976 landing

would result in contamination of the planet Mars by terrestrial microorganisms transported by the spacecraft.

At this time both the USA and the Soviet Union were initiating a program of unmanned exploration of nearby planets. Mars was a particularly important destination, since it was viewed as having the best chance of having indigenous life. After photographs from a flyby mission showed what looked like evidence of past flows by liquid water on the planetary surface, leading scientists questioned whether the planned landing of a spacecraft on the Martian surface might pose an unacceptable risk of introducing living microbes transported from Earth that could proliferate on Mars.

The USA and the Soviet Union then agreed that both nations would carry out their programs so that the risk of planetary contamination, defined as one replication by one microbe in the Martian environment, would not exceed $1/1,000 (10^{-3})$. NASA determined that for the first landing in 1976, Project Viking, the risk would not exceed $1/10,000 (10^{-4})$. It was the job of NASA's Planetary Quarantine Officer (PQO) to veto the mission if the risk exceeded this allowed limit. The author and colleagues carried out an assessment that persuaded the PQO and the distinguished group of scientists advising him that the Viking Lander was below this risk limit.

Given the scant knowledge available about Mars in the early 1970s, how does one deal with the difficulties of assessing the probability that a terrestrial microbe carried aboard the spacecraft could reproduce there? NASA had performed such an assessment, but there were doubts by scientists advising NASA as to whether the results were reasonable. The task was not only to carry out a new assessment but also to communicate with the scientists who had raised these doubts. (This work for NASA is documented in [25] and [41]. A summary is in [39].)

It was well established that the Viking Lander contained a large number of microbes (about 20,000) that were viable in the sense that they were capable of reproducing in a suitable environment. Many were in the form of spores that were encased in the plastic used to protect the spacecraft's solid-state electronics against physical damage from vibration during the mission launch. The scientists believed that microenvironments might be present on Mars where usable water might be present (i.e., in a liquid or near-liquid state, such that microbes could reproduce). The landing site was chosen to be in a dry area, but elsewhere on Mars useable water and nutrients needed for reproduction, such as organic carbon, might be present. Could the microbes get from the place where the spacecraft landed to one of these microenvironments? For example, might dust or sand driven by the high winds on Mars erode the spacecraft, releasing microbes from the plastic in which they were encased? Might a viable microbe then be carried by the Martian winds to a favorable microenvironment in some location, perhaps quite distant from the landing site? For reproduction to occur, the microbe would need to be capable of reproducing in the absence of oxygen (i.e., facultatively anaerobic) and at the extremely low temperature of the Martian surface (facultatively psychrophilic).

The adopted approach used the idea of conditionality and tracked the sequence of events from the landing to a possible reproduction. It is similar to the process for modeling safety issues and the transport process for pollutants emerging from

a source and reaching a receptor area where vulnerable species might be damaged. The model begins with an initiating event, the landing of the spacecraft with its bio-burden of viable terrestrial organisms (VTOs) in different locations on the spacecraft. The next event, release, depends on whether the landing is soft as planned or a hard landing due to equipment malfunction and, also, where on the spacecraft the organisms are located. The analysis team considered three means by which organisms might enter the Martian environment – implantation beneath the surface, such as from being on the bottom of a landing pad or in a buried fragment of the spacecraft that broke up from a hard landing. An organism on the surface of the spacecraft might come off from vibration and land on the soil surface. Wind-driven dust and sand might erode the spacecraft over a long period of time, releasing organisms encased in the plastic or on interior surfaces. The model included all of these modes of release. The next stage is the transport of an organism to a favorable microenvironment. Does such a microenvironment exist, with water near enough the liquid state to be usable for reproduction by the organism? Are the needed nutrients available? Does the organism have the capability to reproduce in the absence of oxygen and at temperatures well below where water normally freezes? The model also included the possibility that the life detection experiment on the spacecraft becomes contaminated, resulting in a large amount of microbial reproduction on the spacecraft and a much larger subsequent release of organisms. This structure for the probabilistic model is shown in Fig. 40.1.

The first box on the left summarized information on the bio-burden: the number of viable terrestrial organisms on board the spacecraft at the time of landing. The estimate took into account the effect of the heat sterilization processes for the components, data on microbial spores in the plastic, estimates on organisms that might be trapped between surfaces as components of the spacecraft were assembled, and recontamination after final assembly, including the potential for contamination of the life detection experiment. The output is the estimated number of organisms in each of five categories of location. The second box includes the outcome for landing, at the planned low velocity with three pads in contact with the Martian soil or a much higher velocity “hard” crash landing that would cause the breakup of the

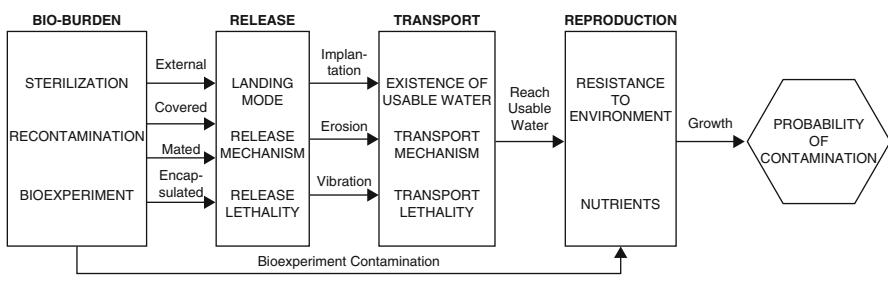
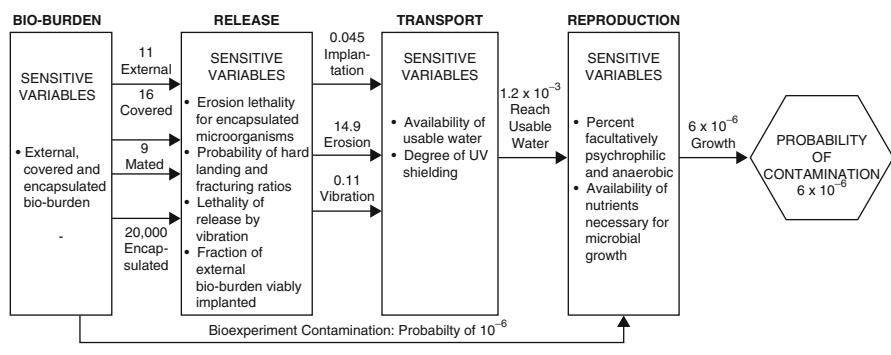


Fig. 40.1 Viking Lander contamination model (Retrieved from <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19750005702.pdf>)

spacecraft with pieces buried deep into Martian soil. The outputs are estimates of the expected number of organisms implanted into the soil, released through wind-driven dust or sand erosion into the Martian atmosphere and released via vibration or other nonviolent processes from the spacecraft onto the Martian surface. The third box deals with transport to a possible favorable microenvironment, a location with water near enough the liquid state at some time in the Martian year (i.e., “useable”) so that reproduction might occur. The inputs are the probability that such a microenvironment exists on Mars and the fraction of the surface area that it covers, given it exists. The third box is a relatively elaborate six-state Markov model allowing for multiple transport events (by a series of Martian dust storms) from release from the spacecraft to contact with useable water. (Details are in Judd et al. [25]). The output of the third box is the expected number of organisms that reach useable water. The fourth box considers availability of nutrients and the characteristics needed for the organisms to reproduce in the cold and without oxygen. If these conditions for reproduction are met, the outcome is “growth,” i.e., planetary contamination.

The model then gives a probability for contamination built up from approximately 20 inputs describing the microbial contents on the landing vehicle, the probabilities assigned to the mission outcome (nominal landing and failure modes), and Martian environmental conditions. The scientists who provided these inputs found it relatively easy to understand the structure of the analysis, the judgments they were being asked to make, and how these judgments would be used. When a complete set of inputs was assembled, the assessment was reviewed with these scientists. The nominal results are shown on the arrows in Fig. 40.2. The nominal case result is a probability of 6×10^{-6} , a factor of about 16 below the mission risk limit.

Extensive sensitivity analyses were carried out to see how the calculated probability of contamination changed with changes in the inputs to the calculation, singly and in combination. An illustrative set are shown in Table 40.1.



Numbers on arrows give the expected numbers of VTOs

Fig. 40.2 Viking Lander Mission contamination model with results (Retrieved from <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19750005702.pdf>)

Table 40.1 Viking Mission contamination sensitivity analysis

Contamination model variables	Probability of contamination						
	Values				Units: = 10** - 6		
	Extreme Low	Intermed. Low	NOMINAL	Intermed. High	Extreme High	Nominal: Low	5.9 High
Bio-burden variables							
1. bio external	2.2	5.5	11	22	55	5	10.7
2. bio covered	3.2	8	16	32	80	3.1	20.2
3. bio encapsulated	4,000	10,000	20,000	40,000	100,000	5	10.4
Release variables							
1. rel hard landing probability	0.0004	0.001	0.002	0.004	0.01	5.2	9.6
3. rel newly exposed/hard, Encaps	0.0001	0.0002	0.001	0.005	0.01	5.4	10.9
4. rel implanted, soft	0.0001	0.0002	0.001	0.005	0.01	5.7	8.7
6. rel VTO/vibration	0.001	0.002	0.01	0.05	0.01	5.4	11.1
9. rel VTO/erosion, Encaps	0.00001	0.00002	0.0001	0.0005	0.001	5.4	10.9
Transport variables							
1. tra survive transit	0.001	0.002	0.01	0.05	0.1	2.2	45.2
2. tra find water	0.0005	0.001	0.005	0.025	0.05	1.5	49.9
4. tra water deposition	0.00005	0.0001	0.0005	0.0025	0.005	5	15.2
5. tra stay lodged	0.1	0.2	0.5	0.8	0.9	5.5	10
Reproduction variables							
1. rep psychrophilic, anaerobic	0.005	0.01	0.05	0.1	0.25	0.6	29.6
2. rep availability of nutrients	0.01	0.02	0.1	0.2	0.5	0.6	29.6

The pattern of numerical results suggested a reason why the calculated risk was more than an order of magnitude below the risk limit of 10^{-4} . Most of the release of viable organisms was through erosion. The Martian atmosphere has no ozone layer to absorb the ultraviolet (UV) radiation in sunlight, so a high UV flux reaches through the Martian atmosphere down to the surface. A terrestrial microorganism would be killed by this UV radiation unless it was protected by shielding, such as being encased in a piece of plastic much larger than the microorganism itself. The density of the Martian atmosphere is well known from observation by telescope from Earth. Only small particles would be transported a significant distance from the spacecraft, even in the periodic Martian dust storms. These small particles would not provide adequate shielding to protect the microbes from the UV radiation. It was therefore highly unlikely that microbes could survive a journey from the Viking landing site to a favorable microenvironment if such microenvironments exist on Mars. This insight was not known previously within NASA and its scientific

advisors, but the scientists advising the Planetary Quarantine Officer and other senior officials at NASA accepted the reasoning. Within a short time the scientists who had expressed concern about the risk of contamination agreed that the objective of holding the risk for the first Viking Mission below 10^{-4} had been met and that the Viking Mission should proceed without the need for further steps to reduce the microbial load on the Viking Lander. The Viking Mission flew on schedule and the landing on Mars in 1976 was successful.

In 2015 more information has become available. Microenvironments have now been observed in the form of dark streaks that appear in several locations on Mars when temperatures are above -23°C . These streaks are thought to be hydrated minerals – a mixture of salts and water. And particles do migrate, even moderately large ones. Sand-grain-size particles have been observed to impact on the solar panels of the Mars Rover vehicles. Theoretical calculations and orbiter photos indicate that sand dunes on Mars move [23, 24, 28].

The authors have not been asked to redo their analysis based on such new information. In 1992 a committee of leading scientists stated, “... it is the unanimous opinion of the task group that terrestrial organisms have almost no chance of multiplying on the surface of Mars and in fact have little chance of surviving for long periods of time, especially if they are exposed to wind and to UV radiation” (Space Science Board [49], page 49). The insight from the analysis had become conventional wisdom. But revisiting the analysis might suggest that NASA should be concerned about the risks of contamination from a rover or a lander near the areas where the streaks that might be liquid water have been observed.

Calculating the probability that a terrestrial microorganism could reproduce on Mars would seem like a situation where there are no data, an example of deep uncertainty and an absence of the information needed for expert judgment. But in this project the analysis team broke the assessment into pieces and assembled judgments from experts via the model described above. Analysis using this model provided a basis for NASA to decide to proceed with the mission.

2.2 Case Study #2: Decision Analysis of Hurricane Modification

The second case study was done a few years earlier. It was on a decision to deploy a new technology in a domain where it was previously assumed that nothing could be done to avert natural disaster but to warn the potentially affected population and, in severe situations, to evacuate them from the area.

Hurricanes are intense cyclonic storms with winds in excess of 74 miles per hour that form over warm ocean water, such as in the Caribbean or adjacent areas of the western Atlantic. In the Pacific and Indian Oceans, these storms are called typhoons, another name for the same natural phenomenon.

In the early 1960s scientists at the National Hurricane Research Center developed a theory that seeding the clouds around the “eye” (the calm circular center) of a hurricane could reduce the intensity of the maximum winds and the storm surge, so that a hurricane impacting a populated coastal area should become less destructive. The theory held that seeding with silver iodide crystals would cause supercooled

water to form into ice crystals, releasing heat that would cause the cloud bank around the eye wall to move outward, reducing the speed of the maximum winds [70]. Small-scale experimental seeding had been carried out on two hurricanes after the theory was proposed, and a computer-implemented model of hurricane dynamics and experiments had been developed. Strong evidence supporting the theory came from cloud seeding on Hurricane Debbie on August 18 and 20 of 1969: After the first day of seeding, a reduction of 31 % in the maximum sustained winds was observed. The hurricane was left alone for a day and then reseeded, with a subsequent wind speed reduction of 15 %.

Our analysis team from SRI International was engaged subsequent to the experiment on Hurricane Debbie. The team was asked by the Assistant Secretary of Commerce for Science and Technology, Myron Tribus, to carry out a decision analysis on whether the government program, Project Stormfury, should be moved from a research project into an “operational” mode that could seed hurricanes threatening coastal areas of the USA.

This analysis was subsequently published in *Science* [19]. The following description is a summary from that publication, with an emphasis on the assessment of uncertainty and the valuation of information from opportunities for further experimental seeding of the kind that had been done for Hurricane Debbie in 1969. While currently available scientific information no longer supports the Stormfury theory, the case study illustrates a number of aspects that apply in other decisions on deploying a new technology in uncertain situations that could evolve negatively as well as positively. The potential for benefit must be weighed against the risk of a negative outcome, and public perceptions become important when a government agency takes responsibility for introducing technology into a natural process that has a potential for harm.

The property damage caused by a hurricane is strongly related to the speed of the maximum sustained winds. These winds do damage directly and through the storm surge, pushing seawater to abnormally high levels as the hurricane impacts a coastal area, causing both flooding and structural damage from wave action.

A simplified view of illustrating the difficulty is shown in Fig. 40.3. Supposing a hurricane is about 12 h away from impacting a populated coastal area. Natural forces might be causing the hurricane to intensify, but with current knowledge the change in the hurricane winds cannot be anticipated. Meteorologists cannot predict accurately how the behavior of the hurricane will evolve over time.

The diagram shows a possible scenario: If the hurricane is seeded, the wind speed is reduced from $w(t_1)$ to $w'(t_1)$, so that seeding has reduced the wind and the property damage. But when the hurricane makes landfall, it has intensified – less if it has been seeded, but the wind speed has increased. Those people suffering property damage might blame a government decision-maker for making the choice to seed.

At the time of our analysis, nearly all the government meteorologists that we questioned said that they would seed a hurricane threatening their homes and families – but only if they could be released from professional liability. The trade-off between accepting the responsibility for seeding and accepting higher probabilities of severe property damage is a crucial issue in this decision. There are many similar

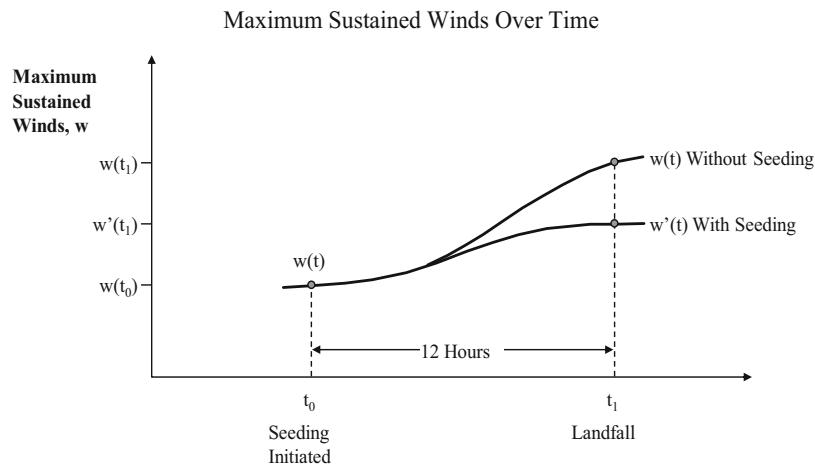


Fig. 40.3 Maximum sustained winds over time

situations where new technology may bring benefits, but not enough is known to assure that there are not substantial risks.

The consequences of seeding and not seeding are characterized with probability distributions. Our team developed these distributions from data on the changes in central pressure of hurricanes, a standard relationship between wind and central barometric pressure used in predicting hurricane winds and verified by wind observations, plus judgment of these government experts on the additional uncertainty introduced by seeding. The team considered three outcomes: (H1) The “beneficial” Stormfury hypothesis: The effect of seeding averaged over many hurricanes is a reduction of the maximum sustained wind speed. (H2) The “null” hypothesis: Seeding on average induces no change in maximum wind speed. (H3) The “detrimental” hypothesis: On average seeding increases the maximum sustained wind speed. These outcomes are mutually exclusive and collectively exhaustive: Only one can be true. Available knowledge is characterized by probabilities over these three hypotheses. Bayes’ theorem allows a logical revision of these probabilities, from before the Debbie experiment, after the results from the Debbie experiment became available (the time of the analysis), and to a future time when more seeding experiments might yield results.

Using the central pressure to wind speed relationship with the available historical data gave a probability distribution for the unseeded and for the “null” hypothesis. This distribution seemed to fit reasonably well to the normal or Gaussian distribution, and to first order, the 12-h wind speed change in percentage terms stayed the same as the initial hurricane wind speed varied. The team decided to stay within the two-parameter normal family of probability distributions for characterizing 12-h wind speed changes, with an important distinction between the average change – the mean – over many hurricanes, and the wind speed change in a specific hurricane, which was estimated with limited precision from the central pressure change data.

Probabilistic Model for 12-Hour Change in Maximum Sustained Wind, Natural Hurricane

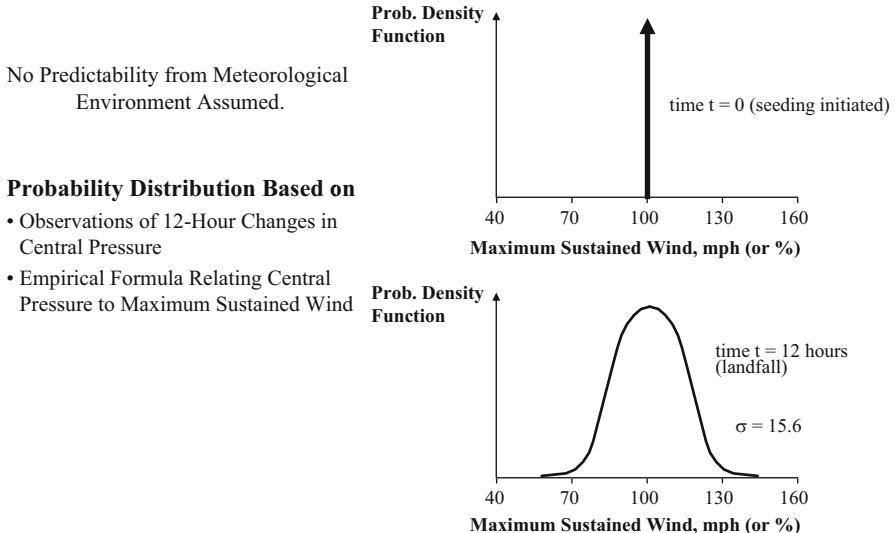


Fig. 40.4 Probabilistic model for 12-h change in maximum sustained wind, natural hurricane

The standard deviation for this unseeded (and “null” if seeded case) was 15.6 %. See Fig. 40.4.

For the “beneficial” seeded case, the team of analysts and experts agreed on a 15 % reduction on average and a standard deviation of 7 % for the effect on individual hurricanes. For the “detrimental” case, the corresponding numbers were a mean increase on 10 % and the same standard deviation, of 7 %. The standard deviation for an individual hurricane under H1 or H3 is the square root of the sum of the squares, 18.6 %.

The team went through an informal process of assessing probabilities on the three hypotheses but using the mathematics of Bayes’ theorem. Establishing probabilities for the three hypotheses was a crucial aspect of characterizing the overall uncertainty on how seeding could reduce wind speed and property damage.

If one starts with equal prior probabilities on the three hypotheses, then using Bayes’ theorem to update after the Debbie results, the posterior probabilities were H1 = 81 %, H2 = 15 %, and H3 = 4 %. The team used a lower value for H1, based on agreement that (1) before Debbie, H1 and H3 were approximately equal but with lower probability than H2, and after Debbie, H1 and H2 were approximately equally likely. The probabilities that met these conditions were H1 = H2 = 0.49 and H3 = 2 %. The government meteorology experts accepted these probabilities as reasonable judgments given the currently available information.

Developing this set of probabilities and probability distributions in Figs. 40.5 and 40.6 was essentially the author’s responsibility as the project leader of the

**Probabilistic Model for 12-Hour Change in Maximum Sustained Wind,
Seeded Hurricane**

Probability Distribution Based on:

- Probability Distribution for 12-Hour Change in Natural Hurricane
- Expert Judgment on Average Effect of Seeding
- Expert Judgment on Fluctuations From Average Effect in Seeding a Particular Storm

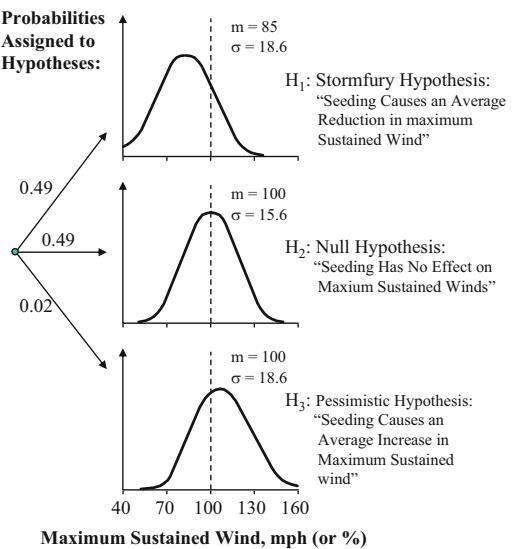


Fig. 40.5 Probabilistic model for 12-h change in maximum sustained wind, seeded hurricane

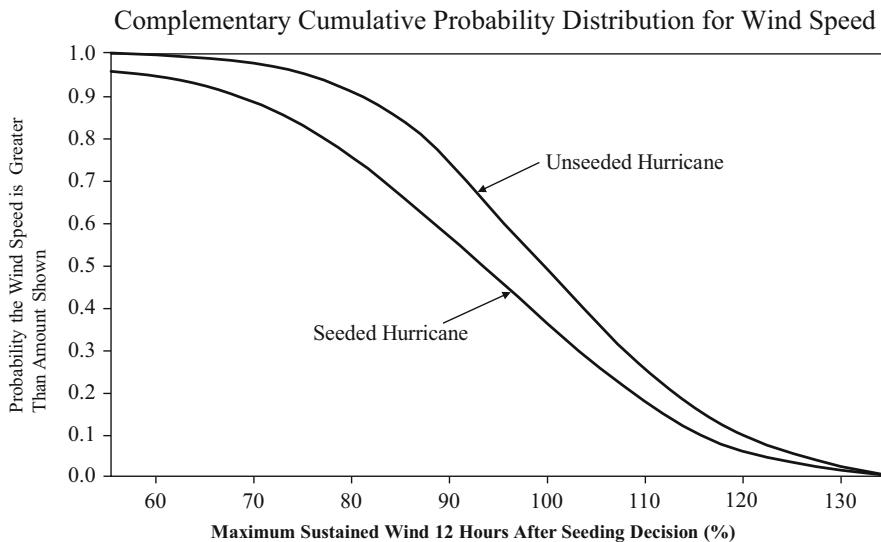


Fig. 40.6 Complementary cumulative probability distribution for wind speed

analysis team. These probabilities provided an imprecise numerical “sketch” designed to give insight on the strategic decision to allow seeding on a hurricane approaching the US coast. A better job of characterizing uncertainty might be done in the tactical situation when seeding became allowed and detailed data on a particular hurricane in a particular location might be available. The sketch was to be used to compare the decision alternatives of seeding and not seeding, with an evaluation on the potential impact on property damage, and of the trade-off between accepting a higher probability on large amounts of property damage and the acceptance of government responsibility, turning a natural disaster into situation modified by the US government action of cloud seeding before the hurricane came ashore. A more refined analysis of the probabilities was left for a later time. As team leader, the author was comfortable that the team had done all that it could do in combining available data and expert judgment within the time and resources of the project.

Early in the project the author learned the following historical anecdote from old hands among the meteorologists. In 1947 the US Navy seeded a hurricane out at sea moving parallel to the coast of Georgia. While the seeding had no apparent effect on its wind intensity, the hurricane made an abrupt left turn and did considerable damage to the city of Savannah. A subsequent investigation determined that the left turn occurred a few hours before the seeding was done. But the accusations that government seeding was responsible for the course change and the resulting property damage lingered in the memories of the scientists. They knew that Dr. Tribus, who had a background that included work on weather modification as well as probabilistic reasoning similar to E.T. Jaynes [56], was eager to go ahead seeding a hurricane threatening the US coast if seeding offered a prospect of reducing property damage. The meteorologists’ preference was for the equivalent of a massive clinical trial; seed about a hundred hurricanes far from shore and get much more experimental data, in addition to the 2 days of observation from seeding Hurricane Debbie. But carrying out such experimental seeding might take many decades, and hurricane damage to the USA was averaging \$400 million per year in 1970.

Wind speed changes needed to be translated into property damage changes. The quantitative methods used provided the equivalent of a sketch. The data available at the time of our project on wind speed versus property damage was assembled, with a correction on property damage for price inflation, and then plotted, as shown in Fig. 40.7. The relationship was steeply nonlinear. A log-log regression gave an exponent of about four and a half (4.363) for property damage as a function of maximum sustained surface wind speed. (For a sensitivity range 3–6 was used.)

With this relationship wind changes could be translated into property damage changes. There was no attempt to establish a baseline in dollars or specify a location such as New Orleans or Miami, but rather to calculate the percentage change in damage. The probability distribution on damage was scaled to 100 for the initial wind speed 12 h from landfall: A “nominal” hurricane had property damage of \$100 million if winds were unchanged, and damage changes estimated based on this exponential relationship of damage to wind speed. Estimates could be scaled

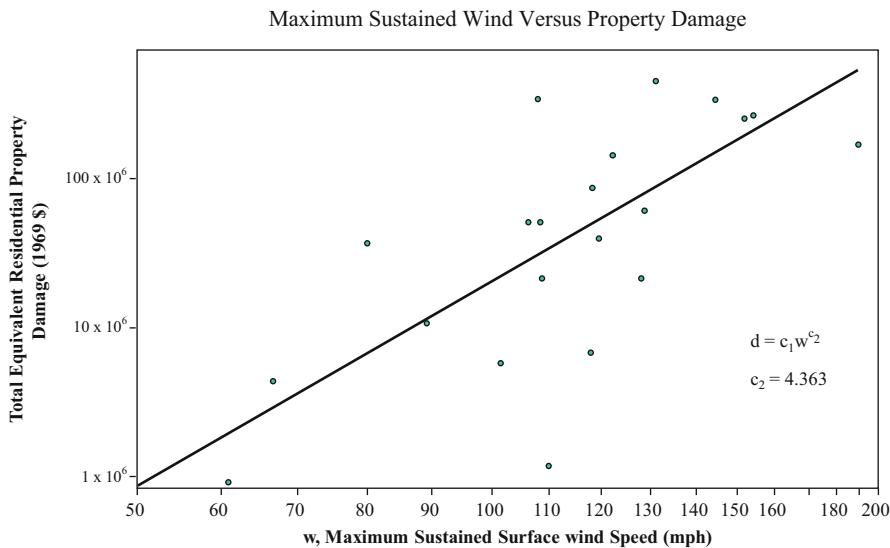


Fig. 40.7 Maximum sustained wind versus property damage

based on property values for different coastal areas. Such adjustment was not done in estimating the exponent from the data on historical hurricanes, but it could be done in a site-specific analysis.

The probability distributions on wind changes and the relationship of wind changes to property damage permitted computation of the expected damage with and without seeding, shown in Fig. 40.8. This calculation could have been done on a computer with great precision. At the time of this analysis, computers were large and took time for access, while handheld calculators were common, as cell phones with calculators have become today. The quantitative characterizations were not precise – they were the equivalent of sketches summarizing the available information. For convenience in explaining and ease for others to reproduce the calculations, the wind speed change outcomes were divided into five intervals: increase of 25 % or more, increase of 10 % to 25 %, little change, +10 % to –10 %, decrease of 10 % to 25 %, and a decrease of more than 25 %. The corresponding representative values for the five intervals are +32 %, +16 %, 0, –16 %, and –34 %. The decision on whether to seed is shown in Fig. 40.9.

Expected values for the wind speed change and property damage were calculated for each of the intervals and then the expected damage, multiplying probability times damage and summing over the five intervals, with and without seeding. This is readily done on a hand calculator. The expected property damage loss was reduced 18.7 % by seeding, a result that pleased Dr. Tribus and confirmed his intuition. The comparison of the alternatives is shown below in decision tree form in Fig. 40.9, with the decision shown as a box, and the resolution of uncertainty on damage from the first seeded hurricane shown with five discrete outcomes. A cost of \$250,000 for seeding was assumed, small compared to damage in the \$100 million range.

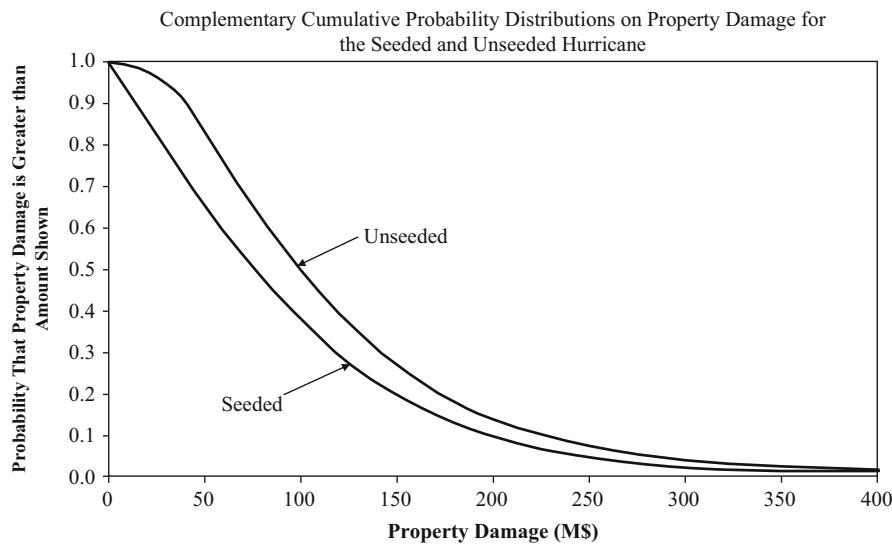


Fig. 40.8 Complementary cumulative probability distributions on property damage for the seeded and unseeded hurricane

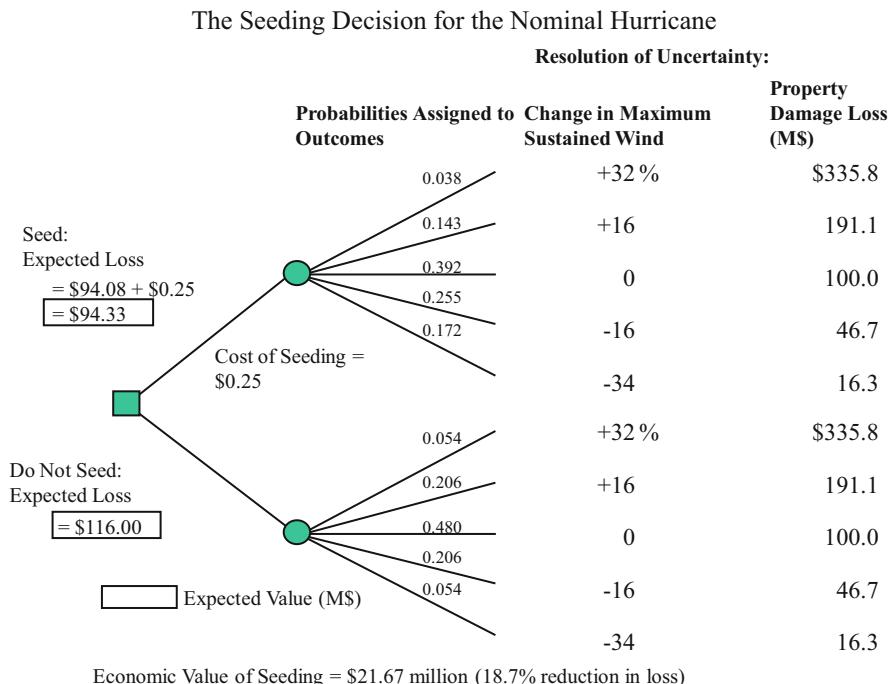


Fig. 40.9 The seeding decision for the nominal hurricane

Consider the government responsibility issue: What if a seeded hurricane intensifies after seeding – or does something else unexpected, such as a sudden change in direction? (The team did not calculate the latter. The meteorologists believed that seeding would not alter the path of the hurricane.) The team calculated that the probability that a seeded hurricane would intensify as 36 % and that it would intensify by at least 10 % as 18 %. (These numbers can be read from the complementary cumulative distribution on wind speed changes, Fig. 40.6.) Would there be public outrage or lawsuits brought against the federal government? The Port of New York Authority communicated informally that it might choose to sue if the US government did not seed a hurricane headed toward New York City.

The author, with responsibility as project leader, was unsure how to get numbers to define this trade-off. The opportunity came after a long day of meetings, having drinks in a hotel bar with a senior career civil servant serving as an advisor to Dr. Tribus. This advisor agreed to give his assessment of what he thought were reasonable trade-off values for the five outcomes. (The drinks and a pledge not to reveal his identity may have helped obtain his agreement.) The discussion went like this: Suppose you as a decision-maker for the federal government had to make a choice between two outcomes: a seeded hurricane that intensifies 16 % between seeding and landfall and an unseeded hurricane intensified even more causing x percent more property damage. Now if x is near zero, I expect you will choose the unseeded alternative. But if x is very large, you might choose to assume the political consequences for taking the responsibility in return for a lot less property damage inflicted on the local population. We want to find the level of x where the choices become equal, your point of indifference. The source gave judgments of $x = 30\%$ for $+16\%$ in wind speed, $x = 50\%$ for $+32\%$ in wind speed, $x = 5\%$ for no change, and $x = 0$ for wind speed reductions. Putting these values into the calculation of expected net value for each alternative makes the results very close, less than a 5 % difference in expected monetary value. The decision tree is shown in Fig. 40.10.

The analysis had now been expanded to include both the property damage minimization objective and the concern over assuming government responsibility for seeding. No one argued that the responsibility costs used in the analysis, obtained from a single source, were unreasonable and should be changed.

The project team decided to hire an experienced law school professor to investigate further. Gary Widman, then with Hastings College of the Law, University of California, wrote what became appendix material for the final report [4]. His findings were as follows:

With respect to Sovereign Immunity, the principle that citizens need a legal basis before they can bring suit against their sovereign government:

In conclusion, existing immunity law provides only partial and unpredictable protection at best. There are also grounds for recognizing that immunity defenses may be avoided in most cases if the plaintiff carefully chooses his remedy, his legal theory, and his forum. Only specific Congressional action offers a prospect of substantial, predictable immunity protection.

The Seeding Decision for the Nominal Hurricane (Government Responsibility Cost Included)

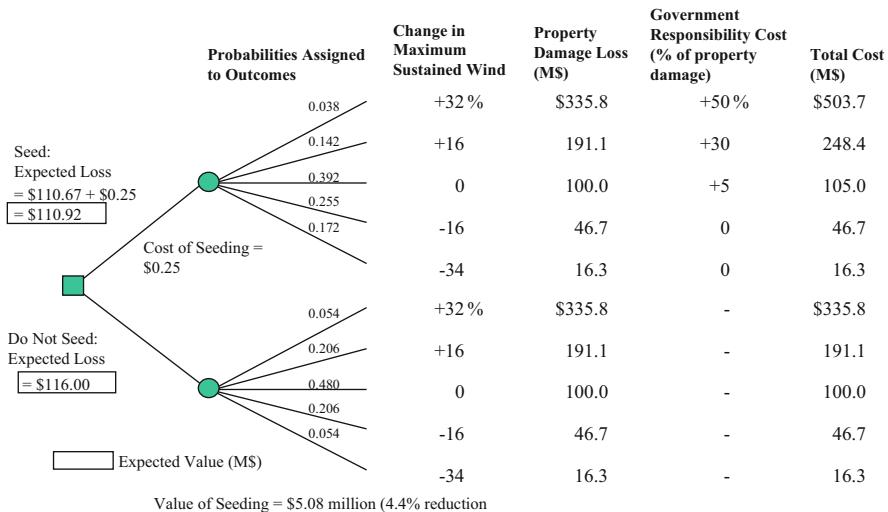


Fig. 40.10 The seeding decision for the nominal hurricane (government responsibility cost included)

And there appeared to be ample potential grounds for bringing a lawsuit:

In conclusion, existing immunity law provides only partial and unpredictable protection at best. There are also grounds for recognizing that immunity defenses may be avoided in most cases if the plaintiff carefully chooses his remedy, his legal theory, and his forum. Only specific Congressional action offers a prospect of substantial, predictable immunity protection.

This legal research persuaded Dr. Tribus and others in the leadership of the National Hurricane Research Laboratory and the National Weather Service that the decision to move to operational seeding would need to be made by the US President or under new legal authority provided by Congress.

There was a further issue to be addressed, and that was getting attention on the opportunities for further seeding experiments such as had been done on Hurricane Debbie. What would it be worth to resolve the uncertainty on which of the three hypotheses were true before a seeding decision needed to be made? Consider that an alternative to resolve this uncertainty by hiring a hypothetical clairvoyant who knows the answer. But the clairvoyant must be paid before we get the clairvoyant's information. What is the clairvoyant's information worth, in the context of the nominal \$100 million hurricane? The calculation of the expected value of information is relatively straightforward once a decision tree has been constructed with values on the outcomes. In the case of perfect information, complete resolution of the uncertainty on which hypothesis is correct, one needs only reverse the sequence, resolution of the uncertainty on which hypothesis before making the choice of whether to seed. The decision tree is shown in Fig. 40.11.

Expected Value of the Clairvoyant's Information: Which Hypothesis Describes the Effect of Seeding?

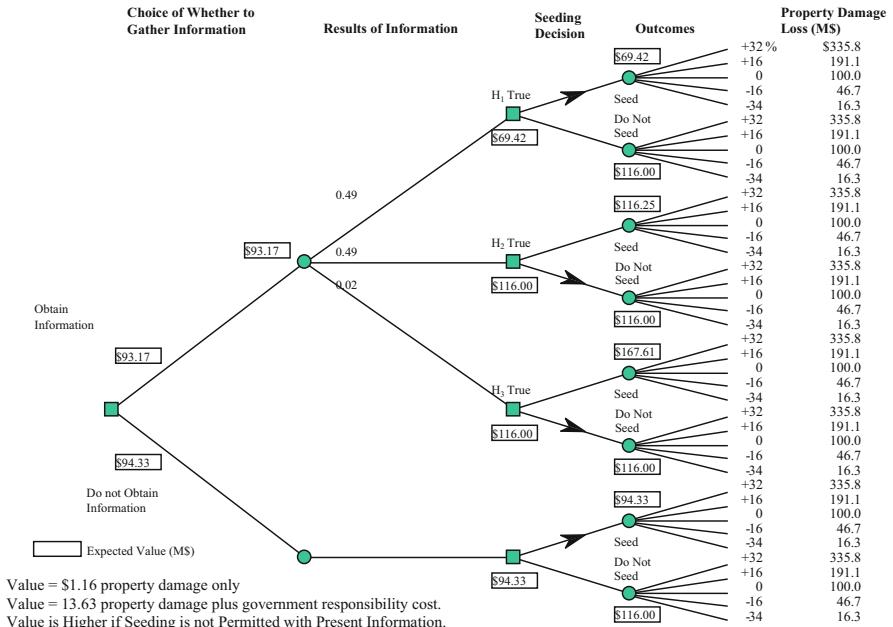


Fig. 40.11 Expected value of the clairvoyant's information: which hypothesis describes the effect of seeding?

A more complex calculation with a decision tree shows the decision to carry out an experimental seeding of the type that was done for Hurricane Debbie. Probabilities are calculated for the outcomes for a seeded hurricane, and Bayes' theorem is used to update the posterior probabilities of the three hypotheses. The overall result, shown in Fig. 40.12, is that for one \$100 million hurricane, the expected value of an experimental seeding is on the order of ten times the cost.

The project final report [4] includes calculated values for multiple seeding experiments and extrapolated to US hurricane seasons. Opportunities for seeding in the Pacific were more plentiful, and both the analysis team and the government sponsors believed that seeding experiments in the Pacific should be done. The analysis was presented to Dr Edward David, the Science Advisor to President Nixon, and the President's Science Advisory Committee (PSAC). President Nixon himself was briefed by Dr. David. The President's reaction is interesting, especially in view of later events in the Nixon Administration. President Nixon considered that if a severe hurricane threatened a US city such as New Orleans, he might wish to go onto television and ask for a plebiscite from affected citizens as to whether, as Commander in Chief, he should order a seeding.

Value of A Seeding Experiment (Government Responsibility Cost Included)

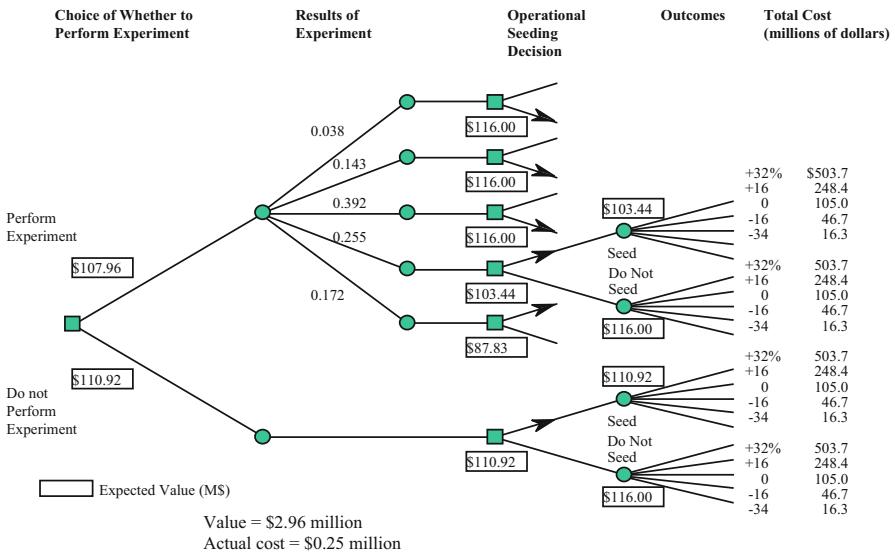


Fig. 40.12 Value of a seeding experiment (government responsibility cost included)

This analysis is believed to be the first use of value of information analysis in a public policy decision context at the Presidential level and the publication in *Science* [19], the first publication of this type on the application of decision analysis with value-of-information results.

In subsequent years, no further seeding experiments were carried out. As a consequence of weather modification activities in connection with the War in Vietnam, relations between the US Navy and the Department of Commerce on cloud seeding became troubled. Further research identified flaws in the Project Stormfury hypothesis: The measured abundance of supercooled water was insufficient to give the needed changes in hurricane structure, and the structural changes observed in Hurricane Debbie were observed to occur in unseeded hurricanes. Project Stormfury was officially cancelled in the early 1980s [70].

3 Conclusions

The reader is reminded again of the importance of process, as opposed to analytical product. In both case studies an analytical team had the task of learning about uncertainties in a highly complex area of emerging science. The teams learned what was needed to structure available knowledge into a quantitative analysis that was accepted by the government scientists and their managers and which subsequently underwent extensive review. There was not extensive public involvement in either case.

The author takes considerable pride in the way the Mars analysis team learned about the importance of UV radiation and particle shielding and the way the hurricane analysis team included government responsibility cost into the hurricane analysis as a way of bringing attention to the importance of the legal issues related to trying to change a weather phenomenon. Both teams dealt with the complexity and uncertainty under time and resource constraints and with difficulties to overcome in getting cooperation from government scientists. Some of the government scientists in the hurricane case study were initially hostile to what they thought was an effort to take seeding operational as part of coastal defense before the technology had been established as acceptably safe. The team learned their concerns and tried to address them openly. The hurricane seeding decision analysis was presented to the committee of leading scientists who advised the President, and the (at the time) innovative Bayesian probability and value of information calculations were well received by these scientists and published in *Science* [19]. The results analysis encouraged support for seeding in the Pacific, which seemed like a very good alternative at the time.

The author has learned over many applications of decision analysis in addition to these two case studies the importance of proper framing: What are the decisions, and what are the consequences that the stakeholders care about? [35, 36].

It is disappointing that there have not been more applications of decision analysis to major public policy problems. There is opportunity: The tools of probabilistic analysis are there, and efforts are ongoing to apply these tools to a range of problems, including the safety of nuclear power plants and the global decisions to deal with the threats posed by greenhouse-gas-induced climate change [40].

Many corporations are now using decision analysis routinely; Chevron received the first Raiffa-Howard Award (named for Howard Raiffa and Ronald Howard) for excellence in decision quality at the 2014 INFORMS meeting in San Francisco. Opportunities to carry out analysis in support decisions depend on the leadership with the decision responsibility. These leaders can ask for the support that decision analysis can provide.

Acknowledgements This chapter draws from presentations on the two case studies at the November 2014 meeting of INFORMS (Institute for Operations Research and the Management Sciences) in San Francisco, on the occasion of the 50th Anniversary of Decision Analysis honoring its founders, Stanford University Professor Ronald Howard and Harvard University Professor Howard Raiffa. The author wishes to acknowledge the collaboration with colleagues in the two case studies, and appreciation to Tony Cox for detailed comments on the initial draft of this chapter. The author expresses gratitude to these and other colleagues over 50 years in decision and risk analysis for the many insights learned about assessing uncertainties and values in a decision support context.

References

1. Arrow, K.J.: A difficulty in the concept of social welfare. *J. Political Econ.* **58**, 328–346 (1950)
2. Arrow, K.J.: Alternative approaches to the theory of choice in risk-taking situations. *Econometrica* **19**(4), 404–437 (1951)
3. Arrow, K.J.: Essays in the Theory of Risk-Bearing. North Holland Pub. Co., Amsterdam (1970)

4. Boyd, D.W., Howard, R.A., Matheson, J.E., North, D.W.: Decision Analysis of Hurricane Modification, Final Report, SRI Project 8503, Stanford Research Institute, Menlo Park (1971)
5. Budnitz, R.J., Apostolakis, G., Boore, D.M., Cluff, L.S., Coppersmith, K.J., Cornell, C.A., Morris, P.A.: Recommendations for Probabilistic Seismic Hazard Analysis: Guidance on Uncertainty and Use of Experts, Report prepared for the Nuclear Regulatory Commission, NUREG/CR-6372, vol. 1, main report, vol. 2, appendices. <http://www.nrc.gov/reading-rm/doc-collections/nuregs/contract/cr6372/> (1997)
6. Budnitz, R.J., Apostolakis, G., Boore, D.M., Cluff, L.S., Coppersmith, K.J., Cornell, C.A., Morris, P.A.: Use of technical expert panels: applications to probabilistic seismic hazard analysis. *Risk Anal.* **18**(4), 463–469 (1998)
7. Cooke, R.M.: Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press, Oxford (1991)
8. Cox, R.T.: Probability, Frequency, and Reasonable Expectation. *Am. J. Phys.* **14**, 1–10 (1946)
9. Cox, R.T.: The Algebra of Probable Inference. Johns Hopkins University Press, Baltimore (1961); See also http://en.wikipedia.org/wiki/Cox%27s_theorem; http://en.wikipedia.org/wiki/Richard_Threlkeld_Cox
10. De Finetti, B.: La Prévision: ses lois logiques, ses sources subjectives. *Annales de L'Institut Henri Poincaré* (1937). An English translation is in *Studies in Subjective Probability*, Kyburg and Smokler (eds.), 1964
11. Fischhoff, B.: The realities of risk-cost-benefit analysis. *Science* **350**(6260) (2015). doi:10.1126/science.aaa6516
12. Fishburn, P.C.: Utility Theory for Decision Making. Wiley, New York (1970)
13. Fishburn, P.C.: Personal communication (1970)
14. Haidt, J.: The Righteous Mind: Why Good People Are Divided by Politics and Religion. Vintage Books, New York (2013)
15. Howard, R.A.: Decision analysis: applied decision theory. In: Hertz, D.B., Melese, J. (eds.) *Proceedings of the Fourth International Conference on Operation Research*, pp. 55–71. Wiley-Interscience, New York (1966)
16. Howard, R.A.: The foundations of decision analysis. *IEEE Trans. Syst. Sci. Cybern.* **SSC-4**(3), 1–9 (1968)
17. Howard, R.A., Abbas, A.E.: Foundations of Decision Analysis. Pearson, New York (2015)
18. Howard, R.A., Matheson, J.E.: Influence diagrams. *Decis. Anal.* **2**(3), 127–143 (2005)
19. Howard, R.A., Matheson, J.E., North, D.W.: The decision to seed hurricanes. *Science* **176**, 1191–1202 (1972)
20. Jaynes, E.T.: Prior probabilities. *IEEE Trans. Syst. Sci. Cybern.* **4**(3), 227–241 (1968)
21. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge (2003). Earlier versions of this book were circulated as colloquium lectures, Socony Mobil Oil Company, 1958, then in a version available online, prior to Jaynes' death in 1998. The Cambridge University Press version was edited by G. Larry Bretthorst and published five years after Jaynes' death
22. Jeffreys, H.: Theory of Probability, 3rd edn. Clarendon Press, Oxford (1939, 1961)
23. Jet Propulsion Laboratory News: <http://mars.nasa.gov/mro/news/whatsnew/index.cfm?FuseAction>ShowNews&NewsID=1183> (2011)
24. Jet Propulsion Laboratory News: <http://www.jpl.nasa.gov/news/news.php?feature=4722> (2015)
25. Judd, B.R., Warner North, D., Pezier, J.P.: Assessment of the Probability of Contaminating Mars., Report No. MSU-2788, prepared for the Planetary Programs Division, National Aeronautics and Space Administration by SRI International, Menlo Park, 156 pp. (1974). Available at: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19740018186.pdf>
26. Kahneman, D.: Thinking, Fast and Slow. Farrar, Strauss, and Giroux, New York (2011)
27. Kahneman, D., Tversky, A., Slovic, P.: Judgment Under Uncertainty: Heuristics and Biases. Cambridge University Press, New York (1982)
28. Kok, J.F., Parteli, E.J.R., Michaels, T.I., Karam, D.B.: The physics of wind-blown sand and dust. *Rep. Prog. Phys.* **75**, 106901 (2012). Available at <http://arxiv.org/ftp/arxiv/papers/1201/1201.4353.pdf>

29. Kyburg, H.E., Jr., Smokler, H.E. (eds.): *Studies in Subjective Probability*. Wiley, New York (1964)
30. Laplace, P.-S., marquis de, *Essai philosophique sur les probabilités* (1814). English translation, Dover, New York (1951)
31. Loèvè, M.: *Probability Theory*, 4th edn. 1963, Springer, New York (1977)
32. Luce, R.D., Howard R.: *Games and Decisions: Introduction and Critical Survey*. Wiley, New York (1957) Reprinted by Dover (1989)
33. Morgan, M.G.: The use (and abuse) of expert elicitation in support of decision making for public policy. *Proc. Natl. Acad. Sci.* **111**, 7176–7184 (2014)
34. Morgan, M.G.: Commentary: our knowledge of the world is not simple: policy makers should deal with it. *Risk Anal.* **35**(1), 19–20 (2015)
35. National Research Council: *Understanding Risk: Informing Decisions in a Democratic Society*. National Academy Press, Washington, D.C. (1996)
36. National Research Council: *Public Participation in Environmental Assessment and Decision Making*. National Academy Press, Washington, D.C. (2008)
37. Navarro, D., Perfors, A.: An introduction to the Beta-Binomial Model, University of Adelaide. https://www.cs.cmu.edu/~10701/lecture/technote2_betabinomial.pdf (undated). Last accessed 2 Feb 2016
38. North, D.W.: The invariance approach to the probabilistic encoding of information. Ph.D. dissertation, Stanford University (1970). Available at: http://www.northworks.net/phd_thesis.pdf
39. North, D.W.: Limitations, definitions, principles, and methods of risk analysis. *Rev. sci. Tech. Off. Int. Epiz.* **14**(4), 913–923 (1995)
40. North, D.W.: Review of five books on climate change. *Risk Anal.* **35**(12), 2221–2227 (2015)
41. North, D.W., Judd, B.R., Pezier, J.P.: New methodology for assessing the probability of contaminating Mars. *Life Sci. Space Res.* **XIII**, 103–109. Academie-Verlag, Berlin (1975)
42. Phillips, L., von Winterfeldt, D.: Reflections on the Contributions of Ward Edwards to Decision Analysis and Behavioral Research, Working paper, London School of Economics. eprints.lse.ac.uk/22711/1/06086.pdf (2006). Chapter 5 in *Advances in Decision Analysis*. Cambridge University Press, Cambridge (2007)
43. Pratt, J.: Risk Aversion in the Small and In the Large. *Econometrica* **32**(1/2), 122–136 (1964)
44. Pratt, J., Raiffa, H., Schlaifer, R.: *Introduction to Statistical Decision Theory*. MIT, Cambridge (1995)
45. Raiffa, H.: *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, Reading (1968)
46. Raiffa, H., Schlaifer, R.: *Applied Statistical Decision Theory*. Cambridge, MA: Graduate School of Business, Harvard University, 1961. Reprinted, Wiley Classics Library, New York (2000)
47. Ramsey, F.P.: Truth and Probability (1926). Reprinted in: *Studies in Subjective Probability*, Kyburg and Smokler (eds.) 1964. Also, the text of this paper appears in the book by Ramsey, *The Foundations of Mathematics and Other Essays*. Harcourt Brace and Company, New York (1931) and is available at: <http://socscerv2.socsci.mcmaster.ca/~econ/ugcm/3li3/ramseyfp/ramsess.pdf>
48. Savage, L.J.: *The Foundations of Statistics*. Wiley, New York (1954). 2nd revised edition, Dover, New York (1972)
49. Space Science Board: National Research Council, *Biological Contamination of Mars*. National Academy Press, Washington, D.C. http://www.nap.edu/catalog.php?record_id=12305 (1992)
50. Spetzler, C., Stael von Holstein, C.-A.S.: Probability encoding in decision analysis. *Manag. Sci.* **22**, 340–358 (1975)
51. Stanford Encyclopedia of Philosophy, Modal Logic entry: <http://plato.stanford.edu/entries/logic-modal/>. Last accessed 2 Feb 2016
52. Taleb, N.N.: *Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets*, 2001, updated second edition. Random House, New York (2005)

53. Taleb, N.N.: *The Black Swan*. Random House, New York (2007)
54. Taleb, N.N.: *Antifragile: Things That Gain from Disorder*. Random House, New York (2012)
55. Tetlock, P.E., Gardner, D.: *Superforecasting: The Art and Science of Prediction*. Crown Publishers, New York (2015)
56. Tribus, M.: *Rational Descriptions, Decisions, and Designs*, Pergamon Press, Elmsford (1969)
57. Vaihinger, H.: *The Philosophy of 'As-If': A System of the Theoretical, Practical, and Religious Fictions of Mankind*. Published in German (1911); in England (1924). Available in a translation by C.K. Ogden, New York, Barnes and Noble. http://en.wikipedia.org/wiki/Hans_Vaihinger (1968). Last accessed 2 Mar 2015
58. Van Horn, K.S.: Constructing a logic of probable inference: a guide to Cox's theorem. *Int. J. Approx. Reason.* **34**(1), 3–24 (2003)
59. Von Neumann, J., Morgenstern, O.: *Theory of Games and Economic Behavior*, 2nd edn. 1947, paperback edition, 2007, Princeton University Press, Princeton
60. Von Winterfeldt, D., Edwards, W.: *Decision Analysis and Behavioral Research*. Cambridge University Press, New York (1986)
61. Wald, A.: *Statistical Decision Functions*. Wiley, New York (1950)
62. Wiki, Arrow's impossibility theorem: https://en.wikipedia.org/wiki/Arrow%27s_impossibility_theorem. Last accessed 30 Jan 2016
63. Wiki, Bayesian network: https://en.wikipedia.org/wiki/Bayesian_network. Last accessed 30 Jan 2016
64. Wiki, Cox's theorem: https://en.wikipedia.org/wiki/Cox%27s_theorem. Last accessed 30 Jan 2016
65. Wiki, De Finetti's theorem: https://en.wikipedia.org/wiki/De_Finetti%27s_theorem. Last accessed 7 Feb 2016
66. Wiki, Influence diagram: https://en.wikipedia.org/wiki/Influence_diagram. Last accessed on 30 Jan 2016
67. Wiki, Maxwell-Boltzmann distribution: https://en.wikipedia.org/wiki/Maxwell%E2%80%99s_Boltzmann_distribution. See also: <http://farside.ph.utexas.edu/teaching/sm1/lectures/node72.html>. Both last accessed 7 Feb 2016
68. Wiki, Monty Hall problem: https://en.wikipedia.org/wiki/Monty_Hall_problem. Last accessed 14 Nov 2015
69. Wiki, Probability interpretations: http://en.wikipedia.org/wiki/Probability_interpretations. Last accessed 17 Nov 2015
70. Wiki, Project Stormfury: https://en.wikipedia.org/wiki/Project_Stormfury. Last accessed 15 Nov 2015
71. Wiki, Voting paradox: https://en.wikipedia.org/wiki/Voting_paradox. Last accessed 30 Jan 2016
72. Winkler, R.: Equal versus differential weighting in combining forecasts. *Risk Anal.* **31**(1), 16–18 (2015)

Validation, Verification, and Uncertainty Quantification for Models with Intelligent Adversaries

41

Jing Zhang and Jun Zhuang

Abstract

Model verification and validation (V&V) are essential before a model can be implemented in practice. Integrating model V&V into the process of model development can help reduce the risk of errors, enhance the accuracy of the model, and strengthen the confidence of the decision-maker in model results. Besides V&V, uncertainty quantification (UQ) techniques are used to verify and validate computational models. Modeling intelligent adversaries is different from and more difficult than modeling non-intelligent agents. However, modeling intelligent adversaries is critical to infrastructure protection and national security. Model V&V and UQ for intelligent adversaries present a big challenge. This chapter first reviews the concepts of model V&V and UQ in the literature and then discusses model V&V and UQ for intelligent adversaries. Some V&V techniques for modeling intelligent adversaries are provided which could be beneficial to model developers and decision-makers facing with intelligent adversaries.

Keywords

Decision making • Intelligent adversaries • Model validation and verification • Validation techniques

Contents

1	Introduction	1402
2	Model Verification vs. Validation	1403
2.1	Terminology	1403
2.2	Relationships Between Verification and Validation	1405
3	Validation, Verification, and UQ in the Literature	1405
3.1	Validation, Verification, and UQ in Model Development Process	1405
3.2	Quantitative Model Validation Techniques	1406

J. Zhang (✉) • J. Zhuang

Department of Industrial and Systems Engineering, New York State University at Buffalo,
Buffalo, NY, USA

e-mail: jzhang42@buffalo.edu; jzhuang@buffalo.edu

4 Validation for Intelligent Adversary Models.....	1407
4.1 Difficulties in Validating Intelligent Adversary Models.....	1407
4.2 Verification and Validation Methods for Intelligent Adversary Models.....	1408
4.3 Validate Intelligent Adversary Models Using Proxy Models	1413
5 Conclusion.....	1415
References.....	1416

1 Introduction

Models have been extensively used in research when describing systems and predicting scenarios. Model verification and validation (V&V) can quantify confidence in the accuracy of model-based predictions under certain assumptions.

Verification refers to building the system right, while validation refers to building the right system [47]. Verification is conducted before validation. The verification process includes assessing code verification and calculation verification. Validation consists of conceptual model validity and operational validity. There are many approaches to validation, such as validation by assumption, validation by results, and validation by common sense. Validation techniques include animation, comparison to models, degenerate tests, event validity, extreme condition, face validity, fixed values, historical data validation, historical methods, internal validity, multistage validation, operational graphics, parameter variability, predictive validation, traces, and turning tests [58]. See the explanations of the techniques in Table 41.1.

Academia, industry, and government have been interested in model validation. Some terms related to the model, such as “reliability,” “credibility,” “confidence,” and “applicability,” have become common in academic and industrial studies, as well as government reports and implementations. A Standards Committee for the development of model V&V procedures for computational solid mechanics models has been formed by the American Society of Mechanical Engineers (ASME); a V&V model for all safety-related nuclear facility design, analyses, and operations has been supported by the Defense Nuclear Facilities Safety Board (DNFSB); and validation of complex models has been a key concern of the military simulation community for over three decades [39].

Considerable attention has been paid to model verification and validation. Numerous articles have appeared in the literature expressing different concerns of the validity of the models that have been proposed. The advances in the techniques of modeling and solution have impacted how people perceive model validation. For details about model V&V and about computational simulation models [57–59], see [44]. Validation methods, procedures for economic and financial models, urban and transportation models, government and criminology models, and medical and physiological models have also been studied [16].

The V&V approach quantifies degree of the accuracy and confidence inferred from the comparison of the prediction from the model with the results from reality

Table 41.1 Common model validation techniques (Source: [58])

Techniques	Explanation
Animation	Use graphs to show the model's behavior through time
Comparison to models	Compare model results to the results of other valid models
Degenerate test	Test model behavior using appropriate values of input/internal parameters
Event validity	Compare model event to real system to see the similarity
Extreme condition	Check model plausibility in extreme and unlikely levels of the system
Face validity	Ask knowledgeable people about the reasonability of the model
Fixed values	Fix values for variables/parameters to check against easily calculated values
Historical data validation	Use part of the data to build model, and the rest of data to test model
Historical methods	Three historical methods: rationalism, empiricism, positive economics
Internal validity	Implement several runs to determine the amount of variability in the model
Multistage validation	Combine the three historical methods into a multistage process
Operational graphics	Display values of various performance measures
Parameter variability	Use sensitivity analysis to determine the parameters' effect
Predictive validation	Check the prediction of the model with the system behavior
Traces	Trace entities in the model to see whether the model logic is correct
Turning tests	Ask people to discriminate the outputs of the model and system

or experiments. There can be no validation if there is no experimental data with which to compare the result of the model [6]. However, for intelligent adversaries, deficiencies of data and the incompleteness of understanding adversaries' behavior hinder the modeler from building the model and obtaining credible predictions. Taking the characteristics of the intelligent adversaries into consideration, is it possible to obtain sufficient data to build and validate such models? If not, is model validation even possible for intelligent adversary analysis in the absence of outcome data? These questions will be discussed in this chapter.

2 Model Verification vs. Validation

2.1 Terminology

We first introduce the terms of “verification” and “validation” before discussing the relationships between them. Model V&V methods and procedures have been defined by multiple organizations. In the development of fundamental concepts and

Table 41.2 Definitions of verification and validation by DMSO and IEEE (Source: [7, 24])

	Verification	Validation
DMSO	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications	The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model
IEEE	The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase	The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements

terminology for V&V, the Department of Defense (DoD) Modeling and Simulation Office (DMSO) has been the leader and played a major role in attempting to standardize the definitions of V&V [7, 8]. In addition, DMSO developed the US fundamental concepts and terminology for model V&V applied to high-level systems such as ballistic missile defense and battle management simulations [71].

There is a variety of formal definitions. The Defense Modeling and Simulation Organization (DMSO) of the Department of Defense (DoD) and the Institute of Electrical and Electronics Engineers (IEEE) give the most widely used definitions of the terms of verification and validation [44]; see Table 41.2.

Software engineering and software quality assurance use the IEEE definitions [44]. By contrast, computational simulations in science and engineering and operation research use the DMSO definitions. The DMSO definitions are widely adopted by [1, 32, 33, 55, 63, 71] as well as in this chapter. Thacker et al. [71] states “Software V&V is fundamentally different from model V&V. Software V&V is required when a computer program or code is the end product. Model V&V is required when a predictive model is the end product. A code is the computer implementation of algorithms developed to facilitate the formulation and approximate solution of a class of models.”

Different papers have defined model V&V differently according to specific contexts. For example, [59] defines model validation as “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model”; according to [15], model validation refers to activities “to establish how closely the model mirrors the perceived reality of the model use/developer team”; [36] defines model as “activities designed to determine the usefulness of a model; i.e., whether it is appropriate for its intended use(s); whether the benefits of improving model usefulness exceed the costs; whether the model contributes to making “better” decisions; and possibly how well the particular model performs compared to alternative models”; [46] defines verification and validation as “the process of determining the accuracy with which a computational model can produce results deliverable by the mathematical model on which it is based: code verification and solution verification” and “the process of determining the accuracy with which a model can predict observed physical events (or the important features of a physical reality)” respectively.

2.2 Relationships Between Verification and Validation

Model verification and validation (V&V) are the primary processes for quantifying and building credibility in numerical models and essential parts of the model development process if models are accepted and used to support decision-making. Verification and validation are often mentioned together in requirements to test models, yet they are fundamentally different.

Verifying a model means checking if the model produces the intended output as a function of the inputs, and whether it is mathematically correct. The focus here is on the model implementation and coding [18]. Verification is a critical activity, but is not the same as validation. Fundamentally, model validation is subjective and different perspectives on model validation could have very different meanings. The assertion “the model was judged valid” can mean almost anything, since the modelers choose “the validity tests, the criteria for passing those tests, what models outputs to validate, what setting to test in, what data to use, etc.”[37].

Verification is a matter of asking “Did I build the thing right?” and “Have the model and the simulation been built so that they fully satisfy the developer’s intent?”. By contrast, validation asks “Did I build the right thing?” and “Will the model be able to adequately support its intended use?” “Is its fidelity appropriate for that?” [49].

The purpose of model verification and validation is to assess and improve credibility, accuracy, and trustworthiness. Model verification and validation cannot certify a model to be accurate for all scenarios; but can provide evidence that a model is sufficiently accurate for its intended use [71].

3 Validation, Verification, and UQ in the Literature

3.1 Validation, Verification, and UQ in Model Development Process

U.S. GAO [76] introduces basic steps in the modeling process, which includes (1) describing problem, (2) isolating system, (3) adopting supporting theory, (4) formulating model, (5) analyzing data requirements, collecting data, (6) developing computer program, (7) debugging computer program, (8) developing alternative solutions, (9) evaluating model output/results, (10) presenting results/plans, (11) developing model maintenance procedures, and (12) transferring system to users. According to [76], steps (1)–(7) are covered by model verification, and steps (1)–(11) are covered by model validation. Yet, the absence of the necessary information often makes it hard to follow the steps to validate models. Gass [15] adopts this modeling process which aims at indicating how “the research community concerned policy models is attempting to develop and test procedures for improving the role of models as decision aids.”

Over the past three decades, new dimensions have been brought to the notion of model V&V by large-scale computer-based mathematical and simulation models

[30]. The “Sargent Circle” in simulation validation is one of the earliest and most influential among all the paradigms of the relationships among V&V activities [56].

In the Sargent Circle [56], conceptual model validation is defined as “determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is reasonable for the intended purpose of the model”; computerized model verification is defined as “assuring that the computer programming and implementation of the conceptual model is correct”; and operational validation is defined as “determining that the model’s output behavior has sufficient accuracy for the model’s intended purpose over the domain of the model’s intended applicability.” Most validation takes place in operational validation. In order to obtain a high degree of confidence in the model and its results, the model developers need to compare the input-output behaviors of the model and the system. There are three basic comparison approaches: (1) graphs of the model and system behavior data, (2) confidence intervals, and (3) hypothesis tests. For details of the methodologies, see [2, 4, 31, 33].

A detailed schematic of the model V&V in model development process is given by [71]. There are two branches in the procedure of model V&V: the first is to obtain relevant and high-quality experimental data via physical testing; and the second is to develop and exercise the model.

Because of inherent randomness, uncertainties cannot be ignored. Uncertainty quantification (UQ) exists in the processes of both performing experiments and developing models, which emphasizes the importance of UQ in improving confidence in both the experiment and model outcomes.

By comparison with experimental data, validation is able to quantify the confidence in the predictive capability of the model. For more definitions, uncertainties, and model explanations, see [1, 45, 52, 53, 70, 71].

3.2 Quantitative Model Validation Techniques

With the development of computing capacity, computational models become extensively used to solve practical problems in various disciplines and play an important role as predictive models for complex systems. Imprecise data and model assumptions could impact the quality of the model prediction. It is important to quantify the uncertainty in the model prediction [55]. Although qualitative validation methods such as graphical comparison between model prediction and experimental data are widely used, statistics-based quantitative methods are essential to systematically account for the uncertainty in both model prediction and experimental observation [43]. Hemez and Doebling [21] claims that the ability to quantify uncertainty is essential for the success of any model validation.

Many previous papers have studied the application of statistical hypothesis testing methods in the context of model validation [22, 51], as well as the validation metrics, which provide quantitative measures of agreement between a predictive model and physical/experimental observations [13, 33, 42]. Yet there remain some unclear issues in the practice of model validation. Ling and Mahadevan [32] studies

the quantitative model validation techniques from the perspectives of both hypothesis testing-based and non-hypothesis testing-based methods and gives a systematic procedure for quantitative model validation. The quantitative model validation techniques include classical hypothesis testing, Bayesian hypothesis testing, confidence intervals, reliability-based metric, and area metric-based method. For details and examples of these quantitative model validation techniques, see [13, 26, 32].

4 Validation for Intelligent Adversary Models

4.1 Difficulties in Validating Intelligent Adversary Models

Model validation, together with verification, is critical for models intended to be used in practice. They are required by many organizations such as the US Department of Defense that uses adversary models [18]. After the attacks on September 11, 2001, billions of dollars have been spent on homeland security. To better understand intelligent adversary behaviors and to better study the strategic interactions between defenders and adversaries (e.g., attackers and terrorists), numerous models have been developed. Unfortunately, because of the deficiency of empirical data, few (if any) such models have yet been validated, which limits the application of those models in practice.

In defense and homeland security, decisions are made about allocating resources to prevent attacks by adversaries and protect the public. Risks from such intelligent adversaries, like terrorists, must be assessed prior to guiding defensive resource allocation; otherwise the effectiveness of resource allocation would be unreliable. Intelligent adversary risk assessment aims to prevent adversary attacks or mitigate the effects of adversary attacks by allocating resource efficiently. The risk assessment has significant importance in protecting the public safety. When a model is adopted to predict the adversaries' behavior and to instruct resource allocations, decision-makers should be confident that the model adequately represents the real situation. Poor risk assessment could lead to ineffective resource allocations and vulnerable targets.

Intelligent adversary risk assessment models are increasingly being developed and studied. Those models need to be validated. Probabilistic risk assessment (PRA)/ event-tree-based methods [12, 74], decision-analytic methods [14, 50], game-theoretic methods [20, 28, 79], and statistical machine-learning methods [11, 35] have been proposed for modeling intelligent adversaries. However, these models are complex and may not be directly tested by comparing model predictions with the outcome of events in the real world, since there are too few comparable adversary attack data to support statistical inferences about the model validity. The Committee on Methodological Improvements to the Department of Homeland Security's Biological Agent Risk Analysis cautions that "there may be insufficient scientific knowledge to verify or validate these models" [41]. When modeling the intelligent adversary, the modeler needs to consider the strategy and the rationality of the adversaries; in addition, it is difficult to model the consequences

resulting from terrorist incidents, since it is hard to assess when, where, and how the terrorists would strike. Meanwhile, as technologies evolve, adaptive terrorists could mount new types of attacks. In the reality of counterterrorism, there are lots of uncertainties; and throughout the built models, there are assumptions and dubious parameters. Unless the models are validated, the model results may not be trustworthy.

It is possible to make empirical observations to compare with the prediction results from the corresponding models. For example, [75] proposes a prospect theory model of coaches's utility and estimates the models' parameters using the data from the 2009 NFL season; [19] studies parking choice models, which are first calibrated based on the collected data from video-recorded observations from a set of parking lots on the University at Buffalo north campus and then used to predict the drivers' behavior. Intelligent adversaries are more difficult to model, since they are adaptive, and may have unknown preferences, beliefs, and capabilities. Guikema [17] indicates that "the key difference between risk assessment for situations with intelligent adversaries and traditional risk assessment problems is that intelligent adversaries adapt. They adapt to observed, perceived, and imputed likely future actions by those defending the system they are attempting to damage. This adaptive behavior must be considered if risk assessment models are to provide accurate estimates of future risk from intelligent adversaries and appropriately support risk management decision making."

Also, [3] claims that adversary risk analysis has three special uncertainties: (1) aleatory uncertainty (randomness of outcomes), (2) epistemic uncertainty (strategic choices of an intelligent adversary), and (3) concept uncertainty (beliefs about how the problems are framed).

Validating models for the probable behaviors of intelligent adversaries may not mean comparing the model results with existing data or experimental data as is in validating traditional models. This is because the data, if any, is often incomplete and sometimes classified. In terms of validating counter-terrorism models, [65] states that "for terrorist acts, validation is only possible in a limited sense and may be more correctly characterized as ensuring the models are reasonable or credible, performing sanity checks, ensuring consistency with what is known about terrorist groups, and not being able to invalidate the model. Validity, in this case is viewed as a range, not a binary valid/invalid assessment."

4.2 Verification and Validation Methods for Intelligent Adversary Models

4.2.1 Basic Necessary Conditions for Intelligent Adversary Risk Analysis

Guikema [17] proposes four basic necessary conditions for intelligent adversary risk analysis:

1. Adversary models must be descriptively accurate representations of future adversary actions to the best of the then-current knowledge of the defender;

2. Adversary models must be computationally tractable to support risk management decisions in the particular situations being addressed;
3. Adversary models must explicitly address uncertainty and represent any uncertainty in the predicted adversary actions;
4. There must exist one or more defensible methods for gaining confidence in the models for practical use.

The methods for gaining confidence in [17] include “validation by common sense” and “limited case-based validation for historic situation when data is available.”

4.2.2 Conceptual Processes to Provide Increased Confidence in Intelligent Adversary Models

Streetman [65] presents three conceptual processes that may be used to provide increased confidence in intelligent adversary models, but may fall short of traditional validation:

1. Minimally required components, which include terrorist objectives, attack logistics, decision criteria, and adaptation. Models that fail to address these key factors are considered less credible or valid;
2. Use of analogy. To obtain the real data could be hard or even impossible. However, we may infer something about the likely future behaviors of intelligent adversaries through appropriate use of existing databases, and historical attacks may be helpful in validating current models. Validating through analogy rather than using direct data is a reasonable approach. “Adversaries that are influenced by bias or have philosophical or religious perspectives will apply those perspectives to all their planning.”
3. Use of uncertainty. In modeling intelligent adversaries, we should consider the uncertainties in the structure and/or parameters. Parameter values should be evaluated and measured properly.

4.2.3 Transparent Risk Assessment to Improve Confidence

“Risk assessment transparency improves confidence” is suggested by [41], where the bioterrorism risk assessment (BTRA) model [10] is reviewed. The current use of the word “transparency” is summarized by [48] as “letting the truth be available for others to see if they so choose, or perhaps think to look, or have the time, means, and skills to look” and involving “active disclosure.”

In establishing confidence and trust in the methods of outputs from risk assessment models, transparency is a major factor. Achieving transparency requires the assumptions, model’s mathematical and structural foundations, and the sources of data used in the analysis to be made explicit. In [41], it is emphasized that “the accuracy of quantitative bioterrorism risk assessment models and the confidence placed in them depend on the validity of the assumptions and the availability of sound data for each of the biological agents being analyzed.”

4.2.4 Importance of Sensitivity Analysis for Validation

NRC [41] suggests that “sensitivity analysis is important for validation.” Saltelli and Tarantola [54] defines sensitivity analysis as the determination of how “uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input.”

There exist a lot of uncertainties or even errors in the model variables and parameters. The decision- maker needs to know how the uncertainties would impact the outputs of the model and therefore the confidence in the model. Many researchers have used sensitivity analysis to test uncertainties and evaluate the validity of the proposed models [60, 61, 80]. Sensitivity analysis has become an important approach to the testing and validation of risk assessment models of complex systems [5].

Recent studies [60, 61, 81] use sensitivity analysis for risk assessment to show how the results of a certain strategy would change when the parameter values change. In the future, it would be important to use more sensitivity analysis in risk management. This would be helpful to see, for example, what countermeasure strategy would be adopted if the modeler knew more about the intelligent adversaries’ behavior.

4.2.5 Comparing Models to Obtain Validity

Many models have been developed to study intelligent adversaries, and there are many variants and examples of these models in the literature [27, 60, 61, 66, 79, 81]. If some model has been proved to be valid, comparing other models with the validated model may be an effective way to do model verification and validation.

In [38], a comparative analysis of probabilistic risk analysis (PRA) and intelligent adversary methods for counterterrorism risk management is conducted. Defender event tree and Bayesian network, attacker event tree and Bayesian network, defender decision tree, attacker decision tree, sequential games, intelligent adversary risk analysis, adversary risk analysis, and simulation games are reviewed. Merrick and Parnell [38] considers each application on the same two illustrative example decisions. With respect to risk assessment, [38] states “Defender event trees and decision trees that represent attacker decisions as probabilities estimate lower expected consequences than attacker event trees and decision trees for the highest expected consequence attack, that is, the one that the attacker would choose.” As for risk communication, it is concluded “Event trees, influence diagrams with just probability nodes, and Bayesian networks with only probability nodes are all equivalent as they are following the laws of probability even though they use different solution algorithms.” At last, in terms of risk management, [38] gets the same conclusion with [41] that “event trees are less useful for assessing the risk posted by intelligent, adaptive adversaries.”

4.2.6 Simulation Validation with Intelligent Adversary Models

Simulation is a powerful tool for the analysis of complex process and systems. A growing number of simulation systems have been created to analyze the threats that terrorist attacks pose for public safety [34].

A National Research Council report [40] urges the Department of Homeland Security (DHS) to better validate its terrorism risk models. Morral et al. [39] reports the RAND's approach to validating the Risk Management Analysis Tool, or RMAT, which is one of the Transportation Security Administration's (TSA) principal terrorism risk modeling tools developed by the TSA and Boeing Company. RMAT is one of the growing class of quantitative models which are complex and could not be validated by comparing model predictions to the outcome of events in the real world, since the reference statistical data is limited. According to [39], "RMAT simulates terrorist behavior and success in attacking vulnerabilities in the domestic commercial air transportation system, drawing on estimates of terrorist resources, capabilities, preferences, decision processes, intelligence collection, and operational planning" and "to estimate the terrorism risk-reduction benefits attributable to new and existing security programs, technologies, and procedures."

Complex simulation is used to test the validity of the RMAT model [39]. In validating the defender model in RMAT, four areas are addressed [39]:

1. Identifying and evaluating the validity of key assumptions implicit in the overall system design;
 2. Comparing the world representation in RMAT to external sources;
 3. Assessing the completeness of the attack scenarios considered in RMAT, including both weapon-target pairings and pathways by which attacks are carried out;
 4. Comparing the attack consequences modeled in RMAT to external sources."
- Regarding the data, diverse forms of evidence are used to validate the data, such as "logic, subject matter expert judgments, and literature searches.

4.2.7 Using Experimental Data to Validate Intelligent Adversary Models

Validation includes comparing the model output with the real data or experimental results. In traditional models, experiments usually mimic the real situation and obtain reliable data. However, for intelligent adversary models, it is typically impossible to find data from experiments in which the conditions correspond exactly to the scenario because of the uncertain and adaptive nature of intelligent adversaries. It may also risk people's lives and public property to do some of such experiments. However, data gained based on laboratory experiments could provide insights into the behaviors of both the defender and attacker during certain hazard and emergency situations, which could be used to validate models.

An experiment was conducted in [23] to assess the extent to which individual decisions are consistent with theoretical predictions of misaligned profiling. The experiments are motivated, in part, by the counterintuitive nature of equilibrium patterns of the randomized strategies. In particular, the theory produces a paradox of misaligned profiling: in equilibrium the high reliability categories are searched more intensively, even though they are used less intensively by the terrorist organization. Field experiments with professional security officials to test these model predictions would be expensive and controversial, if possible at all. The results would be classified. Instead, [23] relies on laboratory experiments, which provide the ability

to replicate and control the environment. The results of the experiment reveal behavioral patterns that are consistent with the predicted patterns. Holt et al. [23] provides theoretical analysis and experimental validation to guide policy makers to improve the effectiveness of targeted profiled screening and investigates the efficient profiling and counterterrorism policy.

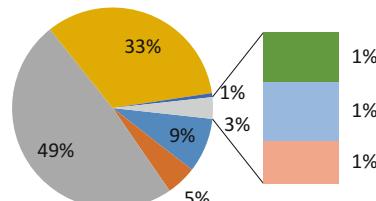
4.2.8 Using Historical Data to Conduct Validation

In addition to experimental data, historical data, which may have been recorded in databases and government reports, can provide other opportunities to validate intelligent adversary models.

In terms of terrorism, there are some databases available that recorded the terrorists' attacks, such as the Global Terrorism Database (GTD) [64], the International Terrorism: Attributes of Terrorist Events (ITERATE) [25], and Terrorism in western Europe: Events Data (TWEED) [69]. Among them, the GTD records incidents from 1970 and includes both domestic and international terror incidents. LaFree and Dugan [29] states that the GTD "have by far the largest number of events than any of the other data sets." Reports from the governments, such as the Federal Emergency Management Agency (FEMA) [72], the National Research Council (NRC) [73], and US Government Accountability Office (U.S. GAO) [77], could also provide useful data to do research on intelligent adversaries.

Zhang and Zhuang [78] presents a class of multi-period and multi-target attacker-defender games where the attackers have multiple attacking options. The attack types considered in [78] include assassination, armed assault, bombing/explosion, facility/infrastructure attack, hijacking, hostage taking, and unarmed assault, which is summarized based on the GTD categories. The percentage of the attack types used by the attackers is shown in Fig. 41.1. Different attack types would impact the attack success probabilities, consequences, as well as the effectiveness of defensive resource allocation. Sequential games are studied when the defender is faced with multiple attack types and adaptive attackers and how the defender would distribute a limited amount of resources to protect multiple urban areas. The objective of the defender is to minimize the total expected loss.

Fig. 41.1 Percentage of attack types in the USA
(Source: [64])



Based on the historical data from the GTD and UASI (the Urban Area Security Initiative, a Department of Homeland Security grant program), the parameter values are estimated in [78], such as the economic loss, fatality loss, success probability, and the defense cost-effectiveness for different attack types and targets. The authors estimate that basing defensive planning on the proposed model results in the lowest expected loss, having an expected loss which is 8–57% lower than the single attack-type model and 82–96% lower than the results of the real allocation [78].

4.3 Validate Intelligent Adversary Models Using Proxy Models

When modeling intelligent adversaries, it is important to construct a representation of preferences. However, sometimes it is difficult or even impossible to get direct elicitation from adversaries. Therefore, we could use indirect elicitation to construct and infer adversary motivations, objectives, preferences, capabilities, and beliefs. Proxy models are used by [27] to infer and validate models of adaptive adversaries. In [27], an adversary objective hierarchy and multi-attribute utility (MAU) models are constructed by proxy, using judgments from an adversary value expert (AVE). Past adversary behavior, public statements by the adversary, adversary web sites, and intelligence could be useful sources for the proxy to validate the behavior of the adversaries.

The proxy MAU models provide a relatively complete and accurate representation of the adversaries' values, including objectives, trade-offs, risk attitudes, and beliefs about consequence impacts. John and Rosoff [27] conducts two validation studies; good convergence between the proxy model and the model assessed by direct contact is demonstrated in both cases, which indicate that the proxy model may provide insights on intelligent adversaries if constructed and implemented properly.

4.3.1 Evaluating Effectiveness of Real-World Deployments to Validate Intelligent Adversary Models

Game theory has been playing an important role in modeling adversary behaviors, and Stackelberg games have been widely used to study terrorism and are in active use for resource deployment scheduling systems by law enforcements around the USA. In a Stackelberg game, there are two players, a leader (defender) and a follower (attacker); the leader chooses a strategy first and the follower subsequently decides his own strategy after observing the leader's strategy. According to [62, 66, 67], “the Stackelberg games models have been used to assist the LAX Airport police in scheduling airport entrance checkpoints and canine patrols of the terminals, the Federal Air Marshals Service (FAMS) to schedule marshals on international flights, the United States Coast Guard (USCG) in scheduling patrols around Boston Harbor, Ports of NY/NJ, and Ports of LA/LB, the Los Angeles Sheriff’s Department (LASD) for patrolling of the Metro trains, and (in-discussion) the patrolling of the Gulf of Mexico for illegal fishing for the USCG.” This system has been expanded to all ports in the USA due to the success of the patrolling schedules [67].

Despite the fact that Stackelberg game-based applications have been deployed in practice, measuring the effectiveness of the applications remains a difficult problem. And the data available about the deterrence of real-world terrorist attacks is very limited. Tambe and Shieh [67] suggests several methods to evaluate the effectiveness of Stackelberg games in real-world deployments, including:

1. Computer simulations of checkpoints and canine patrols;
2. Tests against human subjects, including USC students, an Israeli intelligence unit, and on the Internet Amazon Turk site (which provides some insights into adversary bounded rationality);
3. comparative analysis of predictability of schedules and methodologies before and after implementation of a Stackelberg strategy;
4. Red team/adversary team;
5. Capture rates of guns, drugs, outstanding arrest warrants, and fare evaders;
6. User testimonials.

4.3.2 Validation With Subject Matter Experts

According to [9], the Department of Defense's Modeling and Simulation Coordination Office defines a subject matter expert (SME) as "an individual who, by virtue of position, education, training, or experience, is expected to have a greater-than-normal expertise or insight relative to a particular technical or operational discipline, system or process." However, [68] shows that many SMEs make very poor predictions "at predicting elections, wars, economic collapses, and other events" and are not accountable enough for the accuracy of the forecasts, but the forecasting skills could be improved through learning and practicing.

Experts may offer help to do subjective (and possibly mistaken) model V&V. Experts may have a relatively high level of knowledge about what, when, where, and how the intelligent adversaries may behave. In general, experts may have better knowledge or more plausible-sounding guesses and narratives about the variables and parameters than others and know where the potential uncertainties may exist. Using experts to do model verification and validation is a qualitative technique, and the judgments made by an expert may be subjective and error prone. Also, each expert may have a different understanding of the model and could use different approaches to validate the same model. The benefit of using SME is that the decision-maker may gain different perspectives on the model and would have a comprehensive idea of the situation being modeled.

Louisa and Johnson [34] discusses the validation of a counterterrorism simulation of improvised explosive device (IED) incidents using the SME and concludes "it important to use the expertise of domain experts not only to compare the simulations to previous attacks of which they have knowledge, but also to use their knowledge to create new scenarios that explore the ways in which terrorist attacks could evolve."

Furthermore, the review process of the bioterrorism risk assessment (BTRA) model [10] is an example of using subject matter experts to verify and validate a model. Many good recommendations are given to make the model better [41], such as "The Department of Homeland Security should use an explicit risk analysis

lexicon for defining each technical term appearing in its reports and presentations,” and “To assess the probabilities of terrorist decisions, DHS should use elicitation techniques and decision-oriented models that explicitly recognize terrorists as intelligent adversaries who observe U.S. defensive preparations and seek to maximize the achievement of their own objectives.” DHS [10] concludes that the BTRA is not valid and suggests the DHS not to continue the development of that model.

5 Conclusion

Modeling plays an important role in guiding exploration in scientific research. This chapter has reviewed the concepts of model verification and validation (V&V), illustrated and compared model V&V in the developing process of models, and also discussed techniques for conducting a successful model V&V. Model V&V steps should be integrated with the modeling process and not be separated or treated after the model has been built. In the long run, using a validated model to support decision-making can sometimes improve decisions and make preferred outcomes more likely.

Because of inherent randomness of the systems, uncertainties in the model parameters, and the process of modeling framing, may impact the accuracy of results. Uncertainty quantification (UQ) should be considered in both the modeling process and the experiment process. Having a better understanding of the uncertainties can help the modeler build a more accurate model and thus make the model more effective in practice. This chapter has also illustrated some quantitative model validation techniques in dealing with uncertainty and deciding whether or not to accept the model prediction.

Intelligent adversary model V&V and UQ are different from V&V and UQ for traditional models in literature, where experimental/physical data could be obtained to compare with the model results. Unlike traditional models, the behavior of intelligent adversaries cannot be fully understood, and due to the lack of data and incomplete information about intelligent adversaries’ behavior, it is often the case that neither models nor experimental studies of adversarial behavior can be truly validated. This is a new and challenging area in the literature of model V&V. The model V&V techniques that have been discussed in this chapter include basic necessary conditions for intelligent adversary risk analysis, conceptual processes to provide more confidence in intelligent adversary models, risk assessment transparency, comparing models, simulation, experimental data, historical data, proxy models, evaluating effectiveness of real-world deployments, and using subject matter experts. These techniques attempt to take the uncertainties and adaptiveness of the intelligent adversaries into account, which may be helpful in tackling the dilemma of validating intelligent adversary models.

Many intelligent adversary models have appeared in the literature recently, but research on model V&V and UQ with intelligent adversaries is in many ways still in its infancy, with difficult challenges and limited options for overcoming them. More accurate adversarial models and more sophisticated V&V and UQ procedures with respect to adversarial models should be addressed to better understand the risks from intelligent adversaries and to better assist in surveillance and decision-making.

Acknowledgements This research was partially supported by the United States Department of Homeland Security (DHS) through the National Center for Risk and Economic Analysis of Terrorism Events (CREATE) under award number 2010-ST-061-RE0001. This research was also partially supported by the United States National Science Foundation under award numbers 1200899 and 1334930. However, any opinions, findings, and conclusions or recommendations in this document are those of the authors and do not necessarily reflect views of the DHS, CREATE, or NSF. The authors assume responsibility for any errors.

References

1. AIAA: AIAA guide for the verification and validation of computational fluid dynamics simulation. AIAA-G-077-1998, Reston (1998)
2. Balci, O., Sargent, R.G.: A Methodology for cost-risk analysis in the statistical validation of simulation models. *Commun. ACM.* **24**(4), 190–197 (1981)
3. Banks, D.: Adversarial Risk Analysis: Principles and Practice. Presentation on First Conference on Validating Models of Adversary Behaviors, Buffalo (2013)
4. Banks, J., Carson II J.S., Nelson, B.L.: Discrete-Event System Simulation, 2nd edn. Prentice Hall International, London, UK (1996)
5. Borgonovo, E.: Measuring uncertainty importance: investigation and comparison of alternative approaches. *Risk Anal.* **20**(5), 1349–1361 (2006)
6. Coleman, H.W., Steele, W.G.: Experimentation, Validation, and Uncertainty Analysis for Engineers. Wiley, Hoboken (2009)
7. DoD: DoD directive No 5000.59: Modeling and Simulation (M&S) Management. Defense Modeling and Simulation Office, Office of the Director of Defense Research and Engineering (1994)
8. DoD: Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide. Defense Modeling and Simulation Office, Office of the Director of Defense Research and Engineering (1996)
9. DoD: Special Topic on “Subject Matter Experts and Validation, Verification and Accreditation”, DoD Recommended Practices Guide (RPG) for Modeling and Simulation VV&A, Millennium Edition (2000)
10. DHS: Department of Homeland Security Bioterrorism Risk Assessment: A Call for Change. Available at <http://www.nap.edu/catalog/12206.html> (2006). Accessed in Nov 2015
11. Elovici, Y., Kandel, A., Last, M., Shapira, B., Zaafrany, O.: Using Data mining Techniques for Detecting Terror-Related Activities on the Web. Available at http://www.ise.bgu.ac.il/faculty/mlast/papers/JIW_Paper.pdf. Accessed in Nov 2015
12. Ezell, B.C., Bennett, S.P., Winterfeldt, D., Sokolowski, J., Collins, A.J.: Probabilistic risk analysis and terrorism risk. *Risk Anal.* **30**(4), 575–589 (2010)
13. Ferson, S., Oberkampf, W.: Validation of imprecise probability models. *Int. J. Reliab. Saf.* **3**(1), 3–22 (2009)
14. Garrick, B.J., Hall, J.E., Kilger, M., McDonald, J.C., O’Toole, T., Probst, P.S., Parker, E.R., Rosenthal, R., Trivelpiece, A.W., Arsdale, L.V., Zebroski, E.L.: Confronting the risks of terrorism: making the right decisions. *Reliab. Eng. Syst. Saf.* **86**(2), 129–176 (2004)
15. Gass, S.I.: Decision-aiding models: validation, assessment, and related issues for policy analysis. *Oper. Res.* **31**(4), 603–631(1983)
16. Gruhl, J., Gruhl, H.: Methods and Examples of Model Validation—an Annotated Bibliography. MIT Energy Laboratory Working Paper MIT-EL 78-022WP (1978)
17. Guikema, S.: Modeling intelligent adversaries for terrorism risk assessment: some necessary conditions for adversary models. *Risk Anal.* **32**(7), 1117–1121 (2012)
18. Guikema, S., Reilly, A.: Perspectives on Validation of Terrorism Risk Analysis Models. Presentation on First Conference on Validating Models of Adversary Behaviors, Buffalo (2013)

19. Guo, L., Huang, S., Zhuang, J.: Modeling parking behavior under uncertainty: a static game theoretic versus a sequential neo-additive capacity modeling approach. *Netw. Spat. Econ.* **13**(3), 327–350(2013)
20. Hausken, K., Zhuang, J.: The impact of disaster on the interaction between company and government. *Eur. J. Oper. Res.* **225**(2), 363–376(2013)
21. Hemez, F.M., Doebling, S.W.: Model validation and uncertainty quantification. For publication in the proceeding of IMAC-XIX, the 19th International Model Analysis Conference, Kissimmee, 5–8 Feb 2001
22. Hills, R.G., Leslie, I.H.: Statistical validation of engineering and scientific models: validation experiments to application. Sandia Technical Report (SAND2003-0706) (2003)
23. Holt, C.A., Kydd, A., Razzolini, L., Sheremeta, R.: The Paradox of Misaligned Profiling: Theory and Experimental Evidence. Available at <http://www.people.vcu.edu/~lrazzolini/Profiling.pdf> (2014). Accessed in Nov 2015
24. IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990, New York (1991)
25. International Terrorism: Attributes of Terrorist Events (ITERATE). Available at <http://library.duke.edu/data/collections/iterate>. Accessed in Nov 2015
26. Jiang, X., Mahadevan, S.: Bayesian risk-based decision method for model validation under uncertainty. *Reliab. Eng. Syst. Saf.* **92**(6), 707–718 (2007)
27. John, R., Rosoff, H.: Validation of Proxy Random Utility Models for Adaptive Adversaries. Available at http://psam12.org/proceedings/paper/paper_437_1.pdf (2014). Accessed in November, 2015
28. Jose, V.R.R., Zhuang, J.: Technology Adoption, Accumulation, and Competition in Multi-period Attacker-Defender Games. *Mil. Oper. Res.* **18**(2), 33–47 (2013)
29. LaFree, G., Dugan, L.L.: Introducing the global terrorism database. *Terror. Political Violence* **19**(2), 181–204 (2007)
30. Landry, M., Malouin, J.L., Oral, M.: Model validation in operations research. *Eur. J. Oper. Res.* **14**(3), 207–220 (1983)
31. Law, A.M., Kelton, W.D.: *Simulation Modeling and Analysis*, 2nd edn. McGraw-Hill, New York (1991)
32. Ling, Y., Mahadevan, S.: Quantitative model validation techniques: new insights. *Reliab. Eng. Syst. Saf.* **111**, 217–231 (2013)
33. Liu, Y., Chen, W., Arendt, P., Huang, H.: Toward a better understanding of model validation metrics. *J. Mech. Des.* **133**(7), 1–13(2011)
34. Louisa, N., Johnson, C.W.: Validation of Counter-terrorism Simulation Models. Available at http://www.dcs.gla.ac.uk/~louisa/Publications_files/ISSC09_Paper_2.pdf (2009). Accessed in Nov 2015
35. Mason, R., McInnis, B., Dalal, S.: Machine Learning for the Automatic Identification of Terrorist Incidents in Worldwide News Media. In: 2012 IEEE International Conference on Intelligence and Security Informatics (ISI), Washington, DC, pp. 84–89 (2012)
36. McCarl, B.A.: Model validation: an overview with some emphasis on risk models. *Rev. Market. Agric. Econ.* **52**(3), 153–173 (1984)
37. McCarl, B.A., Spreen, T.H.: Validation of Programming Models. Available at <http://agecon2.tamu.edu/people/faculty/mccarl-bruce/mccspr/new18.pdf> (1997). Accessed in Nov 2015
38. Merrick, J., Parnell, G.S.: A comparative analysis of PRA and intelligent adversary methods for counterterrorism risk management. *Risk Anal.* **31**(9), 1488–1510 (2011)
39. Morral, A.R., Price, C.C., Ortiz, D.S., Wilson, B., LaTourrette, T., Mobley, B.W., McKay, S., Willis, H.H.: Modeling Terrorism Risk to the Air Transportation System: An Independent Assessment of TSA's Risk Management Analysis Tool and Associated Methods. RAND report. Available at http://www.rand.org/content/dam/rand/pubs/monographs/2012/RAND_MG12_41.pdf (2012). Accessed in Nov 2015
40. NRC: Review of the Department of Homeland Security's Approach to Risk Analysis. Available at https://www.fema.gov/pdf/government/grant/2011/fy11_hsgp_risk.pdf (2010). Accessed in Nov 2015

41. NRC: Bioterrorism Risk Assessment. Biological Threat Characterization Center of the National Biodefense Analysis and Countermeasures Center. Fort Detrick, MD (2008)
42. Oberkampf, W., Barone, M.: Measures of agreement between computation and experiment: validation metrics. *J. Comput. Phys.* **217**(1), 5–36 (2006)
43. Oberkampf, W., Trucano, T.: Verification and validation in computational fluid dynamics. *Progr. Aerosp. Sci.* **38**(3), 209–272 (2002)
44. Oberkampf, W.L.: Bibliography for Verification and Validation in Computational Simulation. Sandia Report (1998)
45. Oberkampf, W.L., Trucano, T.G., Hirsch, C.: Verification, validation, and predictive capability in computational engineering and physics. *Appl. Mech. Rev.* **57**(5), 345–384 (2004)
46. Oden, J.T.: A Brief View of Verification, Validation, and Uncertainty Quantification. Available at <http://users.ices.utexas.edu/~serge/WebMMM/Talks/Oden-VVUQ-032610.pdf> (2009). Accessed in Nov 2015
47. O'Keefe, R.M., O'Leary, D.E.: Expert system verification and validation: a survey and tutorial. *Artif. Intell. Rev.* **7**, 3–42 (1993)
48. Oliver, R.W.: What Is Transparency? McGraw-Hill, New York (2004)
49. Pace, D.K.: Modeling and simulation verification and validation challenges. Johns Hopkins APL Technical Digest. **25**(2), 163–172 (2004)
50. Rakesh, K., Sarin, L., Keller, R.: From the editors: probability approximations, anti-terrorism strategy, and bull's-eye display for performance feedback. *Decis. Anal.* **10**(1), 1–5(2013)
51. Rebba, R., Mahadevan, S.: Validation of models with multivariate output. *Reliab. Eng. Syst. Saf.* **91**(8), 861–871 (2006)
52. Roach, P.J.: Verification and Validation in Computational Science and Engineering. Hermosa Publishers, Albuquerque (1998)
53. Salari, K., Knupp, P.: Code Verification by the Method of Manufactured Solutions. Sandia National Laboratories, SAND2000-1444 (2000)
54. Saltelli, A., Tarantola, S.: On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. *J. Am. Stat. Assoc.* **97**(459), 702–709 (2002)
55. Sankararaman, S., Mahadevan, S.: Model validation under epistemic uncertainty. *Reliab. Eng. Syst. Saf.* **96**(9), 1232–1241(2011)
56. Sargent, R.G.: An assessment procedure and a set of criteria for use in the evaluation of computerized models and computer-based modeling tools. Final technical report RADC-TR-80-409, U.S. Air Force (1981)
57. Sargent, R.G.: Some subjective validation methods using graphical displays of data. In: Proceedings of the 1996 Winter Simulation Conference, Coronado, California (1996)
58. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2009 Winter Simulation Conference, Austin, Texas, pp. 162–176 (2009)
59. Schlesinger, S., Crosbie, R.E., Innis, G.S., Lalwani, C.S., Loch, J., Sylvester, R.J., Wright, R.D., Kheir, N., Bartos, D.: Terminology for model credibility. *Simulation* **32**(3), 103–104 (1979)
60. Shan, X., Zhuang, J.: Cost of equity in homeland security resource allocation in the face of a strategic attacker. *Risk Anal.* **33**(6), 1083–1099 (2013)
61. Shan, X., Zhuang, J.: Hybrid defensive resource allocations in the face of partially strategic attackers in a sequential defender-attacker game. *Eur. J. Oper. Res.* **228**(1), 262–272 (2013)
62. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B. , Meyer, G.: PROTECT: a deployed game theoretic system to protect the ports of the United States. In: AAMAS, Valencia, Spain (2012)
63. Sornette, D., Davis, A.B., Vixie, K.R., Pisarenko, V., Kamm, J.R.: Algorithm for model validation: theory and applications. *Proc. Natl. Acad. Sci. U. S. A.* **104**(16), 6562–6567 (2007)
64. START: Global Terrorism Database[data file]. Available at <http://www.start.umd.edu/gtd>. Accessed in Nov 2015
65. Streetman, S.: The Art of the Possible in Validating Models of Adversary Behavior for Extreme Terrorist Acts. Presentation on First Conference on Validating Models of Adversary Behaviors, Buffalo (2013)

66. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press, New York (2011)
67. Tambe, M., Shieh, E.: Stackelberg Games in Security Domains: Evaluating Effectiveness of Real-World Deployments. Presentation on First Conference on Validating Models of Adversary Behaviors, Buffalo (2013)
68. Tetlock, P.E., Gardner, D.: Superforecasting: The Art and Science of Prediction. Crown, New York (2015)
69. Terrorism in Western Europe: Events Data (TWEED). Available at <http://folk.uib.no/sspje/tweed.htm>. Accessed in Nov 2015
70. Thacker, B.H., Riha, D.S., Millwater, H.R., Enright, M.P.: Errors and uncertainties in probabilistic engineering analysis. In: Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Seattle, Washington (2001)
71. Thacker, B.H., Doebling, S.W., Hemez, f. M., Anderson, M.C., Pepin, J.E., Rodriguez, E.A.: Concepts of Model Verification and validation. Available at http://www.ltas-vis.ulg.ac.be/cmsms/uploads/File/LosAlamos_VerificationValidation.pdf (2004). Accessed in Nov 2015
72. The Federal Emergency Management Agency (FEMA). Available at <http://www.fema.gov/>. Accessed in Nov 2015
73. The National Research Council (NRC). Available at <http://www.nationalacademies.org/nrc/>. Accessed in Nov 2015
74. Toubaline, S., Borrión, H., Sage, L.T.: Dynamic generation of event trees for risk modeling of terrorist attacks. In: 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, pp. 111–116 (2012)
75. Urschel, J., J. Zhuang.: Are NFL coaches risk and loss averse? Evidence from their use of kickoff strategies. *J. Quant. Anal. Sports* **7**(3), Article 14(2011)
76. U.S. GAO: Guidelines for Model Evaluation. PAD-79-17, Washington, DC (1979)
77. U.S. Government Accountability Office (U.S. GAO). Available at <http://www.gao.gov/>. Accessed in Nov 2015
78. Zhang, J., Zhuang, J.: Modeling a Multi-period, Multi-target Attacker-defender Game with Multiple attack types. Working paper (2015)
79. Zhang, J., Zhuang, J.: Defending Remote Border Security with Sensors and UAVs based on Network Interdiction Methods. Working paper (2015)
80. Zhuang, J., Bier, V.: Balancing terrorism and natural disasters-defensive strategy with endogenous attacker effort. *Oper. Res.* **55**(5), 976–991(2007)
81. Zhuang, J., Saxton, G., Wu, H.: Publicity vs. impact in nonprofit disclosures and donor preferences: a sequential game with one nonprofit organization and N donors. *Ann. Oper. Res.* **221**(1), 469–491(2014)

Robust Design and Uncertainty Quantification for Managing Risks in Engineering

42

Ron Bates

Abstract

Methods for uncertainty quantification lie at the heart of the robust design process. Good robust design practice seeks to understand how a product or process behaves under uncertain conditions to design out unwanted effects such as inconsistent or below-par performance or reduced life (and hence increased service or total life cycle costs). Understanding these effects can be very costly, requiring a deep understanding of the system(s) under investigation and of the uncertainties to be guarded against.

This chapter explores applications of UQ methods in an engineering design environment, including discussions on risk and decision, systems engineering, and validation and verification. The need for a well-aligned hierarchy of high-quality models is also discussed. These topics are brought together in an Uncertainty Management Framework to provide an overall context for embedding UQ methods in the engineering design process. Lastly, some significant challenges to the approach are highlighted.

Keywords

Systems Engineering • Verification & Validation • Risk • Decision Making • Simulation • Uncertainty Propagation • Robust optimization

Contents

1	Introduction	1422
2	Risk and Decision	1423
3	Systems Engineering and Verification & Validation	1425
3.1	Simulation Verification & Validation	1426

R. Bates (✉)

Design Sciences, Engineering Capability, Rolls-Royce plc., Derby, UK
e-mail: Ron.Bates@Rolls-Royce.com

4	Uncertainty Management Framework	1428
4.1	Quantification of Input Uncertainty	1429
4.2	Modeling and Simulation Framework	1430
4.3	Representation of Uncertainty in Simulation Models	1431
4.4	Propagation of Uncertainty	1431
4.5	Decision Analysis	1431
5	An Example Application: Jet Engine Disc and Blade Design	1432
6	Challenges For the Application of UQ	1434
7	Conclusion	1435
	References	1435

1 Introduction

Applying UQ methods in design is a core robust design activity and can be difficult. Here we discuss the issues around the use of UQ in the design process and the challenges faced by the practitioner. Engineering design is the process of applying science to create a system or process that converts resources to meet desired needs. Typical engineering applications are inherently complex, with many dynamic, nonlinear interactions and multiple objectives and constraints derived from requirements that express those needs. When considering the design and analysis of engineering systems, one needs to consider attributes that are multifaceted. Studies often contain multiple:

- Objectives and constraints,
- Behaviors and failure modes,
- Physical disciplines,
- Scales or levels of fidelity.

Product design involves teams of specialists developing subsystems in parallel with an inherent need to communicate ideas, models, and results between them. Too much focus on individual aspects of a design (single discipline, single objective) will ultimately lead to overall suboptimal performance.

Addressing these issues requires a holistic, systems engineering view. Experience has shown that without a structured approach to product design and development, the inherent complexity of engineering systems can result in undesired emergent behavior, with adverse consequences from significant project cost overrun to much worse, the classic examples being the collapse of the Tacoma Narrows Bridge or the NASA Challenger disaster. Conversely, adopting a systems approach can simplify complex situations, identify key issues, and reduce risk and cost. Examples of this can be seen in the adoption of “lean” methods in manufacturing, construction, and other sectors, where a focus on alignment of requirements and objectives can help drive out waste. A systems approach can also foster innovation by encouraging both revolution and incremental improvement, or the “aggregation of marginal gains”, which is particularly valuable in mature fields where many of the major advances in design and technology have already been made. Importantly for robust design,

this approach provides a natural framework for dealing with uncertainty in the system itself, in the network of models that represent the system, in model form, and in model inputs and in model outputs, which is the main focus of uncertainty quantification methods in engineering.

The rest of this section will describe in more detail the application of UQ in the engineering design environment, followed by a set of challenges that need to be overcome for application to complex engineering systems. The topics to be discussed are:

- Risk and decision
 - Systems engineering and validation and verification
 - Uncertainty management framework
 - Challenges
-

2 Risk and Decision

The relationship between quality, risk, and uncertainty is worth exploring. Robust design has its roots in the quality revolution in postwar Japan (Ishikawa/Deming) where the connection was made between statistical methods applied to industry and management, leading to the famous “14 key principles for management” later embraced by the total quality management revolution. The point here is that we undertake UQ studies to improve our understanding of the effect of uncertainty on the system (or subsystem, or component) in order to make a decision as part of the product development process (PDP). It is all about those decisions. There is a strong hierarchical link between probabilistic analysis, UQ, uncertainty management, and risk (Fig. 42.1).

Taken in isolation, UQ studies on complex engineering systems are costly and time-consuming to conduct. High cost can make it difficult to justify their use, particularly in low-volume industries. Long-running studies may take weeks or even months to complete, which may be too late to influence the design. However, the design process itself is not a linear process, and in general, neither is innovation. Instead ideas are generated, tested, and broken in an iterative process, and design studies need to keep pace with this. Mapping this process to hierarchical models of the system (integrated, multi-scale, multi-fidelity) may facilitate detailed UQ studies early in the PDP, reducing the risk of costly design modifications later. This will be discussed further in later sections.

When considering uncertainty, there are three main categories:

1. Randomness: aleatory uncertainty, or intrinsic randomness,
2. Incompleteness: epistemic uncertainty, or lack of knowledge,
3. Ineptitude, incomprehension, or incompetence.

The field of UQ is most often concerned with (1) and (2). Probabilistic studies can provide a natural framework for dealing with aleatory uncertainty. Probability



Fig. 42.1 From Risk to probabilistic analysis

distributions can be used to model uncertainty as variation, and the impact of variation can be assessed using models of systems. The issue then becomes one of distinguishing between error in model inputs and error in the models themselves (model discrepancy with the physical system).

Where there are deeper uncertainties, it may be difficult to frame that uncertainty in a probabilistic sense. There may be a lack of data, there may not be a sufficiently credible model of the system, or there may not be sufficient knowledge of cause and effect to produce any kind of model. In these cases, knowledge management and systems engineering can help blend qualitative and quantitative information to express these uncertainties in a way that aids decision-making (see Fig. 42.2).

Where there is complexity, or even chaos, it is likely that there is no credible causal model of the system. There may be deep uncertainty about the system and its inputs, with no model, and no well-defined expressions of uncertainty. One way of trying to address this type of problem is to develop parallel plausible scenarios that illustrate these uncertainties and then to analyze their effects. This can be seen, for example, in weather forecasting where several alternative (but equally likely) storm paths may be shown to communicate the uncertainty in the forecast (see, e.g., [5]).

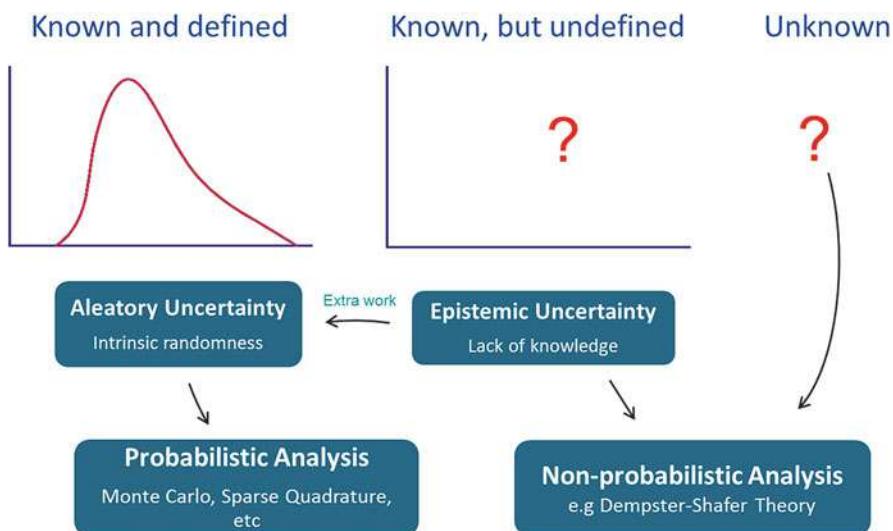


Fig. 42.2 Levels of uncertainty

The Cynefin model [1] provides a framework for integrating and expanding these levels of uncertainty to encompass the wider system view and related decision-making processes and appropriate modes of behavior for dealing with uncertainty. Figure 42.3 shows an attempt at mapping the levels of uncertainty defined in Fig. 42.2 into this framework. Where there is deep uncertainty, one is firmly on the left-hand side of the Cynefin model.

In addition to facilitating UQ studies by helping with the quantification problem, systems engineering can also help to address the issue of ineptitude or incompetence, for example, by providing a framework, or checklist, to facilitate the implementation of best practice or simply by having a structured approach that minimizes data transfer, data processing, model sharing, and model integration. The issue of incomprehension is also worth considering at this point. A proposed solution to a problem may not be adequate if the initial specifications are misunderstood or if the specified system is not correctly modeled. A systems engineering approach can help guard against these problems by checking for inconsistencies and unexpected behavior, which will be explored in the next section. Ultimately, managing uncertainty helps to speed up implementation, reduce error, and reduce cost.

3 Systems Engineering and Verification & Validation

Systems engineering is an approach used to develop complex engineering products. It emphasizes a top-down approach, embracing the whole product life cycle, and is naturally multidisciplinary. Importantly it also places a heavy emphasis on defining requirements (both customer and system) before developing solutions. Why is this

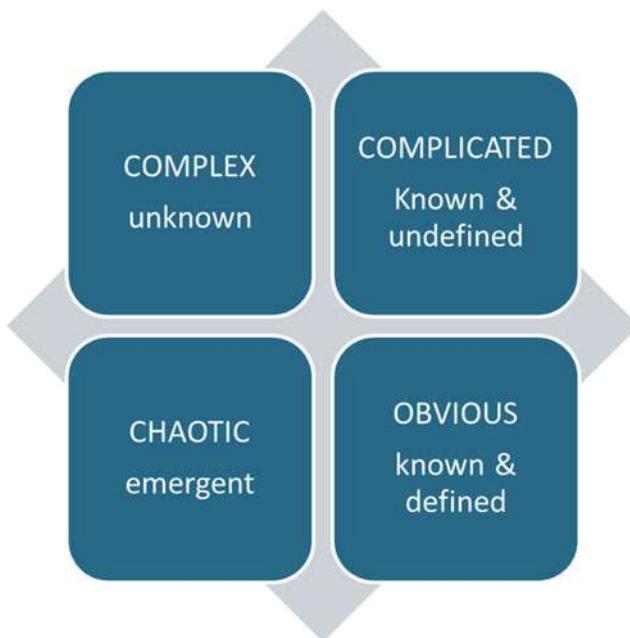


Fig. 42.3 Mapping uncertainty and systems thinking to the Cynefin model

important? Because as systems become more complex, it becomes more difficult to design solutions without encountering undesirable emergent behavior.

Systems engineering promotes a hierarchical approach, embodied in the “V-diagram” shown in Fig. 42.4.

The basic “V” can be applied hierarchically and can operate at many scales in a fractal sense. A typical generic decomposition would be system, subsystem, component, and material. The overall systems engineering framework provides a hierarchical structure to frame UQ studies in the context of risk and decision, as discussed in the previous section. The final step in the system V-diagram is verification, where the system is tested against the original requirements to confirm if it fulfills those requirements: “did we build the thing right?” This is in contrast to system validation which confirms that the requirements are correct and complete: “did we build the right thing?”

System verification and validation (V&V) combines experience, judgment, physical test, and mathematical modeling (simulation). If simulation models are to be used in a V&V context, then they need to be subject to a simulation V&V process, e.g., [4] integrated with the structured systems view.

3.1 Simulation Verification & Validation

In a general sense, a model can be described as a set of assumptions about a given system [2]. Models are used to describe behavior, both static and dynamic, for the

Engineering Tool Guide
Introduction

Roll over or click on the graphic
to learn more about the process.



This is a guide to help you select and apply some useful engineering tools and techniques.

The V-shaped arrangement is applicable at any level, whether system, subsystem or component.

You are strongly encouraged to seek advice on the practical and effective use of these tools and techniques.

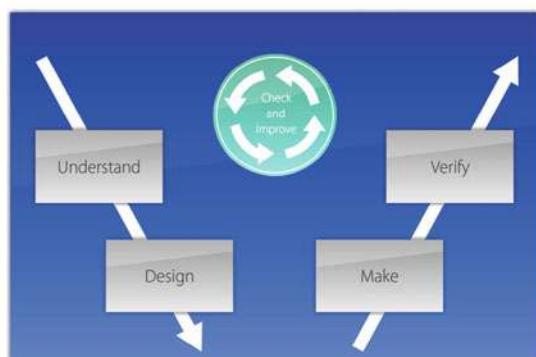


Fig. 42.4 Rolls-Royce engineering tool guide (© Rolls-Royce plc)

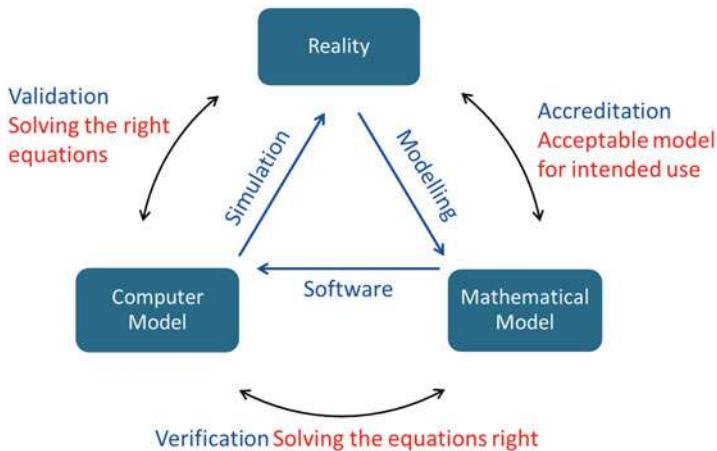


Fig. 42.5 Simulation V&V (A simplified version of the Sargent Cycle [4])

purpose of understanding and controlling (designing) the system. Simulation then is the evaluation of those models to mimic the relevant features of a real process. Once a simulation model is built, it needs to be verified and validated.

In line with the fractal nature of the systems engineering V-diagram, simulation V&V can be viewed as a subsystem of systems V&V (Fig. 42.5), where the goal is to verify the models against experience, judgment, and test data and to validate those models against requirements.

Simulation therefore has multiple roles as it is used throughout the product development process, from design to system validation. Validation can involve testing the simulation model against physical test data for the right physics, the right geometry, the right mesh, the right boundary conditions, etc. This can be achieved with an integrated test and simulation strategy. In the end, the simulation model must be shown to be fit for purpose, and this of course means that the validation process very much depends on the purpose. At this stage one must be mindful of the bounds within which the model is shown to be valid. A model representing a particular design configuration may be calibrated against test data at a particular operating point, but that does not necessarily mean it is a valid model at other design configurations or operating points. If the model is to be used for a design trade study, or for optimization, then the validity of the model over the whole of the defined design space needs to be considered. The discrepancy between model and observed behavior can be modeled, either explicitly or by considering the plausibility of the model as it is evaluated at different locations in the design space.

Where simulation models are used in the early stages of design, they will often need to be evaluated many times, and with many different input configurations (e.g., changes in geometry), it may be possible to use simplified models that are faster to run where less accuracy is required, for example, to detect design trends. Where detailed optimization studies are needed, a more accurate but slower-running model may be necessary to accurately capture behavior. Thus a multi-fidelity approach to modeling may be required, where there are several models of the same system, with different blends of speed and accuracy, used as the situation dictates. Such models need to be aligned with each other, so that they can be interchanged at will. Considering finite element models, this could be achieved by varying the size of the FE mesh, with coarse meshing for faster models and finer meshing for more detailed analysis, subject to model validation. Alternatively, linearized or reduced-order models (ROM) may be used.

4 Uncertainty Management Framework

As previously mentioned, the engineering system can be represented by a network of models. In order to propagate variation through this network, a common parametric model framework is required. The models must be able to represent variation as it is experienced in manufacturing and use. This requires that uncertainty in manufacture and use be defined, measured, and expressed as variation and then propagated as efficiently as possible. The general framework for uncertainty management in analysis is shown in Fig. 42.6.

There are four key pillars to this framework.

1. Quantify input uncertainty (identify sources, elicit, measure)
2. Model the system (validated models, including representation of uncertainty)
3. Propagate uncertainty using a suitable method (see below)
4. Analyze/optimize (sensitivity, objectives, constraints, algorithms)

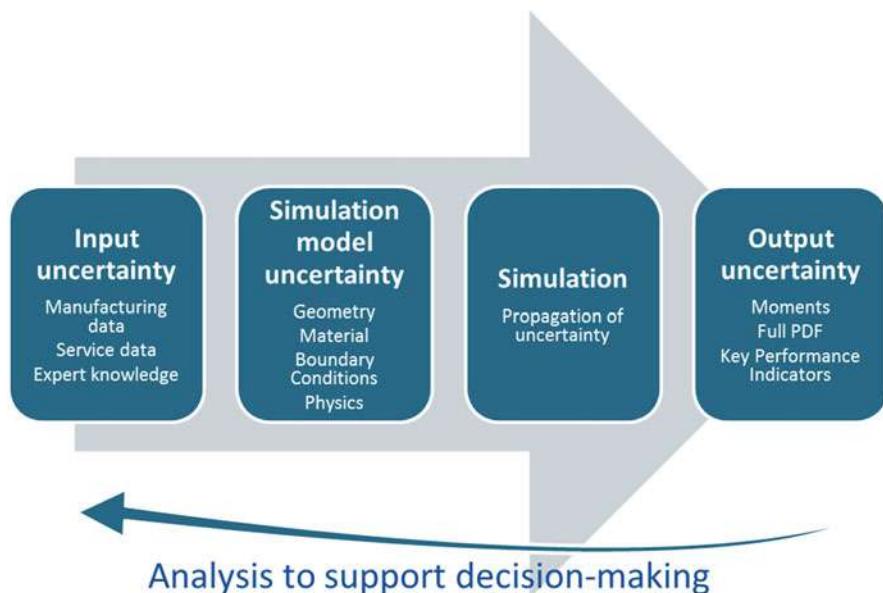


Fig. 42.6 General simulation framework

This simulation framework can be used to solve the forward problem (what is the variation in outputs, given variation in inputs?) and the inverse problem (what input variation caused the observed output variation?). There are significant challenges associated with each pillar of the framework, and these will be further discussed now.

4.1 Quantification of Input Uncertainty

Standard robust design and systems engineering tools can be used to identify sources of input uncertainty, and once identified, these need to be quantified. This usually involves gathering data, although interval-based methods do exist; they are not usually associated with UQ studies, which concentrate on probabilistic analysis. Where deeper uncertainty exists (i.e., uncertainty that cannot be quantified), one can take a scenario-based approach, as previously mentioned. Different types of input uncertainty related to engineering analysis include:

1. Physics. This defines the nature of the problem, e.g., find displacements given forces on objects.
2. Geometry. This defines the domain over which the problem will be solved. This may not necessarily be the fully featured part geometry but could be a simplification.

3. Meshing. The geometry domain needs to be discretized with a mesh as part of the FEM formulation of the problem.
4. Boundary conditions. These define the initial conditions for the model.
5. Solver. In general, the problem needs to be solved numerically, and hence any solution is approximate.
6. Computer hardware. Decomposition of a large problem into subproblems that can be solved with available computing resource (memory, CPU power, bus bandwidth).

4.2 Modeling and Simulation Framework

A system model or network of models that represents the system needs to be:

1. Representative of the relevant system responses,
2. Flexible enough to represent the design space and input uncertainties,
3. Able to run automatically (in “batch mode”), to facilitate design studies and computer experiments.

These modeling and simulation requirements demand flexible, parametric models that can be integrated into workflows containing several simulation codes with data being shared automatically between them. In the case of finite element models, this means automatic generation of geometry, automatic meshing of that geometry, and automatic application of forces and boundary conditions, which are all nontrivial tasks.

Automation is needed for propagation of variation, for example, by using computer experiments [3] where design of experiments (DoE) methods are used to fit empirical models (variously known as emulators, meta-models, surrogates, response surface models) that approximate the behavior of the simulation models, but run much faster and can be used more effectively for design space exploration and optimization. An additional benefit of workflow automation that includes empirical model fitting is that block DoEs can be run in parallel and, in addition, sequential modeling strategies can minimize the number of costly simulation runs needed to fit a good-quality model.

The model network should also be aligned with the broader “systems” view, so that changes to the model can be evaluated by other subsystems (e.g., cost of manufacture) and also so that any resulting design changes can be implemented. Alignment of model networks also facilitates multi-fidelity analysis, where there are multiple models of the same system. Where computational costs are high, a multi-fidelity model framework can be used to achieve accuracy in design search and optimization while reducing the overall computational burden.

4.3 Representation of Uncertainty in Simulation Models

Promoting the parameters of a simulation model from deterministic to stochastic can be very costly from a numerical simulation perspective. Even the most sophisticated propagation methods currently struggle to deal with more than around 20 random variables, when applied to nonlinear systems. Where data exists, variation in individual parameter values can be characterized by standard univariate distributions. Accounting for correlation can be more problematic, but is no less important. Where there is a spatial aspect to the parameters (e.g., boundary conditions along an edge, or a face, material properties, or even geometry), then random fields may be used.

4.4 Propagation of Uncertainty

Methods for the propagation of uncertainty through the model framework include:

- Monte Carlo methods
- Sparse quadrature
- Sigma point methods
- gPC: generalized polynomial chaos
- Stochastic reduced basis methods (for stochastic PDEs)
- Adjoint solvers (for efficient computation of derivatives)

These methods are discussed in detail in other chapters. It is interesting to note that many of these methods are intimately bound to the statistical models that represent the uncertainty, which can make it difficult to develop and deploy generic solutions to the engineering design community.

4.5 Decision Analysis

Evaluation of the model framework provides insight into system performance and design. Specific applications include sensitivity studies, general search in the design space, and numerical optimization. Design information gathered using these techniques can be used to inform decisions during the product development process and to confirm (validate) system behavior as part of a systems V&V process and also for product certification by external authorities. Where information exists on variation in system response (e.g., in-service data, or operational data), then an inverse problem formulation of the model network can attempt to identify the likely variation in system inputs that led to the variation in outputs.

4.5.1 Sensitivity Studies

If one considers the model network as a transfer function, mapping inputs (design parameters) to outputs (system responses), then at its simplest, a sensitivity study can be used to rank the inputs according to their effect on the outputs. The model network evaluates the change in output for a fixed (e.g., unit, or 1%) change in inputs. This does not take into account different levels of uncertainty that each of the inputs may have, which in turn will change the amount by which the outputs will vary. Therefore, in order to be useful, the likely variation of input values must be used. Where these are not known, sensitivity studies are of limited use.

4.5.2 Robust Optimization

Numerical optimization algorithms can be applied to the model network to optimize performance. The lower and upper bounds of the model (input) parameters define a design space which can be explored. If variation in the input parameters is ignored, there is the danger that optimal designs are identified that are at constraint boundaries or that are highly sensitive to small changes in input parameter values (cliff-edge designs). Therefore the robustness of the design to variation in inputs needs to be evaluated, which of course is a prime motivator for UQ studies.

Robust optimization imposes a significant computational burden because a UQ study is potentially needed at every iteration of the optimizer, creating the requirement for fast, efficient UQ methods. The number of objectives is higher for robust optimization as there may well be multiple attributes (e.g., mean, variance), for each response.

5 An Example Application: Jet Engine Disc and Blade Design

The design and development of a new engine is supported by analysis and test operations that rely on models at system, subsystem, and component level. The impact of variation at component level needs to be assessed not only at that level but also at higher levels. This can create a significant challenge for the practical application of UQ methods.

A generic example problem, which is nonetheless based on a real engineering design problem, is presented to illustrate some of the issues that have been discussed. The problem is to design a blade and disc assembly for a jet engine. The idea behind the example is to show the hierarchy of design systems and the multifaceted nature of the design problem.

The disc and blade design problem is addressed at component level, but also needs to be considered at subsystem level (compressor or turbine) and at system (whole engine) level. Figures 42.7, 42.8, and 42.9 show these three levels of design geometry. These geometry models form the basis of the set of analysis models used to inform both design and the verification and validation of the system.

Figure 42.7 shows a schematic of a Rolls-Royce Plc Trent engine. At the core of the engine lies the compressor-combustor-turbine arrangement, where air is compressed, heated, and expanded to produce work. The detailed designs of

Fig. 42.7 Rolls-Royce next-generation advance engine (© Rolls-Royce Image Library)

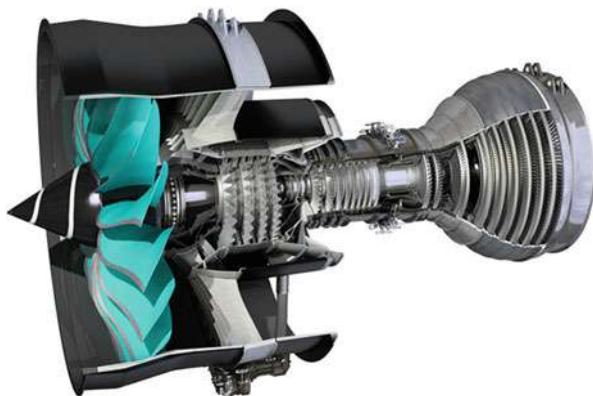
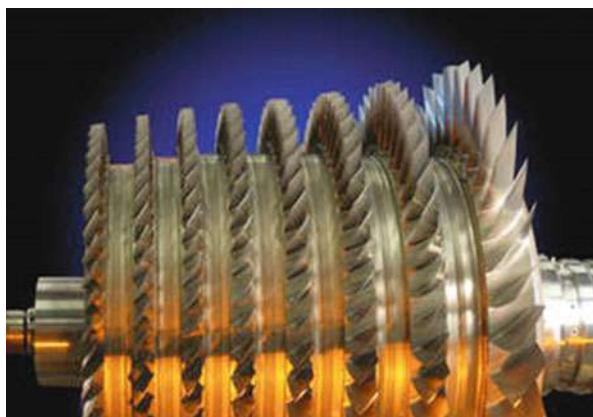


Fig. 42.8 Engine compressor system (©Rolls-Royce Image Library)



the blade and disc arrangements of the compressor and turbine are critical in determining overall engine performance and are subject to multiple design criteria. Performance is evaluated over a full flight profile, with hundreds of complex thermomechanical loading conditions. The blade and disc components need to be evaluated for strength (stress, strain), fatigue, and vibration, requiring the model to be able to represent nonlinear physical phenomena such as creep, plasticity, fatigue, and contact and friction, all in a coupled thermomechanical model.

Not all of these conditions and physics-based models need to be evaluated at the whole engine level. Whole engine analysis is extremely costly – more can be done with available resources if the physical phenomena that drive individual design criteria are modeled at appropriate scale and fidelity. This however creates an overhead in managing the hierarchy of analysis models. The models need to be kept in alignment via common geometry definitions and consistent boundary conditions, all while allowing them to flex to take uncertainty into account and to allow for larger design changes in the search for a good design solution.

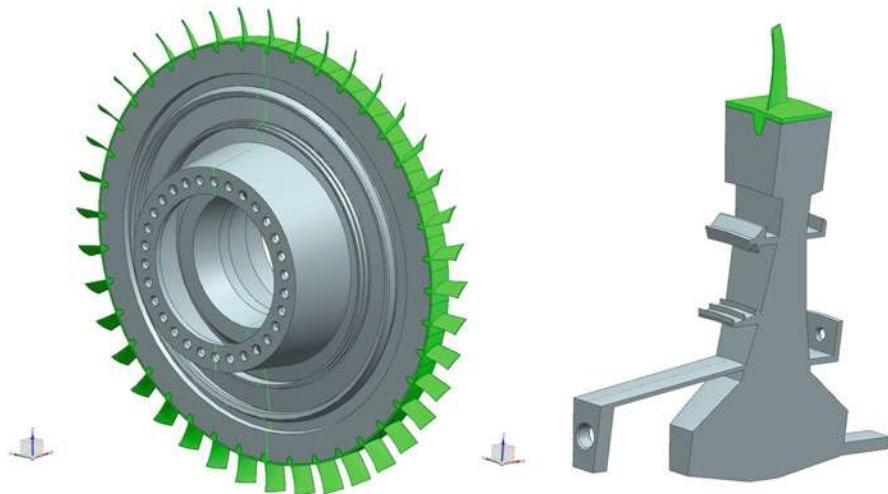


Fig. 42.9 A generic disc and blade model arrangement (*left*) and section thereof (*right*)

The disc and blade arrangement of the compressor subsystem is shown in Fig. 42.8. This subsystem can be modeled and analyzed to determine key performance characteristics, which may involve more detailed modeling at the individual blade and disc level, as depicted in Fig. 42.9

At this detailed level, effects such as contact and friction between blade and disc can be modeled. Thus there is a well-defined modeling pathway from engine to subsystem to component.

6 Challenges For the Application of UQ

In summary, the application of UQ methods to the design of engineering systems poses the following challenges:

1. Simulation
 - a. Multidisciplinary: common parametric models to enable flow of data
 - b. Aligned models: systems of multi-fidelity models
 - c. Fully stochastic simulation models
2. Uncertainty quantification
 - a. Data-led quantification of sources of uncertainty
 - b. Methods for UQ where data are sparse (expensive) and multi-scale
 - c. Propagation methods integrated with simulation models
3. Optimization
 - a. Improved automated geometry creation and meshing (morph geometry or mesh?)
 - b. Increase in the number of objectives considered during optimization

4. Robust design

- a. Tools and methods able to cope with large nonlinear problems
 - b. Hierarchy of variable fidelity models from component to subsystem to system
 - c. Full “System view” of uncertainty quantification and management: decision-making in the face of uncertainty
 - d. Validation: calibration at a test point, extrapolation to a new design vs interpolation
 - e. “Close the loop”: how well did we predict the variation that we see in the field?
-

7 Conclusion

Several aspects of managing uncertainty in engineering design have been discussed. Uncertainty, risk, and decision-making are closely related to each other and also to the process of design and development of new products. By classifying uncertainty into different types (randomness, incompleteness, ineptitude), one can begin to define effective strategies for managing uncertainty. These strategies should be integrated with a systems engineering approach and should ultimately define a hierarchy of models that includes functional models, static and dynamic simulation models, and data, all of which are verified to accurately represent the (sub) systems in question and validated against the original design requirements. This network of models forms the foundation of a framework for managing uncertainty that can be exploited to develop an understanding of how the many forms of variation (randomness) impact design performance.

There are many challenges to making this vision a reality. UQ studies can be costly, both in terms of generating models that are fit for purpose and in evaluating those models to understand how randomness is propagated through them. It can be difficult to develop well-aligned hierarchies of models, and it can also be difficult to understand how the results of a probabilistic simulation study can be used upstream to aid the design decision-making process.

Despite these challenges, the main argument remains that managing risk and uncertainty in engineering design can be achieved using a well-organized, well-structured approach. This can help to minimize risk and create an understanding of uncertainty which can then be used to inform designers on how to make products more robust to variation and more resilient to change.

References

1. Snowden, D.J., Boone, M.E.: A leader’s framework for decision making. *Harv. Bus. Rev.* **85**(11), 68–76 (2007)
2. Hartmann, S.: The world as a process: simulations in the natural and social sciences. In: Hegselmann, R., et al. (eds.) *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View, Theory and Decision Library*. pp. 77–100. Kluwer, Dordrecht (1996)

-
3. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
 4. Sargent, R.G.: Validation of simulation models. In: Highland, H.J., Spiegel, M.F., Shannon, R.E. (eds.) *Proceedings of the 1979 Winter Simulation Conference*. IEEE, Piscataway, pp 497–503 (1979)
 5. Stephens, E.M., Edwards, T.L., Demeritt, D.: Communicating probabilistic information from climate model ensembles—lessons from numerical weather prediction. *WIREs Clim. Change* **3**, 409–426 (2012). doi10.1002/wcc.187

Quantifying and Reducing Uncertainty About Causality in Improving Public Health and Safety

43

Louis Anthony Cox, Jr.

Abstract

Effectively managing uncertain health, safety, and environmental risks requires quantitative methods for quantifying uncertain risks, answering the following questions about them, and characterizing uncertainties about the answers:

- Event detection: What has changed recently in disease patterns or other adverse outcomes, by how much, when?
- Consequence prediction: What are the implications for what will probably happen next if different actions (or no new actions) are taken?
- Risk attribution: What is causing current undesirable outcomes? Does a specific exposure harm human health, and, if so, who is at greatest risk and under what conditions?
- Response modeling: What combinations of factors affect health outcomes, and how strongly? How would risks change if one or more of these factors were changed?
- Decision making: What actions or interventions will most effectively reduce uncertain health risks?
- Retrospective evaluation and accountability: How much difference have exposure reductions actually made in reducing adverse health outcomes?

These are all causal questions. They are about the uncertain causal relations between causes, such as exposures, and consequences, such as adverse health outcomes. This chapter reviews advances in quantitative methods for answering them. It recommends integrated application of these advances, which might collectively be called causal analytics, to better assess and manage uncertain risks. It discusses uncertainty quantification and reduction techniques for causal modeling that can help to predict the probable consequences of different policy

L.A. Cox, Jr., (✉)

Cox Associates and University of Colorado, Denver, CO, USA

e-mail: tcoxdenver@aol.com

choices and how to optimize decisions. Methods of causal analytics, including change-point analysis, quasi-experimental studies, causal graph modeling, Bayesian Networks and influence diagrams, Granger causality and transfer entropy methods for time series, and adaptive learning algorithms provide a rich toolkit for using data to assess and improve the performance of risk management efforts by actively discovering what works well and what does not.

Keywords

Adaptive learning • Change-point analysis (CPA) • Bayesian networks (BN)
• Causal analytics • Causal graph • Causal laws • Counterfactual • Directed acyclic graph (DAG) • DAG model • Dynamic Bayesian networks (DBN)
• Ensemble learning algorithms • Evaluation analytics • Granger causality
• Influence diagram (ID) • Intervention analysis • Interrupted time series analysis • Learning analytics • Marginal structural model (MSM) • Model ensembles • Multi-agent influence diagram (MAID) • Path analysis • Predictive analytics • Prescriptive analytics • Propensity score • Quasi-experiments (QEs) • Simulation • Structural equations model (SEM) • Structure discovery • Transfer Entropy • Uncertainty analytics

Contents

1	Introduction	1439
2	Some Limitations of Traditional Epidemiological Measures for Causal Inference: Uncertainty About Whether Associations Are Causal	1442
2.1	Example: Exposure-Response Relations Depend on Modeling Choices	1443
3	Event Detection and Consequence Prediction: What's New, and So What?	1448
3.1	Example: Finding Change Points in Surveillance Data	1449
4	Causal Analytics: Determining Whether a Specific Exposure Harms Human Health	1453
5	Causes and Effects Are Informative About Each Other: DAG Models, Conditional Independence Tests, and Classification and Regression Tree Algorithms	1454
6	Changes in Causes Should Precede, and Help to Predict and Explain, Changes in the Effects that They Cause	1455
6.1	Change-Point Analysis Can Be Used to Determine Temporal Order	1456
6.2	Intervention Analysis Estimates Effects of Changes Occurring at Known Times, Enabling Retrospective Evaluation of the Effectiveness of Interventions	1456
6.3	Granger Causality Tests Show Whether Changes in Hypothesized Causes Help to Predict Subsequent Changes in Hypothesized Effects	1459
7	Information Flows From Causes to Their Effects over Time: Transfer Entropy	1460
8	Changes in Causes Make Future Effects Different From What They Otherwise Would Have Been: Potential-Outcome and Counterfactual Analyses	1461
9	Valid Causal Relations Cannot Be Explained Away by Noncausal Explanations	1466
10	Changes in Causes Produce Changes in Effects via Networks of Causal Mechanisms	1466
10.1	Structural Equation and Path Analysis Models Model Linear Effects Among Variables	1467

10.2	Bayesian Networks Show How Multiple Interacting Factors Affect Outcome Probabilities.....	1470
10.3	Quantifying Probabilistic Dependencies Among BN Variables.....	1471
10.4	Causal vs. Noncausal BNs.....	1473
10.5	Causal Mechanisms Are Lawlike, Yielding the Same Output Probabilities for the Same Inputs.....	1474
10.6	Posterior Inference in BN Models.....	1475
10.7	Causal Discovery of BNs from Data.....	1477
10.8	Handling Uncertainty in Bayesian Network Models.....	1479
10.9	Influence Diagrams Extend BNs to Support Optimal Risk Management Decision-Making.....	1482
10.10	Value of Information (VOI), Dynamic Bayesian Networks (DBNs), and Sequential Experiments for Reducing Uncertainties Over Time.....	1485
11	Causal Analytics.....	1488
12	Summary and Conclusions: Applying Causal Graph Models to Better Manage Risks and Uncertainties.....	1491
	Cross-References.....	1493
	References.....	1493

1 Introduction

Politically contentious issues often turn on what appear to be technical and scientific questions about cause and effect. Once a perceived undesired state of affairs reaches a regulatory or policy agenda, the question arises of what to do about it to change things for the better. Useful answers require understanding the probable consequences caused by alternative policy actions. For example,

- A century ago, policy makers might have asked whether a prohibition amendment would decrease or increase alcohol abuse.
- A decade ago, policy makers might have wondered whether an invigorated war on drugs would increase or decrease drug abuse.
- Do seatbelts reduce deaths from car accidents, even after accounting for “risk homeostasis” changes in driving behaviors?
- Does gun control reduce deaths due to shootings?
- Does the death penalty reduce violent crime?
- Has banning smoking in bars reduced mortality rates due to heart attacks?
- Do sex education and birth control programs in schools decrease teen pregnancy rates and prevalence of sexually transmitted diseases?
- Has the Clean Air Act reduced mortality rates, e.g., due to lung cancer or coronary heart disease (CHD) or to all causes?
- Will reformulations of childhood vaccines reduce autism?
- Would banning routine antibiotic use in farm animals reduce antibiotic-resistant infections in people?

Policy makers look to epidemiologists, scientists, and risk analysts to answer such questions. They want to know how policy actions will change (or already have changed) outcomes and by how much – how much improvement is caused how

quickly and how long does it last? They want to know what will work best and what has (and has not) worked well in reducing risks and undesirable outcomes without causing unintended adverse consequences. And they want to know how certain or uncertain the answers to these questions are.

Developing trustworthy answers to these questions and characterizing uncertainty about them requires special methods. It is notoriously difficult to quickly and accurately identify events or exposures that cause adverse human health outcomes, quantify uncertainties about causal relations and impacts, accurately predict the probable consequences of a proposed action such as a change in exposure or introduction of a regulation or intervention program, and quantify in retrospect what effects an action actually did cause, especially if other changing factors affected the observed outcomes. The following section explains some limitations of association-based epidemiological and regulatory risk assessment methods that are often used to try to answer these questions. These limitations suggest that association-based methods are not adequate for the task [21], contributing to an unnecessarily widespread prevalence of false positives in current epidemiology that undermines the credibility and value of scientific studies that should be providing trustworthy, crucial information to policy makers [22, 56, 70, 102]. New and better ideas and methods are needed, and are available, to provide better answers. The remaining sections review the current state of the art of methods for answering the following causal questions and quantifying uncertainties about their answers:

1. *Event detection: What has changed recently in disease patterns or other adverse outcomes, by how much, when, and why?* For example, have hospital or emergency room admissions or age-specific mortalities with similar symptoms recently jumped significantly, perhaps suggesting a disease outbreak (or a terrorist bio-attack)?
2. *Consequence prediction: What are the implications for what will probably happen next if different actions (or no new actions) are taken?* For example, how many new illnesses are likely to occur and when? How quickly can a confident answer be developed and how certain and accurate can answers be based on limited surveillance data?
3. *Risk attribution: What is causing current undesirable outcomes? Does a specific exposure harm human health? If so, who is at greatest risk (e.g., children, elderly, other vulnerable subpopulations) and under what conditions (e.g., for what exposure concentrations and durations or for what co-exposures)?* Answering this question is the subject of *hazard identification* in health risk assessment. For example, do ambient concentrations of fine particulate matter or ozone in air (possibly in combination with other pollutants) cause increased incidence rates of heart disease or lung cancer in one or more vulnerable populations? Here, “cause” is meant in the specific sense that reducing exposures would reduce the risks per person per year of the adverse health effects. (The following section contrasts this with other interpretations of “exposure causes disease Y,” such as “exposure X is strongly, consistently, specifically, temporally, and statistically

significantly associated with Y , and the association is biologically plausible and is stronger for greater exposures” or “the fraction of cases of Y attributable to X , based on relative risks or regression models, is significantly greater than zero.” These interpretations do *not* imply that reducing X will reduce Y , as positive associations and large attributable risks may reflect modeling choices or p-hacking, biases, or confounding rather than genuine causation.)

4. *Response modeling: What combinations of factors affect health outcomes and how strongly? How would risks change if one or more of these factors were changed?* For example, what is the quantitative causal relationship between exposure levels and probabilities or rates of adverse health outcomes for individuals and identifiable subpopulations? How well can these relationships be inferred from data, and how can uncertainties about the answers be characterized?
5. *Decision making: What actions or interventions will most effectively reduce uncertain health risks?* How well can the effects of possible future actions be predicted, such as reducing specific exposures, taking specific precautionary measures (e.g., flu shots for the elderly), or other interventions? This is the key information needed to inform risk management decisions before they are made.
6. *Retrospective evaluation and accountability: How much difference have exposure reductions actually made in reducing adverse health outcomes?* For example, has reducing particulate matter air pollution reduced cardiovascular mortality rates over the past decade, or would these reductions have occurred just as quickly without reductions in air pollution (i.e., are these coincident historical trends, or did one cause the other?)

These questions are fundamental in epidemiology and health and safety risk assessment. They are mainly about how changes in exposures affect changes in health outcomes and about how certain the answers are. They can be answered using current methods of causal analysis and uncertainty quantification (UQ) for causal models if sufficient data are available.

The following sections discuss methods for drawing valid causal inferences from epidemiological data and for quantifying uncertainties about causal impacts, taking into account model uncertainty as well as sampling errors and measurement, classification, or estimation errors in predictors. UQ methods based on model ensemble methods, such as Bayesian model averaging (BMA) and various forms of resampling, boosting, model cross validation, and simulation, can help to overcome over-fitting and other modeling biases, leading to wider confidence intervals for the estimated impacts of actions and reducing false-positive rates [50]. UQ has the potential to restore greater integrity and credibility to model-based risk estimates and causal predictions, to reveal the quantitative impacts of model and other uncertainties on risk estimates and recommended risk management actions, and to guide more productive-applied research to decrease key remaining uncertainties and to improve risk management decision-making via active exploration and discovery of valid causal conclusions and uncertainty characterizations.

2 Some Limitations of Traditional Epidemiological Measures for Causal Inference: Uncertainty About Whether Associations Are Causal

Epidemiology has a set of well-developed traditional methods and measures for quantifying associations between observed quantities. These include regression model coefficients and relative risk (RR) ratios (e.g., the ratio of disease rates for exposed and unexposed populations) as well as various quantities derived from them by algebraic rearrangements. Derived quantities include population attributable risks (PARs) and population attributable fractions (PAFs) for the fraction of disease or mortality cases attributable to a specific cause, global burden of disease estimates, etiologic fractions and probability-of-causation calculations, and estimated concentration-response slope factors for exposure-response relations [27, 98]. Although the details of calculations for these measures vary, the key idea for all of them is to observe whether more-exposed people suffer adverse consequences at a higher rate than less-exposed people and, if so, to attribute the excess risks in the more-exposed group to a causal impact of exposure. Conventional statistical methods for quantifying uncertainty about measures of association, such as confidence intervals and *p*-values for RR, PAF, and regression coefficients in logistic regression, Cox proportional hazards, or other parametric or semi-parametric regression models, are typically used to show how firmly the data, together with the assumptions embedded in these statistical models, can be used to reject the null hypothesis of independence (no association) between exposures and adverse health responses. In addition, model diagnostics (such as plots of residuals and formal tests of model assumptions) can reveal whether modeling assumptions appear to be satisfied; more commonly, less informative goodness-of-fit measures are reported to show that the models used do not give conspicuously poor descriptions of the data, at least as far as the goodness-of-fit test can determine. However, goodness-of-fit tests are typically very weak in detecting conspicuously poor fits to data. This is often illustrated by the notorious “Anscombe’s quartet” of qualitatively very different scatter plots giving identical least-squares regression lines and goodness-of-fit test values.

The main limitation of these techniques is that they only address associations, rather than causation. Hence, they typically do not actually quantify the fraction or number of illnesses or mortalities per year that would be prevented by reducing or eliminating specific exposures. Unfortunately, as many methodologists have warned, PAF and probability of causation, as well as regression coefficients, are widely misinterpreted as doing precisely this (e.g., [98]). Large epidemiological initiatives, such as the World Health Organization’s Global Burden of Disease studies, make heavy use of association-based methods that are mistakenly interpreted as if they indicated causal relations. This has become a very common mistake in contemporary epidemiological practice. It undermines the validity, credibility, and practical value of many (some have argued most) causal claims now being published using traditional epidemiological methods [70, 88, 102]. To what extent associations correspond to stable causal laws that can reliably predict future consequences of

policy actions is beyond the power of these traditional epidemiological measures to say [98] doing so requires different techniques.

2.1 Example: Exposure-Response Relations Depend on Modeling Choices

A 2014 article in *Science* [21] noted that “There is a growing consensus in economics, political science, statistics, and other fields that the associational or regression approach to inferring causal relations – on the basis of adjustment with observable confounders – is unreliable in many settings.” To illustrate this point, the authors cite estimates of the effects of total suspended particulates (TSPs) on mortality rates of adults over 50 years old, in which significantly positive associations (regression coefficients) are reported in some regression models that did not adjust for confounders such as age and sex, but significantly negative associations are reported in other regression models that did adjust for confounders by including them as explanatory variables. The authors note that the sign, as well as the magnitude, of reported exposure concentration-response (C-R) relations depends on details of modeling choices about which variables to include as explanatory variables in the regression models. Thus, the quantitative results of risk assessments presented to policy makers as showing the expected reductions in mortality risk per unit decrease in pollution concentrations actually reflect specific modeling choices, rather than reliable causal relations that accurately predict how (or whether) reductions in exposure concentrations would reduce risks.

A distinction from econometrics between structural equations and reduced-form equations [65] is helpful in understanding why different epidemiologists can estimate exposure concentration-response regression coefficients with opposite signs from the same data. The following highly simplified hypothetical example illustrates the key idea. Suppose that cumulative exposure to a chemical increases in direct proportion to age and that the risk of disease (e.g., the average number of illness episodes of a certain type per person per decade) also increases with age. Finally, suppose that the effect of exposure at any age is to decrease risk. These hypothesized causal relations are shown via the following two *structural equations*:

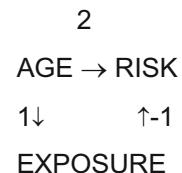
$$\text{EXPOSURE} = \text{AGE}$$

SEM equations

$$\text{RISK} = 2 * \text{AGE} - \text{EXPOSURE}.$$

These are equations with the explicit causal interpretation that a change in the variable on the right side causes a corresponding change in the variable on the left side to restore equality between the two sides (e.g., increasing age increases cumulative exposure and disease risk, but increasing exposure decreases risk at any age). These two structural equations together constitute a *structural equation model* (SEM) that can be diagrammed as in Fig. 43.1:

Fig. 43.1 SEM causal graph model



In this diagram, each variable depends causally only on the variables that point into it, as revealed by the SEM equations. The weights on the arrows (the coefficients in the SEM equations) show how the average value of the variable at the arrow's head will change if the variable at its tail is increased by one unit, for example, increasing AGE by 1 decade (if that is the relevant unit) increases RISK directly by 2 units (e.g., 2 expected illnesses per decade, if that is the relevant unit), increases EXPOSURE by one unit, and thereby decreases RISK indirectly by 1 unit, via the path through EXPOSURE, for a net effect of a 1 unit increase in RISK per unit increase in AGE.

By contrast to such causal SEM models, what is called a *reduced-form model* is obtained by regressing RISK against EXPOSURE. Using the first SEM equation, $\text{EXPOSURE} = 1\text{*AGE}$, to substitute EXPOSURE for AGE in the second SEM equation, $\text{RISK} = 2\text{*AGE} - \text{EXPOSURE}$, yields the following reduced-form equation:

$$\text{RISK} = \text{EXPOSURE}$$

Reduced-form equation

This reduced-form model is a valid *descriptive statistical* model: it reveals that in communities with higher exposure levels, risk should be expected to be greater. But it is not a valid *causal* model: a prediction that reducing exposure would cause a reduction in risk would be mistaken, as the SEM equations make clear. The reduced-form equation is not a structural equation, so it cannot be used to predict correctly how changing the right side would cause the left side to change. The coefficient of EXPOSURE in the linear regression model relating exposure to risk is +1 in the reduced-form model, but is -1 in the SEM model, showing how different investigators might reach opposite conclusions about the sign of “the” exposure-response coefficient based on whether or not they condition on age (or, equivalently, on whether they use structural or reduced-form regression equations).

In current epidemiological practice, the distinction between structural and reduced-form equations is often not clearly drawn. Regression coefficients of various signs and magnitudes, as well as various measures of association based on relative risk ratios, are all presented to policy makers as if they had valid causal interpretations and therefore important implications for risk management policy-making. In air pollution health effects epidemiology, for example, it is standard practice to present regression coefficients as expected reductions in elderly mortality rates (or as expected increases in life span) per unit reduction in air pollution concentrations [24, 28], thereby conflating associations between historical levels

(e.g., pollutant levels and mortality rates both tend to be higher on cold winter days than during the rest of the year, and both have declined in recent decades) with a causal, predictive relation that implies that future reductions in pollution would cause further future reductions in elderly mortality rates. Since such association-based studies are often unreliable indicators of causality [21] or simply irrelevant for determining causality, as in the examples for Fig. 43.1, policy makers who wish to use reliable causal relations to inform policy decisions must seek elsewhere.

These limitations of association-based methods have been well discussed among methodological specialists for decades [98]. Key lessons, such as that the same data set can yield either a statistically significant positive exposure-response regression coefficient or a statistically significant negative exposure-response regression coefficient, depending on the modeling choices made by the investigators, are becoming increasingly appreciated by practitioners [21]. They illustrate an important type of uncertainty that arises in epidemiology, but that is less familiar in many other applied statistical settings: *uncertainty about the interpretation of regression coefficients* (or other association-based measures such as RR, PAF, etc.) as indicating causal relations *vs.* confounded associations or modeling biases *vs.* some of each. This type of uncertainty cannot be addressed by presenting conventional statistical uncertainty measures such as confidence intervals, p-values, regression diagnostics, sensitivity analyses, or goodness-of-fit statistics, since the uncertainty is not about how well a model fits data or about the estimated parameters of the model. Rather, it is about the extent to which the model is only descriptive of the past *vs.* predictive of different futures caused by different choices. Although this is not an uncertainty to which conventional statistical tests apply, it is crucial for the practical purpose of making model-informed risk management decisions. Policy interventions will successfully increase the probabilities of desired outcomes and decrease the frequencies of undesired ones only to the extent that they act causally on drivers of the outcomes and not necessarily to the extent that the models used describe past associations.

One way to try to bridge the gap between association and causation is to ask selected experts what they think about whether or to what extent associations might be causal. However, research on the performance of expert judgments has called into question the reliability of expert judgments, specifically including judgments about causation [61]. Such judgments typically reflect qualitative “weight of evidence” (WoE) considerations about the strength, consistency (e.g., do multiple independent researchers find the claimed associations?), specificity, coherence (e.g., are associations of exposure with multiple health endpoints mutually consistent with each other and with the hypothesis of causality?) temporality (do hypothesized causes precede their hypothesized effects?), gradient (are larger exposures associated with larger risks?), and biological plausibility of statistical associations and the quality of the data sources and studies supporting them. One difficulty is that a strong confounder (such as age in Fig. 43.1) with delayed effects can create strong, consistent, specific, coherent, temporal associations between exposure and risk of an adverse response, with a clear gradient associating larger risks with larger exposures, without providing any evidence that exposure actually causes increased risk.

Showing that an association is strong, for example, does not address whether it is causal, although many WoE systems simply assume that the former supports the latter without explicitly addressing whether the strong associations are instead explained by strong confounding, strong biases, or strong modeling assumptions. Similarly, showing that different investigators find the same or similar association does not necessarily show whether this consistency results from shared modeling assumptions, biases, or confounders. Conflating causal and associational concepts, such as evidence for the strength of an association and evidence for causality of the association, too often makes assessments of causality in epidemiology untrustworthy compared to methods used in other fields, discussed subsequently [51, 83]. Most epidemiologists are trained to treat various aspects of association as evidence for causation, even though they are not, and this undermines the trustworthiness of expert judgments about causation based on WOE considerations [83].

In addition, experts are sometimes asked to judge the *probability* that an association is causal (e.g., [25]). This makes little sense. It neglects the fact that an association may be partly causal and partly due to confounding or modeling biases or coincident historical trends. For example, if exposure does increase risk, but is also confounded by age, then asking for the probability that the regression coefficient relating exposure to risk is causal overlooks the realistic possibility that it reflects both a causal component and a confounding component, so that the probability that it is partly causal might be 1 and the probability that it is completely causal might be 0. A more useful question to pose to experts might be what *fraction* of the association is causal, but this is seldom asked. Common noncausal sources of statistical associations include model selection and multiple testing biases, model specification errors, unmodeled errors in explanatory variables in multivariate models, biases due to data selection and coding (e.g., dichotomizing or categorizing continuous variables such as age, which can lead to residual confounding), and coincident historical trends, which can induce statistically significant-appearing associations between statistically independent random walks – a phenomenon sometimes dubbed as spurious regression [17, 98].

Finally, qualitative subjective judgments and ratings used in many WoE systems are subject to well-documented psychological biases. These include confirmation bias (seeing what one expects to see and discounting or ignoring evidence that might challenge one's preconceptions), motivated reasoning (finding what it benefits one to find and believing what it pays one to believe), and overconfidence (not sufficiently doubting, questioning, or testing one's own beliefs and hence not seeking potentially disconfirming information that might require those beliefs to be revised) [61, 102].

That statistical associations do not in general convey information sufficient for making valid causal predictions has been well understood for decades by statisticians and epidemiologists specializing in technical methods for causal analysis (e.g., [31, 36]). This understanding is gradually percolating through the larger epidemiological and risk analysis communities. Peer-reviewed published papers and reports, including those relied on in many regulatory risk assessments, still too often make the fundamental mistake of reinterpreting empirical exposure-response (ER)

relations between historical levels of exposure and response as if they were causal relations useful for predicting how future changes in exposures would change future responses. Fortunately, this confusion is unnecessary today: appropriate technical methods for causal analysis and modeling are now well developed, widely available in free software such as R or Python, and readily applicable to the same kinds of cross-sectional and longitudinal data collected for association-based studies. Table 43.1 summarizes some of the most useful study designs and methods for valid causal analysis and modeling of causal exposure-response relations.

Despite the foregoing limitations, there is much of potential value in several WoE considerations, especially consistency, specificity, and temporality of associations, especially if they are used as part of a relatively objective, quantitative, data-

Table 43.1 Some formal methods for modeling and testing causal hypotheses

Method and References	Basic Idea	Appropriate study design
Conditional independence tests [31, 32]	Is hypothesized effect (e.g., lung cancer) statistically independent of hypothesized cause (e.g., exposure to chemical), given values of other variables (e.g., education and income)? If so, this undermines causal interpretation.	Cross-sectional data Can also be applied to multi-period data (e.g., in dynamic Bayesian networks)
Panel data analysis [2, 109]	Are changes in exposures followed by changes in the effects that they are hypothesized to help cause? If not, this undermines causal interpretation; if so, this strengthens causal interpretation. Example: Are changes in exposure levels followed (but not preceded) by corresponding changes in mortality rates?	Panel data study: Collect a sequence of observations on same subjects or units over time
Granger causality test [23], transfer entropy [81, 91, 99, 118]	Does the history of the hypothesized cause improve ability to predict the future of the hypothesized effect? If so, this strengthens causal interpretation; otherwise, it undermines causal interpretation. Example: Can lung cancer mortality rates in different occupational groups be predicted better from time series histories of exposure levels and mortality rates than from the time series history of mortality rates alone?	Time series data on hypothesized causes and effects
Quasi-experimental design and analysis [12, 40, 41]	Can control groups and other comparisons refute alternative (noncausal) explanations for observed associations between hypothesized causes and effects? For example, can coincident trends and regression to the mean be refuted as possible explanations? If so, this strengthens causal interpretation.	Longitudinal observational data on subjects exposed and not exposed to interventions that change the hypothesized cause(s) of effects.

(continued)

Table 43.1 (continued)

Method and References	Basic Idea	Appropriate study design
Intervention analysis, change-point analysis [45], Gilmour et al. 2006	<p>Does the best-fitting model of the observed data change significantly at or following the time of an intervention? If so, this strengthens causal interpretation.</p> <p>Do the quantitative changes in hypothesized causes predict and explain the subsequently observed quantitative changes in hypothesized effects? If so, this strengthens causal interpretation.</p> <p>Example: Did lung disease mortality rates fall significantly faster or sooner in workplaces that reduced exposures more or earlier than in workplaces that did not?</p>	<p>Time series observations on hypothesized effects and knowledge of timing of intervention(s)</p> <p>Quantitative time series data for hypothesized causes and effects</p>
Counterfactual and potential outcome models, including propensity scores and marginal structural models (MSMs) [82, 96]	Do exposed individuals have significantly different response probabilities than they would have had if they had not been exposed? Example: Do workers have lower mortality risk after historical exposure reductions than they would have had otherwise?	Cross-sectional and/or longitudinal data, with selection biases and feedback among variables allowed
Causal network models of change propagation [19, 39]	Do changes in exposures (or other causes) create a cascade of changes through a network of causal mechanisms (represented by equations), resulting in changes in the effect variables?	Observations of variables in a dynamic system out of equilibrium
Negative controls (for exposures or for effects) [73]	Do exposures predict health effects better than they predict effects that cannot be caused by exposures more (e.g., reductions in traumatic injuries)?	Observational studies

driven approach to inferring probable causation. The following sections discuss this possibility and show how such traditional qualitative WoE considerations can be fit into more formal quantitative causal analyses.

3 Event Detection and Consequence Prediction: What's New, and So What?

In public health and epidemiology, surveillance data showing changes in hospital or emergency department admission rates for a specific disease or symptom category may provide the first indication that an event has occurred that has caused changes in health outcomes. Initially, the causes of the changes may be uncertain, but if the date of a change can be estimated fairly precisely and matches the date of an event that might have caused the observed effects, then the event might have caused the change

in admissions rates. This causal hypothesis is strengthened if occurrences of the same or similar event in multiple times and places are followed by similar changes in admission rates (consistency and temporality of association) and if these changes in admissions rates do not occur except when the event occurs first (specificity of association). To make this inference sound, the event occurrences must not be triggered by high levels of admissions rates, since otherwise interventions that respond to these high rates might be followed by significant reductions in admission rates due solely to *regression to the mean*, i.e., the fact that exceptionally high levels are likely to be followed by lower levels, even if the interventions have no impact [12].

The technical methods used to estimate when admission rates or other effect have changed significantly, such as counts of accidents or injuries or fatalities per person per week in a population, include several different statistical anomaly-detection and change-point analysis (CPA) algorithms (e.g., [108]). The key idea of these algorithms is to determine whether, for each point in time (e.g., for each week in a surveillance time series), the series is significantly different (e.g., in distribution or trend) before that time point than after it. If so – if a time series jumps at a certain time – that time is called a change point.

3.1 Example: Finding Change Points in Surveillance Data

As an example of change-point detection in surveillance data, consider the following example. Since 2001, when a letter containing anthrax led to 5 deaths and 17 infections from which the victims recovered, the US Environmental Protection Agency (EPA), the Centers for Disease Control and Prevention (CDC), and the Department of Health Services have invested over a billion dollars to develop surveillance methods and prevention and preparedness measures to help reduce or mitigate the consequences of bioterrorism attacks should they occur again [38]. Detecting a significant upsurge in hospital admissions with similar symptoms may indicate that a bioterrorism attack is in progress. The statistical challenge of detecting such changes against the background of normal variability in hospital admissions has motivated the development of computational intelligence methods that seek to reduce the time to detect attacks when they occur, while keeping the rates of false positives acceptably small [11, 106].

Well-developed, sophisticated techniques of statistical uncertainty quantification are currently available for settings in which the patterns for which one is searching are well understood (e.g., a jump in hospitalization rates for patients with similar symptoms that could be caused by a biological agent) and in which enough surveillance data are available to quantify background rates and to monitor changes over time. Figure 43.2 presents a hypothetical example showing weekly counts of hospital admissions with specified symptoms in a certain city. Given such surveillance data, the risk assessment inference task is to determine whether the hospitalization rate increased at some point on time (suggestive of an attack) and, if so, when and by how much. Intuitively, it appears that counts are greater on the

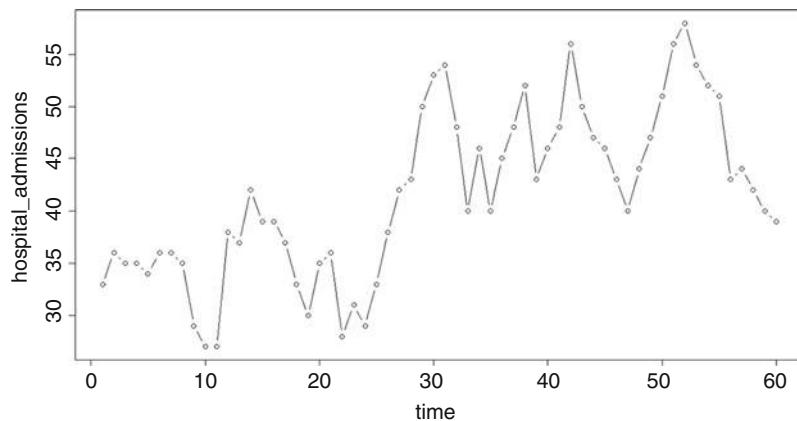


Fig. 43.2 Surveillance time series showing a possible increase in hospitalization rates

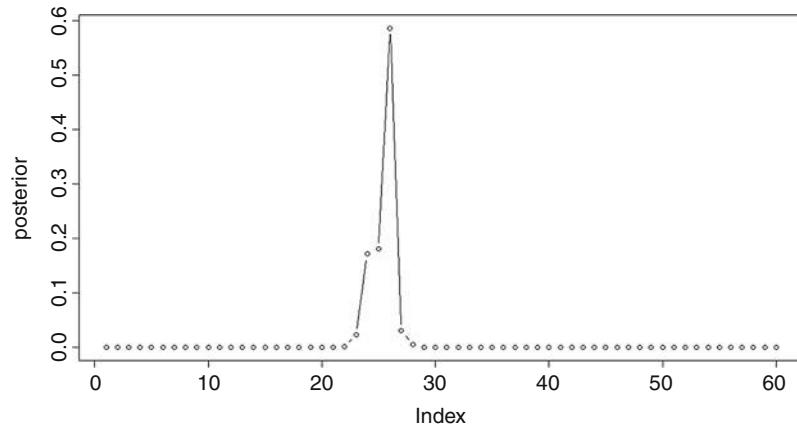


Fig. 43.3 Bayesian posterior distribution for the timing of the increase in Fig. 43.2 if one has occurred

right side of Fig. 43.2 than the left, but might this plausibly just be due to chance, or is it evidence for a real increase in hospitalization rates?

Figure 43.3 illustrates a typical result of current statistical technology (also used in computational intelligence, computational Bayesian, machine learning, pattern recognition, and data mining technologies) for solving such problems by using statistical evidence, together with risk models, to draw inferences about what is probably happening in the real world. The main idea is simple: the highest points indicate the times that are computed to be most likely for when a change in hospitalization rate occurred, based on the data in Fig. 43.2. (Technically, Fig. 43.3 plots the likelihood function of the data, assuming that at most one jump from one level to a different level has occurred for the hospitalization rate. The

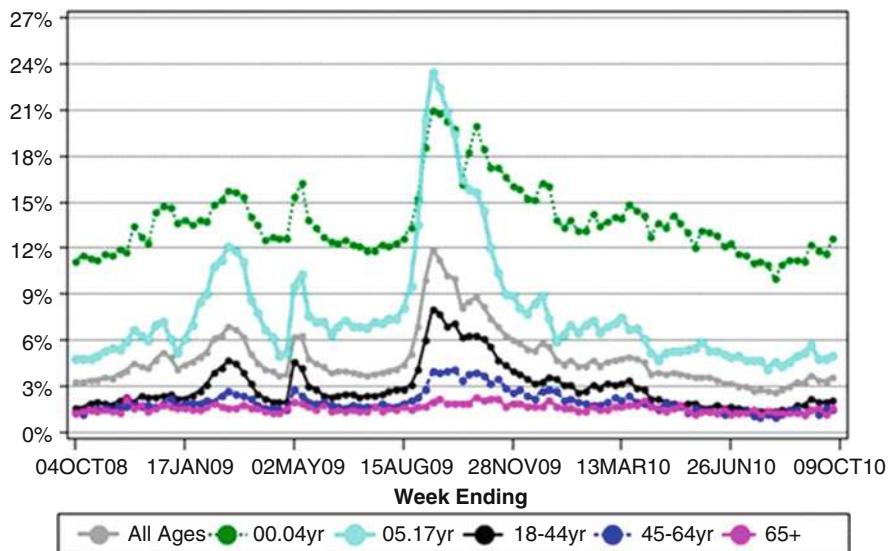


Fig. 43.4 Proportion of emergency department visits for influenza-like illness by age group for October 4, 2008–October 9, 2010, in a US Department of Health and Human Services region (Source: [62], <http://jamia.oxfordjournals.org/content/19/6/1075.long>)

likelihoods are rescaled so that their sum for all 60 weeks is 1, so that they can be interpreted as posterior probabilities if the prior is assumed to be uniform. More sophisticated algorithms are discussed next.) The maximum likelihood-based algorithm accurately identifies both the time of the change (week 25) and the magnitude of its effect to one significant decimal place (not shown in Fig. 43.3). The spread of the likelihood function (or posterior probability distribution) around the most likely value in Fig. 43.3 also shows how precise is the estimation of the change-point time.

Figure 43.4 shows a real-world example of a change point and its consequences for emergency department visits over time. Admissions for flu-like symptoms, especially among infants and children (0–4 and 5–17 year olds), increased sharply in August and declined gradually in each age group thereafter. Being able to identify the jump quickly and then applying a predictive model – such as a stochastic compartmental transition model with susceptible, infected, and recovered subpopulations (SIR model) for each age group – to predict the time course of the disease in the population can help forecast the care resources that will be needed over time for each age group.

More generally, detecting change points can be accomplished by testing the null hypothesis of no change for each time step and correcting for multiple testing bias (which would otherwise inflate false-positive rates, since testing the null hypothesis for each of the many different possible times at which a change might have occurred multiplies the occasions on which an apparently significant change occurs

by chance). Many CPA algorithms use likelihood-based Bayesian methods, as in Fig. 43.3, to identify when a change is most likely to have occurred and whether the hypothesis that it did provides a significantly better explanation (higher likelihood) for the observed data than the null hypothesis of no change. Likelihood-based techniques are fundamental for a wide variety of statistical detection and estimation algorithms. Practitioners can use free, high-quality algorithms available in the R statistical computing environment (e.g., <http://surveillance.r-forge.r-project.org/>; [57]), Python, and other statistics programs and packages to perform CPA analyses.

Algorithms for change-point detection have recently been extended to allow detection of multiple change points within multivariate time series, i.e., in time series containing observations of multiple variables instead of only one [57]. These new algorithms use nonparametric tests (e.g., permutation tests) to determine whether the distributions of the observations before and after the change point differ significantly, even if neither distribution is known, and hence no parametric statistical model can be specified [57]. The development of powerful nonparametric (“model-free”) methods for testing the null hypothesis of no change in the (unknown) distribution enables CPA that is much more robust to uncertainties in modeling assumptions than was possible previously. Assumptions that remain, such as that observations following a change point are drawn from a new distribution, independently of the observations preceding the change point, are statistically testable and weaker than the assumptions (such as approximately normally distributed observations) made in older CPA work.

The use of CPA to search for significant changes in surveillance time series, showing that the number of undesirable events per person per week in a population underwent significant changes at certain times, has allowed the probable causes of observed changes in health and safety to be identified in many applications, providing evidence for or against important causal relations between public policy measures and resulting health and safety effects. For example,

- Nakahara et al. [85] used CPA to assess the impact on vehicle crash fatalities of a program initiated in Japan in 2002 that severely penalized drunk driving. Fatality rates between 2002 and 2006 (the end of the available data series) were significantly lower than between 1995 and 2002. However, the CPA revealed that the change point occurred around the end of 1999, right after a high-profile vehicle fatality that was much discussed in the news. The authors concluded that changes in drunk-driving behavior occurred well before the new penalties were instituted.
- In Finland in 1981–1986, a nationwide oral poliovirus vaccine campaign was closely followed by, and partly overlapped with, a significant increase in the incidence of Guillain-Barré syndrome (GBS). This temporal association raised the important question of whether something about the vaccine might have caused some or all of the increase in GBS. Kinnunen et al. [63] applied CPA to medical records from a nationwide Hospital Discharge Register database. They found that a change point in the occurrence of GBS had probably already taken place before the oral poliovirus vaccine campaign started. They concluded that

there was a temporal association between poliovirus infection and increased occurrence of GBS, but no proof of the suspected causal relation between oral poliovirus vaccines and risk of GBS. This example shows how a precise investigation of the details of temporal associations can both refute some causal hypotheses and suggest others – in this case, that an increase in polio in the population was a common cause of both increased GBS risk and the provision of the vaccine. It also illustrates why a temporal association between an adverse effect and a suspected cause, such as the fact that administration of vaccines preceded increases in GBS risk, should not necessarily be interpreted as providing evidence to support the hypothesis of a causal relation between them.

4 Causal Analytics: Determining Whether a Specific Exposure Harms Human Health

Table 43.2 lists seven principles that have proved useful in various fields for determining whether available data provide valid evidence that some events or conditions cause others. They can be applied to epidemiological data to help determine whether and how much exposures to a hazard contribute causally to subsequent risks of adverse health outcomes in a population, in the sense that reducing exposure would reduce risk – for example, whether and by how much a given reduction in air pollution would reduce cardiovascular mortality rates among the elderly, whether and by how much reducing exposure to television violence in childhood would reduce propensity for violent behavior years later, or whether decreasing high-fat or high-sugar diets in youth would reduce risks

Table 43.2 Principles of causal analytics

1. **Conditional independence principle:** Causes and effects are informative about each other. Technically, there should be positive mutual information (measured in bits and quantified by statistical methods of information theory) between the random variables representing a cause and its effect. This positive mutual information cannot be removed by conditioning on the levels of other variables.
 2. **Granger principle:** Changes in causes should precede, and help to predict and explain, changes in the effects that they cause.
 3. **Transfer entropy principle:** Information flows from causes to their effects over time.
 4. **Counterfactual principle:** Changes in causes make future effects different from what they otherwise would have been.
 5. **Causal graph principle:** Changes in causes produce changes in effects by propagation via one or more paths (sequences of causal mechanisms) connecting them.
 6. **Mechanism principle:** Valid causal mechanisms are lawlike, yielding identically distributed outputs when the inputs are the same.
 7. **Quasi-experiment principle:** Valid causal relations produce differences (e.g., compared to relevant comparison groups) that cannot be explained away by noncausal explanations.
-

of heart attacks in old age. The following sections explain and illustrate these principles and introduce technical methods for applying them to data. They also address the fundamental questions of how to model causal responses to exposure and other factors, how to decide what to do to reduce risk, how to determine how well interventions have succeeded in reducing risks, and how to characterize uncertainties about the answers to these questions.

5 Causes and Effects Are Informative About Each Other: DAG Models, Conditional Independence Tests, and Classification and Regression Tree Algorithms

A key principle for causal analytics is that causes and their effects provide information about each other. If exposure is a cause of increased disease risk, then measures of exposure and of response (i.e., disease risk) should provide *mutual information* about each other, in the sense that the conditional probability distribution for each varies with the value of the other. Software for determining whether this is the case for two variables in a data set is discussed at the end of this section. In addition, if exposures are direct causes of responses, then the mutual information between them cannot be eliminated by conditioning on the values of other variables, such as confounders: a cause provides unique information about its effects. This provides the basis for using statistical *conditional independence tests* to test the observable statistical implications of causal hypotheses: *An effect should never be conditionally independent of its direct causes, given (i.e., conditioned on) the values of other variables.*

As a simple example, if both air pollution and elderly mortality rates are elevated on cold winter days, then if air pollution is a cause of increased elderly mortality rate, the mutual information between air pollution and elderly mortality rates should not be eliminated (“explained away”) by temperature, even though temperature may be associated with each of them. If both temperature and air pollution contribute to increased mortality rates (indicated in causal graph notation as $temperature \rightarrow mortality_rate \leftarrow pollution$), then conditioning on the level of temperature will not eliminate the mutual information between pollution and mortality rate. On the other hand, if the correct causal model were that temperature is a confounder that explains both mortality rate and pollution (e.g., because coal-fired power plants produce more pollution during days with extremely hot and cold weather, and, independently, these temperature extremes lead to greater elderly mortality), diagrammed as $mortality_rate \leftarrow temperature \rightarrow pollution$, then conditioning on the level of temperature would eliminate the mutual information between pollution and mortality rate. Thus, tests that reveal conditional independence relations among variables can also help to discriminate among alternative causal hypotheses.

The notation in these graphs is as follows. Each node in the graph (such as *temperature*, *pollution*, or *mortality_rate* in the preceding example) represents a random variable. Arrows between nodes reveal statistical dependencies (and,

implicitly, conditional independence relations) among the variables. The arrows are usually constrained to form a directed acyclic graph (DAG), meaning that no node can be its own predecessor in the partial ordering of nodes determined by the arrows. The probability distribution of each variable with inward-pointing arrows depends on the values of the variables that point into it, i.e., the conditional probability distribution for the variable at the head of an arrow is affected by the values of its direct “parents” (the variables that point into it) in the causal graph. Conversely, a random variable represented by a node is conditionally independent of all other variables, given the values of the variables that point into it (its parents in the DAG), the values of the variables into which it points (its children), and the values of any other parents of its children (its spouses) – a set of nodes collectively called its Markov blanket in the DAG model.

To illustrate these ideas, suppose that X causes Y and Y causes Z , as indicated by the DAG and $X \rightarrow Y \rightarrow Z$, where X is an exposure-related variable (e.g., job category for an occupational risk or location of a residence for a public health risk), Y is a measure of individual exposure, and Z is an indicator of adverse health response. Then even though each variable is statistically associated with the other two, Z is conditionally independent of X given the value of Y . But Z cannot be made conditionally independent of Y by conditioning on X . One way to test for such conditional independence relations in data is with classification and regression tree algorithms (see, e.g., <https://cran.r-project.org/web/packages/rpart/rpart.pdf> for a free R package and documentation). In this example, a tree for Z would not contain X after splitting on values of Y , reflecting the fact that Z is conditionally independent of X given Y . However, a tree for Z would always contain Y , provided that the data set is large and diverse enough so that the tree-growing algorithm can detect the mutual information between them.

For practitioners, algorithms are now freely available in R, Python, and Google software packages to estimate mutual information, the conditional entropy reduction in one variable when another is observed, and related measures for quantifying how many bits of information observations of one variable provide about another and whether one variable is conditionally independent of another given the values of other variables ([74]; Ince et al. 2009). For example, free R software and documentation for performing these calculations can be found at the following sites:

<https://cran.r-project.org/web/packages/entropy/entropy.pdf>

<https://cran.r-project.org/web/packages/partykit/vignettes/partykit.pdf>.

6 Changes in Causes Should Precede, and Help to Predict and Explain, Changes in the Effects that They Cause

If changes in exposures always precede and help to predict and explain subsequent corresponding changes in health effects, this is consistent with the hypothesis that exposures cause health effects. The following methods and algorithms support formal testing of this hypothesis.

6.1 Change-Point Analysis Can Be Used to Determine Temporal Order

The change-point analysis (CPA) algorithms already discussed can be used to estimate when changes in effects time series occurred. These times can then be compared to the times at which exposures changed (e.g., due to passage of a regulation or to introduction or removal of a pollution source) to determine whether changes in exposures are followed by changes in effects. For example, many papers have noted that bans on public smoking have been followed by significant reductions in risks of heart attacks (acute myocardial infarctions). However, Christensen et al. [14], in a study of the effects of a Danish smoking ban on hospital admissions for acute myocardial infarctions, found that a significant reduction in admissions was already occurring a year *before* the bans started. Thus, the conclusion that bans caused the admissions reductions may be oversimplified. The authors suggest that perhaps some of the decline in heart attack risk could have been caused by earlier improvements in diets or by gradual enactment of smoking bans. Whatever the explanation, checking when reductions began, rather than only whether post-intervention risks are smaller than pre-intervention risks, adds valuable insight to inform potential causal interpretations of the data.

6.2 Intervention Analysis Estimates Effects of Changes Occurring at Known Times, Enabling Retrospective Evaluation of the Effectiveness of Interventions

How much difference exposure reductions or other actions have made in reducing adverse health outcomes or producing other desired outcomes is often addressed using *intervention analysis*, also called *interrupted time series* analysis. The basic idea is to test whether the best description of an effects time series changes significantly when a risk factor or exposure changes, e.g., due to an intervention that increases or reduces it [47, 48, 68]. If the answer is yes, then the size of the changeover time provides quantitative estimates of the sizes and timing of changes in effects following an intervention. For example, an intervention analysis might test whether weekly counts of hospital admissions with a certain set of diagnostic codes or cardiovascular mortalities per person per year among people over 70 fell significantly when exposures fell due to closure of a plant that generated high levels of air pollution. If so, then comparing the best-fitting time series models (e.g., the maximum-likelihood models within a broad class of models, such as the autoregressive integrated moving average (ARIMA) models widely used in time series analysis) describing the data before and after the date of the intervention may help to quantify the size of the effect associated with the intervention. If not, then the interrupted time series does not provide evidence of a detectable effect of the intervention. Free software for intervention analysis is available in R (e.g., [80]; CausalImpact algorithm from Google 2015).

Two main methods of intervention analysis are *segmented regression*, which fits regression lines or curves to the effects time series before and after the intervention and then compares them to detect significant changes in slope or level, and *Box-Tiao analysis*, often called simply intervention analysis, which fits time series models (ARIMA or Box-Jenkins models with models of intervention effects, e.g., jumps in the level, changes in the slope, or ramp-ups or declines in the effects over time) to the effects data before and after the intervention and tests whether proposed effects of interventions are significantly different from zero. If so, the parameters of the intervention effect are estimated from the combined pre- and post-intervention data (e.g., [47, 48]). For effects time series that are stationary (meaning that the same statistical description of the time series holds over time) both before and after an intervention that changes exposure, but that have a jump in mean level due to the intervention, quantifying the difference in effects that the intervention has made can be as simple as estimating the difference in means for the effects time series before and after the intervention, similar to the CPA in Fig. 43.2, but for a known change point. The top panel of Fig. 43.5 shows a similar comparison for heart attack rates before and after a smoking ban. Based only on the lines shown, it appears that

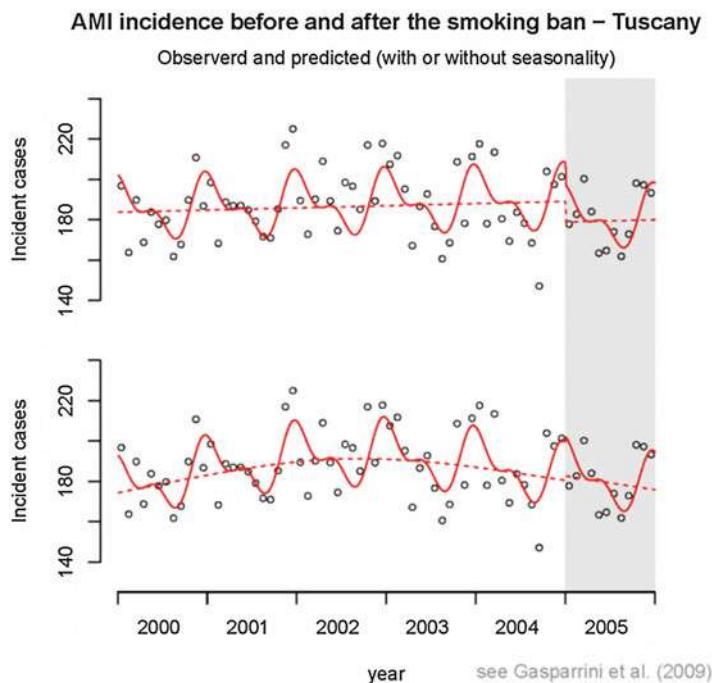


Fig. 43.5 Straight-line extrapolation of the historical trend for heart attack (AMI) rates over-predicts future AMI rates (upper panel) and creates the illusion that smoking bans were followed by reduced AMI rates, compared to more realistic nonlinear extrapolation (lower panel), which shows no detectable benefit from a smoking ban (Source: [33])

heart attack rates dropped following the ban. (If the effects of a change in exposure occur gradually, then distributed-lag models of the intervention's effects can be used to describe the post-intervention observations [47].) In nonstationary time series, however, the effect of the intervention may be obscured by other changes in the time series. Thus, the bottom panel of Fig. 43.5 considers nonlinear trends over time and shows that, in this analysis, any effect of the ban now appears to be negative (i.e., heart attack rates are increased after the ban compared to what is expected based on the nonlinear trend extrapolated from pre-ban data).

Intervention analyses, together with comparisons to time series for comparison populations not affected by the interventions, have been widely applied, with varying degrees of justification and success, to evaluate the impacts caused by changes in programs and policies in healthcare, social statistics, economics, and epidemiology. For example, Lu et al. [75] found that prior authorization policies introduced in Maine to help control the costs of antipsychotic drug treatments for Medicaid and Medicare Part D patients with bipolar disorder were associated with an unintended but dramatic (nearly one third) reduction in initiation of medication regimens among new bipolar patients, but produced no detectable switching of currently medicated patients toward less expensive treatments. Morriss et al. [84] found that the time series of suicides in a district population did not change significantly following a district-wide training program that measurably improved the skills, attitudes, and confidence of primary care, accident and emergency, and mental health workers who received the training. They concluded that “Brief educational interventions to improve the assessment and management of suicide for front-line health professionals in contact with suicidal patients may not be sufficient to reduce the population suicide rate.” [55] used intervention analysis to estimate that the introduction of pedestrian countdown timers in Detroit cut pedestrian crashes by about two thirds. Jiang et al. [59] applied intervention analysis to conclude that, in four Australian states, the introduction of randomized breath testing led to a substantial reduction in car accident fatalities. Callaghan et al. [10] used a variant of intervention analysis, regression-discontinuity analysis, to test whether the best-fitting regression model describing mortality rates among young people changed significantly at the minimum legal drinking age, which was 18 in some provinces and 19 in others. They found that mortality rates for young men jumped upward significantly precisely at the minimum legal drinking, which enabled them to quantify the impact of drinking-age laws on mortality rates. In these and many other applications, intervention analysis and comparison groups have been used to produce empirical evidence for what has worked and what has not and to quantify the sizes over time of effects attributed to interventions when these effects are significantly different from zero.

Intervention analysis has important limitations, however. Even if an intervention analysis shows that an effects time series changed when an intervention occurred, this does not show whether the intervention *caused* the change. Thus, in applications from air pollution bans to gun control, initial reports that policy interventions had significant beneficial effects were later refuted by findings that equal or greater beneficial changes occurred at the same time in comparison populations not affected

by the interventions [44, 64]. Also, more sophisticated methods such as transfer entropy, discussed later, must be used to test and estimate effects in nonstationary time series, since both segmented regression models and intervention analyses that assume stationarity typically produce spurious results for nonstationary time series. For example, as illustrated in Fig. 43.5, Gasparrini et al. [33] in Europe and Barr et al. [8] in the United States found that straight-line projections of what future heart attack (acute myocardial infarction, AMI) rates would have been in the absence of an intervention that banned smoking in public places led to a conclusion that smoking bans were associated with a significant reduction in AMI hospital admission rates following the bans. However, allowing for nonlinearity in the trend, which was significantly more consistent with the data, led to the reverse conclusion that the bans had no detectable impact on reducing AMI admission rates. As illustrated in Fig. 43.5, the reason is that fitting a straight line to historical data and using it to project future AMI rates in the absence of intervention tend to overestimate what those future AMI rates would have been, because the real time series is downward-curving, not straight. Thus, estimates of the effect of an intervention based on comparing observed to model-predicted AMI admission rates will falsely attribute a positive effect even to an intervention that had no effect if straight-line extrapolation is used to project what would have happened in the absence of an intervention, ignoring the downward curvature in the time series. This example illustrates how model specification errors can lead to false inferences about effects of interventions. The transfer entropy techniques discussed later avoid the need for curve-fitting and thereby the risks of such model specification errors.

6.3 Granger Causality Tests Show Whether Changes in Hypothesized Causes Help to Predict Subsequent Changes in Hypothesized Effects

Often, the hypothesized cause (e.g., exposure) and effect (e.g., disease rate) time series both undergo continual changes over time, instead of changing only once or occasionally. For example, pollution levels and hospital admission rates for respiratory or cardiovascular ailments change daily. In such settings of ongoing changes in both hypothesized cause and effect time series, *Granger causality tests* (and the closely related Granger-Sims tests for pairs of time series) address the question of whether the former helps to predict the latter. If not, then the exposure-response histories provide no evidence that exposure is a (Granger) cause of the effects time series, no matter how strong, consistent, etc., the association between their levels over time may be. More generally, a time series variable X is not a Granger cause of a time series variable Y if the future of Y is conditionally independent of the history of X (its past and present values), given the history of Y itself, so that future Y values can be predicted as well from the history of Y values as from the histories of both X and Y . If exposure is a Granger cause of health effects but health effects are not Granger causes of exposures, then this provides evidence that the exposure time series might indeed be a cause of the effects time

series. If exposure and effects are Granger causes of each other, then a confounder that causes both of them is likely to be present. The key idea of Granger causality testing is to provide formal quantitative statistical tests of whether the available data suffice to reject (at a stated level of significance) the null hypothesis that the future of the hypothesized effect time series can be predicted no better from the history of the hypothesized cause time series together with the history of the effect time series than it can be predicted from the history of the effect time series alone. Data that do not enable this null hypothesis to be rejected do not support the alternative hypothesis that the hypothesized cause helps to predict (i.e., is a Granger cause of) the hypothesized effect.

Granger causality tests can be applied to time series on different time scales to study effects of time-varying risk factors. For example, [76] identified a Granger-causal association between fatty diet and risk of heart disease decades later in aggregate (national level) data. Cox and Popken [16] found a statistically significant historical association, but no evidence of Granger causation, between ozone exposures and elderly mortality rates on a time scale of years. Granger causality testing software is freely available in R (e.g., <http://cran.r-project.org/web/packages/MSBVAR/MSBVAR.pdf>).

Originally Granger causality tests were restricted to stationary linear (autoregressive) time series models and to only two time series, a hypothesized cause and a hypothesized effect. However, recent advances have generalized them to multiple time series (e.g., using vector autoregressive (VAR) time series models) and to nonlinear time series models (e.g., using nonparametric versions of the test or parametric models that allow for multiplicative as well as additive interactions among the different time series variables) ([6, 7, 111, 122]; Diks and Wolski 2014). These advances are now being made available in statistical toolboxes for practitioners ([7]). For nonstationary time series, special techniques have been developed, such as vector error-correction (VECM) models fit to first differences of nonstationary variables or algorithms that search for co-integrated series (i.e., series whose weighted averages show zero mean drift). However, these techniques are typically quite sensitive to model specification errors [91]. Transfer entropy (TE) and its generalizations, discussed next, provides a more robust analytic framework for identifying causality from multiple nonstationary time series based on the flow of information among them.

7 Information Flows From Causes to Their Effects over Time: Transfer Entropy

Both Granger causality tests and conditional independence tests apply the principle that causes should be informative about their effects; more specifically, changes in direct causes provide information that helps to predict subsequent changes in effects. This information is not redundant with the information from other variables and cannot be explained away by knowledge of (i.e., by conditioning on the values of) other variables. A substantial generalization and improvement of this information-

based insight is that information flows over time from causes to their effects, but not in the reverse direction. Thus, instead of just testing whether past and present exposures provide information about (and hence help to predict) future health effects, it is possible to quantify the rate at which information, measured in bits, flows from the past and present values of the exposure time series to the future values of the effects time series. This is the key concept of *transfer entropy* (TE) [81, 91, 99, 118]). It provides a nonparametric, or model-free, way to detect and quantify rates of information flow among multiple variables and hence to infer causal relations among them based on the flow of information from changes in causal variables (“drivers”) to subsequent changes in the effect variables that they cause (“responses”). If there is no such information flow, then there is no evidence of causality.

Transfer entropy (TE) is model-free in that it examines the empirically estimated conditional probabilities of values for one time series, given previous values of others, without requiring any parametric models describing the various time series. Like Granger causality, TE was originally developed for only two time series, a possible cause and a possible effect, but has subsequently been generalized to multiple time series with information flowing among them over time (e.g., [81, 91, 99]). In the special case where the exposure and response time series can be described by linear autoregressive (AR) processes with multivariate normal error terms, tests for TE flowing from exposure to response are equivalent to Granger causality tests (Barnett et al. 2009), and Granger tests, in turn, are equivalent to conditional independence tests for whether the future of the response series is conditionally independent of the history of the exposure series, given the history of the response series. Free software packages for computing the TE between or among multiple time series variables are now available for MATLAB [81] and other software (<http://code.google.com/p/transfer-entropy-toolbox/downloads/list>). Although transfer entropy and closely related information-theoretic quantities have been developed and applied primarily within physics and neuroscience to quantify flows of information and appropriately defined causal influences [58] among time series variables, they are likely to become more widely applied in epidemiology as their many advantages become more widely recognized.

8 Changes in Causes Make Future Effects Different From What They Otherwise Would Have Been: Potential-Outcome and Counterfactual Analyses

The insight that changes in causes produce changes in their effects, making the probability distributions for effect variables different from what they otherwise would have been, has contributed to a well-developed field of *counterfactual (potential-outcome) causal modeling* [51]. A common analytic technique in this field is to treat the unobserved outcomes that would have occurred had causes (e.g., exposures or treatments) been different as missing data and then to apply missing-data methods for regression models to estimate the average difference

in outcomes for individuals receiving different treatments or other causes. The estimated difference in responses for treated compared to untreated individuals, for example, can be defined as a measure of the impact caused by treatment at the population level.

To accomplish such counterfactual estimation in situations where randomized assignments of treatments or exposures to individuals is not possible, counterfactual models and methods such as *propensity score matching* (PSM) and *marginal structural models* (MSMs) [96] construct weighted samples that attempt to make the estimated distribution of measured confounders the same as it would have been in a randomized control trial. If this attempt is successful, and if the individuals receiving different treatments or exposures are otherwise statistically identical (more precisely, *exchangeable*), then any significant differences between the responses of subpopulations receiving different treatments or exposures (or other combinations of causes) might be attributed to the differences in these causes, rather than to differences in the distributions of measured confounders [18, 96]. However, this attribution is valid only if the individuals receiving different treatments or exposures are exchangeable – a crucial assumption that is typically neither tested nor easily testable. If treated and untreated individuals differ on unmeasured confounders, for example, then counterfactual methods such as PSM or MSM may produce mistaken estimates of causal impacts of treatment or exposure. Differing propensities to seek or avoid treatment or exposure based in part on unmeasured differences in individual health status could create biased estimates of the impacts of treatment or exposure on subsequent health risks. In general, counterfactual methods for estimating causal impacts of exposures or treatments on health risks make assumptions that imply that estimated differences in health risks between different exposure or treatment groups are caused by differences in the exposures or treatments. The validity of these assumptions is usually unproved. In effect, counterfactual methods assume (rather than establishing) the key conclusion that differences in health risks are caused by differences in treatments or exposures, rather than by differences in unmeasured confounders or by other violations of the counterfactual modeling assumptions.

In marginal structural models (MSMs), the most commonly used sample-weighting techniques (called inverse probability weighting (IPW), as well as refined versions that seek to stabilize the variance of the weights) can be applied at multiple time points to populations for which exposures or treatments, confounders, and individuals entering or leaving the populations are all time-varying. This flexibility, together with emphasis on counterfactuals and missing observations, make MSMs particularly well suited to the analysis of time-varying confounders and effects of treatments or interventions that involve feedback loops, such as when the treatment that a patient receives depends on his or her responses so far and also to analysis of data in which imperfect compliance, attrition from the sample, or other practical difficulties drive a wedge between what was intended and what actually occurred in the treatment and follow-up of patients [96]. For example, MSMs are often applied to *intent-to-treat* data, in which the intent or plan to treat patients in a certain way is taken as the controllable causal driver of outcomes, and what happens next may depend in part on real-world uncertainties.

Despite their advantages in being able in principle to quantify causal impacts in complex time-varying data sets, MSMs have some strong practical limitations. Their results are typically very sensitive to errors in the specification of the regression models used to estimate unobserved counterfactual values, and the correct model specification is usually unknown. Therefore, MSMs are increasingly being used in conjunction with *model ensemble* techniques to address model uncertainty. Model ensemble methods (including Bayesian model averaging, various forms of statistical boosting, k-fold cross-validation techniques, and super-learning, as described next) calculate results using many different models and then combine the results. The use of diverse plausible models avoids the false certainty and potential biases created by selecting a single model. For example, in *super-learning algorithms*, no single regression model is selected. Instead multiple different standard machine-learning algorithms (e.g., logistic regression, random forest, support vector machine, naïve Bayesian classifier, artificial neural network, etc.) are used to predict unobserved values [86] and to estimate IPW weights [37]. These diverse predictions are then combined via weighted averaging, where the weights reflect how well each algorithm predicts known values that have been deliberately excluded (held out for test purposes) from the data supplied to the algorithms – the computational statistical technique known as model cross validation. Applied to the practical problem of estimating the mortality hazard ratio for initiation versus no initiation of combined antiretroviral therapy among HIV-positive subjects, such ensemble learning algorithms produced clearer effects estimates (hazard ratios further below 1, indicating a beneficial effect of the therapy) and narrower confidence intervals than traditional single-model (logistic regression modeling) analysis (*ibid*).

Yet even these advances do not overcome the fact that MSMs requires strong, and often unverifiable, assumptions to yield valid estimates of causal impacts. Typical examples of such assumptions are that there are no unmeasured confounders, that the observed response of each individual (e.g., of each patient to treatment or nontreatment) is in fact caused by the observed risk factors, and that every value for the causal variables occurs for every combination of levels of the confounding variables (e.g., there is no time period before exposures began but confounders were present) [18]. Assuming that these conditions hold may lead to an unwarranted inference that a certain exposure causes an adverse health effect, e.g., that ozone air pollution causes increased asthma-related hospitalizations, even if analyses based only on realistic, empirically verifiable assumptions would reveal no such causal relation [82].

Counterfactual models are often used to assess the effects on health outcomes of medical treatments, environmental exposures, or preventable risk factors by comparing what happened to people who receive the treatments to what models predict would have happened without the treatments. However, a limitation of such counterfactual comparisons is that they are seldom explicit about *why* treatments would not have occurred in the counterfactual world envisioned. Yet, the answer can crucially affect the comparison being drawn [35, 46]. For example, if it is assumed that a patient would not have received a treatment because the physician is confident that it would not have worked and the patient would have died anyway,

then the estimated effect of the treatment on mortality rates might be very different from what it would be if it is assumed that the patient would not have received the treatment because the physician is confident that there is no need for it and that the patient would recover anyway. In practice, counterfactual comparisons usually do not specify in detail the causal mechanisms behind the counterfactual assumptions about treatments or exposures, and this can obscure the precise interpretation of any comparison between what did happen and what is supposed, counterfactually, would have happened had treatments (or exposure) been different. Standard approaches estimate effects under the assumption that those who are treated or exposed are exchangeable with those who are not within strata of adjustment factors that may affect (but are not affected by) the treatment or the outcome, but the validity of this assumption is usually difficult or impossible to prove.

An alternative, increasingly popular, approach to using untested assumptions to justify causal conclusions is the *instrumental variable* (IV) method, originally developed in econometrics [112]. In this approach, an *instrument* is defined as a variable that is associated with the treatment or exposure of interest – a condition that is usually easily verifiable – and that also affects outcomes (e.g., adverse health effects) only through the treatment or exposure variable, without sharing any causes with the outcome. (In DAG notation, such an instrument would be a variable with arrows directed only into Y and Z in the DAG model $X \rightarrow Y \rightarrow Z$, where Z is the outcome variable, Y the treatment or exposure variable, and X a variable that affects exposure, such as job category, residential location, or intent-to-treat.) These latter conditions are typically assumed in IV analyses, but not tested or verified. If they hold, then the effects of unmeasured confounders on the estimated association between Y and Z can be eliminated using observed values of the instrument, and this is the potential great advantage of IV analysis. However, in practice, IV methods applied in epidemiology are usually dangerously misleading, as even minor violations of their untested assumptions can lead to substantial errors and biases in estimates of the effects of different exposures or treatments on outcomes; thus, many methodologists consider it inadvisable to use IV methods to support causal conclusions in epidemiology, despite their wide and increasing popularity for this purpose [112]. Unfortunately, within important application domains in epidemiology, including air pollution health effects research, leading investigators sometimes draw strong but unwarranted causal conclusions using IV or counterfactual (PSM or MSM) methods and then present these dubious causal conclusions and effects estimates to policy-makers and the public as if they were known to be almost certainly correct, rather than depending crucially on untested assumptions of unknown validity (e.g., [104]). Such practices lead to important-seeming journal articles and policy recommendations that are untrustworthy, potentially reflecting the ideological biases or personal convictions of the investigator rather than true discoveries about real-world causal impacts of exposures [103]. Other scientists and policy makers are well advised to remain on guard against such enthusiastic claims about causal impacts and effects estimates promoted by practitioners of IV and counterfactual methods who do not appropriately caveat their conclusions by emphasizing their dependence on untested modeling assumptions.

When the required assumptions for counterfactual modeling cannot be confidently determined to hold, other options are needed for counterfactual analyses to proceed. The simplest and most compelling approach is to use genuine randomized control trials (RCTs), if circumstances permit it. They rarely do, but the exceptions can be very valuable. For example, the state of Oregon in 2008 used a lottery system to expand limited Medicaid coverage for the uninsured by randomly selecting names from a waiting list. Comparing subsequent emergency department use among the randomly selected new Medicaid recipients to subsequent use by those still on the waiting list who had not yet received Medicaid revealed a 40% increase in emergency department usage over the next 18 months among the new Medicaid recipients, including visits for conditions that might better have been treated in primary care physician settings. Because the selection of recipients was random, this increase in usage could be confidently attributed to a causal effect of the Medicaid coverage on increasing emergency department use [114]. The main limitation of such RCTs is not in establishing the existence and magnitude of genuine causal impacts of an intervention in the studied population but rather in determining to what extent the result can be generalized to other populations. While conclusions based on valid causal laws and mechanisms can be transported across contexts, as discussed later, this is not necessarily true of aggregate population-level causal impacts, which may depend on specific circumstances of the studied population.

In the more usual case where random assignment is not an option, use of non-randomized control groups can still be very informative for testing, and potentially refuting, assumptions about causation. Indeed, analyses that estimate the impacts of changes in exposures by comparing population responses before and after an intervention that changes exposure levels can easily be misled unless appropriate comparison groups are used. For example, a study that found a significant drop in mortality rates from the six years prior to a coal burning ban in Dublin county, Ireland, to the six years following the ban concluded that the ban had caused a prompt, significant fall in all-cause and cardiovascular mortality rates [42]. This finding eventually led officials to extend the bans to protect human health. However, such a pre-post comparison study design cannot support a logically valid inference of causality, since it pays no attention to what would have happened to mortality rates in the absence of an intervention, i.e., the coal-burning ban. When changes in all-cause and cardiovascular mortality rates outside the ban area were later compared to those in areas affected by the ban, it turned out that there was no detectable difference between them: contrary to the initial causal inference, the bans appeared to have had no detectable impact on reducing these rates [44]. Instead, the bans took place during a decades-long period over which mortality rates were decreasing, with or without bans, throughout much of Europe and other parts of the developed world, largely due to improvements in early detection, prevention, and treatment of cardiovascular risks. In short, what would have happened in the absence of an intervention can sometimes be revealed by studying what actually did happen in appropriate comparison or control groups – a key idea developed and applied in the field of quasi-experimental (QE) studies, discussed next. Counterfactual causal inferences drawn without such comparisons can easily be misled.

9 Valid Causal Relations Cannot Be Explained Away by Noncausal Explanations

An older, but still useful, approach to causal inference from observational data, developed largely in the 1960s and 1970s, consists of showing that there is an association between exposure and response that cannot plausibly be explained by confounding, biases (including model and data selection biases and specification errors), or coincidence (e.g., from historical trends in exposure and response that move together but that do not reflect causation). *Quasi-experiment* (QE) design and analysis approaches originally developed in social statistics [12] systematically enumerate potential alternative explanations for observed associations (e.g., coincident historical trends, regression to the mean, population selection, and response biases) and provide statistical tests for refuting them with data, if they can be refuted. The interrupted time series analysis studies discussed earlier are examples of quasi-experiments: they do not allow random assignment of individuals to exposed and unexposed populations but do allow comparisons of what happened in different populations before and after an intervention that affects some of the populations but not others (the comparison groups).

A substantial tradition of *refutationist approaches* in epidemiology follows the same general idea of providing evidence for causation by using data to explicitly test, and if possible refute, other explanations for exposure-response associations [77]. As stated by Samet and Bodurow [100], “Because a statistical association between exposure and disease does not prove causation, plausible alternative hypotheses must be eliminated by careful statistical adjustment and/or consideration of all relevant scientific knowledge. Epidemiologic studies that show an association after such adjustment, for example through multiple regression or instrumental variable estimation, and that are reasonably free of bias and further confounding, provide evidence but not proof of causation.” This is overly optimistic, insofar as associations that are reasonably free of bias and confounding do not necessarily provide evidence of causation. For example, strong, statistically significant associations (according to usual tests, e.g., t-tests) typically occur in regression models in which the explanatory and dependent variables undergo statistically independent random walks. The resulting associations do not arise from confounding or bias but from spurious regression, i.e., coincident historical trends created by random processes that are not well described by the assumptions of the regression models. Nonetheless, the recommendation that “plausible alternative hypotheses must be eliminated by careful statistical adjustment and/or consideration of all relevant scientific knowledge” well expresses the refutationist point of view.

10 Changes in Causes Produce Changes in Effects via Networks of Causal Mechanisms

Perhaps the most useful and compelling valid evidence of causation, with the possible exception of differences in effects between treatment and control groups in well-conducted randomized control trials, consists of showing that changes in

exposures propagate through a network of validated lawlike structural equations or mechanisms to produce predictable changes in responses. For example, showing that measured changes in occupational exposures to a workplace chemical consistently produce a sequence of corresponding changes in lung inflammation markers, recruitment rates of activated alveolar macrophages and activated neutrophils to the chronically inflamed lung, levels of tissue-degrading enzymes released by these cell populations, and resulting rates of lung tissue destruction and scarring, leading to onset of lung pathologies and clinically detectable lung diseases (such as emphysema, silicosis, fibrosis, or inflammation-mediated lung cancer), would provide compelling evidence of a causal relation between changes in exposures and changes in those disease rates. Observing the network of mechanisms by which changes in exposures are transduced to changes in disease risks provides knowledge-based evidence of causation that cannot be obtained from any purely statistical analysis of observational data on exposures and responses alone.

Several causal modeling techniques are available to describe the propagation of changes through networks of causal mechanisms. *Structural equation models (SEMs)*, in which changes in right-hand side variables cause adjustments of left-hand side variables to restore all equalities in a system of structural equations, as in Fig. 43.1, provide one way to describe causal mechanisms for situations where the precise time course of the adjustment process is not of interest. Differential equation models, in which flows among compartments change the values of variables representing compartment contents over time (which in turn may affect the rates of flows), eventually leading to new equilibrium levels following an exogenous intervention that changes the compartment content or flow rates, provide a complementary way to describe mechanisms when the time course of adjustment is of interest. Simulation models provide still another way to describe and model the propagation of changes through causal networks. Figure 43.6 illustrates the structure of a simulation model for cardiovascular disease (CVD) outcomes. At each time step, the value of each variable is updated based on the values of the variables that point into it. The time courses of all variables in the model can be simulated for any history of initial conditions and exogenous changes in the input variables (those with only outward-pointing arrows), given the computational models that determine the change in the value of each variable at each time step from the values of its parents in the DAG.

10.1 Structural Equation and Path Analysis Models Model Linear Effects Among Variables

For most of the past century, DAG models such as those in Figs. 43.1 and 43.6, in which arrows point from some variables into others and there are no directed cycles, have been used to explicate causal networks of mechanisms and to provide formal tests for their hypothesized causal structures. For example, *path analysis* methods showing the dependency relations among variables in SEMs have been used for many decades to show how some variables influence others when all relations are assumed to be linear. Figure 43.7 presents an example involving several

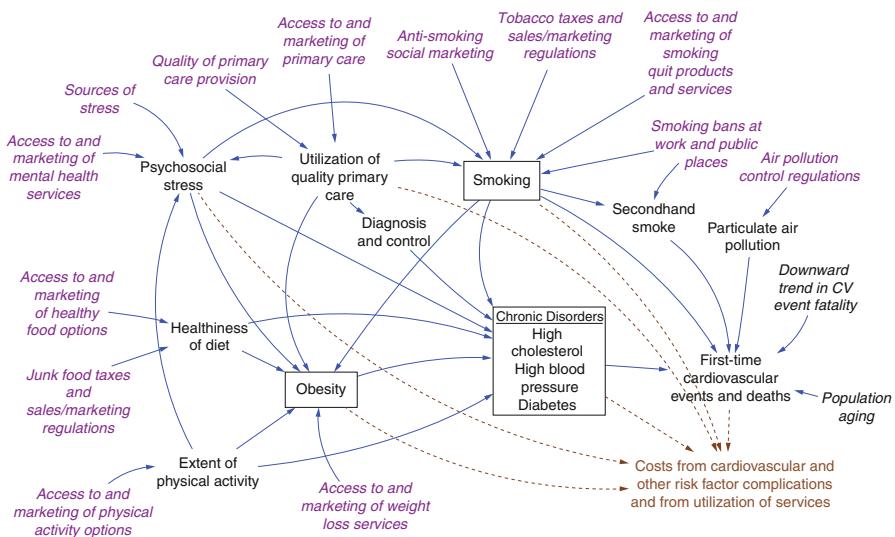


Fig. 43.6 Simulation model for major health conditions related to cardiovascular disease (CVD) and their causes. Boxes represent risk factor prevalence rates modeled as dynamic stocks. Population flows among these stocks – including people entering the adult population, entering the next age category, immigration, risk factor incidence, recovery, cardiovascular event survival, and death – are not shown (Source: [52])

Key: Blue solid arrows: causal linkages affecting risk factors and cardiovascular events and deaths. Brown dashed arrows: influences on costs.

Purple italics: factors amenable to direct intervention.

Black italics (population aging, cardiovascular event fatality): other specified trends.

Black nonitalics: all other variables, affected by italicized variables and by each other

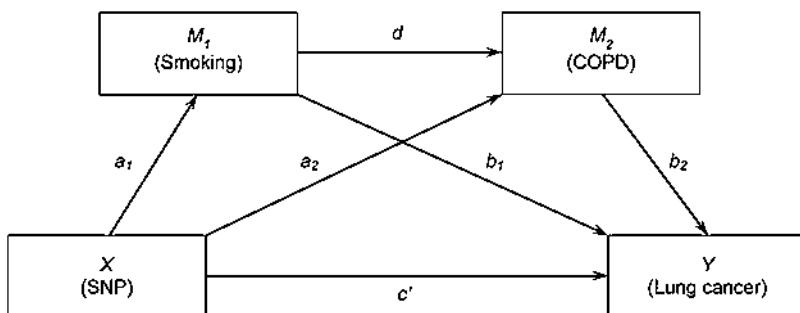


Fig. 43.7 A path diagram with standardized coefficients showing linear effects of some variables on others (Source: [120])

variables that are estimated to significantly predict lung cancer risk: the presence of a particular single nucleotide polymorphism (SNP) (the CHRNA5-A3 gene cluster, a genetic variant which is associated with increased risk of lung cancer), smoking, and presence of chronic obstructive pulmonary disease (COPD) [120].

The path coefficient on an arrow indicates by how much (specifically, by how many standard deviations) the expected value of the variable into which it points would change if the variable at the arrow's tail were increased by one standard deviation, holding all other variables fixed and assuming that all relations are well approximated by linear structural equation regression models, i.e., that changing the variable at the arrow's tail will cause a proportional change in the variable at its head. In this example, the path coefficients are denoted by a_1 , a_2 , b_1 , b_2 , c' , and d . These numbers must be estimated from data to complete the quantification of the path diagram model. Although such path analysis models are derived from correlations, the causal interpretation (i.e., that changing a variable at the tail of an arrow will change the variable at its head in proportion to the coefficient on the arrow between them) is an assumption. It is justified only if the regression equations used are indeed structural (causal) equations and if the assumptions required for multiple linear regression (e.g., additive effects, constant variance, normally distributed errors) hold. For the path diagram in Fig. 43.7, the authors found that the gene variant, X , affected lung cancer risk, Y , by increasing smoking behavior and, separately, by increasing COPD risk, as well as by increasing smoking-associated COPD risk: "The results showed that the genetic variant influences lung cancer risk indirectly through all three different pathways. The percent of genetic association mediated was 18.3% through smoking alone, 30.2% through COPD alone, and 20.6% through the path including both smoking and COPD, and the total genetic variant-lung cancer association explained by the two mediators was 69.1%."

Path diagrams reflect the fact that, if all effects of variables on each other are well approximated by linear regression SEMs, then correlations between variables should be stronger between variables that are closer to each other along a causal chain than between variables that are more remote, i.e., that have more intervening variables. Specifically, the effect of a change in the variable at the start of a path on a variable at the end of it that is transmitted along that path is given by the product of the path coefficients along the path. Thus, in Fig. 43.7, the presence of the SNP should be more strongly correlated with COPD than with COPD-associated lung cancer. Moreover, the effect of a change in an ancestor variable on the value of a remote descendent (several nodes away along one or more causal paths) can be decomposed into the effects of the change in the ancestor variable on any intermediate variables and the effects of those changes in intermediate variables, in turn, on the remote descendent variable. If one variable does *not* point into another, then the SEM/path analysis model implies that the first is not a direct cause of the second. For example, the DAG model $X \rightarrow Y \rightarrow Z$ implies that X is an ancestor (indirect cause) but not a parent (direct cause) of Z . An implication of the causal ordering in this simple DAG model can be tested, as previously noted, by checking whether Z is conditionally independent of X given Y .

In linear SEM/path analysis models, conditional independence tests specialize to testing whether partial correlation coefficients between two variables become zero after conditioning on the values of one or more other variables (e.g., the partial correlation between X and Z , holding Y fixed, would be zero for the path $X \rightarrow Y \rightarrow Z$). This makes information-theoretic methods unnecessary when the

assumptions of linear SEMs and jointly normally distributed error terms relating the value of each variable to the values of its parents hold; analyses based on correlations can then be used instead. Such consistency and coherence constraints can be expressed as systems of equations that can be solved, when identifiability conditions hold, to estimate the path coefficients (including confidence intervals) relating changes in parent variables to changes in their children. Summing these changes over all paths leading from exposure to response variables allows the total effect (via all paths) of a change in exposure on changes in expected responses to be estimated or predicted.

Path analysis and other SEM models are particularly valuable for detecting and quantifying the effects of unmeasured (“latent”) confounders based on the patterns of correlations that they induce among observed variables. SEM modeling methods have also been extended to include quadratic terms, categorical variables, and interaction terms [66]. Standard statistics packages and procedures, such as PROC CALIS in SAS, have made this technology available to modelers for the past four decades, and free modern implementations are readily available (e.g., in the R packages SEM or RAMpath).

10.2 Bayesian Networks Show How Multiple Interacting Factors Affect Outcome Probabilities

Path analysis, which is now nearly a century old, provided the main technology for causal modeling for much of the twentieth century. More recently, DAG models have been generalized so that causal mechanisms need not be described by linear equations for expected values, but may instead be described by arbitrary conditional probability relations. The nodes in such a graph typically represent random variables, stochastic processes, or time series variables in which a decision-maker may intervene from time to time by taking actions that cause changes in some of the time series [4, 23].

Bayesian networks (BNs) are a prominent type of DAG model in which nodes represent constants, random variables, or deterministic functions [3]. Figure 43.8 shows an example of a BN for cardiovascular disease (CVD). As usual, the absence of an arrow between two nodes, such as between *ethnic group* and *CVD* in Fig. 43.8, implies that neither has a probability distribution that depends directly on the value of the other. Thus, *ethnic group* is associated with CVD, but the association is explained by *smoking* as a mediating variable, and the structure of the DAG shows no further dependence of *CVD* on *ethnic group*. Statistically, the random variable indicating cardiovascular disease, *CVD*, is conditionally independent of *ethnic group*, given the value of *smoking*.

Some useful causal implications may be revealed by the structure of a DAG, even before the model is quantified to create a fully specified BN (or other DAG) model. For example, if the DAG structure in Fig. 43.8 correctly describes an individual or population, then elevated systolic BP (blood pressure) is associated with CVD risk, since both have age and ethnicity as ancestors in the DAG. However, changes in statin use, which could affect systolic BP via the intermediate variable

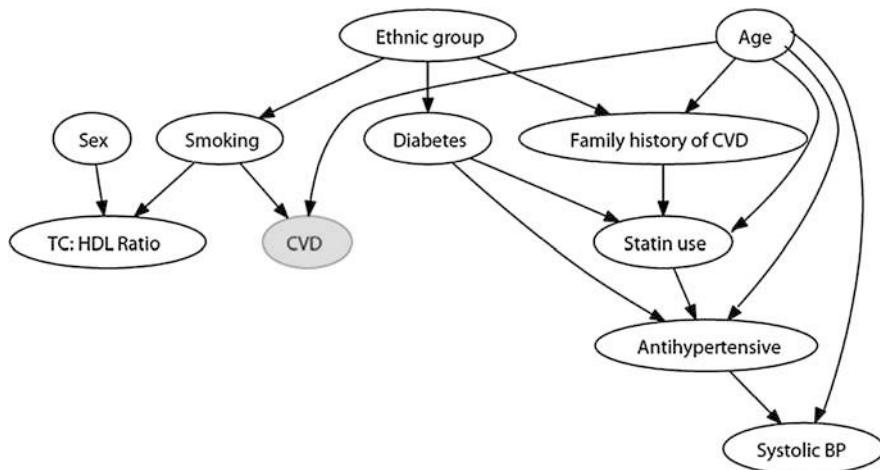


Fig. 43.8 Directed acyclic graph (DAG) structure of a Bayesian network (BN) model for cardiovascular disease (CVD) risk (Source: [115])

antihypertensive, would not be expected to have any effect on CVD risk. Learning the correct DAG structure of causal relations among variables from data – a key part of the difficult task of *causal discovery*, discussed later – can reveal important and unexpected findings about what works and what does not for changing the probabilities of different outcomes, such as CVD in Fig. 43.8. However, uncertainty about the correct causal structure can make sound inference about causal impacts (and hence recommendations about the best choice of actions to produce desired changes) difficult to determine. Such model uncertainty motivates the use of ensemble modeling methods, discussed later.

10.3 Quantifying Probabilistic Dependencies Among BN Variables

For quantitative modeling of probabilistic relations among variables, input nodes in a BN (i.e., nodes with only outward-pointing arrows, such as sex, age, and ethnic group in Fig. 43.8) are assigned marginal (unconditional) probability distributions for the values of the variables they represent. These marginal distributions can be thought of as being stored at the input nodes, e.g., in tables that list the probability or relative frequency of each possible input value (such as male or female for sex, age in years, etc.). They represent the prior probabilities that each input node will have each of its possible values for a randomly selected case or individual described by the BN model, before getting more information about a specific case or individual. For any specific individual to whom the BN model is applied, if the values of inputs such as sex, age, and ethnicity are known, then their values would be specified as inputs and conditioned on at subsequent nodes in applying the model to that individual. Figure 43.9 illustrates this concept.

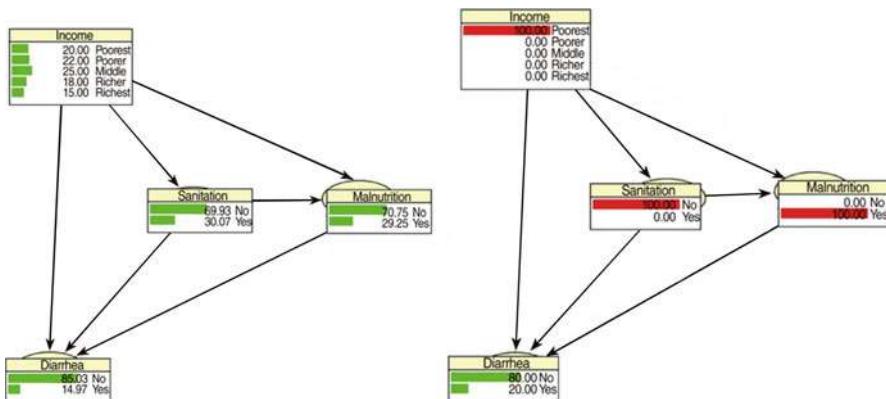


Fig. 43.9 BN model of risk of infectious diarrhea among children under 5 in Cameroon. The left panel shows the unconditional risk for a random child from the population (14.97%); the right panel shows the conditional risk for a malnourished child from a home in the lowest income quintile and with poor sanitation (20.00%) (Source: [87])

The left panel shows a BN model for risk of infectious diarrhea among young children in Cameroon. Each of three risk factors – income quintile, availability of toilets (“sanitation”), and stunted growth (“malnutrition”) – affects the probability that a child will ever have had diarrhea for at least two weeks (“diarrhea”). In addition, these risk factors affect each other, with an observation of low income making observations of poor sanitation and malnutrition more likely and observed poor sanitation also making observed malnutrition more likely at each level of income. The right panel shows an instance of the model for a particular case of a malnourished child from the poorest income quintile living with poor sanitation; these three risk factor values all have probabilities set to 100%, since their values are known. The result is that the risk of diarrhea, conditioned on this information, is increased from an average value of 14.97% in this population of children to 20% when all three risk factors are set to these values.

A BN model stores *conditional probability tables* (CPTs) at nodes with inward-pointing arrow. A CPT simply tabulates the conditional probabilities for the values of the node variable for each combination of values of the variables that point into it (its “parents” in the DAG). For example, the *malnutrition* node in Fig. 43.9 has a CPT with 10 rows, since its two parents, *Income* and *Sanitation*, have 5 and 2 possible levels, respectively, implying ten possible pairs of input values. For each of these ten combinations of input values, the CPT shows the conditional probability for each possible value of *Malnutrition* (here, just *Yes* and *No*, so that the CPT for *Malnutrition* has 2 columns); these conditional probabilities must sum to 1 for each row of the CPT. BN nodes can also represent deterministic functions by CPTs that assign 100% probability to a specific output value for each combination of input values. The conditional probability distribution for the value of a node (i.e., variable) thus depends on the values of the variables that point into it; it can be freely specified (or estimated from data, if adequate data are available) without making any

restrictive assumptions about linearity or normal errors. Such BN models greatly extend the flexibility of practical causal hypothesis testing and causal predictive modeling beyond traditional linear SEM and path analysis models.

In practice, CPTs can usually be condensed into relatively small tables by using classification trees or other algorithms (e.g., rough sets) to bin the potentially large number of combinations of values for a node's parents into just those that predict significantly different conditional probability distributions for the node's value. Instead of enumerating all the combinations of values for the parents, “don't care” conditions (represented by blanks in the CPT entries or by missing splits in a classification tree) can reduce the number of combinations that must be explicitly stored in the CPT. Alternatively, a logistic regression model or other statistical model can be used in place of a CPT at each node. For example, although the *Diarrhea* node in Fig. 43.9 could logically have a CPT with $5 \times 2 \times 2 = 20$ rows, it may be that a simple regression model with only three coefficients for the main effects of the parents, and few or no additional terms for interactions, would adequately approximate the full CPT.

10.4 Causal vs. Noncausal BNs

Any joint probability distribution of multiple random variables can be factored into a product of marginal and conditional probability distributions and displayed in DAG form, usually in several different ways. For example, the joint probability mass function $P(x, y)$ of two discrete random variables X and Y , specifying the probability of each pair of specific values (x, y) for random variables X and Y , can be factored as $P(x)P(y|x)$ or as $P(y)P(x|y)$ and can be displayed in a BN as $X \rightarrow Y$ or as $Y \rightarrow X$, respectively. Here, x and y denote possible values of random variables X and Y , respectively; $P(x, y)$ denotes the joint probability that $X = x$ and $Y = y$; $P(x)$ denotes the marginal probability that $X = x$; $P(y)$ denotes the marginal probability that $Y = y$, and $P(y|x)$ and $P(x|y)$ denote conditional probabilities that $Y = y$ given that $X = x$ and that $X = x$ given that $Y = y$, respectively. Thus, there is nothing inherently causal about a BN. Its nodes need not represent causal mechanisms that map values of inputs to probabilities for the values of outputs caused by those inputs. Even if they do represent such causal mechanisms, they may not explicate how or why the mechanisms work. For example, the direct link from *Income* to *Malnutrition* in Fig. 43.9 gives no insight into how or why changes in income affect changes in malnutrition – e.g., what specific decisions or behaviors are influenced by income that, in turn, results in better or worse nutrition. Thus, it is possible to build and use BNs for probabilistic inference without seeking any causal interpretation of the statistical dependencies among its variables.

However, BNs are often deliberately constructed and interpreted to mean that changes in the value of a variable at the tail of an arrow will cause a change in the probability distribution of the variable into which it points, as described by the CPT at that node. The effect of a change in a parent variable on the probability

distribution of a child variable into which it points may depend on the values of other parents of that node, thus allowing interactions among direct causes at that node to be modeled. For example, in Fig. 43.8, the effects of smoking on CVD risk may be different at different ages, and this would be indicated in the CPT for the CVD node by having different probabilities for the values of the CVD variable at different ages for the same value of the smoking variable. A *causal BN* is a BN in which the nodes represent stable causal mechanisms or laws that predict how changes in input values change the probability distribution of output values. The CPT at a node of a causal BN describes the conditional probability distribution for its value caused by each combination of values of its inputs, meaning that changes in one or more of its input values will be followed by corresponding changes in the probability distribution for the node's value, as specified by the CPT. This is similar to the concept of a causal mechanism in structural equation models, where a change in a right-hand side (explanatory or independent) variable in a structural equation is followed by a change in the left-hand side (dependent or response variable) to restore equality [119].

A causal BN allows changes at input nodes to be propagated throughout the rest of the network, yielding a posterior joint probability distribution for the values of all variables. (If the detailed time course of changes in probabilities is of interest, then differential equations or dynamic Bayesian networks (DBNs), discussed later, may be used to model how the node's probability distribution of values changes from period to period.) The order in which changes propagate through a network provides insight into the (total or partial) causal ordering of variables and can be used to help deduce network structures from time series data [119]. Similarly, in a set of simultaneous linear structural equations describing how equilibrium levels of variables in a system are related, the causal ordering of variables (called the *Simon causal ordering* in econometrics) is revealed by the order in which the equations must be solved to determine the values of all the variables, beginning with exogenous inputs (and assuming that the system of equations can be solved uniquely, i.e., that the values of all variables are uniquely identifiable from the data). Causality flows from exogenous to endogenous variables and among endogenous variables in such SEMs (*ibid*). Exactly how the changes in output probabilities (or in the expected values of left-side variables in an SEM) caused by changes in inputs are to be interpreted (e.g., as changes in the probability distribution of future observed values for a single individual or as changes in the frequency distribution of the variable values in a population of individuals described by the BN) depends on the situation being modeled.

10.5 Causal Mechanisms Are Lawlike, Yielding the Same Output Probabilities for the Same Inputs

A true causal mechanism that has been explicated in enough detail to make reliable predictions can be modeled as a conditional probability table (CPT) that gives the same conditional probabilities of output values whenever the input values are the same. Such a stable, repeatable relation, which might be described as lawlike,

can be applied across multiple contexts as long as the inputs to the node are sufficient to determine (approximately) unique probabilities for its output values. For example, a dose-response relation between radiation exposure and excess age-specific probability (or, more accurately, hazard rate) for first diagnosis with a specific type of leukemia might be estimated from data for one population and then applied to another with similar exposures, provided that the change in risk caused by exposure does not depend on omitted factors. If it depends on age and ethnic group, for example, then these would have to be included, along with exposure, as inputs to the node representing leukemia status. By contrast, unexplained heterogeneity, in which the estimated CPT differs significantly when study designs are repeated by different investigators, signals that a lawlike causal mechanism has not yet been discovered. In that case, the models and the knowledge that the BN represents need to be further refined to discover and express predictively useful causal relations that can be applied to new conditions. The key idea is that, to be transferable across contexts (e.g., populations), the probabilistic relations encoded in CPTs must include all of the input conditions that suffice to make their conditional probabilities accurate, given accurately measured or estimated input values.

A proposed causal relation that turns out to be very heterogeneous, sometimes showing significant positive effects and other times no effects or significant negative effects under the same conditions, does not correspond to a lawlike causal relation and cannot be relied on to make valid causal predictions (e.g., by using mean values averaged over many heterogeneous studies). Thus, the estimated CPTs at nodes in Fig. 43.9 may be viewed as averages of many individual-specific CPTs, and the predictions that they make for any individual case may not be accurate. CPTs that simply summarize historical data on conditional frequency distributions, but that do not represent causal mechanisms, may be no more than mixtures of multiple CPTs for the (perhaps unknown) populations and conditions that contributed to the historical data. They cannot necessarily be generalized to new populations or conditions (sometimes described as being transported to new contexts) or used to predict how outputs will change in response to changes in inputs, unless the relevant mixtures are known. For example, suppose that the *Sanitation* node has a value of 1 for children from homes with toilets and a value of 0 otherwise. If homes may have toilets either because the owners bought them or because a government program supplied them as part of a program along with a child food and medicine program, then the effect of a finding that *Sanitation* = 1 on the conditional probability distribution of *Malnutrition* may depend very much on which of these reasons resulted in *Sanitation* = 1. But this is not revealed by the model in Fig. 43.9. In such a case, the estimated CPTs for the nodes in Fig. 43.9 should not be interpreted as describing causal mechanisms, and the effects on other variables of setting *Sanitation* = 1 by alternative methods cannot be predicted from the model in Fig. 43.9.

10.6 Posterior Inference in BN Models

Once a BN has been quantified by specifying its DAG structure and the probability tables at its nodes, it can be used to draw a variety of useful inferences by applying

any of several well-developed algorithms created and refined over the past three decades [3]. The most essential inference capability of a BN model is that if observations (or “findings”) about the values at some nodes are entered, then the conditional probability distributions of all other nodes can be computed, conditioned on the evidence provided by these observed values. This is called “posterior inference.” In other words, the BN provides computationally practicable algorithms for accomplishing the Bayesian operation of updating prior knowledge or beliefs, represented in the node probability tables, with observations to obtain posterior probabilities. For example, if known values of a patient’s age, sex, and systolic blood pressure were to be entered for the BN in Fig. 43.8, then the conditional probability distributions based on that information could be computed for all other variables, including diabetes status and CVD risk, by BN posterior inference algorithms. In Fig. 43.9, learning that a child is from a home with inadequate sanitation would allow updated (posterior) probabilities for the possible income and nutrition levels, as well as the probability of diarrhea, to be computed using exact probabilistic inference algorithms. The best-known exact algorithm (the junction tree algorithm) is summarized succinctly by [3]. For large BNs, approximate posterior probabilities can be computed efficiently using Monte Carlo sampling methods, such as Gibbs sampling, in which input values drawn from the marginal distributions at input nodes are propagated through the network by sampling from the conditional distributions given by the CPTs, thus building up a sample distribution for any output variable(s) of interest.

The BN in Fig. 43.10 illustrates that different types of data, from demographics (age and sex) to choices and behaviors (smoking) to comorbidities (diabetes) to clinical measurements (such as systolic and diastolic blood pressures (SP and DBP)) and biomarkers (cholesterol levels) can be integrated in a BN model, here built using

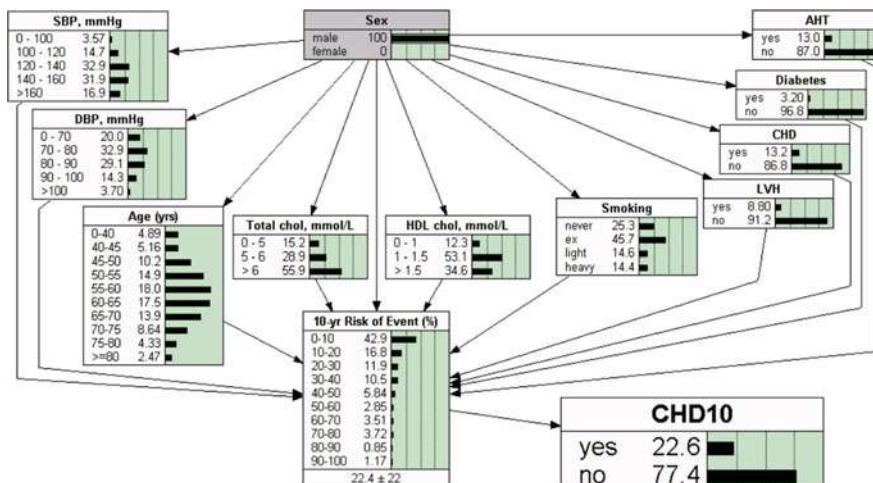


Fig. 43.10 BN model for predicting CHD risk from multiple types of data [117]

the popular *Netica* BN software product, to inform risk estimates for coronary heart disease (CHD) occurring in the next ten years. In addition to posterior inference of entire probability distributions for its variables, BNs can be used to compute the most likely explanations for observed or postulated outcomes (e.g., what are the most likely input values that lead to a specified set of output values) and to study the sensitivity of the probability of achieving (or avoiding) specified target sets of output values to changes in the probabilities of input values.

BN software products such as *Netica*, *Hugin*, or *BayesiaLab* not only provide algorithms to carry out these computations but also integrate them with graphic user interfaces for drawing BNs, populating their probability tables, and reporting results. For example, the DAG in Fig. 43.10, drawn in *Netica*, displays probability distributions for the values of each node. The probabilities are for a man, since the *Sex* node at the top of the diagram has its probability for the value “male” set to 100%. This node is shaded to show that observed or assumed values have been specified by the user, rather than being inferred by the BN model. If additional facts (“findings”) are entered, such as that the patient is a diabetic never-smoker, then the probability distributions at the *10-yr Risk of Event* node and the other nodes would all be automatically updated to reflect (condition upon) this information.

Free BN software packages for creating BNs and performing posterior inference are also available in both R and Python. In R, the *gRain* package allows BNs to be specified by entering the probability tables for their nodes. The resulting BN model can then be queried by entering the variables for which the posterior probability is desired, along with observed or assumed values for other variables. The package will return the posterior probabilities of the query variables, conditioned on the specified observations or assumptions. Both exact and approximate algorithms (such as the junction tree algorithm and Monte Carlo simulation-based algorithms, respectively) for such posterior inference in Bayesian networks are readily available if all variables are modeled as discrete with only a few possible values. For continuous variables, algorithms are available if each node can be modeled as having a normal distribution with a mean that is a weighted sum of the values of its parents, so that each node value depends on its parents’ values through a linear regression equation. Various algorithms based on Monte Carlo simulation are available for the case of mixed discrete and continuous BNs [13].

10.7 Causal Discovery of BNs from Data

A far more difficult problem than posterior inference is to infer or “learn” BNs or other causal graph models directly from data. This is often referred to as the problem of *causal discovery* (e.g., [54]). It includes the *structure discovery* problem of inferring the DAG graph of a BN from data, e.g., by making sure that it shows the conditional independence relations (treated as constraints), statistical dependencies, and order of propagation of changes [119] inferred from data. Structure learning algorithms are typically either constraint-based, seeking to find DAG structures that satisfy the conditional independence relations and other constraints inferred

from data or score-based, seeking to find the DAG structure that maximizes a criterion (e.g., likelihood or posterior probability penalized for complexity) [3,9,20], although hybrid algorithms have also been developed. Learning a BN from data also requires quantifying the probability tables (or other representations of the probabilistic input-output relation) at each node, but this is usually much easier than structure learning. Simply tabulating the frequencies of each output value for each combination of input values may suffice for large data sets if the nodes have been constructed to represent causal mechanisms. For smaller data sets, fitting classification trees or regression models to available data can generate an estimated CPT, giving the conditional probability of each output value for each set of values of the inputs. Alternatively, Bayesian methods can be used to condition priors (typically, Dirichlet priors for multinomial random variables) on available data to obtain posterior distributions for the CPTs [110].

Although many BN algorithms are now available to support learning BNs from data [105], a fundamental limitation and challenge remains that multiple different models often provide approximately equally good explanations of available data, as measured by any of the many scoring rules, information-theoretic measures, and other criteria that have been proposed, and yet they make different predictions for new cases or situations. In such cases, it is better to use an ensemble of BN models instead of any single one to make predictions and support decisions [3]. How best to use common-sense knowledge-based constraints (e.g., that death can be an effect but not a cause of exposure or that age can be a cause but not an effect of health effects) to extract unique causal models, or small sets of candidate models, from data is still an active area of research, but most BN packages allow users to specify both required and forbidden arrows between nodes when these knowledge-based constraints are available. Since it may be impossible to identify a unique BN model from available data, the BN-learning and causal discovery algorithms included in many BN software packages should be regarded as useful heuristics for suggesting possible causal models, rather than as necessarily reliable guides to the truth.

For practical applications, the *bnlearn* package in R [105] provides an assortment of algorithms for causal discovery, with the option of including knowledge-based constraints by specifying directed or undirected arcs that must always be included or that must never be included. For example, in Fig. 43.8, sex, age, and ethnic group cannot have arrows directed into them (they are not caused by other variables), and CVD deaths cannot be a cause of any other variable [115]. The DAG model for cardiovascular disease risk prediction in Fig. 43.8 was discovered using one of the *bnlearn* algorithms (the grow-shrink algorithm for structure learning), together with these knowledge-based constraints. On the other hand, the BN model in Fig. 43.10, which was developed manually based on an existing regression model, has a DAG structure that is highly questionable. Its logical structure is that of a regression model: for men, all other explanatory or independent variables point into the dependent variable *10-year Risk of Event*, and there are no arrows directed between explanatory variables, e.g., from smoking to risk of diabetes. Such additional structure would probably have been discovered had machine learning algorithms for causal discovery such as those in *bnlearn* been applied to the original data. If the

DAG structure of a BN model is incorrect, then the posterior inferences performed using it – e.g., inferences about risks (posterior probabilities) of disease outcomes, and how they would change if inputs such as smoking status were altered – will not be trustworthy. This raises a substantial practical challenge when the correct DAG structure of a BN is uncertain.

10.8 Handling Uncertainty in Bayesian Network Models

BNs and other causal graph model are increasingly used in epidemiology to model uncertain and multivariate exposure-response relations. They are particularly useful for characterizing uncertain causal relations, since they can represent both uncertainty about the appropriate causal structure (DAG model), via the use of multiple DAGs (“ensembles” of DAG models), and uncertainties about the marginal and conditional probabilities at the input and other nodes. As noted by Samet and Bodurow [100], “The uncertainty about the correct causal model involves uncertainty about whether exposure in fact causes disease at all, about the set of confounders that are associated with exposure and cause disease, about whether there is reverse causation, about what are the correct parametric forms of the relations of the exposure and confounders with outcome, and about whether there are other forms of bias affecting the evidence. One currently used method for making this uncertainty clear is to draw a set of causal graphs, each of which represents a particular causal hypothesis, and then consider evidence insofar as it favors one or more of these hypotheses and related graphs over the others.”

An important principle for characterizing and coping with uncertainty about causal models is not to select and use any single model when there is substantial uncertainty about which one is correct [3]. It is more effective, as measured by many performance criteria for evaluating predictive models, such as mean squared prediction error, to combine the predictions from multiple models that all fit the data adequately (e.g., that all have likelihoods at least 10% as large as that of the most likely model). Indeed, the use of multiple models is often essential for accurately depicting model uncertainty when quantifying uncertainty intervals or uncertainty sets for model-based predictions. For example, Table 43.3 presents a

Table 43.3 A machine learning challenge: What outcome should be predicted for case 7 based on the data in cases 1–6?

Case	Predictor 1	Predictor 2	Predictor 3	Predictor 4	Outcome
1	1	1	1	1	1
2	0	0	0	0	0
3	0	1	1	0	1
4	1	1	0	0	0
5	0	0	0	0	0
6	1	0	1	1	1
7	1	1	0	1	?

small hypothetical data set to illustrate that multiple models may provide equally good (in this example, perfect) descriptions of all available data and yet make very different predictions for new cases. For simplicity, all variables in this example are binary (0–1) variables.

Suppose that cases 1–6 constitute a “training set,” with 4 predictors and one outcome column (the right most) to be predicted from them. The challenge for predictive analytics or modeling in this example is to predict the outcome for case 7 (the value, either 0 or 1, in the “?” cell in the lower right of the table). For example, predictors 1–4 might represent various features (1 = present, 0 = absent) of a chemical, or perhaps results of various quick and inexpensive assays for the chemical (1 = positive, 0 = negative) and the outcome might indicate whether the chemical would be classified as a rodent carcinogen in relatively expensive two-year live-animal experiments. A variety of machine-learning algorithms are available for inducing predictive rules or models from training data, from logistic regression to classification trees (or random forest, an ensemble-modeling generalization of classification trees) to BN learning algorithms. Yet, no algorithm can provide trustworthy predictions for the outcome in case 7 based on the training data in cases 1–6, since many different models fit the training data equally well and yet make opposite predictions. For example, the following two models each describe the training data in rows 1–6 perfectly, yet they make opposite prediction for case 7:

Model 1: Outcome = 1 if the sum of predictors 2, 3, and 4 exceeds 1, else 0

Model 2: Outcome = value of Predictor 3.

Likewise, these two models would make opposite predictions for a chemical with predictor values of (0, 0, 1, 0). Model 1 can be represented by a BN DAG structure in which predictors 2, 3, and 4 are the parents of the outcome node (and the CPT is a deterministic function with probabilities of 1 or 0 that the outcome = 1, depending on the values of these predictors). Model 2 would be represented by a BN in which only node 3 is a parent of the outcome node. The following are additional models or prediction rules, e.g.:

Model 3: Outcome is the greater of the values of predictors 1 and 2 except when both equal 1, in which case the outcome is the greater of the values of predictors 3 and 4.

Model 4: Outcome is the greater of the values of predictors 1 and 2 except when both equal 1, in which case the outcome is the lesser of the values of predictors 3 and 4.

These models also provide equally good fits to, or descriptions of, the training data, but make opposite predictions for case 7 and imply yet another BN structure. Thus, it is impossible to confidently identify a single correct model structure from the training data (the data-generating process is *non-identifiable* from the training data), and no predictive analytics or machine learning algorithm can determine from these data a unique model (or set of prediction rules) for correctly predicting the outcome for new cases or situations.

This example illustrates that successful classification or description of reference cases in a training set is a different task from successful prediction of outcomes for new cases outside the training set. It is possible for a computational procedure to have up to 100% accuracy on the former task, while making predictions with no better than a random (50–50) probability of being correct for the latter task. Yet, it is the latter that should be the goal of chief interest to practitioners who want to make predictions or decisions for cases other than those used in building the model. Using ensembles of models can help to characterize the range or set of predicted outcomes for new cases that are consistent with the training data, in the sense of being predicted by models that describe the training data well. They can also provide a basis for procedures that adaptively improve predictions (or decisions) as new cases are observed.

One way to implement this model ensemble approach is via weighted averaging of model-specific predictions, with weights chosen to reflect each model's performance, e.g., how well it explains the data, as assessed by its relative likelihood [3, 78, 79]. Such *Bayesian model averaging* (BMA) of multiple causal graphs avoids the risk of betting predictions on a single model. It demonstrably leads to superior predictions and to reduced model-selection and over-fitting biases in many situations [79]. Similar ideas are included in *super-learning* algorithms, already discussed, which assess model performance and weights via cross validation rather than via likelihood and adaptive learning approaches that learn to optimize not just predictions, but decision rules for making sequences of interventions as outcomes are gradually observed over time (e.g., the *iqLearn* algorithm of Linn et al. [72]). An important application of such decision rule learning algorithms is in sequential multiple assignment randomized trial (SMART) designs for clinical trials. These designs allow treatments or interventions for individual patients to be modified over time as their individual response and covariate histories are observed, in order to increase the probabilities of favorable outcomes for each patient while learning what intervention sequences work best for each type of patient [71].

When the probabilities to be entered into BN node probability tables are unknown, algorithms that propagate *imprecise probabilities* through BN models can be used (e.g., [29, 30]). Both the marginal probabilities at input nodes and the resulting probabilities of different outcomes (or the values at particular output nodes) will then be intervals, representing imprecise probabilities. More generally, instead of specifying marginal and conditional probability tables at the nodes of a BN, uncertainty about the probabilities can be modeled by providing a (usually convex) *set* of probability distributions at each node. BNs generalized in this way are called *credal networks*. Algorithms for propagating sets of probabilities through credal networks have been developed [15] and extended to support optimization of risk management decisions [20].

Alternatively, *second-order probability distributions* (“probabilities of probabilities”) for the uncertain probabilities at BN nodes can be specified. If these uncertainties about probabilities are well approximated by Dirichlet or beta probability distributions (as happens naturally when probabilities or proportions are estimated from small samples using Bayesian methods with uniform or Dirichlet

priors), then Monte Carlo uncertainty analysis can be used to propagate the uncertain probabilities efficiently through the BN model, leading to uncertainty distributions for the posterior probabilities of the values of the variables in the BN (Kleiter 1996). Imprecise Dirichlet models have also been used to learn credal sets from data, resulting in upper and lower bounds for the probability that each variable takes on a given value [15].

Rather than using sets or intervals for uncertain probabilities, it is sometimes possible to simply use best guesses (point estimates) and yet to have confidence that the results will be approximately correct. Henrion et al. (1996) note that, in many situations, the key inferences and insights from BN models are quite insensitive (or “robust”) to variations in the estimated values in the probability tables for the nodes. When this is the case, best guesses (e.g., MLE point estimates) of probability values may be adequate for inference and prediction, even if the data and expertise used to form those estimates are scarce and the resulting point estimates are quite uncertain.

10.9 Influence Diagrams Extend BNs to Support Optimal Risk Management Decision-Making

The BN techniques discussed so far are useful for predicting how output probabilities will change if input values are varied, provided that the DAG structure can be correctly interpreted as showing how changes in the inputs propagate through networks of causal mechanisms to cause changes in outputs. (As previously discussed, this requires that the network is constructed so that the CPTs at nodes represent not merely statistical descriptions of conditional probabilities in historical data but causal relations determining probabilities of output values for each combination of input values.) Once the probabilities of different outputs can be predicted for different inputs, it is natural to ask how the controllable inputs should be set to make the resulting probability distribution of outputs as desirable as possible. This is the central question of decision analysis, and mainstream decision analysis provides a standard answer: choose actions to maximize the expected utility of the resulting probability distribution of consequences.

To modify BN models to support optimal (i.e., expected utility-maximizing) risk management decision-making, the BNs must be augmented with two types of nodes that do not represent random variables or deterministic functions. There is a *utility node*, also sometimes called a *value node*, which is often depicted in DAG diagrams as a hexagon and given a name such as “Decision-maker’s utility.” There must also be one or more *choice nodes*, also called *decision nodes*, commonly represented by rectangles. The risk management decision problem is to make choices at the decision nodes to maximize the expected value of the utility node, taking into account the uncertainties and conditional probabilities described by the rest of the DAG model. Input decision nodes (i.e., decision nodes with only outward-directed arrows) represent inputs whose values are controlled by the decision-maker. Decision nodes with inputs represent *decision rules*, i.e., tables or functions specifying how the decision node’s value is to be chosen, for each

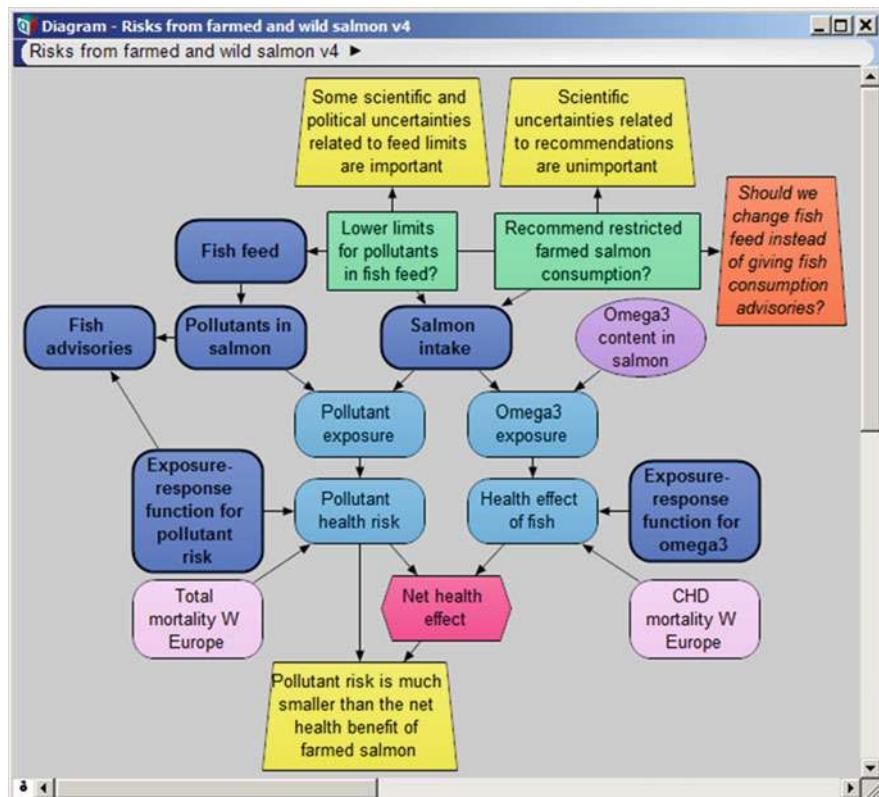


Fig. 43.11 An influence diagram (ID) model with two decision nodes (*green rectangles*) and with *Net health effect* as the value node. (Questions and comments in trapezoids on the periphery are not parts of the formal ID model, but help to interpret it for policy makers) (Source: www.lumina.com/case-studies/farmed-salmon/)

combination of values of its inputs. BNs with choice and value nodes are called *influence diagram* (ID) models. BN posterior inference algorithms can be adapted to solve for the best decisions in an ID, i.e., the choices to make at the choice nodes in order to maximize the expected value of the utility node [3, 123].

Figure 43.11 shows an example of an ID model developed and displayed using the commercial ID software package *Analytica*. Its two decision nodes represent choices about whether to lower the allowed limits for pollutants in fish feed and whether to recommend to consumers that they restrict consumption of farmed salmon, respectively [49]. The two decision nodes are shown as green rectangles, located toward the top of the ID. The value or utility node in Fig. 43.11, shown as a pink hexagon located toward the bottom of the diagram, is a measure of net health effect in a population. It can be quantified in units such as change in life expectancy (added person-year of life) or change in cancer mortality rates caused by different decisions and by the other factors shown in the model. Many of these

factors, such as (a) the estimated exposure-response relations for health harm caused by consuming pollutants in fish and (b) the health benefits caused by consuming omega three fatty acids in fish, are uncertain. The uncertainties are represented by random variables (the dark blue oval-shaped nodes throughout the diagram) and by modeling assumptions that allow other quantities (the light blue oval-shaped nodes) to be calculated from them.

An example of a modeling assumption is that pollutants increase mortality rates in proportion to exposure, with the size of this slope factor being uncertain. Different models (or expert opinions) for relevant toxicology, dietary habits, consumer responses to advisories and recommendations, nutritional benefits of fish consumption, and so forth can contribute to developing the CPTs for different parts of the ID model and characterizing uncertainties about them. IDs thus provide a constructive framework for coordinating and integrating multiple submodels and contributions from multiple domains of expertise and for applying them to help answer practical questions such as how different policies, regulations, warnings, or other actions will affect probable health effects, consumption patterns, and other outcomes of interest.

If multiple decision makers with different jurisdictions or spans of control attempt to control the same outcome, however, then coordinating their decisions effectively may require resolving game-theoretic issues in which each decision maker's best decision depends on what the others do. For example, in Fig. 43.11, if the regulators in charge of setting allowed limits for pollutant contamination levels in fish feed are different from the regulators or public health agencies issuing advisories about what to eat and what not to eat, then each might decide not to take additional action to protect public health if it mistakenly assumes that the other will do so. Problems of risk regulation or management with multiple decision-makers can be solved by generalizing IDs to *multi-agent influence diagrams* (MAIDs) [67, 89, 107]. MAID algorithms recommend what each decision-maker, each with its own utility function and decision variables, should do, taking into account any information it has about the actions of others, when their decisions propagate through a DAG model to jointly determine probabilities of consequences.

Although the idea of extending BNs to include decision and value nodes seems straightforward in principle, understanding which variables are controllable by whom over what time interval may require careful thought in practice. For example, consider a causal graph model (Fig. 43.12) showing factors affecting treatment of tooth defects (central node), such as patient's *age*, *genetics*, *smoking status*, *diabetes*, *use of antibacterials*, *pulpal status*, available surgical *devices*, and *operator skill* [1]. These variables have not been pre-labeled as chance or choice nodes. Even without expertise in dentistry, it is clear that some of the variables, such as *genetics* or *age*, should not ordinarily be modeled as decision variables. Others, such as *Use of antibacterials* and *Pulpal status* (reflecting oral hygiene) may result from a history of previous decisions by patients and perhaps other physicians or periodontists. Still others, such as available surgical *devices* and *operator skill*, are fixed in the short run, but might be considered decision variables over intervals long enough to include the operator's education, training, and experience or if the decisions to be

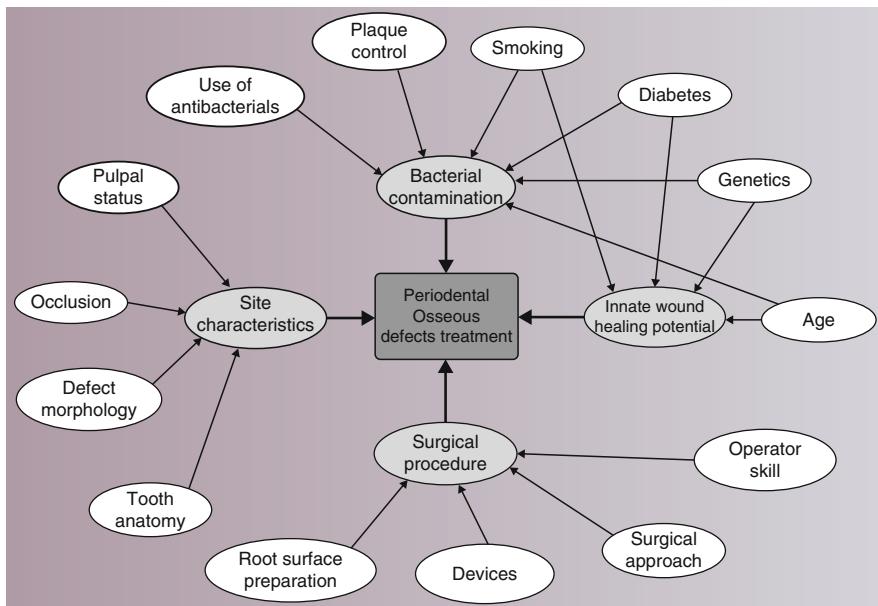


Fig. 43.12 Different variables can be treated as decision variables on different time scales (Source: [1])

made include hiring practices and device acquisition decisions of the clinic where the surgery is performed. *Smoking* and *diabetes* indicators might also be facts about a patient that cannot be varied in the short run, but that might be considered as at least in part determined by past health and lifestyle decisions. In short, even if a perfectly accurate causal graph model were available, the question of who acts upon the world how and over what time frame via the causal mechanisms in the model must still be resolved in formulating an ID or MAID model from a causal BN. In organizations or nations seeking to reduce various risks through policies or regulations, who should manage what, which variables should be taken as exogenously determined, and which should be subjected to control must likewise be resolved before ID or MAID models can be formulated and solved to obtain recommended risk management decisions.

10.10 Value of Information (VOI), Dynamic Bayesian Networks (DBNs), and Sequential Experiments for Reducing Uncertainties Over Time

Once a causal ID model has been fully quantified, it can be used to predict how the probability distributions for different outcomes of interest (such as net health effect in Fig. 43.11) and expected utility will change if different decisions are

made. This what-if capability, in turn, allows decision optimization algorithms to identify which specific decisions and decision rules maximize expected utility and to calculate how sensitive the recommended decisions are to other uncertainties and assumptions in the model. ID software products such as *Analytic* and *Netica* support construction of IDs and automatically solve them for the optimal decisions. For the example in Fig. 43.11, a robust optimal decision is to not recommend restrictions in fish consumption to consumers, as the estimated health benefits of greater fish consumption far outweigh the estimated health risks. This conclusion is unlikely to be reversed by further reductions in uncertainty, i.e., there is little doubt that it is true. By contrast, whether it is worth lowering allowed levels of pollutants in fish feed is much less clear, with the answer depending on modeling assumptions that are relatively uncertain. This implies a positive *value of information* (VOI) for reducing these uncertainties, meaning that doing so might change the best decision and increase expected utility. ID models can represent the option of collecting additional information before making a final decision about what actions to take, such as lowering or not lowering allowed pollutant levels, by including one or more additional decision nodes to represent information acquisition, followed by chance nodes showing what the additional information might reveal.

In an ID with options for collecting more information before taking a final action, the optimal next step based on presently available information might turn out to be to collect additional information before committing to final regulations or other costly actions. This will be the case if and only if the costs of collecting further information next, including any costs of delay that this entails, are less than the benefits from better-informed subsequent decisions, in the sense that collecting more information before acting (e.g., implementing a regulation or issuing a warning in Fig. 43.11) has greater expected utility than taking the best action now with the information at hand. Optimal delay and information acquisition strategies based on explicit VOI calculations often conflict with more intuitive or political criteria. Both individuals and groups are prone to conclude prematurely that there is already sufficient information on which to act and that further delay and information collection are therefore not warranted, due to narrow framing, overconfidence and confirmation biases, groupthink, and other psychological aspects of decision-making [61]. Politicians and leaders may respond to pressure to exhibit the appearance of strong leadership by taking prompt action without first learning enough about their probable consequences. VOI calculations can help to overcome such well-documented limitations of informal decision-making by putting appropriate weight on the value of reducing uncertainty before acting.

To explicitly model the sequencing of information collection, action selection, and resulting changes in outcomes over time, consecutive period-specific BNs or IDs can be linked by information flows, meaning that the nodes in each period's network (or "slice" of the full multi-period model) are allowed to depend on information received in previous periods. The resulting *dynamic Bayesian networks* (DBNs) or dynamic IDs provide a very convenient framework for predicting and optimizing decisions and consequences over time as initial uncertainties are gradually reduced or resolved. They have proved valuable in medical decision-

making for forecasting in detail the probabilities of different time courses of diseases and related quantities, such as probability of first diagnosis with a disease or adverse condition within a specified number of months or years [121], survival times and probabilities for patients with different conditions and treatments, and remaining days of hospitalization or remaining years of life for individual patients being monitored and treated for complex diseases, from cancers to multiple surgeries to sequential organ failures [5, 101].

DBN estimation software is freely available in R packages [69, 94]. It has been developed and used largely by the systems biology community for interpreting time series of gene expressions in systems biology. Biological and medical researchers, electrical engineers, computer scientists, artificial intelligence researchers, and statisticians have recognized that DBNs generalize important earlier methods of dynamic estimation and inference, such as Hidden Markov Models and Kalman filtering for estimation and signal processing [34]. DBNs are also potentially extremely valuable in a wide range of other engineering, regulatory, policy, and decision analysis settings where decisions and their consequences are distributed over time, where feedback loops or other cycles make any static BN inapplicable, or where detailed monitoring of changing probabilities of events is desired so that midcourse changes in actions can be made in order to improve final outcomes.

Development and application of DBN algorithms and various generalizations are fruitful areas of ongoing applied research. Key concepts of DBNs and multi-agent IDs have been successfully combined to model *multi-agent control* of dynamic random processes (modeled as multi-agent partially observable Markov decision processes, POMDPs) [93]. More recently, DBN methods have been combined with ideas from change-point analysis for situations where arcs in the DAG model are gained or lost at certain times as new influences or mechanisms start to operate or former ones cease [97]. These advances further extend the flexibility and realism of DBN models (and dynamic IDs based on them) to apply to description and control of nonstationary time series.

As already discussed, value of information (VOI) calculations, familiar from decision analysis, can be carried out straightforwardly in ID models. Less familiar, but still highly useful, are methods for optimizing the sequential collection of information to better ascertain correct causal models. The best available methods involve design of experiments [116] and of time series of observations [90]. When the correct ID model describing the relation between decisions and consequence probabilities is initially uncertain, then collecting additional information may have value not only for improving specific decisions (i.e., changing decisions or decision rules to increase expected utility) within the context of a specified ID model but also for discriminating among alternative ID models to better ascertain which ones best describe reality. New information can help in learning IDs from data by revealing how the effects of manipulations develop in affected variables over time [119]. For example, Tong and Koller [116] present a Bayesian approach to sequential experimentation in which a distribution of BN DAG structures and CPTs is updated by experiments that set certain variables to new values and monitor the changes in

values of other variables. At each step, the next experiment to perform is selected to most reduce expected loss from incorrect inferences about the presence and directions of arcs in the DAG model. Even in BNs without decision or utility nodes, designing experiments and time series of observations to facilitate accurate learning of BN descriptions can be very valuable in creating and validating models with high predictive accuracy [90].

11 Causal Analytics

The preceding sections have discussed how causal Bayesian networks and other DAG and time series algorithms provide constructive methods for carrying out many risk assessment and risk management tasks, even when there is substantial initial uncertainty about relevant cause-and-effect relations and about the best (expected utility-maximizing) courses of action. Other graphical formalisms for risk analysis and decision-making, such as decision trees, game trees, fault trees, and event trees, which have long been used to model the propagation of probabilistic events in complex systems, can all be converted to equivalent IDs or BNs, often with substantial reductions in computational complexity and with savings in the number of nodes and combinations of variable values that must be explicitly represented [107]. Thus, BNs and IDs provide an attractive unifying framework for characterizing, quantifying, and reducing uncertainties and for deciding what to do under the uncertainties that remain. They, together with time series techniques and machine learning techniques, provide a toolkit for using data to inform inference, prediction, and decision-making with realistic uncertainties. These methods empower the following important and widely used types of analytics for using data to inform decisions:

- *Descriptive analytics:* BNs and IDs describe how the part of the world being modeled probably works, showing which factors influence or determine the probability distributions for which other variables and quantifying the probabilistic relations among variables. If a BN or ID has CPTs that represent the operation of lawlike causal mechanisms – i.e., if it is a causal BN or ID – then it can be used to describe how changes in some variables affect the probability distributions of others and hence how probabilistic causal influences propagate to change the probabilities of outcomes.
- *Predictive analytics:* A BN can be used to predict how the probabilities of future observations change when new evidence is acquired (or assumed). A causal BN or ID predicts how changes made at input nodes will affect the future probabilities of outputs. Dynamic Bayesian networks (DBNs) are used to forecast the probable sequences of future changes that will occur after observed changes in inputs, culminating in a new posterior joint probability distribution for all other variables over time (calculated via posterior inference algorithms). BNs and DBNs are also used to predict and compare the probable consequences (changes in probability distributions of outputs and other variables) caused by alternative hypothetical (counterfactual) scenarios for changes in inputs,

including alternative decisions. Conversely, BNs can predict the most likely explanation for observed data, such as the most likely diagnosis explaining observed symptoms or the most likely sequence of component failures leading to a real or hypothesized failure of a complex system. By predicting the probable consequences of alternative policies or decisions and the most likely causes for undesired outcomes, BNs can inform risk management decision-making and help to identify where to allocate resources to repair or forestall likely failure paths.

- *Uncertainty analytics.* Both BNs and IDs are designed to quantify uncertainties about their predictions by using probability distributions for all uncertain quantities. When model uncertainty is important, model ensemble methods allow the predictions or recommendations from multiple plausible models to be combined to obtain more accurate forecasts and better-performing decision recommendations [3]. DBNs provide the ability to track event probabilities in detail as they change over time, and dynamic versions of MAIDs allow uncertainties about the actions of other decision-makers to be modeled.
- *Prescriptive analytics.* If a net benefit, loss, or utility function for different outcomes is defined, and if the causal DAG relating choices to probabilities of consequences is known, then ID algorithms can be used to solve for the best combination of decision variables to minimize expected loss or maximize expected utility. If more than one decision-maker or policy maker makes choices that affect the outcome, then MAIDS or dynamic versions of MAIDs can be used to recommend what each should do.
- *Evaluation and learning analytics.* Ensembles of BNs, IDs, and dynamic versions and extensions of these can be learned from data and experimentation. Value of information (VOI) calculations determine when a single decision-maker in a situation modeled by a known ID should stop collecting information and take action. Dynamic causal BNs and IDs can be learned from time series data in many settings (including observed responses to manipulations or designed experiments) and current decision rules or policies can be evaluated and improved during the learning process, via methods such as low-regret learning with model ensembles, until no further improvements can be found [107]. Learning about causal mechanisms from the observed time series of responses to past interventions, manipulations, decisions, or policies provides a promising technical approach to using past experience to deliberately improve future decisions and outcomes.

Table 43.4 shows how these various components, which might collectively be called *causal analytics*, provide constructive methods for answering the fundamental questions raised in the introduction. For event detection and consequence prediction, DBNs (especially, nonstationary DBNs) and change-point analysis (CPA) algorithms are well suited for detecting changes in time series of observations and occurrences of unobserved events based on their observable effects. DBNs and causal simulation models, as well as time series models that accurately describe how impacts of changes are distributed over time, are also useful for predicting the probable future consequences of recent changes or “shocks” in the inputs to a system.

Table 43.4 Causal analytics algorithms address fundamental risk management questions under realistic uncertainties

Fundamental questions	Causal analytics algorithms and methods for answering the questions
<i>Event detection: What has changed recently in disease patterns or other adverse outcomes, by how much, when?</i>	<ul style="list-style-type: none"> • Change-point analysis (CPA) algorithms • Dynamic Bayesian networks (DBNs)
<i>Consequence prediction: What are the implications for what will probably happen next if different actions (or no new actions) are taken?</i>	<ul style="list-style-type: none"> • Dynamic Bayesian networks (DBNs) • Simulation modeling • Time series forecasting
<i>Risk attribution: What is causing current undesirable outcomes? Does a specific exposure harm human health, and, if so, who is at greatest risk and under what conditions?</i>	<ul style="list-style-type: none"> • Causal DAG models (e.g., BNs, IDs) • Ensembles of DAG models • Granger causality and transfer entropy (TE) for time series
<i>Response modeling: What combinations of factors affect health outcomes, and how strongly? How would risks change if one or more of these factors were changed?</i>	<ul style="list-style-type: none"> • Causal DAG models, e.g., BN models
<i>Decision making: What actions or interventions will most effectively reduce uncertain health risks?</i>	<ul style="list-style-type: none"> • Influence diagram (ID) algorithms • MAIDs for multiple decision makers • Adaptive learning methods, e.g., iqLearn, if the ID model is uncertain
<i>Retrospective evaluation and accountability: How much difference have exposure reductions or other policies actually made in reducing adverse health outcomes?</i>	<ul style="list-style-type: none"> • Quasi-experimental (QE) studies • Intervention analysis for time series • Ensemble learning algorithms such as iqLearn for continuous improvement

For risk attribution, causal graph models (such as BNs, IDs, and dynamic versions of these) or ensembles of such models can be learned from data and used to quantify the evidence that suspected hazards indeed cause the adverse effects attributed to them (i.e., that there is, with high confidence, a directed arc pointing from a node representing exposure to a hazard into a node representing the effect). If so, the CPT for the effect node quantifies how changes in the exposure node change probabilities of effects, given the levels of other causes with which exposure may interact. Multivariate response modeling, in which the joint distributions of one or more responses vary with the levels of one or more factors that probabilistically cause them, can readily be modeled by DAGs that include the different causal factors and effects. For risk management or regulation under uncertainty, if utility nodes and decision nodes are incorporated into the causal graph models to create known causal ID or MAID models, then the best decisions for risk management (i.e., for inducing the greatest achievable expected utilities) can be identified by well-developed ID solution algorithms, and VOI calculations can be used to optimize costly information collection and the timing of final decisions.

Finally, for retrospective evaluation and accountability, quasi-experiments and intervention analysis of interrupted time series provide traditional methods of analysis, although they require using data (or assumptions) to refute noncausal explanations for changes in time series. More recently developed ensemble-learning methods [3, 107] and adaptive learning algorithms (such as iqLearn for learning to optimize treatment sequences) can be used to continually evaluate and improve the success of current decision rules, policies, or regulations for managing uncertain risks, based on their performance to date and on relative expected costs of switching among them and of failing to do so. Such adaptive evaluation and improvement is possible provided that the consequences of past actions (probably) are monitored and the data are made available and used to update causal IDs, MAIDs, or dynamic versions of such models to allow ongoing learning and optimization. Thus, causal graph methods (including ensemble methods, when appropriate models are uncertain, and time series methods that uncover DAG structures relating time series variables) provide a rich set of tools for addressing fundamental challenges of uncertainty quantification and decision-making under uncertainty.

12 Summary and Conclusions: Applying Causal Graph Models to Better Manage Risks and Uncertainties

The power and maturity of the technical methods in Table 43.4 have spurred their rapid uptake and application in fields such as neurobiology, systems biology, econometrics, artificial intelligence, control engineering, game theory, signal processing, and physics. However, they have so far had relatively limited impact on the practice of uncertainty quantification and risk management in epidemiology, public health, and regulatory science, perhaps because these fields give great deference to the use of subjective judgments informed by weight-of-evidence considerations – an approach widely used and taught since the 1960s, but of unproved and doubtful probative value [83]. Previous sections have illustrated some of the potential of more modern methods of causal analytics, but the vast majority of applied work in epidemiology, public health, and regulatory risk assessment unfortunately still uses older association-based methods and subjective opinions about the extent to which statistically significant differences between risk model coefficients for differently exposed populations might have causal interpretations.

To help close the gap between these poor current practices and the potentially much more objective, reliable, accurate, and sensitive methods of causal analytics in Table 43.4, the following checklist may prove useful in judging the adequacy of policy analyses or quantitative risk assessments (QRAs) that claims to have identified useful predictive causal relations between exposures to risk factors or hazards and resulting risks of adverse effects (responses), i.e., causal exposure-response (E-R) relations.

1. *Does the QRA show that changes in exposures precede the changes in health effects that they are said to cause?* Are results of appropriate technical analyses (e.g., change-point analyses, intervention analyses and other quasi-experimental comparisons, and Granger causality tests or transfer entropy results) presented, along with supporting data? If effects turn out to precede their presumed causes, then unmeasured confounders or residual confounding by confounders that the investigators claim were statistically “controlled for” may be at work.
2. *Does the QRA demonstrate that health effects cannot be made conditionally independent of exposure* by conditioning on other variables, especially potential confounders? Does it present the details, data, and results of appropriate statistical tests (e.g., conditional independence tests and DAGs) showing that health effects and exposures share mutual information that cannot be explained away by any combination of confounders?
3. *Does the QRA present and test explicit causal graph models*, showing the results of formal statistical tests of the causal hypotheses implied by the structure of the model (i.e., which variables point into which others)? Does it identify which alternative causal graph models are most consistent with available data (e.g., using the Occam’s Window method of [78])? Most importantly, does it present clear evidence that changes in exposure propagate through the causal graph, causing successive measurable changes in the intermediate variables along hypothesized causal paths? Such coherence, consistency, and biological plausibility demonstrated in explicit causal graph models showing how hypothesized causal mechanisms dovetail with each other to transduce changes in exposures to changes in health risks can provide compelling objective evidence of a causal relation between them, thus accomplishing what older and more problematic WoE frameworks have long sought to provide [95].
4. *Have noncausal explanations for statistical relations among observed variables (including exposures, health effects, and any intermediate variables, modifying factors, and confounders) been explicitly identified and convincingly refuted* using well-conducted and reported statistical tests? Especially, have model diagnostics (e.g., plots of residuals and discussions of any patterns) and formal tests of modeling assumptions been presented that show that the models used appropriately describe the data to which the QRA applies them and that claimed associations are not caused by model selection biases or specification errors, failures to model errors in exposure estimates and other explanatory variables, omitted confounders or other latent variables, uncorrected multiple testing bias, or coincident historical trends (e.g., spurious regression, if the exposure and health effects time series in longitudinal studies are not stationary)?
5. *Have all causal mechanisms postulated in the QRA modeling been demonstrated to exhibit stable, uniform, lawlike behavior*, so that there is no substantial unexplained heterogeneity in estimated input-output (e.g., E-R or C-R) relations? If the answer is no, then missing factors may need to be identified and their effects modeled before valid predictions can be made based on the assumption that changes in causes will yield future changes in effects that can be well described and predicted based on estimates of cause-effect relations from past data.

If the answers to these five diagnostic questions are all yes, then the QRA has met the burden of proof of showing that the available data are consistent with a causal relation and that other (noncausal) explanations are not plausible. It can then proceed to quantify the estimated changes in probability distributions of outputs, such as future health effects, that would be caused by changes in controllable inputs (e.g., future exposure levels) using the causal models developed to show that exposure causes adverse effects. The effort needed to establish valid evidence of a causal relation between historical levels of inputs and outputs by being able to answer yes to questions 1–5 pays off at this stage. Causal graph models (e.g., Bayesian networks with validated causal interpretations for their CPTs), simulation models based on composition of validated causal mechanisms, and valid path diagrams and SEM causal models can all be used to predict quantitative changes in outputs that would be caused by changes in inputs, e.g., changes in future health risks caused by changes in future exposure levels, given any scenario for the future values of other inputs.

Conversely, if the answer to any of the preceding five diagnostic questions is no, then it is premature to make causal predictions based on the work done so far. Either the additional work needed to make the answers yes should be done or results should be stated as contingent on the as-yet unproved assumption that this can eventually be done.

Cross-References

► [Rare-Event Simulation](#)

References

1. Alpiste Illueca, F.M., Buitrago Vera, P., de Grado Cabanilles, P., Fuenmayor Fernandez, V., Gil Loscos, F.J.: Periodontal regeneration in clinical practice. *Med. Oral Patol. Oral Cir. Bucal.* **11**(4), e3:82–e3:92 (2006)
2. Angrist, J.D., Pischke, J.-S.: *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, Princeton (2009)
3. Ashcroft, M.: Performing decision-theoretic inference in Bayesian network ensemble models In: Jaeger,M., Nielsen, T.D., Viappiani, P. (eds.) Twelfth Scandinavian Conference on Artificial Intelligence, Aalborg, vol. 257, pp. 25–34 (2013)
4. Arnold, A., Liu, Y., Abe, N.: Temporal causal modeling with graphical Granger methods. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD-07), San Jose, 12–15 Aug 2007. ACM, New York. <http://dl.acm.org/citation.cfm?id=1281192&picked=prox>
5. Azhar, N., Ziraldo, C., Barclay, D., Rudnick, D.A., Squires, R.H., Vodovotz, Y., Pediatric Acute Liver Failure Study Group: Analysis of serum inflammatory mediators identifies unique dynamic networks associated with death and spontaneous survival in pediatric acute liver failure. *PLoS One.* **8**(11), e78202 (2013). doi:10.1371/journal.pone.0078202
6. Bai, Z., Wong, W.K., ZhangB.: Multivariate linear and nonlinear causality tests. *Math. Comput. Simul.* **81**(1), 5–17 (2010)
7. Barnett, L., Seth, A.K.: The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference. *J. Neurosci. Methods* 223 (2014)

8. Barr, C.D., Diez, D.M., Wang, Y., Dominici, F., Samet, J.M.: Comprehensive smoking bans and acute myocardial infarction among Medicare enrollees in 387 US counties: 1999–2008. *Am. J. Epidemiol.* **176**(7), 642–648 (2012). Epub 17 Sep 2012
9. Brenner E, Sontag D. (2013) SparsityBoost: a new scoring function for learning Bayesian network structure. In: 29th Conference on Uncertainty in Artificial Intelligence (UAI2013). Westin Bellevue Hotel, Washington, DC, 11–15 July 2013. <http://auai.org/uai2013/prints/papers/30.pdf>
10. Callaghan, R.C., Sanches, M., Gatley, J.M., Stockwell, T.: Impacts of drinking-age laws on mortality in Canada, 1980–2009. *Drug Alcohol Depend.* **138**, 137–145 (2014). doi:10.1016/j.drugalcdep.2014.02.019
11. Cami, A., Wallstrom, G.L., Hogan, W.R.: Measuring the effect of commuting on the performance of the Bayesian Aerosol Release Detector. *BMC Med. Inform. DecisMak.* **9**(Suppl 1), S7 (2009)
12. Campbell, D.T., Stanley, J.C.: Experimental and Quasi-experimental Designs for Research. Rand McNally, Chicago (1966)
13. Chang, K.C., Tian, Z.: Efficient inference for mixed Bayesian networks. In: Proceedings of the Fifth International Conference on Information Fusion, Annapolis, vol. 1, pp 527–534, 8–11 July 2002. IEEE. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1021199>
14. Christensen, T.M., Møller, L., Jørgensen, T., Pisinger, C.: The impact of the Danish smoking ban on hospital admissions for acute myocardial infarction. *Eur. J. PrevCardiol.* **21**(1), 65–73 (2014). doi:10.1177/2047487312460213
15. Corani, G., Antonucci, A., Zaffalon, M.: Bayesian networks with imprecise probabilities: theory and application to classification. In: Holmes, D.E., Jaim, C. (eds.) Data Mining: Foundations and Intelligent Paradigms. Intelligent Systems Reference Library, vol. 23, pp. 49–93 (2012)
16. Cox, L.A. Jr., Popken, D.A.: Has reducing fine particulate matter and ozone caused reduced mortality rates in the United States? *Ann. Epidemiol.* **25**(3), 162–173 (2015)
17. Cox, L.A. Jr., Popken, D.A., Berman, D.W.: Causal versus spurious spatial exposure-response associations in health risk analysis. *Crit. Rev. Toxicol.* **43**(Suppl 1), 26–38 (2013)
18. Crowson, C.S., Schenck, L.A., Green, A.B., Atkinson, E.J., Therneau, T.M.: The basics of propensity scoring and marginal structural models. Technical report #84, 1 Aug 2013. Department of Health Sciences Research, Mayo Clinic, Rochester. http://www.mayo.edu/research/documents/biostat-84-pdf/doc_20024406
19. Dash, D., Drudzsel, M.J.: A note on the correctness of the causal ordering algorithm. *Artif. Intell.* **172**, 1800–1808 (2008). <http://www.pitt.edu/~drudzsel/psfiles/aij08.pdf>
20. De Campos C.P., Ji, Q.: Efficient structure learning of Bayesian networks using constraints. *J. Mach. Learn. Res.* **12**, 663–689 (2011)
21. Dominici, F., Greenstone, M., Sunstein, C.R.: Science and regulation. Particulate matter matters. *Science*. **344**(6181), 257–259 (2014). doi:10.1126/science.1247348
22. The Economist: Trouble at the Lab: scientists like to think of science as self-correcting. To an alarming degree, it is not. www.economist.com/news/briefing/21588057-scientists-think-science-self-correcting-alarming-degree-it-not-trouble, 19 Oct 2013
23. Eichler, M., Didelez, V.: On Granger causality and the effect of interventions in time series. *Lifetime Data Anal.* **16**(1), 3–32 (2010). Epub 26 Nov 2009. <http://www.ncbi.nlm.nih.gov/pubmed/19941069>
24. EPA (U.S. Environmental Protection Agency): The Benefits and Costs of the Clean Air Act from 1990 to 2020. Final Report – Rev. A. Office of Air and Radiation, Washington, DC (2011)
25. EPA: Expanded expert judgment assessment of the concentration-response relationship between PM2.5 exposure and mortality. www.epa.gov/ttn/ecas/regdata/Uncertainty/pm_ee_report.pdf (2006)
26. Exarchos, K.P., Goletsis, Y., Fotiadis, D.I.: A multiscale and multiparametric approach for modeling the progression of oral cancer. *BMC Med. Inform. DecisMak.* **12**, 136 (2012). doi:10.1186/1472-6947-12-136.

27. Ezzati, M., Hoorn, S.V., Lopez, A.D., Danaei, G., Rodgers, A., Mathers, C.D., Murray, C.J.L.: Comparative quantification of mortality and burden of disease attributable to selected risk factors. In: Lopez, A.D., Mathers, C.D., Ezzati, M., Jamison, D.T., Murray, C.J.L. (eds.) Global Burden of Disease and Risk Factors, chapter 4. World Bank, Washington, DC (2006)
28. Fann, N., Lamson, A.D., Anenberg, S.C., Wesson, K., Risley, D., Hubbell, B.J.: Estimating the national public health burden associated with exposure to ambient PM_{2.5} and Ozone. *Risk Anal.* **32**(1), 81–95 (2012)
29. Ferson, S., Donald, S.: Probability bounds analysis. In: Mosleh, A., Bari, R.A. (eds.) Probabilistic Safety Assessment and Management, pp. 1203–1208. Springer, New York (1998)
30. Ferson, S., Hajagos, J.G.: Arithmetic with uncertain numbers: rigorous and (often) best possible answers. In: Helton, J.C., Oberkampf, W.L. (eds.) Alternative Representations of Epistemic Uncertainty. Reliability Engineering & System Safety, vol. 85, pp. 135–152; 1–369 (2004)
31. Freedman, D.A.: Graphical models for causation, and the identification problem. *Eval. Rev.* **28**(4), 267–293 (2004)
32. Friedman, N., Goldszmidt, M.: Learning Bayesian networks with local structure. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 421–459. MIT, Cambridge (1998)
33. Gasparrini, A., Gorini, G., Barchielli, A.: On the relationship between smoking bans and incidence of acute myocardial infarction. *Eur. J. Epidemiol.* **24**(10), 597–602 (2009)
34. Ghahramani, Z.: Learning dynamic Bayesian networks. In: Giles, C.L., Gori, M. (eds.) Adaptive Processing of Sequences and Data Structures. International Summer School on Neural Networks "Caianniello, E.R." Vietri sul Mare, Salerno, 6–13 Sept 1997. Tutorial Lectures. Lecture Notes in Computer Science, vol. 1387 (1998). <http://link.springer.com/sbook/10.1007/BFb0053992>, <http://link.springer.com/bookseries/558>, <http://mlg.eng.cam.ac.uk/zoubin/SALD/learnDBNs.pdf> (1997)
35. Greenland, S.: Epidemiologic measures and policy formulation: lessons from potential outcomes. *Emerg. Themes Epidemiol.* **2**, 5 (2005)
36. Greenland, S., Brumback, B.: An overview of relations among causal modelling methods. *Int. J. Epidemiol.* **31**(5), 1030–1037 (2002). <http://www.ncbi.nlm.nih.gov/pubmed/12435780>
37. Gruber, S., Logan, R.W., Jarrín, I., Monge, S., Hernán, M.A.: Ensemble learning of inverse probability weights for marginal structural modeling in large observational datasets. *Stat. Med.* **34**(1), 106–117 (2015)
38. Grundmann, O.: The current state of bioterrorist attack surveillance and preparedness in the US. *Risk Manag. Health Policy.* **7**, 177–187 (2014)
39. Hack, C.E., Haber, L.T., Maier, A., Shulte, P., Fowler, B., Lotz, W.G., Savage, R.E., Jr.: A Bayesian network model for biomarker-based dose response. *Risk Anal.* **30**(7), 1037–1051 (2010)
40. Harris, A.D., Bradham, D.D., Baumgarten, M., Zuckerman, I.H., Fink, J.C., Perencevich, E.N.: The use and interpretation of quasi-experimental studies in infectious diseases. *Clin. Infect Dis.* **38**(11), 1586–1591 (2004)
41. Harris, A.D., McGregor, J.C., Perencevich, E.N., Furuno, J.P., Zhu, J., Peterson, D.E., Finkelstein, J.: The use and interpretation of quasi-experimental studies in medical informatics. *J. Am. Med. Inform. Assoc.* **13**(1), 16–23 (2006)
42. Harvard School of Public Health: Press Release: Ban On Coal Burning in Dublin Cleans the Air and Reduces Death Rates www.hsph.harvard.edu/news/press-releases/archives/2002-releases/press10172002.html (2002)
43. Health Effects Institute (HEI): Impact of Improved Air Quality During the 1996 Summer Olympic Games in Atlanta on Multiple Cardiovascular and Respiratory Outcomes. HEI Research Report #148 (2010). Authors: Jennifer L. Peel, Mitchell Klein, W. Dana Flanders, James A. Mulholland, and Paige E. Tolbert. Health Effects Institute. Boston, MA. <http://pubs.healtheffects.org/getfile.php?u=564>

44. Health Effects Institute (HEI): Did the Irish Coal Bans Improve Air Quality and Health? HEI Update. <http://pubs.healtheffects.org/getfile.php?u=929> (Summer, 2013). Last Retrieved 1 Feb 2014
45. Helfenstein, U.: The use of transfer function models, intervention analysis and related time series methods in epidemiology. *Int. J. Epidemiol.* **20**(3), 808–815 (1991)
46. Hernán, M.A., Taubman, S.L.: Does obesity shorten life? The importance of well-defined interventions to answer causal questions. *Int. J. Obes. (Lond.)* **32**(Suppl 3), S8–S14 (2008)
47. Hibbs, D.A., Jr.: On analyzing the effects of policy inter ventions: Box-Jenkins and Box-Tiao vs. structural equation models. *Sociol. Methodol.* **8**, 137–179 (1977). <http://links.jstor.org/sici?doi=0081-1750%281977%298%3C137%3AOATEOP%3E2.0.CO%3B2-K>
48. Hipel, K.W., Lettenmaier, D.P., McLeod, I.: Assessment of environmental impacts part one: *Interv. Anal. Environ. Manag.* **2**(6), 529–535 (1978)
49. Hites, R.A., Foran, J.A., Carpenter, D.O., Hamilton, M.C., Knuth, B.A., Schwager, S.J.: Global assessment of organic contaminants in farmed salmon. *Science* **303**(5655), 226–229 (2004)
50. Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging. *Stat. Sci.* **14**, 382–401 (1999)
51. Höfler, M.: The Bradford Hill considerations on causality: a counterfactual perspective. *Emerg. Themes Epidemiol.* **2**, 11 (2005)
52. Homer, J., Milstein, B., Wile, K., Trogdon, J., Huang, P., Labarthe, D., et al.: Simulating and evaluating local interventions to improve cardiovascular health. *Prev. Chronic Dis.* **7**(1), A18 (2010). www.cdc.gov/pcd/issues/2010/jan/08_0231.htm. Accessed 3 Nov 2015
53. Hora, S.: Eliciting probabilities from experts. In: Edwards, W., Miles, R.F., von Winterfeldt, D. (eds.) *Advances in Decision Analysis: From Foundations to Applications*, pp. 129–153. Cambridge University Press, New York (2007)
54. Hoyer, P.O., Hyvärinen, A., Scheines, R., Spirtes, P., Ramsey, J., Lacerda, G., Shimizu, S.: Causal discovery of linear acyclic models with arbitrary distributions. In: *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence - UAI*, Helsinki, Conference held 9–12 July 2008, pp. 282–289. <http://arxiv.org/ftp/arxiv/papers/1206/1206.3260.pdf>
55. Huitema, B.E., Van Houten, R., Manal, H.: Time-series intervention analysis of pedestrian countdown timer effects. *Accid Anal Prev.* **72**, 23–31 (2014). doi:10.1016/j.aap.2014.05.025
56. Ioannidis, J.P.A.: Why most published research findings are false. *PLoS Med.* **2**(8), e124 (2005). doi:10.1371/journal.pmed.0020124
57. James, N.A., Matteson, D.S.: ecp: an R package for nonparametric multiple change point analysis of multivariate data. *J. Stat. Softw.* **62**(7) (2014). <http://www.jstatsoft.org/v62/i07/paper>
58. Janzing, D., Balduzzi, D., Grosse-Wentrup, M., Scholkopf, B.: Quantifying causal influences. *Ann. Stat.* **41**(5), 2324–2358 (2013). doi:10.1214/13-AOS1145
59. Jiang, H., Livingston, M., Manton, E.: The effects of random breath testing and lowering the minimum legal drinking age on traffic fatalities in Australian states. *Inj. Prev.* **21**(2), 77–83 (2015). doi:10.1136/injuryprev-2014-041303
60. Joffe, M., Gambhir, M., Chadeau-Hyam, M., Vineis, P.: Causal diagrams in systems epidemiology. *Emerg. Themes Epidemiol.* **9**(1), 1 (2012). doi:10.1186/1742-7622-9-1
61. Kahneman, D.: *Thinking, Fast and Slow*. Farrar, Straus, and Giroux, New York (2011)
62. Kass-Hout, T.A., Xu, Z., McMurray, P., Park, S., Buckeridge, D.L., Brownstein, J.S., Finelli, L., Groseclose, S.L.: Application of change point analysis to daily influenza-like illness emergency department visits. *J. Am. Med. Inform. Assoc.* **19**(6), 1075–1081 (2012). doi:10.1136/amiainjnl-2011-000793
63. Kinnunen, E., Junntila, O., Haukka, J., Hovi, T.: Nationwide oral poliovirus vaccination campaign and the incidence of Guillain-BarréSyndrome. *Am. J. Epidemiol.* **147**(1), 69–73 (1998)
64. Kleck, G., Britt, C.L., Bordua, D.: The emperor has no clothes: an evaluation of interrupted time series designs for policy impact assessment. *J. Firearms Public Policy* **12**, 197–247 (2000)

65. Klein, L.R.: Regression systems of linear simultaneous equations. In: A Textbook of Econometrics, 2nd edn, pp. 131–196. Prentice-Hall, Englewood Cliffs (1974). ISBN:0-13-912832-8
66. Kline, R.B.: Principles and Practice of Structural Equation Modeling. Guilford Press, New York (1998)
67. Koller, D., Milch, B.: Multi-agent influence diagrams for representing and solving games. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (2001)
68. Lagarde, M.: How to do (or not to do) ... Assessing the impact of a policy change with routine longitudinal data. *Health Policy Plan.* **27**(1), 76–83 (2012). doi: 10.1093/heapol/czr004.
69. Lebre, S.: Package 'G1DBN': a package performing dynamic Bayesian network inference. CRAN repository, 19 Feb 2015. <https://cran.r-project.org/web/packages/G1DBN/G1DBN.pdf>
70. Lehrer, J.: Trials and errors: why science is failing us. *Wired.* <http://www.wired.co.uk/magazine/archive/2012/02/features/trials-and-errors?page=all>, 28 Jan 2012
71. Lei, H., Nahum-Shan, I., Lynch, K., Oslin, D., Murphy, S.A.: A "SMART" design for building individualized treatment sequences. *Ann. Rev. Clin. Psychol.* **8**, 14.1–14.28 (2012)
72. Linn, K.A., Laber, E.B., Stefanski LA.: iqLearn: interactive Q-learning in R. <https://cran.r-project.org/web/packages/iqLearn/vignettes/iqLearn.pdf> (2015)
73. Lipsitch, M., Tchetgen Tchetgen, E., Cohen, T.: Negative controls: a tool for detecting confounding and bias in observational studies. *Epidemiology* **21**(3), 383–388 (2010)
74. Lizier, J.T.: JIDT: an information-theoretic toolkit for studying the dynamics of complex systems. *Front. Robot. AI* **1**, 11 (2014); doi:10.3389/frobt.2014.00011 (preprint: arXiv:1408.3270), <http://arxiv.org/pdf/1408.3270.pdf>
75. Lu, C.Y., Soumerai, S.B., Ross-Degnan, D., Zhang, F., Adams, A.S.: Unintended impacts of a Medicaid prior authorization policy on access to medications for bipolar illness. *Med Care.* **48**(1), 4–9 (2010). doi:10.1097/MLR.0b013e3181bd4c10.
76. Lynch, W.D., Glass, G.V., Tran, Z.V.: Diet, tobacco, alcohol, and stress as causes of coronary artery heart disease: an ecological trend analysis of national data. *Yale J. Biol. Med.* **61**(5), 413–426 (1988)
77. MacLure, M.: Taxonomic axes of epidemiologic study designs: a refutationist perspective. *J. Clin. Epidemiol.* **44**(10), 1045–1053 (1991)
78. Madigan, D., Raftery, A.: Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Am. Stat. Assoc.* **89**, 1535–1546 (1994)
79. Madigan, D., Andersson, S.A., Perlman, M.D., Volinsky, C.M.: Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Commun. Stat. Theory Methods* **25**, 2493–2519 (1996)
80. McLeod et al. (2011) Time series analysis with R. <http://www.stats.uwo.ca/faculty/aim/tsar/tsar.pdf>
81. Montalto, A., Faes, L., Marinazzo, D.: MuTE: a MATLAB toolbox to compare established and novel estimators of the multivariate transfer entropy. *PLoS One* **9**(10), e109462 (2014). doi:10.1371/journal.pone.0109462
82. Moore, K.L., Neugebauer, R., van der Laan, M.J., Tager, I.B.: Causal inference in epidemiological studies with strong confounding. *Stat Med.* (2012). doi:10.1002/sim.4469
83. Morabia, A.: Hume, Mill, Hill, and the *sui generis* epidemiologic approach to causal inference. *Am. J. Epidemiol.* **178**(10), 1526–1532 (2013)
84. Morriss, R., Gask, L., Webb, R., Dixon, C., Appleby, L.: The effects on suicide rates of an educational intervention for front-line health professionals with suicidal patients (the STORM project). *Psychol. Med.* **35**(7), 957–960 (2005)
85. Nakahara, S., Katanoda, K., Ichikawa, M.: Onset of a declining trend in fatal motor vehicle crashes involving drunk-driving in Japan. *J. Epidemiol.* **23**(3), 195–204 (2013)
86. Neugebauer, R., Fireman, B., Roy, J.A., Raebel, M.A., Nichols, G.A., O'Connor, P.J.: Super learning to hedge against incorrect inference from arbitrary parametric assumptions in marginal structural modeling. *J. Clin. Epidemiol.* **66**(8 Suppl):S99–S109 (2013). doi:10.1016/j.jclinepi.2013.01.016

87. Nguefack-Tsague, G.: Using Bayesian networks to model hierarchical relationships in epidemiological studies. *Epidemiol. Health* **33**, e2011006 (2011). doi:10.4178/epih/e2011006. Epub 17 Jun 2011. <http://e-epih.org/journal/view.php?doi=10.4178/epih/e2011006>
88. Nuzzo, R.: Scientific method: statistical errors. P values, the 'gold standard' of statistical validity, are not as reliable as many scientists assume. *Nature* **506**, 150–152 (2014). doi:10.1038/506150a
89. Owczarek, T.: On modeling asymmetric multi-agent scenarios. In: IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Rende (Cosenza), 21–23 Sept 2009
90. Page, D., Ong, I.M.: Experimental design of time series data for learning from dynamic Bayesian networks. *Pac. Symp. Biocomput.* **2006**, 267–278 (2006)
91. Papana, A., Kyrtsov, C., Kugiumtzis, D., Cees, D.: Detecting causality in non-stationary time series using partial symbolic transfer entropy: evidence in financial data. *Comput. Econ.* **47**(3), 341–365 (2016). <http://link.springer.com/article/10.1007%2Fs10614-015-9491-x>
92. Pearl, J.: An introduction to causal inference. *Int. J. Biostat.* **6**(2), Article 7 (2010). doi:10.2202/1557-4679.1203
93. Polich, K., Gmytrasiewicz, P.: Interactive dynamic influence diagrams. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, New York. Article No. 34. <http://dl.acm.org/citation.cfm?id=1329166>
94. Rau, A.: Package 'ebdbNet': empirical Bayes estimation of dynamic Bayesian networks. CRAN repository, 19 Feb 2015. <https://cran.r-project.org/web/packages/ebdbNet/ebdbNet.pdf>
95. Rhomberg, L.: Hypothesis-based weight of evidence: an approach to assessing causation and its application to regulatory toxicology. *Risk Anal.* **35**(6), 1114–1124 (2015)
96. Robins, J.M., Hernán, M.A., Brumback, B.: Marginal structural models and causal inference in epidemiology. *Epidemiology* **11**(5), 550–560 (2000)
97. Robinson, J.W., Hartemink, A.J.: Learning non-stationary dynamic Bayesian networks. *J. Mach. Learn. Res.* **11**, 3647–3680 (2010)
98. Rothman, K.J., Lash, L.L., Greenland, S.: Modern Epidemiology, 3rd edn. Lippincott, Williams, & Wilkins. New York (2012)
99. Runge, J., Heitzig, J., Petoukhov, V., Kurths, J.: Escaping the curse of dimensionality in estimating multivariate transfer entropy. *Phys. Rev. Lett.* **108**, 258701. Published 21 June 2012
100. Samet, J.M., Bodurow, C.C. (eds.): Improving the Presumptive Disability Decision-Making Process for Veterans. Committee on Evaluation of the Presumptive Disability Decision-Making Process for Veterans, Board on Military and Veterans Health, Institute of Medicine. National Academies Press, Washington, DC (2008)
101. Sandri, M., Berchialla, P., Baldi, I., Gregori, D., De Blasi, R.A.: Dynamic Bayesian networks to predict sequences of organ failures in patients admitted to ICU. *J. Biomed. Inform.* **48**, 106–113 (2014). doi:10.1016/j.jbi.2013.12.008
102. Sarewitz, D.: Beware the creeping cracks of bias. *Nature* **485**, 149 (2012)
103. Sarewitz, D.: Reproducibility will not cure what ails science. *Nature* **525**(7568), 159 (2015)
104. Schwartz, J., Austin, E., Bind, M.A., Zanobetti, A., Koutrakis, P.: Estimating causal associations of fine particles with daily deaths in Boston. *Am. J. Epidemiol.* **182**(7), 644–650 (2015)
105. Scutari, M.: Learning Bayesian networks with the bnlearn R package. *J. Stat. Softw.* **35**(3) (2010). www.jstatsoft.org/v35/i03/paper. Last accessed 5 May 2015
106. Shen, Y., Cooper, G.F.: A new prior for Bayesian anomaly detection: application to biosurveillance. *Methods Inf. Med.* **49**(1), 44–53 (2010)
107. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2010)
108. Skrøvseth, S.O., Bellika, J.G., Godtliebsen, F.: Causality in scale space as an approach to change detection. *PLoS One.* **7**(12), e52253 (2012). doi:10.1371/journal.pone.0052253
109. Stebbings, J.H., Jr.: Panel studies of acute health effects of air pollution. II. A methodologic study of linear regression analysis of asthma panel data. *Environ. Res.* **17**(1), 10–32 (1978)

110. Steck, H.: Learning the Bayesian network structure: Dirichlet prior versus data. In: Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI2008), University of Helsinki City Centre Campus, Helsinki, 9–12 July 2008
111. Sun, X.: Assessing nonlinear granger causality from multivariate time series. *Mach. Learn. Knowl. Discov. Databases. Lect. Notes Comput. Sci.* **5212**, 440–455 (2008)
112. Swanson, S.A., Hernán, M.A.: How to report instrumental variable analyses (suggestions welcome). *Epidemiology* **24**(3), 370–374 (2013)
113. Tashiro, T., Shimizu, S., Hyvärinen, A., Washio T.: ParceLiNGAM: a causal ordering method robust against latent confounders. *Neural Comput.* **26**(1), 57–83 (2014)
114. Taubman, S.L., Allen, H.L., Wright, B.J., Baicker, K., Finkelstein, A.N.: Medicaid increases emergency-department use: evidence from Oregon's health insurance experiment. *Science.* **343**(6168), 263–268 (2014). doi:10.1126/science.1246183
115. Thornley, S., Marshall, R.J., Wells, S., Jackson, R.: Using directed acyclic graphs for investigating causal paths for cardiovascular disease. *J. Biomet. Biostat.* **4**, 182 (2013). doi:10.4172/2155-6180.1000182
116. Tong, S., Koller, D.: Active learning for structure in Bayesian networks. In: International Joint Conference on Artificial Intelligence (IJCAI), Seattle (2001)
117. Twardy, C.R., Nicholson, A.E., Korb, K.B., McNeil, J.: Epidemiological data mining of cardiovascular Bayesian networks. *J. Health Inform.* **1**(1), e3:1–e3:13 (2006)
118. Vicente, R., Wibral, M., Lindner, M., Pipa, G.: Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *J. Comput. Neurosci.* **30**(1), 45–67 (2011)
119. Voortman, M., Dash, D., Drudzsel, M.J.: Learning causal models that make correct manipulation predictions with time series data. In: Guyon, I., Janzing, D., Schölkopf, B. (eds.) JMLR Workshop and Conference Proceedings, vol. 6, pp. 257–266. NIPS 2008 Workshop on Causality. <http://jmlr.csail.mit.edu/proceedings/papers/v6/voortman10a/voortman10a.pdf> (2008)
120. Wang, J., Spitz, M.R., Amos, C.I., et al.: Method for evaluating multiple mediators: Mediating effects of smoking and COPD on the association between the CHRNA5-A3 Variant and Lung Cancer Risk. de Torres JP, ed. *PLoS One.* **7**(10), e47705 (2012). doi:10.1371/journal.pone.0047705
121. Watt, E.W., Bui, A.A.: Evaluation of a dynamic Bayesian belief network to predict osteoarthritic knee pain using data from the osteoarthritis initiative. *AMIA Annul. Symp. Proc.* **2008**, 788–792 (2008)
122. Wen, X., Rangarajan, G., Ding, M.: Multivariate Granger causality: an estimation framework based on factorization of the spectral density matrix. *Philos. Trans. R. Soc. A* **371**, 20110610 (2013). <http://dx.doi.org/10.1098/rsta.2011.0610>
123. Zhang, N.L.: Probabilistic inference in influence diagrams. *Comput. Intell.* **14**, 475–497 (1998)

Part VI

Codes of Practice and Factors of Safety

Conceptual Structure of Performance Assessments for Geologic Disposal of Radioactive Waste

44

Jon C. Helton, Clifford W. Hansen, and Cédric J. Salaberry

Abstract

A conceptual structure for performance assessments (PAs) for radioactive waste disposal facilities and other complex engineered facilities based on the following three basic conceptual entities is described: EN1, a probability space that characterizes aleatory uncertainty; EN2, a function that predicts consequences for individual elements of the sample space for aleatory uncertainty; and EN3, a probability space that characterizes epistemic uncertainty. This structure provides a basis for (i) a separation of the effects and implications of aleatory and epistemic uncertainty; (ii) the design, computational implementation, and documentation of a PA for a complex system; and (iii) informative uncertainty and sensitivity analyses. The implementation of this structure is illustrated with results from PAs for the Waste Isolation Pilot Plant for transuranic radioactive waste and the proposed Yucca Mountain repository for spent nuclear fuel and high-level radioactive waste.

Keywords

Aleatory uncertainty • Epistemic uncertainty • Performance assessment • Radioactive waste • Sensitivity analysis • Uncertainty analysis

J.C. Helton (✉)

Thermal Sciences and Engineering Department, Sandia National Laboratories, Albuquerque, NM, USA

e-mail: jchelto@sandia.gov

C.W. Hansen

Photovoltaics and Distributed Systems Department, Sandia National Laboratories, Albuquerque, NM, USA

e-mail: cwhanse@sandia.gov

C.J. Salaberry

Applied Systems Analysis and Research Department, Sandia National Laboratories, Albuquerque, NM, USA

e-mail: cnsalla@sandia.gov

Contents

1	Introduction	1504
2	Characterization of Uncertainty	1504
3	EN1, Representation of Aleatory Uncertainty	1506
4	EN2, Model That Estimates Consequences	1508
5	EN3, Representation of Epistemic Uncertainty	1509
6	Propagation and Display of Uncertainty	1513
7	Sensitivity Analysis	1530
8	Summary	1533
	References	1534

1 Introduction

A conceptual structure for performance assessments (PAs) for the geologic disposal of radioactive waste is described. Illustrations of this structure are provided by past PAs for (i) a repository for transuranic radioactive waste near Carlsbad, New Mexico, known as the Waste Isolation Pilot Plant (WIPP) [1] and (ii) a proposed repository for spent nuclear fuel and high-level radioactive waste at Yucca Mountain (YM), Nevada [2]. Probabilistic risk assessments (PRAs) for nuclear power plants and other complex engineered facilities also have a similar conceptual structure [3–5].

The following topics are addressed: (i) characterization of uncertainty (Sect. 2), (ii) representation of aleatory uncertainty (Sect. 3), (iii) estimation of consequences (Sect. 4), (iv) representation of epistemic uncertainty (Sect. 5), (v) propagation and display of uncertainty (Sect. 6), and (vi) sensitivity analysis (Sect. 7). The presentation then ends with a brief summary (Sect. 8).

This presentation is an edited and updated adaption of two prior conference presentations [6, 7] and also draws on additional material in two special journal issues devoted to the indicated WIPP and YM PAs [1, 2].

2 Characterization of Uncertainty

A PA for a geologic repository for radioactive waste or any other complex facility is an analysis intended to answer three questions about the facility and one additional question about the analysis itself. The initial three questions are Q1, “What could happen?”; Q2, “How likely is it to happen?”; and Q3, “What are the consequences if it does happen?”. Formally, the answers to the preceding three questions can be represented by a set of ordered triples of the form

$$(\mathcal{S}_i, p\mathcal{S}_i, \mathbf{cS}_i), i = 1, 2, \dots, n_S, \quad (44.1)$$

where (i) \mathcal{S}_i is a set of similar occurrences, (ii) the sets \mathcal{S}_i are disjoint (i.e., $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$) and $\cup_i \mathcal{S}_i$ contains everything that could potentially occur at the particular facility under consideration, (iii) $p\mathcal{S}_i$ is the probability for \mathcal{S}_i , and (iv) \mathbf{cS}_i is a vector of consequences associated with \mathcal{S}_i [8]. The fourth question is Q4,

“What is the uncertainty in the answers to the first three questions?” or, equivalently, “What is the level of confidence in the answers to the first three questions?”.

The set of ordered triples in Eq. (44.1) is often referred to as the Kaplan-Gärrick ordered triple representation for risk and, in essence, is an intuitive representation for a function f (i.e., a random variable) defined in association with a probability space $(\mathcal{A}, \mathbb{A}, p_A)$, where (i) \mathcal{A} is the set of everything that could occur in the particular universe under consideration, (ii) \mathbb{A} is the collection of subsets of \mathcal{A} for which probability is defined, and (iii) p_A is the function that defines probability for the elements of \mathbb{A} ([9], Sect. IV.3). Specifically, the sets \mathcal{S}_i are elements of \mathbb{A} with $\cup_i \mathcal{S}_i = \mathcal{A}$; $p_A(\mathcal{S}_i)$ is the probability $p_{\mathcal{S}_i}$ of \mathcal{S}_i ; and $f(\mathbf{a}_i)$ for a representative element \mathbf{a}_i of \mathcal{S}_i defines \mathbf{cS}_i . Another less commonly used possibility is that \mathbf{cS}_i is the expected value of f conditional on \mathcal{S}_i .

The uncertainty characterized by the sets \mathcal{S}_i and associated probabilities $p_{\mathcal{S}_i}$ in Eq. (44.1) introduced as answers to questions Q1 and Q2 is often referred to as aleatory uncertainty and results from a perceived randomness in future occurrences that could take place at the facility under consideration. The descriptor aleatory derives from the Latin *āleae*, which refers to games of dice, and is the reason for the notational use of “A” in the definition of the probability space $(\mathcal{A}, \mathbb{A}, p_A)$. Alternative designators for aleatory uncertainty include stochastic, type A, and irreducible. Question Q4 relates to uncertainty that results from a lack of knowledge with respect to the appropriateness of assumptions and/or parameter values used in an analysis. The basic idea is that an analysis has been developed to the point that it has a well-defined overall structure with fixed values for models and parameters, but uncertainty remains with respect to appropriate parameter values and possibly submodel structure or choice for use in this overall structure. This form of uncertainty is usually indicated by the descriptor epistemic, which derives from the Greek *epistēmē* for knowledge. Alternative descriptors for epistemic uncertainty include subjective, state of knowledge, type B, and reducible. Most analyses use probability to characterize epistemic uncertainty, which in turn means that in some way a probability space $(\mathcal{E}, \mathbb{E}, p_E)$ characterizing epistemic uncertainty must be defined and incorporated into the analysis. The representation of aleatory and epistemic uncertainty is an important, and sometimes contentious, part of the analysis of a complex system. As a result, an extensive literature exists on the incorporation of aleatory and epistemic uncertainty into analyses of complex systems (e.g., [10–23]).

A PA for a geologic repository for radioactive waste or any other complex facility is a very involved analysis. The understanding of such an analysis is greatly facilitated if it can be seen “in the large” before having to work through all the details of the implementation of the analysis. Fortunately, such an analysis is typically underlain by three basic entities or components: EN1, a probability space $(\mathcal{A}, \mathbb{A}, p_A)$ that characterizes aleatory uncertainty; EN2, a function f that estimates consequences for individual elements \mathbf{a} of the sample space \mathcal{A} for aleatory uncertainty; and EN3, a probability space $(\mathcal{E}, \mathbb{E}, p_E)$ that characterizes epistemic uncertainty [7, 24–26]. A recognition and understanding of these three basic entities makes it possible to understand the conceptual and computational structure of a

large PA without having basic concepts obscured by fine details of the analysis and leads to an analysis that results in informative uncertainty and sensitivity analyses. The nature and role of these three basic entities in PA and associated uncertainty and sensitivity analyses are elaborated on and illustrated in following sections.

3 EN1, Representation of Aleatory Uncertainty

As indicated in Sect. 2, aleatory uncertainty is formally characterized by a probability space $(\mathcal{A}, \mathbb{A}, p_A)$. The elements \mathbf{a} of \mathcal{A} are vectors

$$\mathbf{a} = [a_1, a_2, \dots, a_{nA}] \quad (44.2)$$

characterizing individual futures that potentially could occur at the facility under consideration. The set \mathcal{A} and the individual futures \mathbf{a} contained in \mathcal{A} are typically defined for some specified time interval. For example, the time interval $[0, 10^4 \text{ yr}]$ is specified in the regulations for WIPP [24, 27], and time intervals of $[0, 10^4 \text{ yr}]$ and $[0, 10^6 \text{ yr}]$ are specified in different parts of the regulations for the proposed YM repository ([26], Sect. 6). In contrast, futures are, in effect, defined for 1 year time intervals in PAs for nuclear power plants [3–5]. In practice, the probability space $(\mathcal{A}, \mathbb{A}, p_A)$ is usually defined by specifying distributions for the individual elements of \mathbf{a} . For notational purposes, it is often convenient to represent the distribution associated with the elements \mathbf{a} of \mathcal{A} with a density function $d_A(\mathbf{a})$.

For the 1996 PA supporting the Compliance Certification Application (CCA) for the WIPP, the only disruptions with sufficient potential for occurrence over the 10^4 yr regulatory period to merit inclusion in the analysis were drilling for petroleum resources and mining of potash above the excavated waste disposal drifts [27, 28]. As a consequence, the individual futures underlying the 1996 WIPP PA have the form

$$\mathbf{a} = \underbrace{[t_1, l_1, e_1, b_1, p_1, \mathbf{a}_1]}_{\text{1st intrusion}}, \underbrace{[t_2, l_2, e_2, b_2, p_2, \mathbf{a}_2], \dots, [t_n, l_n, e_n, b_n, p_n, \mathbf{a}_n]}_{\text{2nd intrusion}}, \dots, \underbrace{[t_n, l_n, e_n, b_n, p_n, \mathbf{a}_n]}_{\text{nth intrusion}}, t_{\min}, \quad (44.3)$$

where (i) n is the number of drilling intrusions, (ii) t_i is the time (yr) of the i th intrusion, (iii) l_i designates the location of the i th intrusion, (iv) e_i designates the penetration of an excavated or nonexcavated area by the i th intrusion, (v) b_i designates whether or not the i th intrusion penetrates pressurized brine in the Castile Formation, (vi) p_i designates the borehole plugging procedure used with the i th intrusion, (vii) \mathbf{a}_i designates the type of waste penetrated by the i th intrusion, and (viii) t_{\min} is the time (yr) at which potash mining occurs within the land withdrawal boundary. The distributions for times t_i for drilling intrusions and t_{\min} for potash mining are defined by Poisson processes; the distributions for the remaining elements of \mathbf{a} are defined on the basis of the properties of the waste disposal drifts associated with the WIPP and the geologic environment that surrounds these drifts.

Additional information on the elements of \mathbf{a} and their probabilistic characterization is available in Table III of Ref. [29] and in Ref. [27].

For the 2008 PA supporting the license application for the proposed YM repository,

$$\mathbf{a} = [nEW, nED, nII, nIE, nSG, nSF, \mathbf{a}_{EW}, \mathbf{a}_{ED}, \mathbf{a}_{II}, \mathbf{a}_{IE}, \mathbf{a}_{SG}, \mathbf{a}_{SF}], \quad (44.4)$$

where (i) nEW = number of early waste package (WP) failures, (ii) nED = number of early drip shield (DS) failures, (iii) nII = number of igneous intrusive events, (iv) nIE = number of igneous eruptive events, (v) nSG = number of seismic ground motion events, (vi) nSF = number of seismic fault displacement events, (vii) \mathbf{a}_{EW} = vector defining the nEW early WP failures, (viii) \mathbf{a}_{ED} = vector defining the nED early DS failures, (ix) \mathbf{a}_{II} = vector defining the nII igneous intrusive events, (x) \mathbf{a}_{IE} = vector defining the nIE igneous eruptive events, (xi) \mathbf{a}_{SG} = vector defining the nSG seismic ground motion events, and (xii) \mathbf{a}_{SF} = vector defining the nSF fault displacement events. In turn, the vectors \mathbf{a}_{EW} , \mathbf{a}_{ED} , \mathbf{a}_{II} , \mathbf{a}_{IE} , \mathbf{a}_{SG} , and \mathbf{a}_{SF} are of the form

$$\mathbf{a}_{EW} = [\mathbf{a}_{EW,1}, \mathbf{a}_{EW,2}, \dots, \mathbf{a}_{EW,nEW}], \mathbf{a}_{ED} = [\mathbf{a}_{ED,1}, \mathbf{a}_{ED,2}, \dots, \mathbf{a}_{ED,nED}], \quad (44.5)$$

$$\mathbf{a}_{II} = [\mathbf{a}_{II,1}, \mathbf{a}_{II,2}, \dots, \mathbf{a}_{II,nII}], \mathbf{a}_{IE} = [\mathbf{a}_{IE,1}, \mathbf{a}_{IE,2}, \dots, \mathbf{a}_{IE,nIE}], \quad (44.6)$$

and

$$\mathbf{a}_{SG} = [\mathbf{a}_{SG,1}, \mathbf{a}_{SG,2}, \dots, \mathbf{a}_{SG,nSG}], \mathbf{a}_{SF} = [\mathbf{a}_{SF,1}, \mathbf{a}_{SF,2}, \dots, \mathbf{a}_{SF,nSF}], \quad (44.7)$$

where (i) $\mathbf{a}_{EW,j}$ = vector defining early WP failure j for $j = 1, 2, \dots, nEW$, (ii) $\mathbf{a}_{ED,j}$ = vector defining early DS failure j for $j = 1, 2, \dots, nED$, (iii) $\mathbf{a}_{II,j}$ = vector defining igneous intrusive event j for $j = 1, 2, \dots, nII$, (iv) $\mathbf{a}_{IE,j}$ = vector defining igneous eruptive event j for $j = 1, 2, \dots, nIE$, (v) $\mathbf{a}_{SG,j}$ = vector defining seismic ground motion event j for $j = 1, 2, \dots, nSG$, and (vi) $\mathbf{a}_{SF,j}$ = vector defining seismic fault displacement event j for $j = 1, 2, \dots, nSF$. Definitions of the vectors $\mathbf{a}_{EW,j}$, $\mathbf{a}_{ED,j}$, $\mathbf{a}_{II,j}$, $\mathbf{a}_{IE,j}$, $\mathbf{a}_{SG,j}$, and $\mathbf{a}_{SF,j}$ and their associated probabilistic characterizations are given in Refs. [30–32].

As an example, the individual vectors $\mathbf{a}_{EW,j}$, $j = 1, 2, \dots, nEW$, appearing in the definition of \mathbf{a}_{EW} in Eq. (44.5) are defined by

$$\mathbf{a}_{EW,j} = [t_j, b_j, d_j] \quad (44.8)$$

and characterize the properties of failed WP j , where (i) t_j designates WP type (i.e., $t_j = 1 \sim$ commercial spent nuclear fuel (CSNF) WP, $t_j = 2 \sim$ codisposed (CDSF) WP), (ii) b_j designates percolation bin in which failed WP is located (i.e., $b_j = k \sim$ location of failed WP in percolation bin k for $k = 1, 2, 3, 4, 5$; see Fig. 44.2, Ref. [33]), and (iii) d_j designates whether the failed WP experiences nondripping or dripping conditions (i.e., $d_j = 0 \sim$ nondripping conditions and $d_j = 1 \sim$ dripping

conditions). In turn, the associated probabilities are based on the assumption that the number of early failed WPs follows a binomial distribution with the failed WPs distributed randomly over WP types, percolation bins, and nondripping/dripping conditions.

Although potentially complex, representations of aleatory uncertainty of the form illustrated for the WIPP CCA PA and the YM license application PA permit a complete and unambiguous description of what constitutes aleatory uncertainty and the probabilistic characterization of this uncertainty.

4 EN2, Model That Estimates Consequences

As indicated in Sect. 2, the second entity that underlines a PA for a complex system is a function f that estimates a vector $f(\mathbf{a})$ of consequences for individual elements \mathbf{a} of the sample space \mathcal{A} for aleatory uncertainty. In most analyses, the function f corresponds to a sequence of models for multiple physical processes that must be implemented and numerically evaluated in one or more computer programs. However, for notational and conceptual purposes, it is useful to represent this sequence of models as a function of elements of the sample space \mathcal{A} for aleatory uncertainty. In most analyses associated with radioactive waste disposal, the analysis results of interest are functions of time. In this situation, the model used to predict system behavior can be represented by $f(\tau|\mathbf{a})$, where τ corresponds to time and the indication of conditionality (i.e., “ $|\mathbf{a}$ ”) emphasizes that the results at time τ depend on the particular element of \mathcal{A} under consideration. In most analyses, $f(\tau|\mathbf{a})$ corresponds to a large number of results. In general, τ could also correspond to designators other than time such as spatial coordinates; however, it is possible that spatial coordinates and associated location-dependent results would simply be viewed as part of a very large number of results corresponding to $f(\tau|\mathbf{a})$.

As an example, the sequence of linked models that corresponds to f in the 1996 WIPP PA is indicated in Fig. 44.1. As summarized in Table 44.1, the models indicated in Fig. 44.1 represent a variety of physical processes and also involve a variety of mathematical structures (see Refs. [1, 28] for additional information on the models used in the 1996 WIPP PA).

As another example, a subset of the models that correspond to f in the 2008 PA for the proposed YM repository is indicated in Fig. 44.2. The particular configuration of models shown in Fig. 44.2 was used in the YM PA to calculate consequences associated with elements of the sample space \mathcal{A} for aleatory uncertainty that involved seismic disruptions. Similar model configurations were used to calculate consequences for early WP failures, early DS failures, and igneous intrusive events; a very different suite of models was used to calculate consequences for igneous eruptive events. The number of individual models used in the 2008 YM PA is too great to permit their individual description here. However, descriptions of these models are available in Refs. [35, 36] and in the more detailed model-specific technical reports cited in Refs. [35, 36].

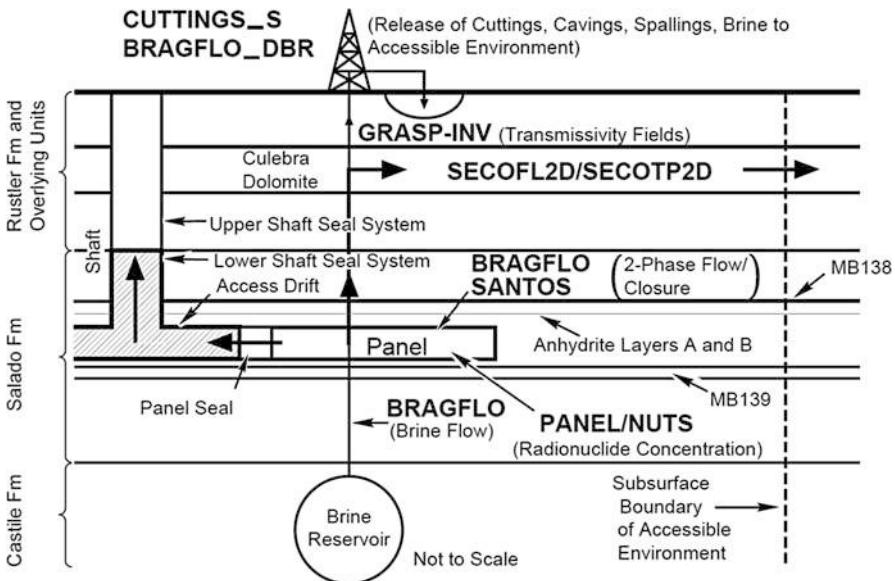


Fig. 44.1 Models used in the 1996 WIPP PA ([34], Fig. 44.4); see Table 44.1 for a description of individual models

As is evident from Fig. 44.1, Fig. 44.2, and Table 44.1, the function f that constitutes the second of the three entities that underlie a PA for radioactive waste repository can be quite complex.

5 EN3, Representation of Epistemic Uncertainty

As indicated in Sect. 2, epistemic uncertainty is formally characterized by a probability space $(\mathcal{E}, \mathbb{E}, p_E)$. The elements \mathbf{e} of \mathcal{E} are vectors of the form

$$\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M] = [e_1, e_2, \dots, e_{nE}], \quad (44.9)$$

where \mathbf{e}_A is a vector of epistemically uncertain quantities involved in the definition of the probability space $(\mathcal{A}, \mathbb{A}, p_A)$ for aleatory uncertainty that constitutes the first of the three entities that underlies a PA and \mathbf{e}_M is a vector of epistemically uncertain quantities involved in the definition of the function f that constitutes the second of the three entities that underlies a PA. The probability space $(\mathcal{E}, \mathbb{E}, p_E)$ for epistemic uncertainty and its associated density function $d_E(\mathbf{e})$ are usually developed through an expert review process that involves assigning a distribution to each element e_j of \mathbf{e} [37–44]. With the introduction of the epistemically uncertain quantities that constitute the elements of $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$, the notation for the density function $d_A(\mathbf{a})$ associated with the probability space $(\mathcal{A}, \mathbb{A}, p_A)$ for aleatory uncertainty becomes $d_A(\mathbf{a}|\mathbf{e}_A)$ to indicate the dependence on \mathbf{e}_A ; similarly, the notation for the time-dependent values for the function f becomes $f(\tau|\mathbf{a}, \mathbf{e}_M)$.

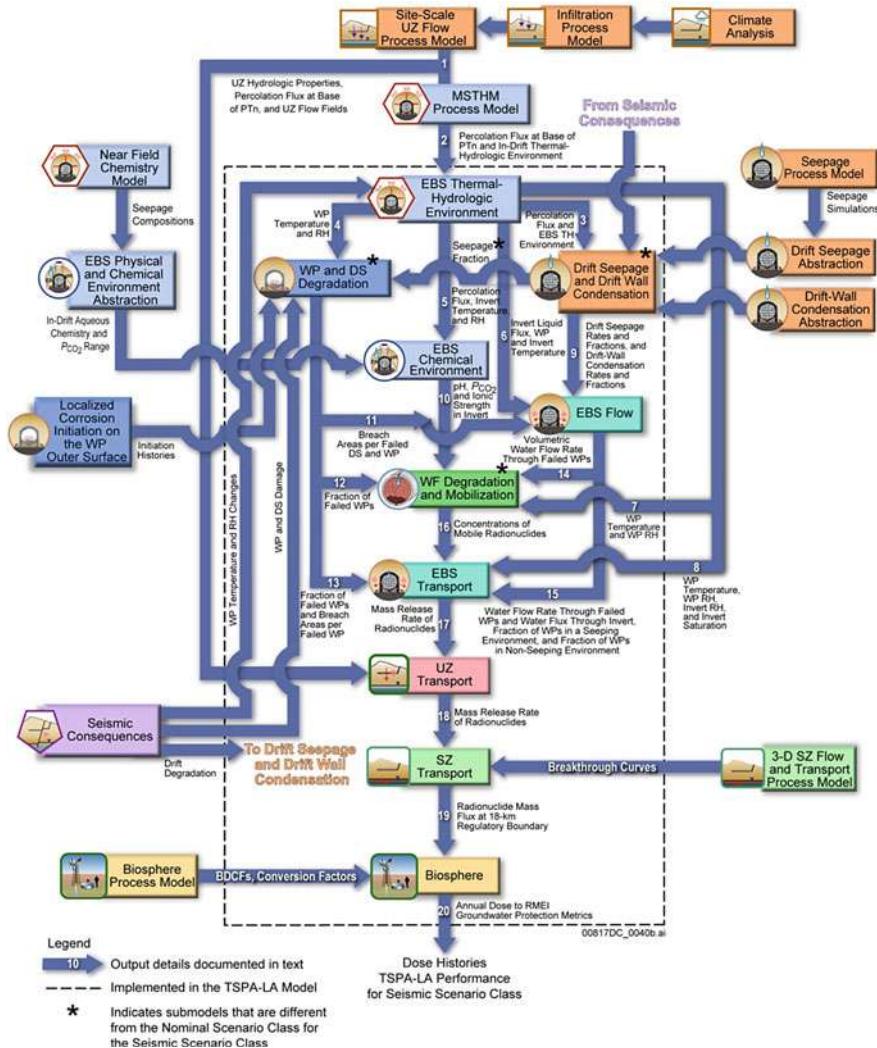


Fig. 44.2 Models used in the 2008 YM PA for seismic disruptions ([35], Fig. 6.1.4–6)

As examples, the probability spaces for epistemic uncertainty in the 1996 WIPP PA and the 2008 YM PA involved $nE = 57$ and $nE = 392$ epistemically uncertain quantities, respectively. Example elements of the vector \mathbf{e} for the WIPP and YM PAs are described in Tables 44.2 and 44.3. The example variables in Tables 44.2 and 44.3 were selected to include all variables that appear in the example sensitivity analyses discussed in Sect. 7.

As done in the WIPP and YM PAs, probability is the most widely used mathematical structure for the representation of epistemic uncertainty. However, a number of additional structures for the representation of epistemic uncertainty in

Table 44.1 Summary of individual models used in the 1996 WIPP PA ([34], Table 1)

BRAGFLO: Calculates multiphase flow of gas and brine through a porous heterogeneous reservoir. Uses finite difference procedures to solve system of nonlinear partial differential equations (PDEs) that describes the conservation of gas and brine along with appropriate constraint equations, initial conditions, and boundary conditions.
BRAGFLO-DBR: Special configuration of BRAGFLO model used to calculate dissolved radionuclide releases to the surface at the time of a drilling intrusion. Uses initial value conditions obtained from calculations performed with BRAGFLO and CUTTINGS_S.
CUTTINGS_S: Calculates quantity of radioactive material brought to the surface in cuttings, cavings, and spallings generated by a drilling intrusion. Spalling calculation uses initial value conditions obtained from calculations performed with BRAGFLO.
GRASP-INV: Generates transmissivity fields conditioned on measured transmissivity values and calibrated to steady-state and transient pressure data at well locations using an adjoint sensitivity and pilot-point technique.
NUTS: Solves system of PDEs for dissolved radionuclide transport in the vicinity of the repository. Uses brine volumes and flows calculated by BRAGFLO as input.
PANEL: Calculates rate of discharge and cumulative discharge of radionuclides from a waste panel through an intruding borehole. Uses brine volumes and flows calculated by BRAGFLO as input.
SANTOS: Determines quasistatic, large deformation, inelastic response of two-dimensional solids with finite element techniques. Used to determine porosity of waste as a function of time and cumulative gas generation, which is an input to calculations performed with BRAGFLO.
SECOFL2D: Calculates single-phase Darcy flow for groundwater flow in two dimensions based on a PDE for hydraulic head. Uses transmissivity fields generated by GRASP-INV.
SECOTP2D: Simulates transport of radionuclides in a fractured porous medium. Solves two PDEs: one provides a two-dimensional representation for convective and diffusive radionuclide transport in fractures and the other provides a one-dimensional representation for diffusion of radionuclides into the rock matrix surrounding the fractures. Uses flow fields calculated by SECOFL2D.

Table 44.2 Examples of the $nE = 57$ elements of the vector \mathbf{e} of epistemically uncertain variables in the 1996 WIPP PA (see [45], Table 1, for a complete listing of the indicated 57 epistemically uncertain variables and sources of detailed information on the individual variables)

<i>ANHPRM</i> – Logarithm of anhydrite permeability (m^2). <i>Distribution</i> : Student's with 5 degrees of freedom. <i>Range</i> : -21.0 to -17.1 . <i>Correlation</i> : -0.99 rank correlation with <i>ANHCOMP</i> (anhydrite compressibility).
<i>BHPRM</i> – Logarithm of borehole permeability (m^2). <i>Distribution</i> : Uniform. <i>Range</i> : -14.0 to -11.0 .
<i>HALPOR</i> – Halite porosity (dimensionless). <i>Distribution</i> : Piecewise uniform. <i>Range</i> : 0.001 to 0.03 .
<i>SHBCEXP</i> – Brooks-Corey pore distribution parameter for shaft (dimensionless). <i>Distribution</i> : Piecewise uniform. <i>Range</i> : 0.11 to 8.10 .
<i>WRGSSAT</i> – Residual gas saturation in waste (dimensionless). <i>Distribution</i> : Uniform. <i>Range</i> : 0 to 0.15 .
<i>WMICDFLG</i> – Pointer variable for microbial degradation of cellulose (dimensionless). <i>Distribution</i> : Discrete. $WMICDFLG = 1, 2, 3$ implies no microbial degradation of cellulose, microbial degradation of only cellulose, microbial degradation of cellulose, plastic and rubber.
<i>WTAUFAIL</i> – Shear strength of waste (Pa). <i>Distribution</i> : Uniform. <i>Range</i> : 0.05 to 10 .

Table 44.3 Examples of the $nE = 392$ elements of the vector **e** of epistemically uncertain variables in the 2008 YM PA (see [26], App. B, for a complete listing of the indicated 392 epistemically uncertain variables and sources of detailed information on the individual variables)

<i>CORRATSS.</i> Stainless steel corrosion rate ($\mu\text{m}/\text{yr}$). <i>Distribution:</i> Truncated log normal. <i>Range:</i> 0.01 to 0.51. <i>Mean/Median/Mode:</i> 0.267. <i>Standard Deviation:</i> 0.209.
<i>CSNFMASS.</i> Scale factor used to characterize uncertainty in radionuclide content of commercial spent nuclear fuel (CSNF) (dimensionless). <i>Distribution:</i> Uniform. <i>Range:</i> 0.85 to 1.4.
<i>DSNFMASS.</i> Scale factor used to characterize uncertainty in radionuclide content of defense spent nuclear fuel (DSNF) (dimensionless). <i>Distribution:</i> Triangular. <i>Range:</i> 0.45 to 2.9. <i>Mode:</i> 0.62.
<i>DTDRHUNC.</i> Selector variable used to determine the collapsed drift rubble thermal conductivity (dimensionless). <i>Distribution:</i> Discrete. <i>Range:</i> 1 to 2.
<i>HLWDRACD.</i> Effective rate coefficient (affinity term) for the dissolution of high-level waste (HLW) glass in codisposed spent nuclear fuel (CDSP) waste packages (WPs) under low pH conditions ($\text{g}/(\text{m}^2\text{d})$). <i>Distribution:</i> Triangular. <i>Range:</i> 8.41E 03 to 1.15E 07. <i>Mode:</i> 8.41E 03.
<i>IGRATE.</i> Frequency of intersection of the repository footprint by a volcanic event (yr^{-1}). <i>Distribution:</i> Piecewise uniform. <i>Range:</i> 0 to 7.76E-07.
<i>INFIL.</i> Pointer variable for determining infiltration conditions: 10th, 30th, 50th, or 90th percentile infiltration scenario (dimensionless). <i>Distribution:</i> Discrete. <i>Range:</i> 1 to 4.
<i>INRFRCTC.</i> The initial release fraction of ^{99}Tc in a CSNF waste package (dimensionless). <i>Distribution:</i> Triangular. <i>Range:</i> 0.0001 to 0.0026. <i>Mode:</i> 0.001.
<i>MICC14.</i> Groundwater Biosphere Dose Conversion Factor (BDCF) for ^{14}C in modern interglacial climate ($(\text{Sv}/\text{year})/(\text{Bq}/\text{m}^3)$). <i>Distribution:</i> Discrete. <i>Range:</i> 7.18E-10 to 2.56E-08. <i>Mean:</i> 1.93E-09. <i>Standard Deviation:</i> 1.85E-09.
<i>MICTC99.</i> Groundwater BDCF for ^{99}Tc in modern interglacial climate ($(\text{Sv}/\text{year})/(\text{Bq}/\text{m}^3)$). <i>Distribution:</i> Discrete. <i>Range:</i> 5.28E-10 to 2.85E-08. <i>Mean:</i> 1.12E-09. <i>Standard Deviation:</i> 1.26E-09.
<i>SCCTHR.</i> Residual stress threshold for stress corrosion cracking (MPa). <i>Distribution:</i> Uniform. <i>Range:</i> 315.9 to 368.55.
<i>SCCTHRP.</i> Residual stress threshold for stress corrosion crack (SCC) nucleation of Alloy 22 (as a percentage of yield strength in MPa) (dimensionless). <i>Distribution:</i> Uniform. <i>Range:</i> 90 to 105.
<i>SZFIPOVO.</i> Logarithm of flowing interval porosity in volcanic units (dimensionless). <i>Distribution:</i> Piecewise uniform. <i>Range:</i> -5 to -1. <i>Mean/Median/Mode:</i> -3.
<i>SZGWSPDPM.</i> Logarithm of the scale factor used to characterize uncertainty in groundwater-specific discharge (dimensionless). <i>Distribution:</i> Piecewise uniform. <i>Range:</i> -0.951 to 0.951.
<i>THERMCON.</i> Selector variable for one of three host-rock thermal conductivity scenarios (low, mean, and high) (dimensionless). <i>Distribution:</i> Discrete. <i>Range:</i> 1 to 3.
<i>WDGCA22.</i> Temperature-dependent slope term of Alloy 22 general corrosion rate (K). <i>Distribution:</i> Truncated normal. <i>Range:</i> 666 to 7731. <i>Mean:</i> 4905. <i>Standard Deviation:</i> 1413.
<i>WDGCUA22.</i> Variable for selecting distribution for general corrosion rate (low, medium, or high) (dimensionless). <i>Distribution:</i> Discrete. <i>Range:</i> 1 to 3.
<i>WDNSCC.</i> Stress corrosion cracking growth rate exponent (repassivation slope) (dimensionless). <i>Distribution:</i> Truncated normal. <i>Range:</i> 0.935 to 1.395. <i>Mean:</i> 1.165. <i>Standard Deviation:</i> 0.115.
<i>WDZOLID.</i> Deviation from median yield strength range for outer lid (dimensionless). <i>Distribution:</i> Truncated normal. <i>Range:</i> -3 to 3. <i>Mean:</i> 0. <i>Standard Deviation:</i> 1.

the presence of limited information have also been developed, including interval analysis, evidence theory, and possibility theory (e.g., [46–55]). As yet, these uncertainty structures have not been used in large-scale PAs.

6 Propagation and Display of Uncertainty

Two possibilities exist when the propagation and display of uncertainty are considered in a PA that involves a separation of aleatory uncertainty and epistemic uncertainty: (i) presentation of the effects of epistemic uncertainty conditional on a specific realization of aleatory uncertainty and (ii) presentation of the effects of aleatory uncertainty conditional on a specific realization of epistemic uncertainty. The presentation of the effects of epistemic uncertainty conditional on a specific realization of aleatory uncertainty is considered first.

Effects of epistemic uncertainty conditional on a specific realization of aleatory uncertainty. Epistemic uncertainty in an outcome of a PA conditional on a specific realization \mathbf{a} of aleatory uncertainty can be formally summarized with a cumulative distribution function (CDF) or a complementary cumulative distribution function (CCDF). For a real-valued analysis outcome $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$, the CDF and CCDF for y resulting from epistemic uncertainty in \mathbf{e}_M are defined by

$$p_E(\tilde{y} \leq y|\mathbf{a}) = \int_{\mathcal{E}} \underline{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_M)] d_E(\mathbf{e}_M) d\mathcal{E} \quad (44.10)$$

and

$$p_E(y < \tilde{y}|\mathbf{a}) = 1 - p_E(\tilde{y} \leq y|\mathbf{a}) = \int_{\mathcal{E}} \bar{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_M)] d_E(\mathbf{e}_M) d\mathcal{E}, \quad (44.11)$$

respectively, where (i) \mathbf{e}_M is a vector of epistemically uncertain quantities affecting the function f as indicated in conjunction with Eq. (44.9), (ii) $d_E(\mathbf{e}_M)$ is the density function associated with the probability space that characterizes the epistemic uncertainty associated with \mathbf{e}_M , (iii) \mathcal{E} and $d_E(\mathbf{e}_M)$ are restricted to possible values for \mathbf{e}_M , and (iv)

$$\underline{\delta}_y(\tilde{y}) = \begin{cases} 1 & \text{for } \tilde{y} \leq y \\ 0 & \text{otherwise} \end{cases} \quad \bar{\delta}_y(\tilde{y}) = 1 - \underline{\delta}_y(\tilde{y}) = \begin{cases} 1 & \text{for } y \leq \tilde{y} \\ 0 & \text{otherwise.} \end{cases} \quad (44.12)$$

Specifically, plots of the points $[y, p_E(\tilde{y} \leq y|\mathbf{a})]$ and $[y, p_E(y \leq \tilde{y}|\mathbf{a})]$ define the CDF and CCDF for $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$, with these plots being conditional on the vector \mathbf{a} from the sample space for aleatory uncertainty. In addition,

$$E_E(y|\mathbf{a}) = \int_{\mathcal{E}} f(\tau|\mathbf{a}, \mathbf{e}_M) d_E(\mathbf{e}_M) d\mathcal{E} \quad (44.13)$$

defines the expected value of $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$ over epistemic uncertainty.

In most PAs, the integrals in Eqs. (44.10), (44.11) and (44.13) are too complex to estimate with formal quadrature procedures, with the result that these integrals are typically estimated with procedures based on simple random sampling or Latin hypercube sampling. In turn, the indicated sampling procedures result in estimates of the form

$$p_E(\tilde{y} \leq y|\mathbf{a}) \cong \sum_{k=1}^n \underline{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_{Mk})]/n, \quad (44.14)$$

$$p_E(y < \tilde{y}|\mathbf{a}) \cong \sum_{k=1}^n \bar{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_{Mk})]/n, \quad (44.15)$$

and

$$E_E(y|\mathbf{a}) \cong \sum_{k=1}^n f(\tau|\mathbf{a}, \mathbf{e}_{Mk})/n, \quad (44.16)$$

where \mathbf{e}_{Mk} , $k = 1, 2, \dots, n$, is a sample from \mathcal{E} obtained in consistency with the density function $d_E(\mathbf{e}_M)$ for epistemic uncertainty.

Dose (mrem/yr) to the reasonably maximally exposed individual (RMEI) determined in the 2008 YM PA under the assumption of nominal (i.e., undisturbed) conditions provides an illustration of the results formally defined in Eqs. (44.10–44.16). In the 2008 YM PA, nominal conditions correspond to the occurrence of the aleatory future \mathbf{a}_N defined in Eq. (44.4) for which no disruptions of any type occur (i.e., the future in which $nEW = nED = nII = nIE = nSG = nSF = 0$). The 2008 YM PA used a Latin hypercube sample (LHS)

$$\mathbf{e}_k = [\mathbf{e}_{Ak}, \mathbf{e}_{Mk}], k = 1, 2, \dots, n = 300, \quad (44.17)$$

of size 300 in the propagation of epistemic uncertainty [26, 56, 57]. Latin hypercube sampling operates in the following manner to generate a sample of size n from the distributions D_1, D_2, \dots, D_{nE} associated with the elements of $\mathbf{e} = [e_1, e_2, \dots, e_{nE}]$. The range of each variable e_l is divided into n disjoint intervals of equal probability, and one value e_{kl} is randomly selected from each interval. The n values for e_1 are randomly paired without replacement with the n values for e_2 to produce n pairs. These pairs are then randomly combined without replacement with the n values for e_3 to produce n triples. This process is continued until a set of $n nE$ -tuples $\mathbf{e}_k = [e_{k1}, e_{k2}, \dots, e_{knE}], k = 1, 2, \dots, n$, is obtained, with this set constituting the LHS. Owing to its efficient stratification properties, Latin hypercube sampling has also been used for the propagation of epistemic uncertainty in the 1996 WIPP PA [45], the NUREG-1150 probabilistic risk assessments for five nuclear power stations [3, 4], and many other analyses.

As indicated in Sect. 5, only elements of the vector \mathbf{e}_M in Eq. (44.17) are involved in this example for dose to the RMEI as the elements of the vector \mathbf{e}_A are involved in the definition of probability distributions related to the characterization of aleatory uncertainty. In consistency with the notation used in the 2008 YM PA, $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ is used to represent dose to the RMEI at time τ conditional on (i) the future corresponding to \mathbf{a}_N and (ii) the values for epistemically uncertain quantities contained in the vector \mathbf{e}_M . A complex sequence of calculations similar to those indicated in Fig. 44.2 is used to determine

$$D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}) \text{ for } 0 \leq \tau \leq 10^6 \text{ yr and } k = 1, 2, \dots, n = 300. \quad (44.18)$$

In turn, approximations to the CDF, CCDF, and expected value for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ as defined in Eqs. (44.14), (44.15) and (44.16) are given by

$$p_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M) \leq D] \cong \sum_{k=1}^{n=300} \underline{\delta}_D[D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})]/n, \quad (44.19)$$

$$p_E[D < D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)] \cong \sum_{k=1}^{n=300} \bar{\delta}_D[D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})]/n, \quad (44.20)$$

$$E_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)] \cong \sum_{k=1}^{n=300} D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})/n, \quad (44.21)$$

and illustrated in Fig. 44.3a for $\tau = 600,000$ yr.

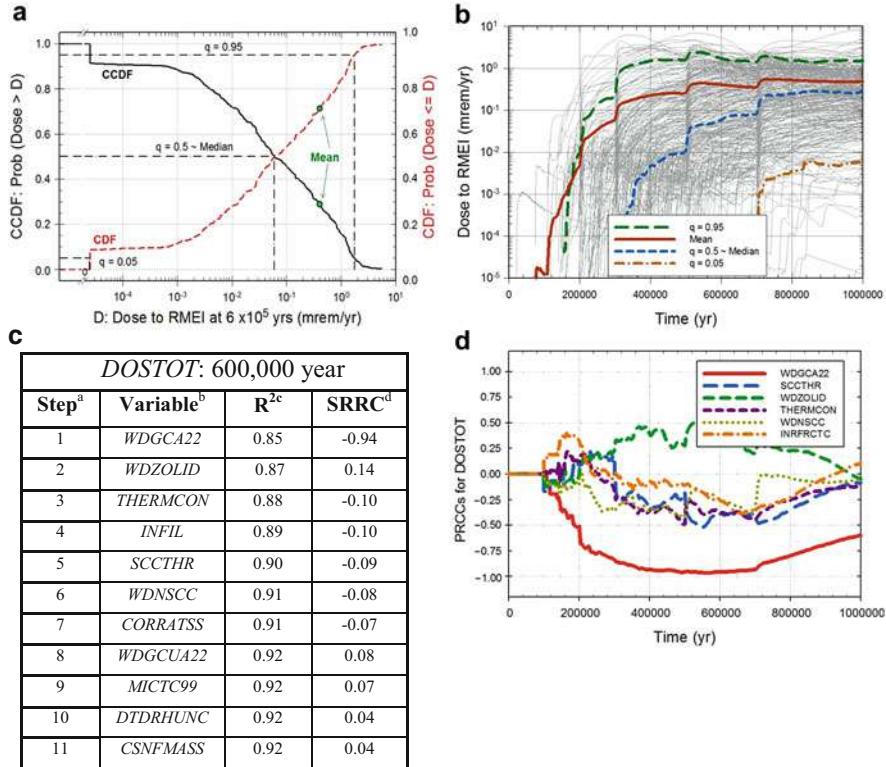
Also identified in Fig. 44.3a are selected quantile values for $q = 0.05, 0.5$, and 0.95 . Once a CDF is available, quantile values are easily determined as indicated in Fig. 44.3a by (i) starting at a desired quantile value q on the ordinate of the CDF plot, (ii) drawing a horizontal line to the CDF, and (iii) then dropping a vertical line to the abscissa to obtain the q -quantile of the variable under consideration. More formally and with the same notation as used in Eqs. (44.10–44.16), the q quantile value $Q_{Eq}[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ for $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$ conditional on \mathbf{a} and arising from the epistemic uncertainty associated with \mathbf{e}_M is the value of y such that

$$\begin{aligned} q &= \int_{\mathcal{E}} \underline{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_M)] d_E(\mathbf{e}_M) d\mathcal{E} \\ &\cong \sum_{k=1}^n \underline{\delta}_y[f(\tau|\mathbf{a}, \mathbf{e}_{Mk})]/n. \end{aligned} \quad (44.22)$$

Specifically, the quantiles $Q_{Eq}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ at $\tau = 600,000$ yr indicated in Fig. 44.3a are defined by the values of D such that

$$q \cong \sum_{k=1}^{n=300} \underline{\delta}_D[D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})]/n \quad (44.23)$$

for $q = 0.05, 0.5$, and 0.95 .



- a: Steps in stepwise rank regression analysis
 b: Variables listed in order of selection in stepwise regression
 c: Cumulative R^2 value with entry of each variable into regression model
 d: Standardized rank regression coefficients (SRRCs) in final regression model

Fig. 44.3 Estimates obtained with an LHS of size $n = 300$ of the epistemic uncertainty in dose $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ to RMEI conditional on the realization \mathbf{a}_N of aleatory uncertainty corresponding to nominal conditions: (a) CDF and CCDF for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ with $\tau = 600,000$ yr ([58], Fig. 1b); (b) individual results $D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})$, $k = 1, 2, \dots, n$, and associated expected (mean) values $E_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ and quantile values $Q_{E_q}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$, $q = 0.05, 0.5, 0.95$, for $0 \leq \tau \leq 10^6$ yr ([58], Fig. 1a); (c) stepwise rank regressions for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ at $\tau = 600,000$ yr ([33], Table 10); and (d) partial rank correlation coefficients (PRCCs) for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ for $0 \leq \tau \leq 10^6$ yr ([33], Fig. 23b)

As is the case for many PA results, $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ is a function of time. Specifically, $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ is determined for $0 \leq \tau \leq 10^6$ yr in the 2008 YM PA. Although plots of the form shown in Fig. 44.3a provide a complete display of the effects of epistemic uncertainty at the time under consideration (e.g., $\tau = 600,000$ yr in Fig. 44.3a), they are not practical for displaying the effects of epistemic uncertainty at a large number of times. In such situations, an effective presentation format is to use a single plot to present both (i) the

individual results $f(\tau|\mathbf{a}, \mathbf{e}_M), k = 1, 2, \dots, n$, calculated for the time interval $t_{mn} \leq \tau \leq t_{mx}$ under consideration and (ii) the expected value $E_E[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ and selected quantile values $Q_{Eq}[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ for $f(\tau|\mathbf{a}, \mathbf{e}_M)$ also calculated for the time interval $t_{mn} \leq \tau \leq t_{mx}$. As an example, this presentation format is illustrated in Fig. 44.3b for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ and $0 \leq \tau \leq 10^6$ yr, where (i) the lighter lines correspond to the results $D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}), k = 1, 2, \dots, n = 300$, calculated as indicated in conjunction with Eq. (44.18) and (ii) the heavier lines correspond to the expected value $E_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ calculated as indicated in Eq. (44.21) and the quantiles $Q_{Eq}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)], q = 0.05, 0.5, 0.95$, calculated as indicated in Eq. (44.23). The presentation format illustrated in Fig. 44.3b provides an effective way to present a large amount of information in a small amount of space. The sensitivity analysis results in Fig. 44.3c and d are discussed in Sect. 7.

Latin hypercube sampling and the propagation of epistemic uncertainty. Latin hypercube sampling was developed for use in the analysis of computationally demanding analysis structures for which computational demands severely limit the number of system simulations that can be performed ([56]; [59], App. A). As a result, the LHS size in use in such analyses is often perceived as being small relative to the number of epistemically uncertain variables being sampled (e.g., an LHS of size 100 from 57 variables in the 1996 WIPP PA and an LHS of size 300 from 392 variables in the 2008 YM PA). In response to this concern, a replicated sampling procedure has been developed to provide a way to assess the adequacy of the LHS size used in the analysis of a complex and computationally demanding analysis structure ([60]; [45], Sect. 7). Specifically, the goal of the procedure is to determine how much variability exists in estimated analysis outcomes when different LHSs of size n are used.

The procedure is based on independently generating LHSs of size n with different random seeds nR times. At an intuitive level, the adequacy of the sample size can be assessed by simply comparing the results obtained with the individual samples. At a more formal level, the t -distribution can be used to place confidence intervals around summary results (e.g., expected values over epistemic uncertainty) obtained from individual samples. Specifically, assume that the LHS has been replicated nR times to produce $r = 1, 2, \dots, nR$ independently generated LHSs

$$\mathbf{e}_{rk} = [\mathbf{e}_{A_{rk}}, \mathbf{e}_{M_{rk}}], k = 1, 2, \dots, n, \quad (44.24)$$

of size n and that C_r is an analysis result of interest obtained for replicated sample r . For example, C might be the expected value $E_E[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ at time τ that is estimated with the use of Latin hypercube sampling, and $C_r = E_{Er}[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ would be the estimate for $C = E_E[f(\tau|\mathbf{a}, \mathbf{e}_M)]$ obtained with replicated sample r . Then,

$$\bar{C} = \sum_{r=1}^{nR} C_r / nR \text{ and } SE(\bar{C}) = \left\{ \sum_{r=1}^{nR} [C_r - \bar{C}]^2 / nR(nR - 1) \right\}^{1/2} \quad (44.25)$$

provide an additional estimate of C and an estimate of the standard error associated with this additional estimate. The t -distribution with $nR - 1$ degrees of freedom can now be used to place confidence intervals around the estimate \bar{C} for C in Eq. (44.25). Specifically, the $1 - \alpha$ confidence interval for C is given by $\bar{C} \pm t_{1-\alpha/2}SE(\bar{C})$, where $t_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the t -distribution with $nR - 1$ degrees of freedom. For example, $t_{1-\alpha/2} = 4.303$ for $\alpha = 0.05$ and $nR = 3$.

The 2008 YM PA used $nR = 3$ replicated LHSs of size 300 to assess the adequacy of an LHS of size 300 from the 392 epistemically uncertain variables under consideration. As an example, the results obtained for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ are shown in Fig. 44.4. Specifically, the results for the individual replicates (i.e., $E_{Er}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ and $Q_{Eqr}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ for $q = 0.05, 0.5, 0.95$) are shown in Fig. 44.4a, and the time-dependent 95% confidence intervals for $E_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ are shown in Fig. 44.4b. As indicated by the results in Fig. 44.4, an LHS of size 300 is adequate for assessing the epistemic uncertainty present in estimates for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$. There is a substantial amount of variability in the LHS results prior to about 300,000 yr; however, this is due to the presence of results that are either 0 or so small as to be below the resolution of the calculation (see Fig. 44.3b).

As part of the 2008 YM PA, the effects of sample size for a large number of different analysis outcomes were assessed with the three indicated replicated LHSs (e.g., see [30], Fig. 10; [31], Figs. 10 and 19; [32], Figs. 11 and 23; [61], Figs. 5 and 10; [62]). In each case, an LHS of size 300 is adequate for assessing the epistemic uncertainty present in the analysis outcome under consideration.

Another approach to assess sample size adequacy is to observe the results obtained with a sequence of LHSs of increasing size. An adequate sample size then corresponds to the sample size at which estimates of the analysis results of primary interest stabilize with respect to increasing sample size (e.g., [63], Figs. 13 and 14). Unlike simple random sampling, the size of an LHS cannot be extended by sampling additional sample elements without consideration of the sample elements already present in the sample. However, a technique does exist to extend an LHS of a given size to an LHS of a larger size [64].

Another concern that arises from the small sample sizes used with Latin hypercube sampling is that these small sample sizes could result in spurious correlations between variables within a sample. However, this problem can be controlled with a restricted pairing technique developed by Iman and Conover to control rank correlations between variables within random samples and LHSs [65, 66]. Specifically, this technique can be used to insure that a sample has (i) rank correlations close to zero for uncorrelated variables and (ii) desired rank correlations between correlated variables.

Effects of aleatory uncertainty conditional on a specific realization of epistemic uncertainty. The formal representation of the effects of aleatory uncertainty conditional on a specific realization of epistemic uncertainty is similar to the formal

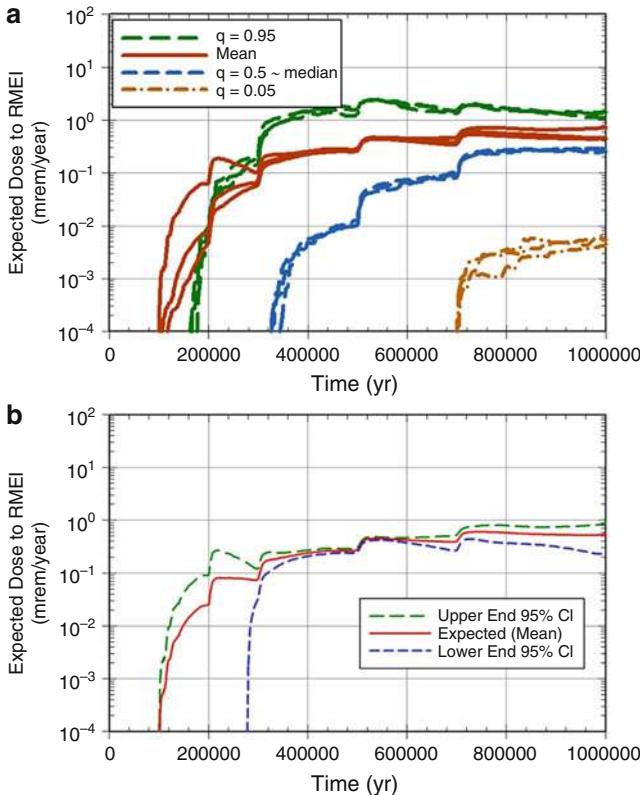


Fig. 44.4 Illustration of replicated sampling to determine the adequacy of an LHS of size 300 for the assessment of the epistemic uncertainty present in estimates for the dose $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ to the RMEI conditional on undisturbed (i.e., nominal) repository conditions as indicated by the vector \mathbf{a}_N ([58], Fig. 4): (a) expected values $E_{E_T}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ and quantiles $Q_{Eqr}[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)], q = 0.05, 0.5, 0.95$, for $D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)$ obtained for each of the three replicated samples indicated in Eq. (44.24) and (b) time-dependent 95% confidence intervals (i.e., for $\alpha = 0.05$) for $E_E[D_N(\tau|\mathbf{a}_N, \mathbf{e}_M)]$ obtained as indicated in conjunction with Eq. (44.25)

representation of the effects of epistemic uncertainty conditional on a specific realization of aleatory uncertainty but with the roles of the probability spaces for epistemic uncertainty and aleatory uncertainty reversed. Specifically, aleatory uncertainty in an outcome of a PA conditional on a specific realization $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ of epistemic uncertainty can be formally summarized with a CDF or a CCDF. For a real-valued analysis outcome $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$, the CDF and CCDF for y resulting from aleatory uncertainty are defined by

$$p_A(\tilde{y} \leq y|\mathbf{e}) = \int_{\mathcal{A}} \delta_y [f(\tau|\mathbf{a}, \mathbf{e}_M)] d_A(\mathbf{a}|\mathbf{e}_A) d\mathcal{A} \quad (44.26)$$

and

$$p_A(y < \tilde{y}|\mathbf{e}) = 1 - p_A(\tilde{y} \leq y|\mathbf{e}) = \int_{\mathcal{A}} \bar{\delta}_y [f(\tau|\mathbf{a}, \mathbf{e}_M)] d_A(\mathbf{a}|\mathbf{e}_A) d\mathcal{A}, \quad (44.27)$$

respectively, where (i) \mathbf{a} is a vector defining one realization of aleatory uncertainty, (ii) $d_A(\mathbf{a}|\mathbf{e}_A)$ is the density function conditional on \mathbf{e}_A associated with the probability space that characterizes the aleatory uncertainty associated with \mathbf{a} , and (iii) $\underline{\delta}_y(\tilde{y})$ and $\bar{\delta}_y(\tilde{y})$ are defined the same as in Eq.(44.12). In turn, plots of the points $[y, p_A(\tilde{y} \leq y|\mathbf{e})]$ and $[y, p_A(y \leq \tilde{y}|\mathbf{e})]$ define the CDF and CCDF for $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$, with these plots being conditional on the vector $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ from the sample space for epistemic uncertainty. Similarly,

$$E_A(y|\mathbf{e}) = \int_{\mathcal{A}} f(\tau|\mathbf{a}, \mathbf{e}_M) d_A(\mathbf{a}|\mathbf{e}_A) d\mathcal{A} \quad (44.28)$$

defines the expected value of $y = f(\tau|\mathbf{a}, \mathbf{e}_M)$ over aleatory uncertainty.

In most PAs, the integrals in Eqs. (44.26), (44.27) and (44.28) are too complex to estimate with formal quadrature procedures, with the result that these integrals are typically estimated with procedures based on some variant of random sampling or stratified sampling. In turn, the indicated sampling procedures result in estimates of the form

$$p_A(\tilde{y} \leq y|\mathbf{e}) \cong \begin{cases} \sum_{i=1}^m \underline{\delta}_y [f(\tau|\mathbf{a}_i, \mathbf{e}_M)] / m \\ \sum_{i=1}^{\tilde{m}} \underline{\delta}_y [f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_M)] p_A(\mathcal{A}_i|\mathbf{e}_A), \end{cases} \quad (44.29)$$

$$p_A(y < \tilde{y}|\mathbf{e}) \cong \begin{cases} \sum_{i=1}^m \bar{\delta}_y [f(\tau|\mathbf{a}_i, \mathbf{e}_M)] / m \\ \sum_{i=1}^{\tilde{m}} \bar{\delta}_y [f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_M)] p_A(\mathcal{A}_i|\mathbf{e}_A), \end{cases} \quad (44.30)$$

and

$$E_A(y|\mathbf{e}) \cong \begin{cases} \sum_{i=1}^m f(\tau|\mathbf{a}_i, \mathbf{e}_M) / m \\ \sum_{i=1}^{\tilde{m}} f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_M) p_A(\mathcal{A}_i|\mathbf{e}_A), \end{cases} \quad (44.31)$$

where (i) $\mathbf{a}_i, i = 1, 2, \dots, m$, is a sample from \mathcal{A} generated in consistency with the density function $d_A(\mathbf{a}|\mathbf{e}_A)$ for aleatory uncertainty and (ii) the sets (i.e., strata) $\mathcal{A}_i, i = 1, 2, \dots, \tilde{m}$, satisfy $\cup_i \mathcal{A}_i = \mathcal{A}$ with $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for $i \neq j$, $\tilde{\mathbf{a}}_i$ is a representative element of \mathcal{A}_i , and $p_A(\mathcal{A}_i|\mathbf{e}_A)$ is the probability of \mathcal{A}_i . The summations with \mathbf{a}_i and $\tilde{\mathbf{a}}_i$ correspond to approximations with simple random or quasi-random sampling and stratified sampling, respectively.

Use of the approximations with the sets \mathcal{A}_i and associated values $\tilde{\mathbf{a}}_i$ in Eqs. (44.29), (44.30) and (44.31) corresponds to use of the ordered triple representation for risk in Eq.(44.1) with $\mathcal{S}_i = \mathcal{A}_i$, $p_{\mathcal{S}_i} = p_A(\mathcal{A}_i|\mathbf{e}_A)$, and $c_{\mathcal{S}_i} = f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_M)$. In effect, representations of this form are being employed when an event tree is used to represent the effects of aleatory uncertainty, with each path

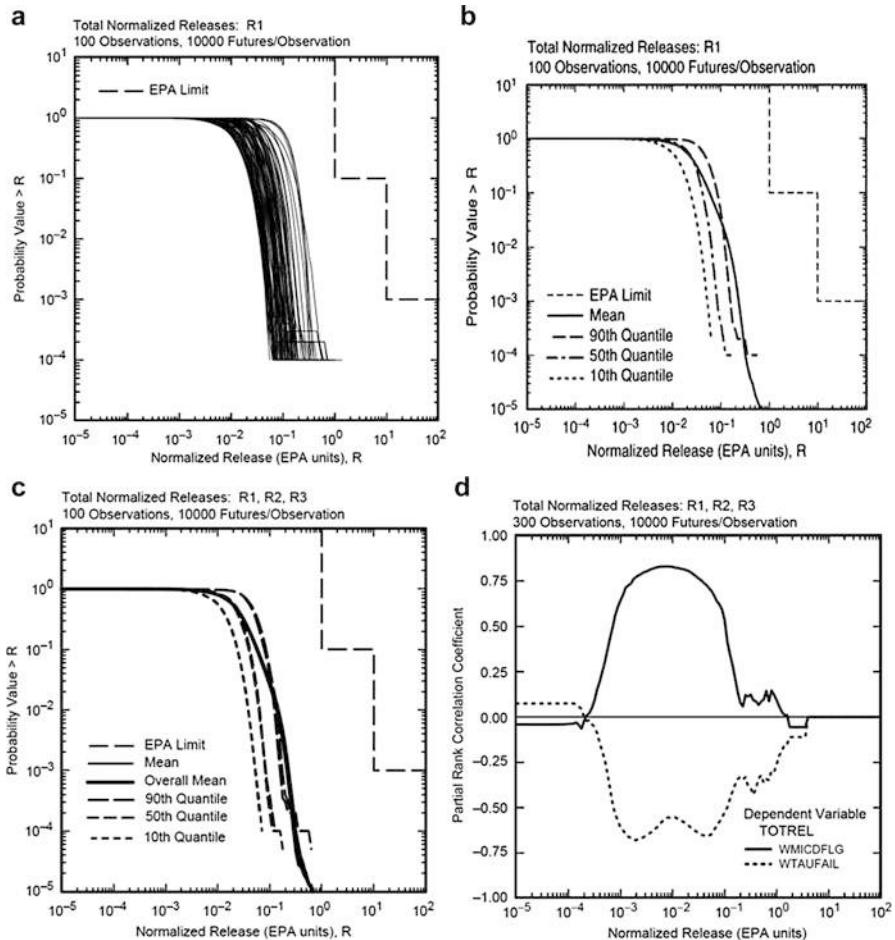


Fig. 44.5 Distribution of CCDFs for normalized release to the accessible environment over 10^4 yr obtained in the 1996 WIPP PA: (a) individual CCDFs for LHS of size 100 ([67], Fig. 1a), (b) mean and quantile curves for CCDFs in Frame a ([67], Fig. 1b), (c) replicated mean and quantile curves for CCDFs in Frame a for three independent LHSs (i.e., R1, R2, R3) of size 100 ([67], Fig. 2a), and (d) partial rank correlation coefficients (PRCCs) for all 300 CCDFs used to generate results in Frame c ([67], Fig. 5)

through the tree being used to both define one set \mathcal{A}_i and determine the probability of this set.

As an example, the CCDFs for normalized radionuclide release to the accessible environment obtained in the 1996 WIPP PA are shown in Fig. 44.5a. In this example, the individual CCDFs are calculated as indicated in Eq. (44.30) with a random sample of size 10^4 from the sample space for aleatory uncertainty, and the distribution of CCDFs results from repeating this calculation for each element $\mathbf{e}_k = [\mathbf{e}_{Ak}, \mathbf{e}_{Mk}]$ of an LHS of size $n = 100$ from the sample space for epistemic

uncertainty (see Ref. [67] for computational details). As indicated immediately after Eq. (44.27), each CCDF in Fig. 44.5a is a plot of points of the form $[y, p_A(y \leq \tilde{y}|\mathbf{e}_k)]$ with y corresponding to normalized release R at 10,000 yr. Specifically, the indicated release is the integrated (i.e., total) release that takes place from repository closure (i.e., time 0 yr) out to 10,000 yr; thus, in the formal model representation $f(\tau|\mathbf{a}, \mathbf{e}_M)$ used in Eqs. (44.10–44.31), the time variable τ corresponds to 10,000 yr in Fig. 44.5a.

Displays of the form shown in Fig. 44.5a provide a visual impression of the effects of epistemic uncertainty in the determination of the presented results. A more quantitative presentation of the effects of epistemic uncertainty is provided by a plot of the expected value and selected quantile values (e.g., $q = 0.05, 0.5, 0.95$) for the exceedance probabilities associated with individual values on the abscissa (e.g., normalized release R as in Fig. 44.5a). In general and with the same notation as used in Eqs. (44.29), (44.30) and (44.31), the indicated expected value $E_E[p_A(y < \tilde{y}|\mathbf{e})]$ is given by

$$\begin{aligned} E_E[p_A(y < \tilde{y}|\mathbf{e})] &= \int_{\mathcal{E}} \left\{ \int_{\mathcal{A}} \bar{\delta}_y [f(\tau|\mathbf{a}, \mathbf{e}_M)] d_A(\mathbf{a}|\mathbf{e}) d\mathcal{A} \right\} d_E(\mathbf{e}) d\mathcal{E} \\ &\cong \begin{cases} \sum_{k=1}^n \left\{ \sum_{i=1}^m \bar{\delta}_y [f(\tau|\mathbf{a}_i, \mathbf{e}_{Mk})] / m \right\} / n \\ \sum_{k=1}^n \left\{ \sum_{i=1}^{\tilde{m}} \bar{\delta}_y [f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_{Mk})] p_A(\mathcal{A}_i|\mathbf{e}_{Ak}) \right\} / n \end{cases} \quad (44.32) \end{aligned}$$

and the q quantile value $Q_{Eq}[p_A(y < \tilde{y}|\mathbf{e})]$ is the value of $p = p_A(y < \tilde{y}|\mathbf{e})$ such that

$$\begin{aligned} q &= \int_{\mathcal{E}} \underline{\delta}_p [p_A(y < \tilde{y}|\mathbf{e})] d_E(\mathbf{e}) d\mathcal{E} \\ &= \int_{\mathcal{E}} \underline{\delta}_p \left\{ \int_{\mathcal{A}} \bar{\delta}_y [f(\tau|\mathbf{a}, \mathbf{e}_M)] d_A(\mathbf{a}|\mathbf{e}_A) d\mathcal{A} \right\} d_E(\mathbf{e}) d\mathcal{E} \\ &\cong \begin{cases} \sum_{k=1}^n \underline{\delta}_p \left\{ \sum_{i=1}^m \bar{\delta}_y [f(\tau|\mathbf{a}_i, \mathbf{e}_{Mk})] / m \right\} / n \\ \sum_{k=1}^n \underline{\delta}_p \left\{ \sum_{i=1}^{\tilde{m}} \bar{\delta}_y [f(\tau|\tilde{\mathbf{a}}_i, \mathbf{e}_{Mk})] p_A(\mathcal{A}_i|\mathbf{e}_{Ak}) \right\} / n. \end{cases} \quad (44.33) \end{aligned}$$

As an example, mean and quantile (i.e., percentile) results for the CCDFs in Fig. 44.5a are presented in Fig. 44.5b.

Results analogous to those illustrated in Fig. 44.5a, b can also be obtained and presented for CDFs. However, CCDFs rather than CDFs usually provide the preferred presentation format for results obtained in PAs for two reasons. First, CCDFs provide an answer to questions of the form “How likely is it to be this bad

or worse?”, which is typically the type of question of most interest in a PA. Second, CCDFs facilitate the display of the probabilities associated with low probability but high consequence aleatory occurrences (especially when a \log_{10} scale is used on the ordinate); in contrast, such probabilities are difficult to obtain from a CDF owing to the lack of resolution in displayed probabilities as cumulative probability approaches 1.0.

As previously indicated, the 1996 WIPP PA used an LHS of size 100 in the propagation of epistemic uncertainty. To assess the adequacy of this sample size, the analysis was repeated with three replicated LHSs of size 100 denoted R1, R2, and R3. As shown in Fig. 44.5c, there is little variability in the results obtained with the individual replicated samples. The sensitivity analysis results with PRCCs in Fig. 44.5d are discussed in Sect. 7.

An interesting potential regulatory requirement is also illustrated in Fig. 44.5a, b. The two-step boundary labeled “EPA limit” in the upper right corner of Fig. 44.5a derives from a licensing regulation for the WIPP established by the US Environmental Agency (EPA) which requires for the time interval $[0, 10^4 \text{ yr}]$ after repository closure that (i) the probability of exceeding a normalized release of size $R = 1.0$ shall be less than 0.1 and (ii) the probability of exceeding a normalized release of size $R = 10.0$ shall be less than 0.001. As specified by the EPA, the indicated requirement is violated if the mean CCDF in Fig. 44.5b crosses the indicated boundary line (see Ref. [24] for additional details). A requirement of this type places stronger restrictions on system outcomes as the severity of these outcomes increases and is known as the Farmer limit line approach to the definition of acceptable risk [68–70].

An additional example of the representation of the effects of aleatory uncertainty conditional on a specific realization of epistemic uncertainty is provided by the treatment in the 2008 YM PA of expected dose $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ over aleatory uncertainty to the RMEI conditional on a realization $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ of epistemic uncertainty. In the preceding, $D(\tau|\mathbf{a}, \mathbf{e}_M)$ represents dose (mrem/yr) to the RMEI at time τ (yr) conditional on the “future” defined by the vector \mathbf{a} of aleatory quantities (see Eq. (44.4)) and the vector \mathbf{e}_M of epistemically uncertain quantities (see Table 44.3) used in the calculation of dose to the RMEI; and $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ is defined as indicated in Eq. (44.28). Specifically,

$$E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)] = \int_{\mathcal{A}} D(\tau|\mathbf{a}, \mathbf{e}_M) d_A(\mathbf{a}|\mathbf{e}_A) d\mathcal{A}. \quad (44.34)$$

The time-dependent expected values $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ for $D(\tau|\mathbf{a}, \mathbf{e}_M)$ were estimated for the time intervals $[0, 20,000 \text{ yr}]$ and $[0, 10^6 \text{ yr}]$ for each of the LHS sample elements $\mathbf{e}_k = [\mathbf{e}_{Ak}, \mathbf{e}_{Mk}]$ indicated in Eq. (44.17).

The resultant expected dose curves $[\tau, E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]], k = 1, 2, \dots, n = 300$, the associated curves for the quantiles $Q_{Eq}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}, q = 0.05, 0.5, 0.95$, and the expected value $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ for dose over aleatory and epistemic uncertainty are shown in Fig. 44.6a. The quantities $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ and $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ are both expected doses but with

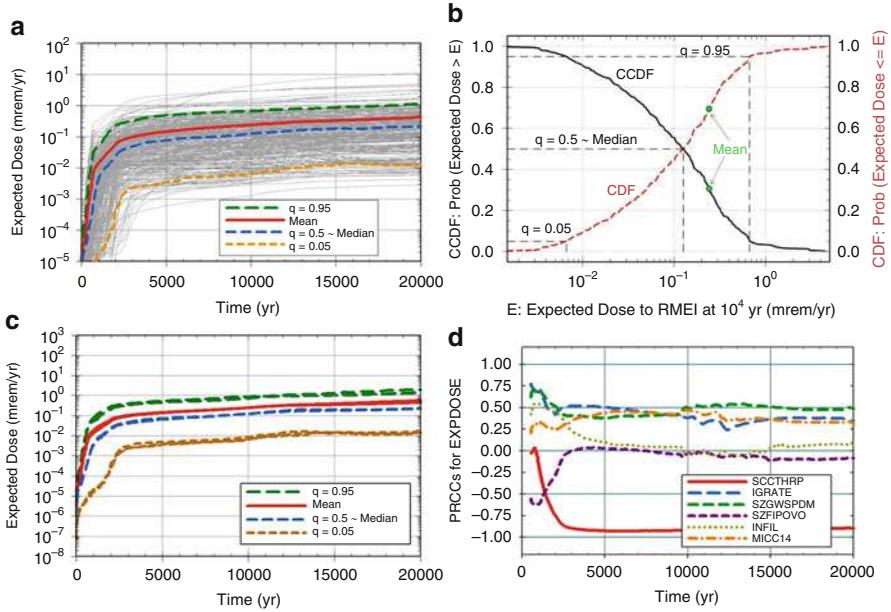


Fig. 44.6 Estimates obtained with an LHS of size $n = 300$ of expected dose $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ to the RMEI in the 2008 YM PA for the time interval $[0, 20,000]$ yr: (a) expected dose $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$, $k = 1, 2, \dots, n = 300$, expected (mean) dose $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, and quantiles $Q_{E_q}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, $q = 0.05, 0.5, 0.95$ ([61], Fig. 1a); (b) cumulate probabilities $p_E\{E_A[D(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A] \leq D\}$, exceedance probabilities $p_E\{D < E_A[D(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, expected (mean) dose $E_E\{E_A[D(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, and quantiles $Q_{E_q}\{E_A[D(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, $q = 0.05, 0.5, 0.95$, for $\tau = 10^4$ yr ([61], Fig. 1b); (c) estimates of $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ and $Q_{E_q}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, $q = 0.05, 0.5, 0.95$, obtained with three replicated LHSs of size $n = 300$ as indicated in Eq. (44.24) ([61], Fig. 5); and (d) partial rank correlation coefficients (PRCCs) for $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ ([61], Fig. 6)

expectations calculated with respect to different probability spaces. In consistency with regulatory requirements that place bounds on $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ for the YM repository, the 2008 YM PA refers to $E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ and $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ as “expected dose” and “expected (mean) dose,” respectively [26]. A more detailed summary of the distribution results for $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ at $\tau = 10^4$ yr is presented in Fig. 44.6b, and the stability of the results obtained with the three replicated LHSs of size 300 is illustrated in Fig. 44.6c by the similarity of the resultant plots for estimates of $Q_{Eq}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$, $q = 0.05, 0.5, 0.95$, and $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$. The sensitivity analysis results in Fig. 44.6d are discussed in Sect. 7.

The numerics of determining the expected doses $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Fig. 44.6a are quite complicated and will be discussed briefly after this paragraph. Once the expected doses $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ are available, the estimation of $Q_{Eq}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ and $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ is straightforward.

Specifically, $Q_{Eq}\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ corresponds to the value y defined and approximated by

$$\begin{aligned} q &= \int_{\mathcal{E}} \delta_y \{E_A [D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\} d_E(\mathbf{e}) d\mathcal{E} \\ &\cong \sum_{k=1}^n \delta_y \{E_A [D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]\} / n, \end{aligned} \quad (44.35)$$

and $E_E\{E_A[D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ is defined and approximated by

$$\begin{aligned} E_E \{E_A [D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\} &= \int_{\mathcal{E}} E_A [D(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A] d_E(\mathbf{e}) d\mathcal{E} \\ &\cong \sum_{k=1}^n E_A [D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}] / n. \end{aligned} \quad (44.36)$$

Similarly, $Q_{Eq}\{E_A[y(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ and $E_E\{E_A[y(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ for an arbitrary analysis result $y(\tau|\mathbf{a}, \mathbf{e}_M)$ are defined and approximated as indicated in Eqs. (44.35) and (44.36) with $y(\tau|\mathbf{a}, \mathbf{e}_M)$ and $y(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ replacing $D(\tau|\mathbf{a}, \mathbf{e}_M)$ and $D(\tau|\mathbf{a}, \mathbf{e}_{Mk})$.

The numerics of determining the expected doses $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ are now considered. In concept, $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ can be approximated by

$$E_A [D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_A] \cong \sum_{i=1}^m D(\tau|\mathbf{a}_i, \mathbf{e}_{Mk}) / m, \quad (44.37)$$

where $\mathbf{a}_i, i = 1, 2, \dots, m$, is a sample from the probability space \mathcal{A} for aleatory uncertainty (see Eq. (44.4)) generated in consistency with the density function $d_A(\mathbf{a}|\mathbf{e}_A)$. As is the case in many complex analyses, the overall structure of the 2008 YM PA was too complex (see Eq. (44.4) and Fig. 44.2) for this sampling-based approach for the estimation of $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ to be practicable. In such situations, it is necessary to (i) decompose the analysis into a number of intermediate calculations smaller than a complete calculation of the result of interest (e.g., $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Eq. (44.37) and Fig. 44.6) and then (ii) combine these intermediate results in a computationally efficient manner to obtain the final result of interest. The indicated combination of intermediate results involves taking advantage of computational efficiencies that derive from specific properties of the analysis and can include (i) division of the original integral over the sample space \mathcal{A} for aleatory uncertainty into a sum of integrals over subsets of \mathcal{A} ; (ii) use of additivity, linearity, and interpolations in conjunction with intermediate results to obtain additional results; and (iii) development of one or more metamodels that are approximations to the original model.

In the 2008 YM PA, the first step in the evaluation of $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ was to divide the defining integral for $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Eq. (44.34) into a sum of integrals over selected subsets of the sample space \mathcal{A} for aleatory uncertainty. These subsets are referred to as scenario classes (and sometimes as modeling cases) in the 2008 YM PA and are denoted and defined with the notation used in Eqs. (44.4–44.7) by (i) $\mathcal{A}_{EW} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nEW \geq 1\}$ for the early WP failure scenario class, (ii) $\mathcal{A}_{ED} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nED \geq 1\}$ for the early DS failure scenario class, (iii) $\mathcal{A}_{II} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nII \geq 1\}$ for the igneous intrusion scenario class, (iv) $\mathcal{A}_{IE} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nIE \geq 1\}$ for the igneous eruption scenario class, (v) $\mathcal{A}_{SG} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nSG \geq 1\}$ for the seismic ground motion scenario class, (vi) $\mathcal{A}_{SF} = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nSF \geq 1\}$ for the seismic fault displacement scenario class, and (vii) $\mathcal{A}_N = \{\mathbf{a}|\mathbf{a} \in \mathcal{A} \text{ and } nEW = nED = nII = nIE = nSG = nSF = 0\} = \{\mathbf{a}_N\}$ for the nominal scenario class. In turn, the indicated sum of integrals is given by

$$\begin{aligned} & E_A [D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}] \\ &= \int_{\mathcal{A}} D(\tau|\mathbf{a}, \mathbf{e}_{Mk}) d_A(\mathbf{a}|\mathbf{e}_{Ak}) d\mathcal{A} \\ &\cong D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}) + \sum_{C \in \mathcal{MC}} \int_{\mathcal{A}_C} D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk}) d_A(\mathbf{a}|\mathbf{e}_{Ak}) d\mathcal{A} \quad (44.38) \\ &= D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}) + \sum_{C \in \mathcal{MC}} \int_{\mathcal{A}_C} D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk}) d_A(\mathbf{a}|\mathbf{e}_{Ak}) d\mathcal{A} \\ &= D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}) + \sum_{C \in \mathcal{MC}} E_A [D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}], \end{aligned}$$

with (i) $\mathcal{MC} = \{EW, ED, II, IE, SG, SF\}$ and (ii) the doses $D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ for scenario class \mathcal{A}_C calculated with use of only the elements of \mathbf{a} and $\mathbf{e} = [\mathbf{e}_A, \mathbf{e}_M]$ that constitute part of the defining conditions for scenario class \mathcal{A}_C (i.e., early WP failure, early DS failure, igneous intrusion, igneous eruption, seismic ground motion, and seismic fault displacement). The preceding approximation to $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ was justified for use in the 2008 YM PA on the basis of trade-offs between the effects of high probability-low consequence scenario classes and low probability-high consequence scenario classes ([61], Sect. 5).

Use of the decomposition in Eq. (44.38) reduces the determination of $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ from the evaluation of a single very complex integral to the evaluations of (i) $D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})$ and (ii) six individual integrals (i.e., one integral for each element of the set \mathcal{MC}). The six integrals for $D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ over \mathcal{A}_C for $C \in \mathcal{MC}$ that need to be evaluated are still complex, but less complex than a single integral for $D(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ over \mathcal{A} . The advantage of the decomposition in Eq. (44.38) is that the integrals for $D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ over \mathcal{A}_C for $C \in \mathcal{MC}$ can be implemented in ways that obtain computational efficiencies that take advantage of specific properties of the dose function $D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ and the restriction of the density function to the set \mathcal{A}_C for scenario class C .

As a single example, a summary description of the evaluation of $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ over the time interval [0, 20,000 yr] for the seismic ground motion scenario class will be used for illustration; a full description of the analysis for $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ is given in Refs. [32, 71, 72]. In effect, the consideration of only seismic ground motion events reduces the sample space \mathcal{A} for aleatory uncertainty to the previously indicated set \mathcal{A}_{SG} with elements \mathbf{a} defined by

$$\mathbf{a} = [n_{SG}, t_1, v_1, A_1, t_2, v_2, A_2, \dots, t_{n_{SG}}, v_{n_{SG}}, A_{n_{SG}}], \quad (44.39)$$

where (i) n_{SG} = number of seismic ground motion events in 20,000 yr, (ii) t_i = time (yr) of event i , (iii) v_i = peak ground motion velocity (m/s) for event i , (iv) A_i = damaged area (m^2) on individual WPs for peak ground motion velocity v_i , (v) the occurrence of seismic ground motion events is characterized by a hazard curve for peak ground motion velocity, and (vi) damaged area is characterized by distributions conditional on peak ground motion velocity. In effect, \mathcal{A}_{SG} is the sample space for aleatory uncertainty when only seismic ground motion events are considered over the time interval [0, 20,000 yr].

To evaluate $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$, it is necessary to integrate the function $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ over the set \mathcal{A}_{SG} with \mathbf{a} defined as indicated in Eq. (44.39). In full detail, $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ is defined by the model system indicated in Fig. 44.2. Evaluation of this system is too computationally demanding to permit its evaluation 1000's of times for each element $\mathbf{e}_k = [\mathbf{e}_{Ak}, \mathbf{e}_{Mk}]$ of the LHS in Eq. (44.17). This is a common situation in analyses of complex systems, where very detailed physical models are developed which then turn out to be too computationally demanding to be naively used in the propagation of aleatory uncertainty. In such situations, it is necessary to find ways to efficiently use the results of a limited number of model evaluations to predict outcomes for a large number of different possible realizations of aleatory uncertainty.

For the seismic ground motion scenario class and the time interval [0, 20,000 yr], the needed computational efficiency was achieved by evaluating $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ at a sequence of seismic event times (i.e., 100, 1000, 3000, 6000, 12,000, 18,000 yrs) and for a sequence of damaged areas (i.e., 10^{-8+s} (32.6 m^2) for $s = 1, 2, \dots, 5$ with 32.6 m^2 corresponding to the surface area of a WP) at each of the indicated times (Fig. 44.7a). This required $6 \times 5 = 30$ evaluations of the system indicated in Fig. 44.2 for each LHS element in Eq. (44.17). Once obtained, these evaluations can be used with appropriate interpolation and additive procedures to evaluate $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ for different values of \mathbf{a} for each LHS element $\mathbf{e}_k = [\mathbf{e}_{Ak}, \mathbf{e}_{Mk}]$ (see Ref. [32], Sect. 4, for computational details).

The individual CCDFs in Fig. 44.7b are defined by probabilities of the form shown in Eq. (44.27) with (i) $D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})$ and \mathbf{e}_{Ak} replacing $y(\tau|\mathbf{a}, \mathbf{e}_M)$ and \mathbf{e}_A and (ii) $\tau = 10^4$ yr. Numerically, the integrals that define exceedance probabilities for the individual CCDFs are approximated with (i) random sampling from the possible values for \mathbf{a} as indicated in Eq. (44.30) and (ii) estimated values $\hat{D}_{SG}(10^4|\mathbf{a}_i, \mathbf{e}_{Mk})$ for $D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})$ constructed from results of

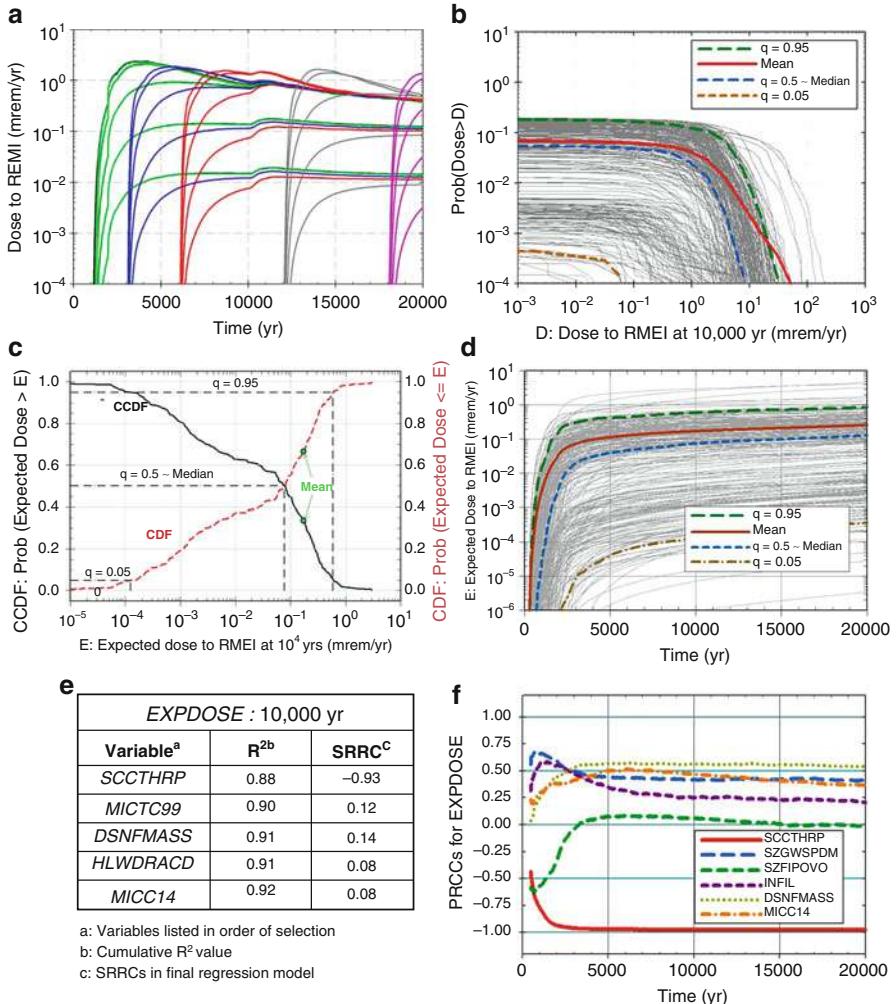


Fig. 44.7 Example results for dose (mrem/yr) to RMEI for seismic ground motion scenario class: (a) dose for seismic events occurring at different times and causing different damaged areas on WPs ([32], Fig. 4a), (b) CCDFs for dose at 10,000 yr ([32], Fig. 10), (c) CCDF and CDF for expected dose at 10,000 yr ([32], Fig. 6b), (d) time-dependent expected dose ([32], Fig. 6a), (e) stepwise rank regression for expected dose at 10,000 yr ([72], Table 4), and (f) time-dependent PRCCs for expected dose ([72], Fig. 4b)

the form shown in Fig. 44.7a (see Ref. [32], Sect. 4, for computational details). Specifically,

$$\hat{p}_A [y < D_{SG}(10^4 | \mathbf{a}, \mathbf{e}_{Mk}) | \mathbf{e}_{Ak}] = \sum_{i=1}^m \bar{\delta}_y [\hat{D}_{SG}(10^4 | \mathbf{a}_i, \mathbf{e}_{Mk})] / m, \quad (44.40)$$

with the \mathbf{a}_i , $i = 1, 2, \dots, m$, sampled in consistency with the density function $d_A(\mathbf{a}|\mathbf{e}_{Ak})$ for vectors of the form shown in Eq. (44.39). The mean and quantile curves in Fig. 44.7b are (i) defined and approximated as indicated in Eqs. (44.32) and (44.33) and (ii) provide a summary of the epistemic uncertainty present in the estimation of exceedance probabilities (i.e., $p_A[y < D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$) for $D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})$.

As indicated in Eqs. (44.28) and (44.31), the expected value $E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ of $D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})$ over aleatory uncertainty can also be defined and estimated, with the estimates $\hat{E}_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ obtained with the random samples used to construct the CCDFs in Fig. 44.7b. The expected values $E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ and their corresponding estimates are the result of reducing each CCDF in Fig. 44.7b to a single number. As illustrated in Fig. 44.7c, the epistemic uncertainty associated $E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$ can be summarized by (i) an expected (mean) value $E_E\{E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ over epistemic uncertainty, (ii) a CDF defined by the cumulative probabilities for $E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$, or (iii) a CCDF defined by the complementary cumulative probabilities for $E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]$. The indicated expected value, CDF and CCDF are defined and approximated in a manner analogous to that used to obtain the results in Fig. 44.3a with $\hat{E}_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ replacing $D_N(600,000|\mathbf{a}_N, \mathbf{e}_M)$. The expected (mean) value $E_E\{E_A[D_{SG}(10^4|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ illustrated in Fig. 44.7c is the outcome of reducing all the information in Fig. 44.7b to a single number.

Expected doses $E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ for individual LHS elements correspond to the lighter lines in Fig. 44.7d, and quantile values $Q_{Eq}\{E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]\}$ and expected (mean) values $E_E\{E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_M)|\mathbf{e}_A]\}$ for $E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ that summarize the effects of epistemic uncertainty correspond to the darker dashed and solid lines. The results in Fig. 44.7d at 10,000 years correspond to the results shown in more detail in Fig. 44.7c. For reasons of computational efficiency, the individual expected dose curves $E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ for $0 \leq \tau \leq 20,000$ yr in Fig. 44.7d were estimated with a quadrature procedure that used the results in Fig. 44.7a rather than with a sampling-based procedure as illustrated in Fig. 44.7b; details of this procedure are given in Ref. [32]. The sensitivity analysis results in Fig. 44.7e and f are discussed in Sect. 7.

The expected doses $E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Fig. 44.7d constitute one of the six expected doses appearing in the approximation to $E_A[D_{SG}(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Eq. (44.38) for $0 \leq \tau \leq 20,000$ yr. The other five expected doses $E_A[D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$, $C = EW, ED, II, IE, SF$, were determined in a generally similar manner in which a representative set of calculations analogous to those indicated in Fig. 44.7a were performed for each LHS element and then used to estimate $E_A[D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$; details of these calculations are given in Refs. [30–32]. The resultant values for all six expected doses $E_A[D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ for the LHS in Eq. (44.17) are shown in Fig. 44.8. As indicated in Eq. (44.38), the doses $E_A[D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ in Fig. 44.8 are added to obtain the approximations to $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ for $0 \leq \tau \leq 20,000$ yr in Fig. 44.6a. Associated analyses

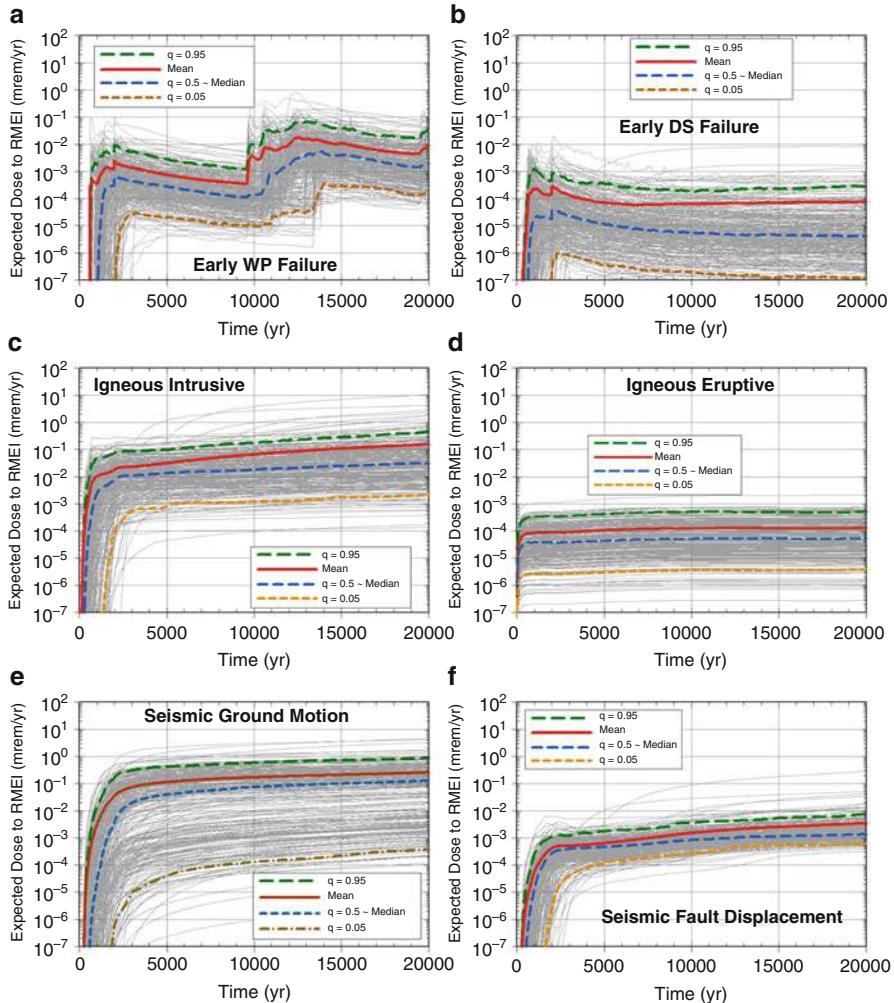


Fig. 44.8 Expected dose $E_A[D_C(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ to RMEI for $0 \leq \tau \leq 20,000$ yr for $C \in \{EW, ED, II, IE, SG, SF\}$: (a) early WP failure, (b) early DS failure, (c) igneous intrusion, (d) igneous eruption, (e) seismic ground motion, and (f) seismic fault displacement ([61], Fig. 2)

Showed that $D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk}) = 0$ mrem/yr for $0 \leq \tau \leq 20,000$ yr (see Fig. 44.3a); thus, $D_N(\tau|\mathbf{a}_N, \mathbf{e}_{Mk})$ made no contribution to $E_A[D(\tau|\mathbf{a}, \mathbf{e}_{Mk})|\mathbf{e}_{Ak}]$ for $0 \leq \tau \leq 20,000$ yr.

7 Sensitivity Analysis

Sensitivity analysis is an important part of a PA and, indeed, any analysis. Specifically, while uncertainty analysis is intended to determine the total uncertainty

in analysis outcomes, sensitivity analysis is intended to determine the contributions of individual uncertain analysis inputs to the total uncertainty in analysis results of interest. The descriptor sensitivity analysis is usually used in reference to the determination of the effects of epistemic uncertainty on analysis outcomes of interest. As a result, sensitivity analysis is typically closely tied to the previously indicated propagation of epistemic uncertainty. In particular, a sampling-based propagation of epistemic uncertainty generates a mapping between uncertain analysis inputs and uncertain analysis results that can then be explored with a variety of sensitivity analysis procedures, including examination of scatterplots, correlation analysis, regression analysis, partial correlation analysis, rank transformations, statistical tests for patterns based on gridding, entropy tests for patterns based on gridding, non-parametric regression analysis, squared rank differences/rank correlation test, two-dimensional Kolmogorov-Smirnov test, tests for patterns based on distance measures, top-down coefficient of concordance, and variance decomposition [73, 74].

Example sensitivity analyses based on partial rank correlation coefficients (PRCCs) are presented in Figs. 44.3d, 44.5d, 44.6d, and 44.7f. In these examples, the PRCCs are determined by analyzing the uncertainty associated with the analysis results above individual values on the abscissas in the indicated figures and then connecting these results to form curves of sensitivity analysis results for individual analysis inputs ([73], Sects. 6.4 and 6.5; [75]). As a result of the use of a rank transformation ([73], Sect. 6.5; [76]), a PRCC provides a measure of the strength of the monotonic effect of an uncertain analysis input on an analysis result after the removal of the monotonic effects of all other uncertain analysis inputs. In the presence of nonlinear but monotonic relationships between a dependent variable and multiple independent (i.e., sampled) variables, use of the rank transformation can substantially improve the resolution of sensitivity analysis results. Specifically, a sensitivity analysis based on rank-transformed data operates in the following manner: (i) the smallest value for each variable is assigned a rank of 1, (ii) next largest value is assigned a rank of 2, (iii) tied values are assigned their average rank, and (iv) so on up to the largest value, which is assigned a rank equal to the sample size in use; then, the analysis is performed with these rank-transformed values. The rank transformation is a very useful procedure in many sensitivity analyses as it converts nonlinear but monotonic relationships into linear relationships.

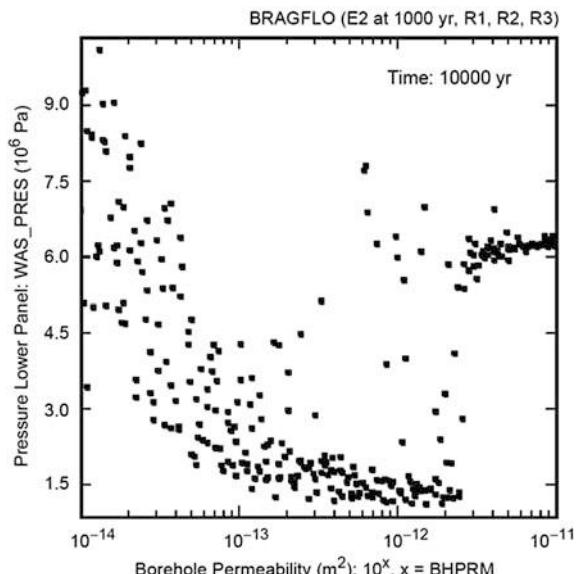
The results in Figs. 44.3d, 44.5d, 44.6d, and 44.7f illustrate a spectrum of the possibilities in which sensitivity analyses can be performed in a PA, including: (i) results conditional on a specific realization of aleatory uncertainty (Fig. 44.3d), (ii) exceedance probabilities that define CCDFs that summarize the effects of aleatory uncertainty (Fig. 44.5d), (iii) expected values over aleatory uncertainty (Fig. 44.6d), and (iv) intermediate results that are part of a very complex calculation (Fig. 44.7f).

As another example, stepwise rank regressions are presented in Figs. 44.3c and 44.7e for results at specific points in time. In stepwise regression, a regression model is first constructed with the most influential variable (i.e., x_1 as determined based on R^2 values for regression models containing only single variables). Then, a regression model is constructed with x_1 and the next most influential variable (i.e., x_2 as determined based on R^2 values for regression models containing x_1 and

each of the remaining variables). The process then repeats to determine x_3 in a similar manner and continues until no more variables with an identifiable effect on the dependent variable can be found. Variable importance (i.e., sensitivity) is then indicated by (i) the order in which variables are selected in the stepwise process, (ii) the changes in cumulative R^2 values as additional variables are added to the regression model, and (iii) the standardized regression coefficients for the variables in the final regression model ([73], Sect. 6.3). In a rank regression, the variables are rank-transformed before the regression analysis is performed.

When relationships between individual independent variables and a dependent variable are both nonlinear and nonmonotonic, sensitivity analyses with partial correlation coefficients and stepwise regression procedures will perform poorly with both untransformed data and rank-transformed data. In such situations, examination of scatterplots may be sufficient to reveal the effects of nonlinear relationships between individual independent variables and a dependent variable ([73], Sect. 6.6; [77]). Specifically, for a sample $\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{k1,nV}]$, $k = 1, 2, \dots, n$, from nV uncertain analysis inputs (e.g., as in Eq. (44.17)) and a corresponding analysis result $y_k = y(\mathbf{x}_k)$, the scatterplot for variable x_l is a plot of the points (x_{kl}, y_k) for $k = 1, 2, \dots, n$. As an example, the well-defined relationship between BHPRM and pressure is clearly shown by the scatterplot in Fig. 44.9 but missed by partial correlation and regression analyses with untransformed and rank-transformed data. More formal procedures to identify nonlinear relationships in sampling-based analyses are provided by grid-based procedures ([73], Sect. 6.6; [78]), nonparametric regression procedures ([73], Sect. 6.8; [79–82]), and the squared rank differences/rank correlation test ([73], Sect. 6.9; [83]).

Fig. 44.9 Example of a scatterplot obtained in the sampling-based uncertainty/sensitivity analysis for the WIPP ([92], Fig. 28; see [92], Sect. 4, for additional discussion of this example)



Many additional examples of sampling-based sensitivity analysis results are available in Ref. [1] for the 1996 WIPP PA and in Ref. [2] for the 2006 YM PA. Additional information on both sampling-based approaches to uncertainty and sensitivity analysis and additional approaches to uncertainty and sensitivity analysis (e.g., differential analysis and variance decomposition) is available in a number of reviews [73, 74, 84–91].

8 Summary

As described and illustrated in this presentation, a conceptual structure and an associated computational structure for PAs for radioactive waste disposal facilities and other complex engineered facilities can be based on the following three conceptual entities : EN1, a probability space ($\mathcal{A}, \mathbb{A}, p_A$) that characterizes aleatory uncertainty; EN2, a function f that estimates consequences for individual elements **a** of the sample space \mathcal{A} for aleatory uncertainty; and EN3, a probability space ($\mathcal{E}, \mathbb{E}, p_E$) that characterizes epistemic uncertainty [7, 8, 24–26]. A recognition and understanding of these three basic entities makes it possible to understand the conceptual and computational structure of a large PA without having basic concepts obscured by fine details of the analysis and leads to an analysis that results in insightful uncertainty and sensitivity analyses. For example, calculations that are basically integrations of very complex functions can be recognized and described as such before the introduction of the often highly-complex numerical procedures needed to perform such integrations.

The indicated conceptual structure also provides a basis and organizational structure for the production of quality documentation for a PA. Everyone will not agree with the results of a complex PA that supports an important societal decision. However, everyone should be able to know what was done and obtained in such a PA. Such knowledge is (i) essential to an informed discussion of a PA and its results and (ii) dependent on quality documentation. A poorly documented PA is likely to be extensively criticized as result of (i) frustrations over trying to determine what was done and (ii) resultant misassumptions about what was done.

Use of the described uncertainty and sensitivity analysis procedures provide important insights into a PA and the factors that affect its outcomes. Uncertainty analysis results enhance the quality of decisions that can be made on the basis of a PA and also enhance the credibility of a PA by showing the resolution (i.e., uncertainty) in its results. Sensitivity analysis provides guidance on both (i) the dominant contributors to the uncertainty in results obtained in a PA and (ii), if necessary, where resources can be best invested to reduce the uncertainty in the results of a PA.

In addition, sampling-based uncertainty and sensitivity analysis procedures are important analysis verification tools. The extensive exercising of the models used in a PA in a sampling-based uncertainty and sensitivity analysis provides a very effective test of the operation of these models. Further, the associated sensitivity results are effective in revealing inappropriate relationships between variables that

are indicative of errors in the formulation or implementation of the PA and its underlying models.

Owing to both the insights that can be gained and the potential for the identification of analysis errors, the examination of uncertainty and sensitivity results should never be limited to only the final outcomes of greatest interest in a PA. Rather, uncertainty and sensitivity results should be examined at multiple points in the chain of computations that lead to the final result or results of greatest interest (e.g., regulatory requirements). Examples of the analysis of intermediate results in a PA are provided by Refs. [92–97] for the WIPP PA and Refs. [33, 72, 98, 99] for the YM PA. Extensive experience with complex PAs has shown that an initial sampling-based uncertainty and sensitivity analysis will almost always reveal errors that need to be corrected before the results of the PA are ready for presentation.

References

1. Helton, J.C., Marietta, M.G. (eds.): Special issue: the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 1–451 (2000)
2. Helton, J.C., Hansen, C.W., Swift, P.N. (eds.): Special issue: performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 1–456 (2014)
3. Helton, J.C., Breeding, R.J.: Calculation of reactor accident safety goals. *Reliab. Eng. Syst. Saf.* **39**(2), 129–158 (1993)
4. Breeding, R.J., Helton, J.C., Gorham, E.D., Harper, F.T.: Summary description of the methods used in the probabilistic risk assessments for NUREG-1150. *Nucl. Eng. Des.* **135**(1), 1–27 (1992)
5. Breeding, R.J., Helton, J.C., Murfin, W.B., Smith, L.N., Johnson, J.D., Jow, H.-N., Shiver, A.W.: The NUREG-1150 probabilistic risk assessment for the Surry Nuclear Power Station. *Nucl. Eng. Des.* **135**(1), 29–59 (1992)
6. Helton, J.C., Hansen, C.W., Sallaberry, C.J.: Conceptual structure of performance assessments for the geologic disposal of radioactive waste. In: 10th International Probabilistic Safety Assessment & Management Conference, Seattle (2010)
7. Helton, J.C., Sallaberry, C.J.: Uncertainty and sensitivity analysis: from regulatory requirements to conceptual structure and computational implementation. In: 10th IFIP WG 2.5 Working Conference on Uncertainty Quantification in Scientific Computing, Boulder. IFIP Advances in Information and Communication Technology (AICT), vol. 377, pp. 60–76 (2012)
8. Kaplan, S., Garrick, B.J.: On the quantitative definition of risk. *Risk Anal.* **1**(1), 11–27 (1981)
9. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 2, 2nd edn. Wiley, New York (1971)
10. Parry, G.W., Winter, P.W.: Characterization and evaluation of uncertainty in probabilistic risk analysis. *Nucl. Saf.* **22**(1), 28–42 (1981)
11. Parry, G.W.: The characterization of uncertainty in probabilistic risk assessments of complex systems. *Reliab. Eng. Syst. Saf.* **54**(2–3), 119–126 (1996)
12. Apostolakis, G.: The concept of probability in safety assessments of technological systems. *Science* **250**(4986), 1359–1364 (1990)
13. Apostolakis, G.: The distinction between aleatory and epistemic uncertainties is important: an example from the inclusion of aging effects into PSA. In: Proceedings of the International Topical Meeting on Probabilistic Safety Assessment, PSA '99: Risk Informed and Performance-Based Regulation in the New Millennium, vol. 1, pp. 135–142. American Nuclear Society, La Grange Park (1999)

14. Helton, J.C.: Treatment of uncertainty in performance assessments for complex systems. *Risk Anal.* **14**(4), 483–511 (1994)
15. Helton, J.C., Burmaster, D.E.: Guest editorial: treatment of aleatory and epistemic uncertainty in performance assessments for complex systems. *Reliab. Eng. Syst. Saf.* **54**(2–3), 91–94 (1996)
16. Helton, J.C.: Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *J. Stat. Comput. Simul.* **57**(1–4), 3–76 (1997)
17. Helton, J.C., Johnson, J.D., Sallaberry, C.J.: Quantification of margins and uncertainties: example analyses from reactor safety and radioactive waste disposal involving the separation of aleatory and epistemic uncertainty. *Reliab. Eng. Syst. Saf.* **96**(9), 1014–1033 (2011)
18. Hoffman, F.O., Hammonds, J.S.: Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. *Risk Anal.* **14**(5), 707–712 (1994)
19. Paté-Cornell, M.E.: Uncertainties in risk analysis: six levels of treatment. *Reliab. Eng. Syst. Saf.* **54**(2–3), 95–111 (1996)
20. Winkler, R.L.: Uncertainty in probabilistic risk assessment. *Reliab. Eng. Syst. Saf.* **54**(2–3), 127–132 (1996)
21. Ferson, S., Ginzburg, L.: Different methods are needed to propagate ignorance and variability. *Reliab. Eng. Syst. Saf.* **54**(2–3), 133–144 (1996)
22. Aven, T.: On different types of uncertainties in the context of the precautionary principle. *Risk Anal.* **31**(10), 1515–1525 (2011)
23. Aven, T., Reniers, G.: How to define and interpret a probability in a risk and safety setting. *Saf. Sci.* **51**, 223–231 (2013)
24. Helton, J.C., Anderson, D.R., Jow, H.-N., Marietta, M.G., Basabilvazo, G.: Conceptual structure of the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 151–165 (2000)
25. Helton, J.C.: Mathematical and numerical approaches in performance assessment for radioactive waste disposal: dealing with uncertainty. In: Scott, E.M. (ed.) *Modelling Radioactivity in the Environment*, pp. 353–390. Elsevier Science, New York (2003)
26. Helton, J.C., Hansen, C.W., Sallaberry, C.J.: Conceptual structure and computational organization of the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 223–248 (2014)
27. Helton, J.C., Davis, F.J., Johnson, J.D.: Characterization of stochastic uncertainty in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 167–189 (2000)
28. U.S. Department of Energy: Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant. DOE/CAO-1996-2184, vols. I–XXI. U.S. Department of Energy, Carlsbad Area Office, Waste Isolation Pilot Plant, Carlsbad (1996)
29. Helton, J.C., Anderson, D.R., Jow, H.-N., Marietta, M.G., Basabilvazo, G.: Performance assessment in support of the 1996 compliance certification application for the Waste Isolation Pilot Plant. *Risk Anal.* **19**(5), 959–986 (1999)
30. Helton, J.C., Hansen, C.W., Sallaberry, C.J.: Expected dose for the early failure scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 297–309 (2014)
31. Sallaberry, C.J., Hansen, C.W., Helton, J.C.: Expected dose for the igneous scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 339–353 (2014)
32. Helton, J.C., Gross, M.B., Hansen, C.W., Sallaberry, C.J., Sevougian, S.D.: Expected dose for the seismic scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 380–398 (2014)
33. Hansen, C.W., Behie, G.A., Bier, A., Brooks, K.M., Chen, Y., Helton, J.C., Hommel, S.P., Lee, K.P., Lester, B., Mattie, P.D., Mehta, S., Miller, S.P., Sallaberry, C.J., Sevougian, S.D., Vo, P.: Uncertainty and sensitivity analysis for the nominal scenario class in the 2008 performance

- assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 272–296 (2014)
34. Helton, J.C., Johnson, J.D., Jow, H.-N., McCurley, R.D., Rahal, L.J.: Stochastic and subjective uncertainty in the assessment of radiation exposure at the Waste Isolation Pilot Plant. *Hum. Ecol. Risk Assess.* **4**(2), 469–526 (1998)
35. Sandia National Laboratories: Total System Performance Assessment Model/Analysis for the License Application. MDL-WIS-PA-000005 Rev 00, AD 01. U.S. Department of Energy Office of Civilian Radioactive Waste Management, Las Vegas (2008)
36. Hansen, C.W., Birkholzer, J.T., Blink, J., Bryan, C.R., Chen, Y., Gross, M.B., Hardin, E., Houseworth, J., Howard, R., Jarek, R., Lee, K.P., Lester, B., Mariner, P., Mattie, P.D., Mehta, S., Perry, F.V., Robinson, B., Sassani, D., Sevougian, S.D., Stein, J.S., Wasiolek, M.: Overview of total system model used for the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 249–266 (2014)
37. Hora, S.C., Iman, R.L.: Expert opinion in risk analysis: the NUREG-1150 methodology. *Nucl. Sci. Eng.* **102**(4), 323–331 (1989)
38. Thorne, M.C., Williams, M.M.R.: A review of expert judgement techniques with reference to nuclear safety. *Prog. Nucl. Saf.* **27**(2–3), 83–254 (1992)
39. Budnitz, R.J., Apostolakis, G., Boore, D.M., Cluff, L.S., Coppersmith, K.J., Cornell, C.A., Morris, P.A.: Use of technical expert panels: applications to probabilistic seismic hazard analysis. *Risk Anal.* **18**(4), 463–469 (1998)
40. McKay, M., Meyer, M.: Critique of and limitations on the use of expert judgements in accident consequence uncertainty analysis. *Radiat. Prot. Dosim.* **90**(3), 325–330 (2000)
41. Meyer, M.A., Booker, J.M.: *Eliciting and Analyzing Expert Judgment: A Practical Guide*. SIAM, Philadelphia (2001)
42. Ayyub, B.M.: *Elicitation of Expert Opinions for Uncertainty and Risks*. CRC Press, Boca Raton (2001)
43. Cooke, R.M., Goossens, L.H.J.: Expert judgement elicitation for risk assessment of critical infrastructures. *J. Risk Res.* **7**(6), 643–656 (2004)
44. Garthwaite, P.H., Kadane, J.B., O'Hagan, A.: Statistical methods for eliciting probability distributions. *J. Am. Stat. Assoc.* **100**(470), 680–700 (2005)
45. Helton, J.C., Martell, M.-A., Tierney, M.S.: Characterization of subjective uncertainty in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 191–204 (2000)
46. Baudrit, C., Dubois, D.: Practical representations of incomplete probabilistic knowledge. *Comput. Stat. Data Anal.* **51**(1), 86–108 (2006)
47. Dubois, D., Prade, H.: Possibility theory and its applications: a retrospective and prospective view. In: Riccia, G.D., Dubois, D., Kruse, R., Lenz, H.-J. (eds.) *Decision Theory and Multi-agent Planning*. CISM International Centre for Mechanical Sciences, vol. 482, pp. 89–109 (2006)
48. Klir, G.J.: *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley-Interscience, New York (2006)
49. Ross, T.J.: *Fuzzy Logic with Engineering Applications*, 2nd edn. Wiley, New York (2004)
50. Helton, J.C., Johnson, J.D., Oberkampf, W.L.: An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliab. Eng. Syst. Saf.* **85**(1–3), 39–71 (2004)
51. Ross, T.J., Booker, J.M., Parkinson, W.J. (eds.): *Fuzzy Logic and Probability Applications: Bridging the Gap*. Society for Industrial and Applied Mathematics, Philadelphia (2002)
52. Klir, G.J., Wierman, M.J.: *Uncertainty-Based Information*. Physica-Verlag, New York (1999)
53. Bardossy, G., Fodor, J.: *Evaluation of Uncertainties and Risks in Geology*. Springer, New York (2004)
54. Helton, J.C., Johnson, J.D., Oberkampf, W.L., Sallaberry, C.J.: Representation of analysis results involving aleatory and epistemic uncertainty. *Int. J. Gen. Syst.* **39**(6), 605–646 (2010)

55. Helton, J.C., Johnson, J.D.: Quantification of margins and uncertainties: alternative representations of epistemic uncertainty. *Reliab. Eng. Syst. Saf.* **96**(9), 1034–1052 (2011)
56. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
57. Helton, J.C., Davis, F.J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliab. Eng. Syst. Saf.* **81**(1), 23–69 (2003)
58. Helton, J.C., Hansen, C.W., Sallaberry, C.J.: Expected dose for the nominal scenario class in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 267–271 (2014)
59. Helton, J.C., Davis, F.J.: Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. SAND2001-0417. Sandia National Laboratories, Albuquerque (2002)
60. Iman, R.L.: Statistical methods for including uncertainties associated with the geologic isolation of radioactive waste which allow for a comparison with licensing criteria. In: Kocher, D.C. (ed.) *Proceedings of the Symposium on Uncertainties Associated with the Regulation of the Geologic Disposal of High-Level Radioactive Waste*, Gatlinburg, 9–13 Mar 1981, pp. 145–157. U.S. Nuclear Regulatory Commission, Directorate of Technical Information and Document Control, Washington, DC (1982)
61. Helton, J.C., Hansen, C.W., Sallaberry, C.J.: Expected dose and associated uncertainty and sensitivity analysis results for all scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 421–435 (2014)
62. Hansen, C.W., Helton, J.C., Sallaberry, C.J.: Use of replicated Latin hypercube sampling to estimate sampling variance in uncertainty and sensitivity analysis results for the geologic disposal of radioactive waste. *Reliab. Eng. Syst. Saf.* **107**, 139–148 (2012)
63. Helton, J.C., Sallaberry, C.J.: Computational implementation of sampling-based approaches to the calculation of expected dose in performance assessments for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **94**(3), 699–721 (2009)
64. Sallaberry, C.J., Helton, J.C., Hora, S.C.: Extension of Latin hypercube samples with correlated variables. *Reliab. Eng. Syst. Saf.* **93**, 1047–1059 (2008)
65. Iman, R.L., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat. Simul. Comput.* **B11**(3), 311–334 (1982)
66. Iman, R.L., Davenport, J.M.: Rank correlation plots for use with correlated input variables. *Commun. Stat. Simul. Comput.* **B11**(3), 335–360 (1982)
67. Helton, J.C., Anderson, D.R., Basabilvazo, G., Jow, H.-N., Marietta, M.G.: Summary discussion of the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 437–451 (2000)
68. Farmer, F.R.: Reactor safety and siting: a proposed risk criterion. *Nucl. Saf.* **8**(6), 539–548 (1967)
69. Cox, D.C., Baybutt, P.: Limit lines for risk. *Nucl. Technol.* **57**(3), 320–330 (1982)
70. Munera, H.A., Yadigaroglu, G.: On farmer's line, probability density functions, and overall risk. *Nucl. Technol.* **74**(2), 229–232 (1986)
71. Helton, J.C., Gross, M.B., Sallaberry, C.J.: Representation of aleatory uncertainty associated with the seismic ground motion scenario class in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 399–405 (2014)
72. Hansen, C.W., Behie, G.A., Bier, A., Brooks, K.M., Chen, Y., Helton, J.C., Hommel, S.P., Lee, K.P., Lester, B., Mattie, P.D., Mehta, S., Miller, S.P., Sallaberry, C.J., Sevougian S.D., Vo, P.: Uncertainty and sensitivity analysis for the seismic scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 406–420 (2014)

73. Helton, J.C., Johnson, J.D., Sallaberry, C.J., Storlie, C.B.: Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1175–1209 (2006)
74. Helton, J.C.: Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliab. Eng. Syst. Saf.* **42**(2–3), 327–367 (1993)
75. Iman, R.L., Shortencarier, M.J., Johnson, J.D.: A FORTRAN 77 Program and User's Guide for the Calculation of Partial Correlation and Standardized Regression Coefficients. NUREG/CR-4122, SAND85-0044. Sandia National Laboratories, Albuquerque (1985)
76. Iman, R.L., Conover, W.J.: The use of the rank transform in regression. *Technometrics* **21**(4), 499–509 (1979)
77. Cooke, R.M., van Noortwijk, J.M.: Graphical methods. In: Saltelli, A., Chan, K., Scott, E.M. (eds.) *Sensitivity Analysis*, pp. 245–264. Wiley, New York (2000)
78. Kleijnen, J.P.C., Helton, J.C.: Statistical analyses of scatterplots to identify important factors in large-scale simulations, 1: review and comparison of techniques. *Reliab. Eng. Syst. Saf.* **65**(2), 147–185 (1999)
79. Storlie, C.B., Helton, J.C.: Multiple predictor smoothing methods for sensitivity analysis: description of techniques. *Reliab. Eng. Syst. Saf.* **93**(1), 28–54 (2008)
80. Storlie, C.B., Helton, J.C.: Multiple predictor smoothing methods for sensitivity analysis: example results. *Reliab. Eng. Syst. Saf.* **93**(1), 55–77 (2008)
81. Storlie, C.B., Swiler, L.P., Helton, J.C., Sallaberry, C.J.: Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliab. Eng. Syst. Saf.* **94**(11), 1735–1763 (2009)
82. Storlie, C.B., Reich, B.J., Helton, J.C., Swiler, L.P., Sallaberry, C.J.: Analysis of computationally demanding models with continuous and categorical inputs. *Reliab. Eng. Syst. Saf.* **113**(1), 30–41 (2013)
83. Hora, S.C., Helton, J.C.: A distribution-free test for the relationship between model input and output when using Latin hypercube sampling. *Reliab. Eng. Syst. Saf.* **79**(3), 333–339 (2003)
84. Hamby, D.M.: A review of techniques for parameter sensitivity analysis of environmental models. *Environ. Monit. Assess.* **32**(2), 135–154 (1994)
85. Frey, H.C., Patil, S.R.: Identification and review of sensitivity analysis methods. *Risk Anal.* **22**(3), 553–578 (2002)
86. Ionescu-Bujor, M., Cacuci, D.G.: A comparative review of sensitivity and uncertainty analysis of large-scale systems—I: deterministic methods. *Nucl. Sci. Eng.* **147**(3), 189–2003 (2004)
87. Cacuci, D.G., Ionescu-Bujor, M.: A comparative review of sensitivity and uncertainty analysis of large-scale systems—II: statistical methods. *Nucl. Sci. Eng.* **147**(3), 204–217 (2004)
88. Cacuci, D.G.: *Sensitivity and Uncertainty Analysis, Volume 1: Theory*. Chapman and Hall/CRC Press, Boca Raton (2003)
89. Cacuci, D.G., Ionescu-Bujor, M., Navon, I.M.: *Sensitivity and Uncertainty Analysis, Volume 2: Application to Large-Scale Systems*. Chapman and Hall/CRC Press, Boca Raton (2005)
90. Saltelli, A., Chan, K., Scott, E.M. (eds.): *Sensitivity Analysis*. Wiley, New York (2000)
91. Saltelli, A., Ratto, M., Tarantola, S., Campolongo, F.: Sensitivity analysis for chemical models. *Chem. Rev.* **105**(7), 2811–2828 (2005)
92. Helton, J.C., Bean, J.E., Economy, K., Garner, J.W., MacKinnon, R.J., Miller, J., Schreiber, J.D., Vaughn, P.: Uncertainty and sensitivity analysis for two-phase flow in the vicinity of the repository in the 1996 performance assessment for the Waste Isolation Pilot Plant: disturbed conditions. *Reliab. Eng. Syst. Saf.* **69**(1–3), 263–304 (2000)
93. Helton, J.C., Bean, J.E., Economy, K., Garner, J.W., MacKinnon, R.J., Miller, J., Schreiber, J.D., Vaughn, P.: Uncertainty and sensitivity analysis for two-phase flow in the vicinity of the repository in the 1996 performance assessment for the Waste Isolation Pilot Plant: undisturbed conditions. *Reliab. Eng. Syst. Saf.* **69**(1–3), 227–261 (2000)
94. Berglund, J.W., Garner, J.W., Helton, J.C., Johnson, J.D., Smith, L.N.: Direct releases to the surface and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant: cuttings, cavings and spillings. *Reliab. Eng. Syst. Saf.* **69**(1–3), 305–330 (2000)

95. Stoelzel, D.M., O'Brien, D.G., Garner, J.W., Helton, J.C., Johnson, J.D., Smith, L.N.: Direct releases to the surface and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant: direct brine release. *Reliab. Eng. Syst. Saf.* **69**(1–3), 343–367 (2000)
96. Stockman, C.T., Garner, J.W., Helton, J.C., Johnson, J.D., Shinta, A., Smith, L.N.: Radionuclide transport in the vicinity of the repository and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 370–396 (2000)
97. Ramsey, J.L., Blaine, R., Garner, J.W., Helton, J.C., Johnson, J.D., Smith, L.N., Wallace, M.: Radionuclide and colloid transport in the Culebra Dolomite and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliab. Eng. Syst. Saf.* **69**(1–3), 397–420 (2000)
98. Hansen, C.W., Behie, G.A., Bier, A., Brooks, K.M., Chen, Y., Helton, J.C., Hommel, S.P., Lee, K.P., Lester, B., Mattie, P.D., Mehta, S., Miller, S.P., Sallaberry, C.J., Sevougian, S.D., Vo, P.: Uncertainty and sensitivity analysis for the early failure scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 310–338 (2014)
99. Sallaberry, C.J., Behie, G.A., Bier, A., Brooks, K.M., Chen, Y., Hansen, C.W., Helton, J.C., Hommel, S.P., Lee, K.P., Lester, B., Mattie, P.D., Mehta, S., Miller, S.P., Sevougian, S.D., Vo, P.: Uncertainty and sensitivity analysis for the igneous scenario classes in the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab. Eng. Syst. Saf.* **122**, 354–379 (2014)

Redundancy of Structures and Fatigue of Bridges and Ships Under Uncertainty

45

Dan M. Frangopol, Benjin Zhu, and Mohamed Soliman

Abstract

Bridges and ships are key components of civil and marine infrastructure systems, respectively. Due to the nature of their in-service role, failure of such structures may lead to very high consequences. Preventing such failures requires rigorous design and life-cycle management techniques. In order to maintain satisfactory performance for these structures throughout their service life, various uncertainties associated with the design and management processes should be properly accounted for. Among the important design considerations for bridges and ships that highly depend on the proper identification of these uncertainties are the structural redundancy quantification and fatigue life assessment. In this chapter, quantification of redundancy and its integration into the design process of structural components are discussed. Additionally, the probabilistic fatigue assessment problem and the sources of uncertainties associated with the fatigue life prediction models are presented.

Keywords

Redundancy • Reliability • Uncertainty • Fatigue • Life-cycle

D.M. Frangopol (✉)

Department of Civil and Environmental Engineering, Lehigh University, Engineering Research Center for Advanced Technology for Large Structural Systems (ATLSS Center), Bethlehem, PA, USA

e-mail: dan.frangopol@lehigh.edu

B. Zhu

Rizzo Associates, Inc., Pittsburgh, PA, USA

e-mail: benjin.zhu@rizzoassoc.com

M. Soliman

School of Civil and Environmental Engineering, Oklahoma State University, Stillwater, OK, USA

e-mail: mohamed.soliman@okstate.edu

Contents

1	Introduction	1542
2	Quantitative Redundancy Measures	1543
2.1	Time-Variant Redundancy	1545
2.2	Redundancy Factor for Structural Component Design	1547
3	Fatigue in Bridges and Ships	1552
3.1	Fatigue Assessment Based on Fracture Mechanics	1553
3.2	The S-N Approach	1557
4	Conclusions	1562
	References	1562

1 Introduction

Structures are expected to maintain desired levels of serviceability and safety during their service life. Due to the unavoidable uncertainties in structural design and performance assessment, probabilistic performance indicators, such as reliability and redundancy, were introduced. Redundancy, in general, indicates the ability of a structural system to redistribute the load after the failure of one component or subsystem. Adequate redundancy should be considered in structural design. System redundancy has attracted a lot of research interest in recent decades [19, 21, 22, 26, 44].

Even though redundancy is a highly desired attribute for a structure, quantitative guidance is not commonly available in the structural design specifications. Load and resistance factor design (LRFD) replaced allowable stress design within the last decades with better understanding of structural performance and system reliability theory [2]. Since LRFD approach considers the uncertainties associated with the resistances and load effects using separate factors, it provides better means of incorporating system reliability and redundancy concepts in the structural design.

Several measures for quantifying structural redundancy are available in the literature. Frangopol and Curley [19] defined the system redundancy as the ratio of the reliability index of the intact system to the difference between the reliability indices of the intact and the damaged systems. Rabi et al. [44] defined the redundancy by comparing the probability of system collapse to the probability of first member failure. Ghosn and Moses [22] proposed three redundancy measures that are defined in terms of relative reliability indices. Since redundancy is regarded as an important performance indicator, research on the evaluation of system redundancy in real structures has been conducted. Tsopelas and Husain [52] studied the system redundancy of reinforced concrete buildings. Wen and Song [53] investigated the redundancy of special moment resisting frames and dual systems under seismic excitations. Kim [27] evaluated the redundancy of steel box-girder bridge by using a nonlinear finite element model.

Fatigue is another critical mechanism which can adversely affect the safety of metallic structures. Fatigue is the process of crack initiation and propagation resulting from repetitive loads on the structure [17]. These cracks, if not effectively detected and repaired, may cause sudden failure of the damaged component. If the structure does not have an adequate redundancy level, a catastrophic structural

failure may occur due to the failure of the cracked component. Although methodologies for fatigue assessment and design are already established in literature and design specifications, fatigue cracks exist in various types of structures such as bridges, offshore structures, and naval vessels. This can be attributed to the fact that a large number of factors contribute to the fatigue crack initiation and propagation including the structural configuration of the detail, presence of initial defects, environmental conditions, and loading conditions, among others. Additionally, several sources of uncertainty affect the fatigue crack initiation and propagation. Such uncertainties must be well-understood and should be accounted for in order to correctly assess the performance of a structure with respect to fatigue. Accordingly, several researchers address the fatigue life estimation and performance assessment on probabilistic basis [28, 41, 54].

This chapter presents a brief review of several proposed redundancy measures. A methodology to assess the life-cycle redundancy of a structure is described in details in context of system reliability. Next, several quantifications of the redundancy factor used for structural component design are discussed. The chapter also discusses the probabilistic aspects of the fatigue life estimation problem.

2 Quantitative Redundancy Measures

Redundancy is a performance indicator that measures the reserve capacity of structures after the failure of a component or subsystem and provides an indication on the presence of alternative load paths within the structure. For a redundant system, the failure of a single member will not result in the collapse of the entire structure if it is a redundant system. Several redundancy measures were proposed in the literature, including deterministic and probabilistic performance indicators. Frangopol and Klisinski [20] defined a reserve strength factor as the ratio of the load-carrying capacity of the intact structure or member, C , to the nominal applied load on the structure or the effect of this load on the member, Q ,

$$R_1 = \frac{C}{Q} \quad (45.1)$$

The reserve strength factor varies from a value of infinity, when the intact structure has no load, to a value of 1.0, when the nominal load on the intact structure equals its capacity.

The residual strength factor provides a measure for the strength of the system in a damaged condition compared to the intact system. It is defined as the ratio of the capacity of the damaged structure or member, C_d , to the capacity of the intact structure or member, C , and can be expressed as [20]

$$R_2 = \frac{C_d}{C} \quad (45.2)$$

This factor takes values between 0, when damaged structure has zero capacity, and 1.0, when damaged structure does not have any reduction in its load-carrying

capacity. Frangopol and Klisinski [20] also proposed the following deterministic measure of redundancy:

$$R_3 = \frac{1}{1 - R_2} \quad (45.3)$$

where R_2 is the residual strength factor.

Ghosn and Moses [22] defined redundancy as the capability of a structure to continue to carry loads after the damage or failure of one or more members. In order to provide objective measures of system redundancy, they investigated the load multipliers LF_u , LF_f , LF_d , and LF_1 . It is assumed that the live load applied has a certain configuration (e.g., AASTHO HS-20 truck), and then the first main member will fail when the reference load is multiplied by a factor LF_1 . Similarly, LF_u is the load multiplier when the ultimate capacity of the intact structure is reached. LF_f is the load multiplier when large vertical deformations rendering the structure unfit for use are reached, in other words when the functionality is lost. LF_d is the load multiplier when the ultimate capacity of the damaged structure is reached. Ghosn and Moses [22] defined three deterministic measures of the system's capacity as compared to the most critical member's capacity as:

$$R_u = \frac{LF_u}{LF_1} \quad (45.4)$$

$$R_f = \frac{LF_f}{LF_1} \quad (45.5)$$

$$R_d = \frac{LF_d}{LF_1} \quad (45.6)$$

where R_u is the system reserve ratio for the ultimate limit state, R_f is the system reserve ratio for the functionality limit state, and R_d is the system reserve ratio for the damage condition.

In order to account for the uncertainties in system and member capacities, as well as the applied loads, Frangopol and Curley [19] proposed a probabilistic measure for redundancy as

$$RI = \frac{P_{f(dmg)} - P_{f(sys)}}{P_{f(sys)}} \quad (45.7)$$

where RI is the system redundancy index, $P_{f(dmg)}$ is the probability of damage occurrence to the system, and $P_{f(sys)}$ is the probability of system failure. In addition, they defined redundancy index in terms of reliability index as

$$I_R = \frac{\beta_{intact}}{\beta_{intact} - \beta_{damaged}} \quad (45.8)$$

where I_R is redundancy index, β_{intact} is the reliability index of the intact system, and $\beta_{damaged}$ represents the reliability index of the damaged system.

Ghosn and Moses [22] also proposed probabilistic measures of redundancy. They examined the differences between the reliability indices of the systems expressed in terms of the reliability index associated with the ultimate capacity of the intact structure, $\beta_{ultimate}$; the reliability index associated with the functionality loss, $\beta_{functionality}$; the reliability index associated with the ultimate capacity of the damaged structure, $\beta_{damaged}$; and the reliability index of the most critical member of the structure, β_{member} . As measures of redundancy levels, they investigated the relative reliability indices for the ultimate $\Delta\beta_u$, functionality $\Delta\beta_f$, and damaged $\Delta\beta_d$ limit states and are defined as:

$$\Delta\beta_u = \beta_{ultimate} - \beta_{member} \quad (45.9)$$

$$\Delta\beta_f = \beta_{functionality} - \beta_{member} \quad (45.10)$$

$$\Delta\beta_d = \beta_{damaged} - \beta_{member} \quad (45.11)$$

2.1 Time-Variant Redundancy

The majority of studies on system redundancy assume that both the loads and resistances are time independent. In other words, the probability density functions (PDF) of the random variables are kept constant during the lifetime of a structure due to the complexity involved in dealing with time-dependent systems. Nevertheless, accounting for the effects of time on structures may have significant impact on the redundancy measures and is necessary for accurate performance assessment of structural systems.

Prediction of both the time-variant loads and resistances is required to include the time factor in the performance prediction model. The resistances of structural components and systems are subjected to deterioration in time. Deterioration can be caused by various mechanisms, such as corrosion and fatigue. Furthermore, the applied loading may increase over time (e.g., traffic loads on bridges). The combined effect of the decrease in resistance and increase in load will lead to significant drop of structural reliability and redundancy.

In this section, a procedure for evaluating the time-variant redundancy for structural systems under uncertainty and deterioration, based on the redundancy measure given in Eq. (45.8), is presented. As indicated in this equation, the reliabilities (or failure probabilities) associated with the intact and damaged structures must be determined to evaluate redundancy. A common approach for assessing the time-variant reliability is the point-in-time method in which the reliability index is computed at discrete time points to establish a lifetime reliability profile. Alternatively, the reliability and redundancy can be assessed in continuous time periods using survivor functions [40]. In the point-in-time method, the limit state functions are required to be updated at every time point according to the predictive models for deteriorated resistance and increased traffic loads.

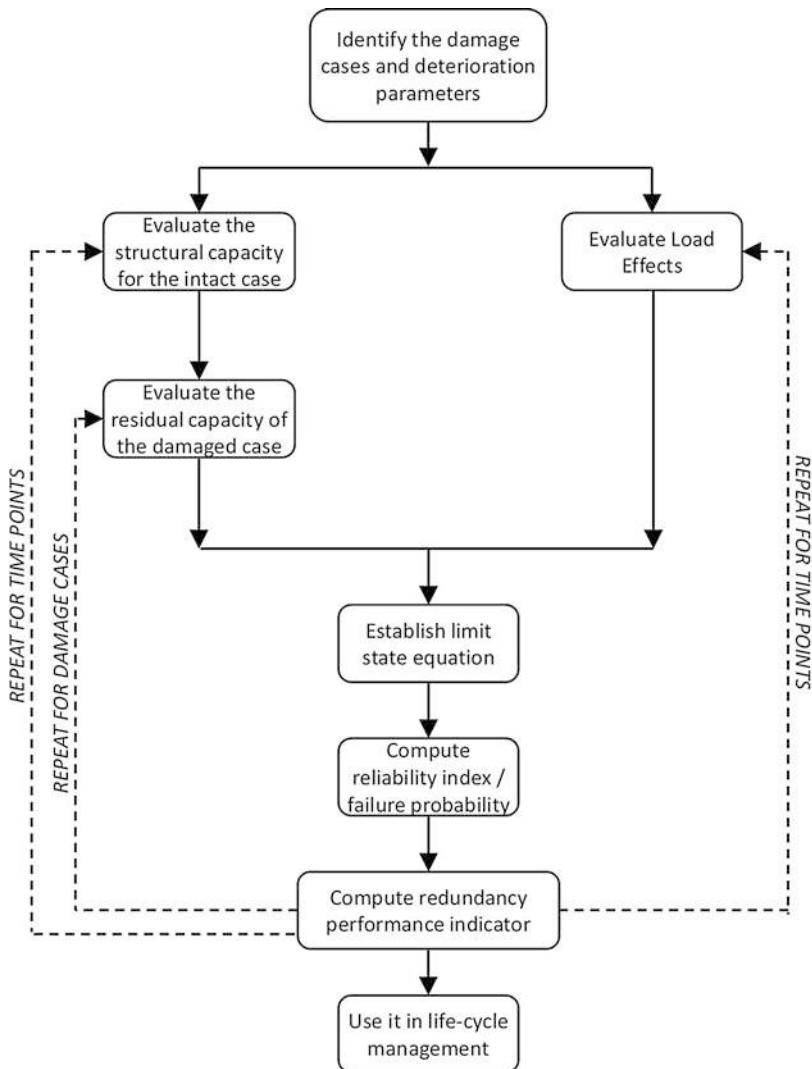


Fig. 45.1 Methodology for assessing time-variant redundancy

The methodology for assessing the time-variant redundancy of deteriorating structures is illustrated in Fig. 45.1. In this methodology, the term redundancy is associated with a certain damage case (e.g., failure of a critical member). The first step of the methodology is identifying the damage cases and deterioration parameters to be considered in the life-cycle performance evaluation. The next steps are the computations for the resistance and the load effects. The random variables associated with the resistance must be identified. The load capacity of the structure associated with the considered damage case should be computed accounting for

uncertainties. The variation of the live load over time is required in order to perform time-dependent reliability analysis. Then, the limit state equation consisting of the resistance and the load effects needs to be established. The instantaneous probability of failure and/or reliability index associated with a damage case can be obtained by first-order reliability method (FORM), second-order reliability method (SORM), or Monte Carlo simulation. Once the reliability index or probability of failure is obtained for a time point, the redundancy index can be calculated based on the adopted formulation (e.g., Eq. (45.8)). This procedure should be repeated at different time steps with time-variant values of resistance and load effects in order to obtain the lifetime redundancy profile. In addition, redundancy associated with different damage cases must be evaluated to identify the importance of components within the system.

The procedure for system reliability assessment with emphasis on bridges is further illustrated herein. The system reliability of bridges can be assessed based on appropriate assumptions associated with the interaction of individual components. In general, a bridge system can be modeled as a series, parallel, or series-parallel system. The reliability of a bridge structural system is often evaluated by considering the system failure as series-parallel combination of the components' failures. After determining the random variables and their statistical parameters for component reliability analysis, the limit states of components can be included in the system model. A bridge component may have several failure modes. For example, a girder can fail in bending or shear. The flexural or shear capacity for girders and the slab can be calculated based on the strength limit state relations given in AASHTO LRFD Bridge Design Specifications [1].

Another approach for redundancy assessment of bridges makes use of finite element (FE) method. An appropriate statistical distribution for the desired output of FE analysis (e.g., stress, displacement, bending moment) can be obtained by repeating the analysis for a large number of samples of the random variables associated with the structure. However, for complex structures, it may be computationally expensive to repeat the FE analysis several times. In such cases, the response surface method (RSM) can be used to approximate the relation between the desired output of FE analysis and random variables by performing analyses for a significantly less number of samples [40]. In this method, the response surface function is generated by fitting a mathematical model to the simulation outcomes in terms of input variables of the simulation model. Support points for each design variable are selected to estimate the parameters of the response surface function. This task can be performed by using experimental design concepts.

2.2 Redundancy Factor for Structural Component Design

Structures are purposefully designed much stronger than what is usually needed to accommodate intended loading. The main objective is to create a margin of safety under extreme events with high consequences, unexpected loadings, and structural deterioration. The extent by which the structural capacity of a system exceeds the

expected loads or actual loads is described as factor of safety. Safety factor is applied to the material strength of structural component in the conventional allowable stress design (ASD) to obtain the allowable strength. Although this ASD has been the primary approach used for structural component design since the early 1900s, it is not a rational and economical design method as the uncertainties involved in the structural design process are not properly considered.

With the increased understanding of structural behavior and loading process, the design philosophy evolved to the load factor design (LFD) around the 1970s. Compared to the ASD, LFD recognizes that the live load was more variable than dead load, and, therefore, uncertainties in load prediction were considered by introducing the load factors. However, deterministic approaches were still used when calibrating these factors. Due to the development and application of probability-based reliability theory in structural engineering, a design philosophy referred to as load and resistance factor design (LRFD) was introduced in the 1980s. Compared with the previous design methods, LRFD is more rational because the uncertainties associated with resistances and loads are quantitatively incorporated in the design process [51].

Structural components in a system do not behave independently; instead, they interact with other components under external loads to form a structural system. However, the previous design methods ignore this system effect in the component design. The aforementioned definitions of redundancy indicate that it is connected to system behavior. Therefore, consideration of system redundancy is essential for component design. Structures with sufficient redundancy can be rewarded by allowing their components to have less conservative designs, while structures that are nonredundant would be required to have higher component capacities. The existence of uncertainties in resistances and loads requires quantifying the system redundancy in a probabilistic manner. Since the LRFD approach was developed based on the probability theory, it provides a basis to incorporate system reliability and redundancy concepts in structural component design.

In the current LRFD design specifications, load and resistances factors are developed using statistical knowledge of the variability of loads and structural performance. In the AASHTO LRFD bridge design specifications, three system factors relating to ductility, redundancy, and operation classification, respectively, are applied on the load side of the strength limit state. The factor relating to redundancy, denoted as η_R , is included to account for the effect of system redundancy on the component design. Its value is determined based on three redundancy levels [1], as follows:

$$\eta_R \begin{cases} \geq 1.05 & \text{for nonredundant member} \\ = 1.00 & \text{for conventional level of redundancy} \\ \geq 0.95 & \text{for exceptional levels of redundancy} \end{cases} \quad (45.12)$$

This classification of the redundancy levels is very general, and the evaluation of the factor relating to redundancy is very subjective. In fact, this factor is affected by several parameters, such as system type, number of components in

a structure, and correlations among the resistances of members, among others (Hendawi and Frangopol 1994; [22]). Therefore, although the LRFD code has been continuously refined and revised since its initial publication in 1994, there is room for further improvement in the classification of the redundancy level and quantification of the associated values [51], as indicated in section 1.3.2.1 in AASHTO [1]: “improved quantification of ductility, redundancy, and operational classification may be attained with time, and possibly leading to a rearranging of Eq. 1.3.2.1-1, in which these effects may appear on either side of the equation or on both sides.”

Although quantification of redundancy factors is a very complex task, significant effort has been devoted by researchers to achieve this goal. Hendawi and Frangopol [26] proposed a system factor modifier placed on the resistance side of the limit states to account for the system redundancy in the design of individual components. Similarly, Ghosn and Moses [22] presented a system factor to serve the same purpose. Four limit states are investigated to determine the system factors:

1. Member failure limit state: to check the safety of an individual component using elastic analysis.
2. Ultimate limit state: to determine the ultimate capacity of the intact bridge system.
3. Functionality limit state, which is defined as a maximum acceptable displacement of a main component under live load.
4. Damaged condition limit state: to find the ultimate capacity of a damaged bridge system.

The aforementioned load multipliers LF_1 , LF_u , LF_f , and LF_d correspond to the four limit states, respectively. They are calculated using incremental analysis considering the elastic and inelastic material behavior of structural components. Finally, the system factors are determined by comparing the strength reserve ratios (i.e., R_u , R_f , and R_d) related to different limit states with the required redundancy ratios that are obtained through a system reliability calibration to provide adequate redundancy. The system factors are calibrated for bridges with typical configurations: (a) simple-span, prestressed concrete and steel I-beam bridges with 4, 6, 8, 10, and 12 beams, (b) simple-span, prestressed concrete bridges with 2, 3, and 5 spread boxes as well as multi-box beam bridges with up to 11 adjacent boxes, and (c) two-span continuous I-beam bridges associated with three system limit states (ultimate, functionality, and damaged). The calibration process of the system factor is complicated because it requires finite element modeling and nonlinear incremental structural analysis. The factors obtained are only for common types of bridges with limited number of components. For other bridges with unique geometries and nontypical material properties, the associated system factors are not provided.

Zhu and Frangopol [56–58] proposed a new definition of redundancy factor applied to the load side in the strength limit state for component design. This redundancy factor is defined as the ratio of the mean resistance of a component in a system when the system reliability is prescribed to the mean resistance of the

same component when its reliability index is the same as that of the system. For example, consider a single component with resistance R and load P which are treated as random variables. Given the distribution types of R and P , mean value of load $E(P)$, coefficients of variation of resistance and load $V(R)$ and $V(P)$, and the prescribed component reliability index $\beta_c = 3.5$, the mean value of the component resistance $E_c(R)$ can be determined (e.g., by using Monte Carlo simulation). In two specific cases where both R and P follow normal or lognormal distribution, $E_c(R)$ can be directly calculated by solving Eqs. (45.13) and (45.14), respectively:

$$\beta_c = \frac{E_c(R) - E(P)}{\sqrt{(E_c(R) \cdot V(R))^2 + (E(P) \cdot V(P))^2}} \quad (45.13)$$

$$\beta_c = \frac{\ln \left[\frac{E_c(R)}{E(P)} \sqrt{\frac{1+V^2(P)}{1+V^2(R)}} \right]}{\sqrt{\ln [(1 + V^2(R)) (1 + V^2(P))]} \quad (45.14)}$$

For a three-component parallel system whose components are the same as the single component just described (e.g., geometry and material properties, distribution parameters), the load acting on the system is assumed to be $3P$, and the correlation coefficient between the resistances of components i and j is assumed to be $\rho(R_i, R_j)$. According to the definition of redundancy factor, the target system reliability index should be the same as the component reliability index, which is 3.5 herein. Therefore, the mean value of component resistance in the system, denoted as $E_{cs}(R)$, can be calculated by using Monte Carlo simulation. Having obtained the mean resistance of a component in a system when the system reliability index is 3.5, $E_{cs}(R)$, and the mean resistance of the same component when the component reliability index is 3.5, $E_c(R)$, the redundancy factor, denoted as η_R , which is defined as the ratio of $E_{cs}(R)$ to $E_c(R)$, can be determined.

Since system reliability is affected by system modeling, number of components in the system, correlations among the component resistances, and post-failure material behavior of components, the effects of these parameters on the system redundancy should be considered in the quantification of the redundancy factor. Three brief examples are presented herein to illustrate these effects. The first example is a four-component structure. Based on different definitions of system failure, three types of systems can be modeled: (45.1) series model, the system fails if any component fails; (45.2) parallel model, the system fails only if all components fail; and (c) series-parallel model, the system fails if components 1 and 2 fail simultaneously or components 3 and 4 fail at the same time. The resistance R and load effect P associated with each component are assumed to be the same and to follow normal distribution. The coefficients of variation of R and P are considered 10%. The mean value of load affect, denoted $E(P)$, is assumed to be 10. Three correlation cases among the resistances of components are investigated herein: (a) $\rho(R_i, R_j) = 0$, no correlation; (b) $\rho(R_i, R_j) = 0.5$, partial correlation; and (c) $\rho(R_i, R_j) = 1.0$, perfect correlation. Therefore, the mean values of resistance of a single component when its reliability is 3.5 are obtained for the

three correlation cases using Eq. (45.13). Similarly, the mean values of component resistances $E_{cs}(R)$ when the system reliability index is also 3.5 can be calculated for the three systems associated with different correlation cases. Finally, the redundancy factors η_R for series, parallel, and series-parallel systems are obtained as (45.1) 1.062, 0.785, and 0.878 when $\rho(R_i, R_j) = 0$; (45.2) 1.059, 0.859, and 0.933 when $\rho(R_i, R_j) = 0.5$; and (45.3) 1.0, 1.0, and 1.0 when $\rho(R_i, R_j) = 1.0$. It is noticed that the redundancy factor associated with series system is the highest, while its counterpart associated with parallel system is the lowest. As the correlation among the component resistances increases, the redundancy factor decreases in the series system but increases in the parallel and series-parallel systems. The redundancy factor in the perfect correlation case is independent of the system modeling type.

These observations indicate that the component in the series system requires larger redundancy factor in the design to reach the targeted system reliability level. However, in the parallel system, components can be designed more economically. Higher correlation among the component resistances leads to higher redundancy factor in the parallel and series-parallel systems, which implies that the component needs to be designed stronger. An opposite conclusion is drawn for the series system. In the perfect correlation case, the redundancy factor is always 1.0. This is expected because a system whose components are identical and their failure modes are perfectly correlated can be reduced to a single component. Therefore, the redundancy factor does not change with the system type.

The second example is to study the effect of a number of components on the redundancy factor. An additional three-component structure is investigated, and its results will be compared with those from the four-component systems just discussed. Two types of systems are formed for this three-component structure: series and parallel. The components are considered as identical and the loads acting on them are assumed to be similar. The statistical parameters associated with resistances and loads (i.e., $V(R)$, $V(P)$, and $E(P)$) in the previous four-component structure are used herein. Assuming the target reliability index both for single component and the systems is 3.5, the redundancy factors of the series and parallel systems are found to be: (45.1) 1.049 and 0.812 when $\rho(R_i, R_j) = 0$; (45.2) 1.047 and 0.879 when $\rho(R_i, R_j) = 0.5$; and (45.3) 1.0 and 1.0 when $\rho(R_i, R_j) = 1.0$. By comparing these factors with the values in the four-component systems, it is seen that as the number of components increases from three to four, the redundancy factor associated with series system increases, while its counterpart associated with the parallel system decreases. However, this conclusion is not valid in the perfect correlation case where the redundancy factor is constant.

All the materials used in practical cases have their own behaviors. The ability of components to carry loads beyond the elastic limits affects the redistribution of loads within a system and the availability of alternative load paths. Therefore, the post-failure material behavior of components needs to be considered in the quantification of the redundancy factor [58]. The last example in this section studies the impact of post-failure behavior on the redundancy factor using the aforementioned three-component parallel system. The load acting on the system is $3P$. The resistances and loads of components follow normal distribution with the parameters mentioned

previously. Two types of post-failure behavior are examined: ductile and brittle. A ductile component can still take load after yielding; however, a brittle component completely loses its capacity when it fails. This difference results in different limit state equations of the parallel system in the ductile and brittle cases, and therefore, the mean resistances in these two cases will be different. If the target reliability index is predefined as 3.5, the redundancy factors associated with the ductile and brittle cases are obtained as: (45.1) 0.98 and 1.033 when $\rho(R_i, R_j) = 0$; (45.2) 0.989 and 1.026 when $\rho(R_i, R_j) = 0$; and (45.3) 1.0 and 1.0 when $\rho(R_i, R_j) = 1.0$. It is noticed that the redundancy factor of the ductile system is much lower than that of the brittle system in the no correlation and partial correlation cases. This means the component with brittle material needs to be designed much stronger than the one with ductile material.

It is seen from the three examples that the redundancy factor is significantly affected by the system modeling, number of components, correlations among the component resistances, and post-failure material behavior. This emphasizes the necessity of quantifying redundancy factors for a variety of systems with different parameters mentioned previously. To achieve this goal, nondeterministic idealized systems are used to evaluate the redundancy factors for ductile and brittle systems with up to 500 components considering different system types, several correlation cases, and two probability distribution types. As shown in Zhu et al. [59], the obtained redundancy factors in ductile and brittle systems can be often larger than the upper bound (i.e., 1.05) or lower than the lower bound (i.e., 0.95) as defined in the AASHTO LRFD Bridge Design Specifications.

3 Fatigue in Bridges and Ships

Fatigue damage accumulation is a major concern for several types of metallic structures including highway bridges, railway bridges, offshore structures, and both steel and aluminum naval vessels. Fatigue can be defined as “the tendency of a metal to break under repeated cyclic loading at a stress considerably less than the tensile strength in a static test” [45]. Repeated fluctuating loads which cause fatigue damage accumulation include traffic loads on bridges, wind loads on masts, and wave loads on ships. At regions of stress concentrations where an initial crack or crack-like defect exists, crack propagation may occur when elastic stresses are applied. This stress concentration causes plastic zones to be formed at the crack tip. Initiation and propagation of cracks in the plastic localized region occur due to the cumulative damage acting over a certain number of stress fluctuations. These cracks can eventually cause the fracture of the component [4]. In modern steel bridges and naval vessels, stress concentration occur at different components due to the geometric configuration, the presence of initial material flaws, or as a result of the welding process.

Design guidelines and specifications provide methods to compute the fatigue life associated with various details widely employed in steel and aluminum structures (see, e.g., BS5400 [5], BS7910 [6], Eurocode 9 [15], Eurocode 3 [14], AASHTO [1]). These design guidelines mainly implement the $S-N$ (i.e., stress-life) approach

for fatigue design and assessment due to its simplicity and sufficient agreement with experimental test results. Fatigue assessment approaches based on fracture mechanics can also be used to assess the crack conditions at a certain detail and predict the time-variant crack size.

A brief discussion on the $S-N$ approach and the linear elastic fracture mechanics in fatigue assessment is presented next. Uncertainties associated with different model parameters are also discussed. Additionally, the widely implemented methodologies for quantifying and reducing these uncertainties are presented.

3.1 Fatigue Assessment Based on Fracture Mechanics

In this method, the stresses near the crack tip are related to the range of the stress intensity factor, ΔK . Paris' law [42] relates the crack growth rate to the range of the stress intensity factor as follows:

$$\frac{da}{dN} = C \cdot (\Delta K)^m \quad (45.15)$$

where a is crack size, N is number of cycles, and ΔK is range of the stress intensity factor. C and m are material parameters. The range of the stress intensity factor can be expressed as

$$\Delta K = K(a) \cdot S_{re} \cdot \sqrt{\pi a} \quad (45.16)$$

where S_{re} is the stress range and $K(a)$ is the generalized stress intensity factor which depends on the crack orientation and shape. This factor can be expressed as [16]

$$K(a) = F_e \cdot F_s \cdot F_w \cdot F_g \quad (45.17)$$

in which F_e , F_s , F_w , and F_g are correction factors taking into account the effects of the elliptical crack shape, free surface, finite width, and nonuniform stress acting on the crack, respectively. Solutions for these correction factors can be found in Tada et al. [50].

Using Eqs. (45.15) and (45.16), the number of cycles associated with a growth in the crack size from an initial size of a_o to a size of a_t can be calculated as

$$N = \frac{1}{C \cdot S_{re}^m} \cdot \int_{a_o}^{a_t} \frac{1}{(K(a) \cdot \sqrt{\pi a})^m} da \quad (45.18)$$

Considering the average daily number of cycles N_{avg} to be constant, the time interval (in years) associated with a crack growth from a_o to a size of a_t can be calculated as

$$t = \frac{1}{365 \cdot N_{avg} \cdot C \cdot S_{re}^m} \cdot \int_{a_o}^{a_t} \frac{1}{(K(a) \cdot \sqrt{\pi a})^m} da \quad (45.19)$$

The integration in Eq. (45.19) can be performed numerically to find the crack growth profile for the analyzed detail. However, the cases where $K(a)$ is not constant may take considerable computational effort especially if probabilistic performance

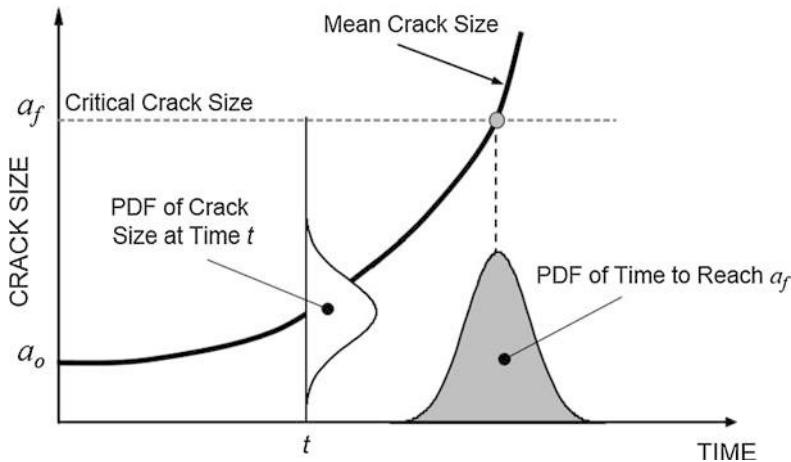


Fig. 45.2 Crack growth under uncertainty

assessment is required. For such probabilistic analysis, uncertainties associated with the parameters C , m , a_o , N_{avg} , and S_{re} should be accounted for by considering their associated probabilistic distributions. The parameters C and m are material properties whose descriptors can be found through the results of experimental programs, while the stress range and the number of cycles depend on the loading conditions at the detail. The final outcome of stochastic fatigue crack growth analysis is to obtain the probabilistic crack growth profile with respect to time. Other useful information can also be found from such analysis. For instance, the PDF of the time required for the crack to reach a given size can be identified. Accordingly, such analyses can yield the PDF of the time to reach the critical crack size (or the time to failure) which can be employed to find an estimate of the reliability and failure hazard associated with the detail based on lifetime functions [32]. Furthermore, the PDF of the crack size at a given time instance can be also be quantified. Figure 45.2 shows schematically the results of such process.

Although the model based on Paris' law does not consider factors such as the effect of variable amplitude loading, mean stress, and the stress ratio, it has been widely used for predicting the crack growth profile, analyzing the fatigue reliability, and establishing the optimum inspection and maintenance plans. In order to identify the probabilistic fatigue crack growth profile by using Paris' law, several stochastic approaches have been proposed. Guedes Soares and Garbatov [23] proposed a probabilistic approach based on Paris' law to study the time-variant fatigue reliability of steel ship details. Assuming that $K(a)$ is constant, an expression for the expected crack size over time was given by

$$a(t) = \left[a_o^{1-\frac{m}{2}} + \left(1 + \frac{m}{2} \right) \cdot C \cdot (S_{re})^m \cdot K(a)^m \cdot \pi^{\frac{m}{2}} \cdot N(t) \right]^{\frac{1}{(1-m/2)}} \quad (45.20)$$

where $N(t)$ represents the time cumulative number of cycles.

In this approach, the crack growth was predicted by implementing Taylor-series expansion of the crack size at the mean values of C , a_o , and S_{re} . Other parameters of the crack growth model were treated as deterministic. A similar approach was implemented by Chung et al. [8] to predict the fatigue crack growth and obtain the optimum nondestructive inspection times for fatigue-prone details in steel bridges. However, their approach implemented Monte Carlo simulation to predict the crack growth profile. These approaches only predict the crack propagation under uncertainty, and the crack initiation can be incorporated by assuming that an initial defect with size a_o exists in the detail.

The fracture mechanics approach was also employed by Kim and Frangopol [28] to find the optimum inspection schedule for fatigue critical details which minimize the damage detection delay. This delay is defined as the time elapsed between the damage occurrence and the damage detection. In their study, Eq. (45.19) was used in a Monte Carlo simulation process to predict the time-dependent crack growth. The study was extended in Kim and Frangopol [29] to find the inspection schedule which simultaneously minimizes the damage detection delay and the life-cycle inspection cost. Soliman et al. [49] also accounted for the uncertainties in the fatigue crack growth model parameters by using Monte Carlo simulation in a study to find the optimum nondestructive inspection times and types for a bridge structure with multiple critical fatigue details. Although the parameter m in Eq. (45.19) was considered deterministic in the previous studies, several studies show that this parameter should be considered as a random variable with high correlation between the parameters C and m . However, Kim et al. [30] shows that considering the uncertainty associated with m will have a minor effect on the fatigue crack growth simulation results.

Since the descriptors of input parameters will have a significant effect on the simulation results, the proper implementation of uncertainty quantification and reduction techniques is necessary to improve the accuracy of such prediction model. With respect to the stress range and the number of cycles, perhaps the most effective technique for uncertainty quantification is to use the information obtained from structural health monitoring of analyzed details [7,31,33]. In such programs, sensors are installed at the critical members to record the strain histories under normal loading conditions for a sufficient period of time. Next, a cycle counting algorithm such as the rainflow [12] method is implemented to establish the stress-range bin histograms and the cycle count. This information can be used to find an accurate estimate of the stress range and number of cycles [11] which is used for fatigue analysis.

Other parameters, such as the initial crack size, can be identified using field inspection outcomes for surface growing cracks. However, in most of the cases, an initial crack size is assumed depending on the capabilities of the inspection method. For visually inspected components, it can be assumed to be 2.5 mm [45], while a smaller value (e.g., 0.5 mm) can be considered for other nondestructive inspection techniques. However, for probabilistic analysis, this parameter is generally considered as a random variable. Yazdani and Albrecht [55] suggest that the mean value should be 0.5 mm. Other smaller values have been also reported in literature, for

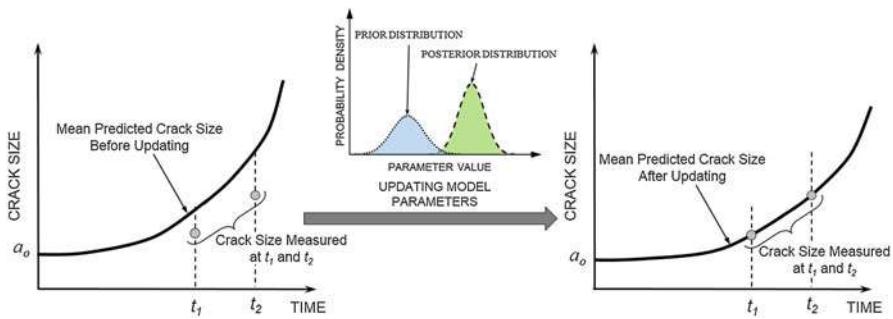


Fig. 45.3 Fatigue crack growth updating based on inspection outcomes

Instance, Moan and Song [38] considered the mean value to be 0.11 mm. Several distribution types for this parameters have also been reported including lognormal distribution [30, 49] and exponential distribution [38]. Additionally, the coefficient of variation of the initial crack size has been reported to range from 0.2 to 0.5 [8, 30]. Accordingly, a significant variability in the value of the initial crack size can be seen. Similar variability in the descriptors of the random variables C and m can also be found. Accordingly, attempts to quantify the uncertainties associated with these parameters should be made to improve the accuracy of the stochastic analysis.

To help quantifying the uncertainties associated with such model parameters, the Bayesian approach can be used to find a better representation of the model parameters' descriptors given field inspection information. In this approach, the information from inspection actions is used to represent the likelihood function which can be combined with the prior knowledge of the model parameters to yield an updated posterior distribution of the model parameters. Hence, the performance prediction is performed using the posterior parameters to achieve more reliable results. Figure 45.3 schematically shows this process. This approach was investigated by [24], in which the probability of detection was used in an updating procedure to predict the posterior distribution of the initial crack size at a certain point in time, and the measurement was used as the new data to update the PDF of the initial crack size. However, inspection optimization and scheduling was not considered as a goal of their study. Perrin et al. [43] used Bayesian techniques and Markov Chain Monte Carlo (MCMC) for fatigue crack growth analysis based on data collected during experimental investigations. Their results showed the feasibility of updating the model parameters based on crack size measurements. Li et al. [34] used Bayesian updating to study the effect of the sensor degradation on the estimation of the remaining useful life of structures. Soliman and Frangopol [47] proposed an approach to establish integrated management plans that provide optimal intervention times and types while making use of the available inspection and monitoring information to improve the performance prediction process, and, hence, better and effective decisions can be made. In this process, information from field measurement is used to represent the likelihood function which can be

combined with the prior information on model parameters to find their posterior distributions. The posterior distributions can be found as

$$P(\boldsymbol{\theta}|\mathbf{d}) = \frac{P(\boldsymbol{\theta}) \cdot P(\mathbf{d}|\boldsymbol{\theta})}{\int P(\boldsymbol{\theta}) \cdot P(\mathbf{d}|\boldsymbol{\theta}) d\boldsymbol{\theta}} \quad (45.21)$$

where $P(\boldsymbol{\theta}|\mathbf{d})$ is the posterior distribution of model parameters $\boldsymbol{\theta}$ given the additional information \mathbf{d} , $P(\boldsymbol{\theta})$ represents the prior distribution of model parameters, $P(\mathbf{d}|\boldsymbol{\theta})$ is the likelihood function of obtaining information \mathbf{d} conditioned by $\boldsymbol{\theta}$, and \mathbf{d} and $\boldsymbol{\theta}$ are the vectors of observed data and model parameters, respectively.

By knowing the prior distributions and the likelihood function, the posterior distributions can be established using sampling approaches based on Markov chain Monte Carlo simulation such as the Metropolis algorithm [36] or the slice sampling algorithm [39]. The considered likelihood function is [43]

$$P(\mathbf{d}|\boldsymbol{\theta}) = \prod_{i=1}^n \left[\frac{1}{\sqrt{2\pi} \cdot \sigma_e} \cdot \exp \left(-\frac{1}{2} \left(\frac{d_i - a_{p,i}}{\sigma_e} \right)^2 \right) \right] \quad (45.22)$$

where d_i and $a_{p,i}$ are the observed and predicted data, respectively, at the i th inspection; σ_e represents a single-error term combining the measurement and modeling errors which is assumed to follow a normal distribution with zero mean and a standard deviation σ_e (i.e., $N(0, \sigma_e)$). The vector of observed data, consisting of crack size measurements identified during fatigue inspections, is

$$\mathbf{d} = \{a_{ins,1}, a_{ins,2}, a_{ins,3}, \dots, a_{ins,n}\} \quad (45.23)$$

where $a_{ins,1}$ = measured crack size at the i th inspection and n = number of inspections. Based on the sample space of inspection outcomes, subsequent inspection times can be found. In Soliman and Frangopol [47], this updating process was integrated in a comprehensive life-cycle management approach to find updated inspection times based on field measurements. An optimization process, executed iteratively for possible crack size measurements, was constructed and solved to find the optimum inspection time which minimizes the total cost which included the inspection cost as well as the failure cost. The outcome of this process is shown schematically in Fig. 45.4.

Although fatigue assessment models based on fracture mechanics are well established, the standardization and formalization of these approaches, especially with respect to stochastic analysis, is still required.

3.2 The S-N Approach

The $S-N$ approach is adopted by most of the specifications for design and assessment of fatigue critical components in bridges and naval vessels. The $S-N$ approach provides a linear or piecewise linear relationship between the logarithm

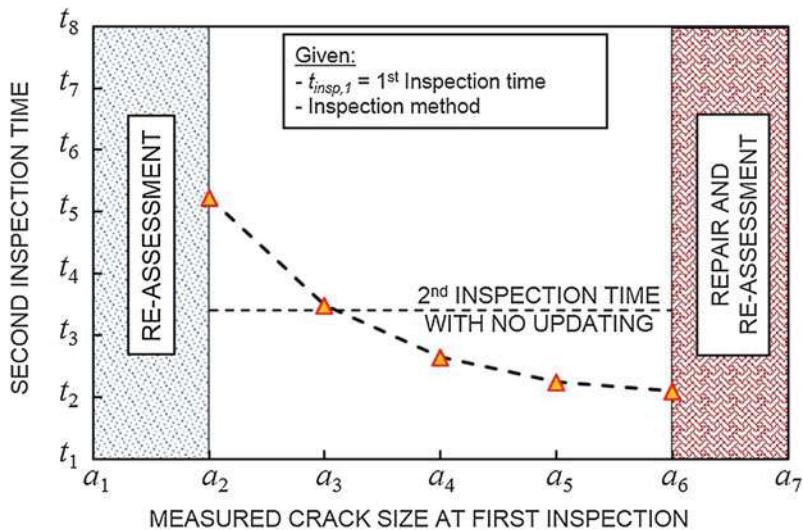


Fig. 45.4 Updated inspection times based on field measurements

of the stress range and the logarithm of the expected number of cycles to failure for several types of details. In order to construct the $S-N$ line for a certain detail, several specimens are tested in the laboratory under different cyclic stress amplitudes until a crack with predefined size propagates through the detail. The number of cycles to failure is plotted against the applied stress range of each specimen, and regression analysis is performed to plot the mean $S-N$ line in the logarithmic scale. In order to achieve a satisfactory probability of survival for designed structures, a design line is usually defined in the codes through shifting the mean line horizontally to the left by a certain amount.

The $S-N$ relationship within each portion of the $S-N$ lines can be expressed as

$$N = A \cdot S^{-m} \quad (45.24)$$

where N is the number of cycles to failure under the stress range S , A is a fatigue detail coefficient for each category, and m is the slope parameter of the $S-N$ line.

In most of the structural components in bridges and ships, fatigue-prone details are normally subjected to variable amplitude stress-range cycles. Accordingly, an equivalent constant amplitude stress range is computed to calculate the fatigue damage under this loading. Miner's rule [37] is the most widely employed method to quantify the fatigue damage accumulation due to variable amplitude loading. If the histogram of stress range is identified for the detail, and assuming a linear damage accumulation, Miner's damage accumulation index is

$$D = \sum_{i=1}^{n_{ss}} \frac{n_i}{N_i} \quad (45.25)$$

where n_{ss} is the number of stress-range bins in a stress-range histogram, n_i is the number of stress cycles in the i th bin with stress range S_i , and N_i is the number of cycles to failure under the stress range S_i . In this model, the failure of the detail occurs when $D = 1.0$. It should be noted that this approach neglects the effect of loading sequence (i.e., the order in which the cycles are applied) which can be significant in some cases [46]. However, for most structural engineering problems where the residual stresses are high and the plasticity is restricted, this model is sufficient to achieve reasonable accuracy [17].

Based on Miner's damage accumulation rule, an equivalent constant amplitude stress range can be defined as

$$S_{re} = \left[\sum_{i=1}^{n_{ss}} \frac{n_i}{N_T} \cdot S_i^m \right]^{\frac{1}{m}} \quad (45.26)$$

where $N_T = \sum_{i=1}^{n_{ss}} n_i$. By knowing the average annual number of cycles N_{avg}^A , an estimate of the fatigue life can be computed as

$$t \text{ (years)} = \frac{N}{N_{avg}^A} \quad (45.27)$$

Several stress analysis methods can be used for the fatigue assessment of steel and aluminum details, namely, the nominal stress, structural hot spot stress, and notch stress. The choice of the stress type and its corresponding $S-N$ relationships depends on the type of details, available measurement data, and the used specifications. The nominal stress approach is adopted by several design and assessment guides such as the Eurocode 9 [15] and the AASHTO [1]. In this approach, the stress range refers to the stresses acting on the considered location neglecting the stress concentration arising from both the weld effects and the structural configuration. Accordingly, The $S-N$ lines inherently consider these effects. This approach is characterized by the ease of application since the calculation of nominal stresses can be simply computed for most structural engineering problems. However, assessing fatigue life of a specific detail using this approach requires the existence of a similar match in the design or assessment guidelines, which requires experimental testing of a large number of details. For steel bridges, the $S-N$ lines in the design and assessment guidelines cover most of the details adopted in practice. However, for steel and aluminum ships, several of the details adopted in practice may not be found in the design guidelines which increases the difficulty of applying the nominal stress approach and calls for the application of either the hot spot or notch stress approaches.

In the structural hot spot stress approach, the stress induced in the proximity of the weld is used. This stress includes the stress concentration due to the structural configuration but not due to the weld itself. Accordingly, this stress is compared to $S-N$ lines which similarly incorporate the effect of weld stress concentration.

Accordingly, computing the hot spot stress requires more advanced structural analysis when compared to the nominal stress. However, using this approach requires a lower number of $S-N$ lines to be developed, which reduces the cost of experimental investigations. To compute the stress without the excluding the concentration due to weld, a single reference point at a prescribed distance from the weld toe can be used. Alternatively, the hot spot stress can be extrapolated by measurement performed at multiple reference points.

In the notch stress approach, the total stress acting at the crack initiation location including the stress concentration due to both the structural configuration and the weld geometry is considered. The notch stress is usually more difficult to obtain; however, it can be used to find the fatigue life of the structural detail using the $S-N$ curve for a base non-welded metal.

For design purposes, it is generally preferred to use the nominal stress approach which requires an exact match of the detail to be found in the design specifications. When dealing with SHM data, it may not be practical to find the stress concentration at the weld toe using strain measurement, due to the high stress gradient at this location. Thus, depending on the available data, the nominal stress approach can be used if a similar detail can be found in design guides. Otherwise, the structural hot spot stress approach can be used.

Although the fatigue life estimation by using the $S-N$ approach is straightforward, uncertainties associated with the fatigue damage process requires more robust probabilistic analysis to evaluate the fatigue life of structural components. Accordingly, reliability-based methodologies for fatigue life assessment have been proposed. These methodologies use the time-variant reliability index as the performance indicator. Through the definition of minimum acceptable reliability levels, the fatigue life is considered to be reached when the reliability reaches its minimum threshold.

The reliability index β is related to the probability of failure P_f (i.e., the probability of violating a certain limit state), through the following relationship:

$$\beta = \Phi^{-1}(1 - P_f) \quad (45.28)$$

in which $\Phi^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function. For the estimating the reliability-based remaining fatigue life, the following performance function can be used [3, 25, 31, 48]:

$$g(t) = \Delta - D(t) \quad (45.29)$$

where Δ is Miner's critical damage accumulation index, indicating the allowable accumulated damage, and $D(t)$ is Miner's damage accumulation index indicating the demand. According to Miner's damage accumulation model, the failure of the detail will occur at $\Delta = 1.0$. However, significant variability has been reported in this value. For instance, Collette [10] reports values of Miner's critical damage accumulation for various test programs ranging from 0.5 to more than 10 with most of the values close to one. Accordingly, Miner's critical damage accumulation index is generally treated as a random variable which follow a lognormal distribution.

Most of the studies recommend the mean value to be 0.9 [3] or 1.0 ([9, 48]; [31]) with a coefficient of variation of 0.48 [3, 9] or 0.3 [54].

Based on Eq. (45.24), Miners's damage accumulation index can be computed as

$$D(t) = \frac{N(t)}{A} \cdot S_{re}^m = \frac{t \cdot N_{avg}}{A} \cdot S_{re}^m \quad (45.30)$$

where A and m are the $S-N$ relationship parameters, S_{re} is the equivalent constant amplitude stress range, t is the operational time of the structure, and N_{avg} is average number of cycles. In this model, the $S-N$ parameter A can also be treated as a random variable following a lognormal distribution whose parameters can also be established based on regression analysis of test data.

Based on Eqs. (45.29) and (45.30) and assuming that all the random variables (i.e., S_{re} , A , and Δ) follow the lognormal distribution [3, 31], the fatigue reliability index β can be derived as follows:

$$\beta(t) = \frac{\lambda_\Delta + \lambda_A - m \cdot \lambda_{S_{re}} - \ln N(t)}{\sqrt{\xi_\Delta^2 + \xi_A^2 + (m \cdot \xi_{S_{re}})^2}} \quad (45.31)$$

where λ and ξ are the lognormal parameters associated with different random variables. If the target reliability threshold β_{target} is known, the fatigue life t_f can be determined as follows [48]:

$$t_f = \frac{e^{k-m \cdot \lambda_{S_{re}}}}{N_{avg}} \quad (45.32)$$

where

$$k = \lambda_\Delta + \lambda_A - \beta_{target} \cdot \sqrt{\xi^2} \quad (45.33)$$

and

$$\xi^2 = \xi_\Delta^2 + \xi_A^2 + (m \cdot \xi_{S_{re}})^2 \quad (45.34)$$

Eq. (45.32) represents a direct way to estimate the reliability-based fatigue life for a specific detail once the associated stress-range distribution is known. The performance function in Eq. (45.29) can be used to calculate the reliability index using computer software such as RELSYS [13]. Alternatively, Eq. (45.31) can be used directly to calculate the reliability index under the assumption that all the random variables have a lognormal PDF. After establishing the time-variant fatigue reliability profile, the service life is considered to be achieved when the reliability level reaches the minimum allowable reliability level. With respect to fatigue, this reliability threshold is considered to be ranging from two to four [35].

In newly constructed bridges and naval vessels, fatigue-prone components are designed to have either an infinite fatigue life or a fatigue life that is long enough to avoid failure during the service life. For bridges, fatigue cracks should be repaired as soon as they are discovered in order to prevent the occurrence of sudden failures.

In some structures, such as cable stayed bridges where it is difficult to inspect all the details frequently, monitoring devices can be installed to provide warning when alarming crack growth is detected. However, for large-scale structures which rely to a great extent on welding connections such as ships, thousands of crack initiation locations exist and fatigue cracking is inevitable. In such cases, fatigue repairs and inspections can significantly impact the life-cycle operational cost of a vessel, and the structure can be operating with undetected growing cracks. Methodologies to identify the effect of these growing cracks on the structural performance, reliability, and redundancy levels are still required.

4 Conclusions

The ultimate goal of the design process is minimizing the life-cycle cost while maintaining safety at desired levels [18]. Structural systems are subjected to damage and deterioration in time which may lead to sudden failures. Therefore, in addition to basic safety measures such as the reliability index, quantifying the structural redundancy is beneficial. This chapter briefly discussed the redundancy of structural systems in a life-cycle context. Several measures for system redundancy were reviewed. Then, the time-dependent aspects of redundancy are described, and a methodology for assessing time-dependent redundancy is presented. Finally, several quantifications of the redundancy factor used for structural component design are discussed. The chapter also discussed the probabilistic aspects of the fatigue life estimation problem.

In a probabilistic context, system reliability methods should be incorporated to examine structural system behavior. There are several factors which might have significant impact on the redundancy of structural systems, such as material behavior, system modeling type, and correlation effects, among others. These factors need to be included not only in the redundancy assessment of existing structures but also in the design of new structures.

Much research is still necessary to solve uncertainty quantification problems related to structural redundancy and fatigue in real-life applications. Aleatoric and epistemic uncertainties have to be considered and both forward uncertainty propagation and inverse uncertainty quantification have to be used in order to design robust structures and assess and predict the safety of existing structures.

References

1. American Association of State Highway and Transportation Officials (AASHTO): AASHTO LRFD Bridge Design Specifications, 7th edn. American Association of State Highway and Transportation Officials, Washington, DC (2014)
2. Ang, A.H-S., Tang, W.H.: Probability Concepts in Engineering Planning and Design, vol. 2. Wiley, New York (1984)

3. Ayyub, B.M., Assakkaf, I.A., Kihl, D.P., Siev, M.W.: Reliability-based design guidelines for fatigue of ship structures. *Naval Eng. J.* **114**(2), 113–138 (2002)
4. Barsom, J.M., Rolfe, S.T.: Fracture and Fatigue Control in Structures: Applications of Fracture Mechanics. ASTM, West Conshohocken (1999)
5. British Standards Institute (BSI): Steel, Concrete, and Composite Bridges: Code of Practice for Fatigue. 5400-Part 10. British Standards Institute, London (1980)
6. British Standards Institute (BSI): Guide to Methods for Assessing the Acceptability of Flaws in Metallic Structures. BS7910, British Standards Institute, London (2005)
7. Chan, T.H.T., Li, Z.X., Ko, J.M.: Fatigue analysis and life prediction of bridges with structural health monitoring data – Part II: application. *Int. J. Fatigue* **23**(1), 55–64 (2001)
8. Chung, H., Manuel, L., Frank, K.: Optimal inspection scheduling of steel bridges using nondestructive testing techniques. *J. Br. Eng.* **113**, 305–319 (2006)
9. Collette, M., Incecek, A.: An approach for reliability-based fatigue design of welded joints in aluminum high-speed vessels. *J. Ship Res.* **50**(3), 85–98 (2006)
10. Collette, M.: Strength and Reliability of Aluminum Stiffened Panels, pp. 139–198, A Thesis submitted for the Degree of Doctor of Philosophy, School of Marine Science and Technology, Faculty of Science, Agriculture and Engineering, University of Newcastle (2005)
11. Connor, R.J., Fisher, J.W.: Identifying effective and ineffective retrofits for distortion fatigue cracking in steel bridges using field instrumentation. *J. Br. Eng.* **11**(6), 745–52 (2006)
12. Downing, S.D., Socie, D.F.: Simple rainflow counting algorithms. *Int. J. Fatigue* **4**(1), 31–40 (1982)
13. Estes, A.C., Frangopol, D.M.: RELSYS: A computer program for structural system reliability analysis. *Struct. Eng. Mech. Techno – Press* **6**(8), 901–919 (1998)
14. Eurocode 3: Design of Steel Structures Part 1–9, Fatigue Strength. CEN – European Committee for Standardisation, Brussels (2010)
15. Eurocode 9: Design of Aluminium Structures Part 1–3, Additional Rules for Structures Susceptible to Fatigue. CEN – European committee for Standardisation, Brussels (2009)
16. Fisher, J.W.: Fatigue and Fracture in Steel Bridges, Case Studies. Wiley, New York (1984)
17. Fisher, J.W., Kulak, G.L., Smith, I.F.: A fatigue primer for structural engineers, National Steel Bridge Alliance, Chicago (1998)
18. Frangopol, D.M.: Life-cycle performance, management, and optimization of structural systems under uncertainty: Accomplishments and challenges. *Struct. Infrastruct. Eng.* **7**(6), 389–413 (2011)
19. Frangopol, D.M., Curley, J.P.: Effects of damage and redundancy on structural reliability. *J. Struct. Eng.* **113**(7), 1533–1549 (1987)
20. Frangopol, D.M., Klisinski, M.: Material behavior and optimum design of structural systems. *J. Struct. Eng.* **115**(5), 1054–1075 (1989)
21. Frangopol, D.M., Nakib, R.: Redundancy in highway bridges. *Eng. J.* **28**(1), 45–50 (1991). American Institute of Steel Construction, Chicago
22. Ghosh, M., Moses, F.: Redundancy in Highway Bridge Superstructures. NCHRP Report 406. Transportation Research Board, Washington, DC (1998)
23. Guedes Soares, C., Garbatov, Y.: Fatigue reliability of the ship hull girder. *Mar. Struct.* **9**(3–4), 495–516 (1996)
24. Heredia-Zavoni, E., Montes-Iturriaga, R.: A Bayesian model for the probability distribution of fatigue damage in tubular joints. *J. Offshore Mech. Arctic Eng.* **126**(3) 243–249 (2004)
25. Jensen, J.J.: In: Bhattacharyya, R., McCormick, M.E. (eds.) Load and Global Response of Ships. Ocean Engineering Series, vol. 4. Elsevier, Oxford, UK (2001)
26. Hendawi, S., Frangopol, D.M.: System reliability and redundancy in structural design and evaluation. *Struct. Saf.* **16**(1+2), 47–71 (1994)
27. Kim, J.: Finite Element Modeling of Twin Steel Box-Girder Bridges for Redundancy Evaluation. Dissertation, The University of Texas at Austin, Austin (2010)
28. Kim, S., Frangopol, D.M.: Optimum inspection planning for minimizing fatigue damage detection delay of ship hull structures. *Int. J. Fatigue* **33**(3), 448–459 (2011)

29. Kim, S., Frangopol, D.M.: Probabilistic bicriterion optimum inspection/monitoring planning: application to naval ships and bridges under fatigue. *Struct. Infrastruct. Eng.* **8**(10), 912–927 (2012)
30. Kim, S., Frangopol, D.M., Soliman, M.: Generalized probabilistic framework for optimum inspection and maintenance planning. *J. Struct. Eng.* **139**(3), 435–447 (2013)
31. Kwon, K., Frangopol, D.M., Soliman, M.: Probabilistic fatigue life estimation of steel bridges by using a bilinear S-N approach. *J. Bridge Eng.* **17**(1), 58–70 (2012)
32. Leemis, L.M.: Reliability, Probabilistic Models and Statistical Methods. Prentice Hall, Englewood Cliffs (1995)
33. Li, Z.X., Chan, T.H.T., Ko, J.M.: Fatigue analysis and life prediction of bridges with structural health monitoring data – Part I: methodology and strategy. *Int. J. Fatigue* **23**(1), 45–53 (2001)
34. Li, Z., Zhang, Y., Wang, C.: A sensor-driven structural health prognosis procedure considering sensor performance degradation. *Struct. Infrastruct. Eng.* **9**(8), 764–776 (2013)
35. Mansour, A.E., Wirsching, P.H., White, G.J., Ayyub, B.M.: Probability-Based Ship Design: Implementation of Design Guidelines. SSC 392. Ship Structures Committee, Washington (1996)
36. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953)
37. Miner, M.A.: Cumulative damage in fatigue. *J. Appl. Mech.* **12**(3), 159–164 (1945)
38. Moan, T., Song, R.: Implications of inspection updating on system fatigue reliability of offshore structures. *J. Offshore Mech. Arctic Eng.* **122**(3), 173–180 (2000)
39. Neal, R.M.: Slice sampling. *Ann. Stat. Inst. Math. Stat.* **31**(3), 705–767 (2003)
40. Okasha, N.M., Frangopol, D.M.: Advanced modeling for efficient computation of life-cycle performance prediction and service-life estimation of bridges. *J. Comput. Civil Eng.* **24**(6), 548–556 (2010)
41. Paik, J., Wang, G.: Time-dependent risk assessment of ageing ships accounting for general/pit corrosion, fatigue cracking and local dent damage. World Maritime Technology Conference, San Francisco (2003)
42. Paris, P.C., Erdogan, F.A.: Critical analysis of crack propagation laws. *J. Basic Eng.* **85**(Series D), 528–534 (1963)
43. Perrin, F., Sudret, B., Pendola, M.: Bayesian updating of mechanical models-application in fracture mechanics. 18 è, Grenoble, 27–31 (2007)
44. Rabi, S., Karamchandani, A., Cornell, C.A.: Study of redundancy of near-ideal parallel structural systems. In: Proceeding of the 5th International Conference on Structural Safety and Reliability, pp. 975–982. ASCE, New York (1989)
45. Sharp, M.L., Nordmark, G.E., Menzemer, C.C.: Fatigue Design of Aluminum Components and Structures. McGraw-Hill, New York (1996)
46. Shigley, J., Mischke, C.: Mechanical Engineering Design, 5th edn. McGraw-Hill, New York (1989)
47. Soliman, M., Frangopol D.M.: Life-cycle management of fatigue sensitive structures integrating inspection information. *J. Infrastruct. Syst.* **20**(2), 04014001 (2014)
48. Soliman, M., Barone, G., Frangopol, D.M.: Fatigue reliability and service life prediction of aluminum high-speed naval vessels based on structural health monitoring. *Struct. Health Monit.* **14**(1), 3–19 (2015)
49. Soliman, M., Frangopol, D.M., Kim, S.: Probabilistic optimum inspection planning of steel bridges based on multiple fatigue sensitive details. *Eng. Struct.* **49**, 996–1006 (2013)
50. Tada, H., Paris, P.C., Irwin, G.R.: The Stress Analysis of Cracks Handbook. The American Society of Mechanical Engineers, 3rd edn. Three Park Avenue, New York (2000)
51. Tobias, D.H.: Perspectives on AASHTO load and resistance factor design. *J. Br. Eng.* **16**(6), 684–692 (2011)
52. Tsopelas, P., Husain, M.: Measures of structural redundancy in reinforced concrete buildings. II: redundancy response modification factor R_R . *J. Struct. Eng.* **130**(11), 1659–1666 (2004)
53. Wen, Y.K., Song, S.-H.: Structural reliability/redundancy under earthquakes. *J. Struct. Eng.* **129**, 56–67 (2003)

54. Wirsching, P.H.: Fatigue reliability for offshore structures. *J. Struct. Eng.* **110**(10), 2340–2356 (1984)
55. Yazdani, N., Albrecht, P.: Risk analysis of fatigue failure of highway steel bridges. *J. Struct. Eng.* **113**(3), 483–500 (1987)
56. Zhu, B., Frangopol, D.M.: Effects of post-failure material behavior on redundancy factor for design of structural components in nondeterministic systems. *Struct. Infrastruct. Eng.* **11**(4), 466–485 (2014)
57. Zhu, B., Frangopol, D.M.: Redundancy-based design of nondeterministic systems, chapter 23. In: Frangopol, D.M., Tsompanakis, Y. (eds.) *Maintenance and Safety of Aging Infrastructure. Structures and Infrastructures*, vol. 10, pp. 707–738. CRC Press/Balkema, Taylor & Francis Group, London (2014)
58. Zhu, B., Frangopol, D.M.: Effects of post-failure material behavior on the reliability of systems. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part A: Civil Eng.* **1**(1), 04014002, 1–13 (2015)
59. Zhu, B., Frangopol, D.M., Kozy, B.M.: System reliability and the redundancy factor by simplified modeling. In: Furuta, H., Frangopol, D.M., Akiyama, M. (eds.) *Assessment, Maintenance and Management*, p. 148. CRC Press/Balkema, Taylor & Francis Group, London (2015) and Full Paper on DVD, Taylor & Francis Group PLC, London, pp. 614–618 (2015)

Matthew Collette

Abstract

Uncertainty quantification in ship structural performance remains uneven. While the ship structure community has long developed probabilistic models for the ocean environment and structural responses, complete quantification frameworks remain the exception not the rule. Much of the current uncertainty focus is around design code development, and more piecemeal responses in areas where current design codes are not adequate. This chapter briefly reviews the current state of the art in ship structural performance, including examples of failures in service. Then, the current design code landscape is reviewed. In design codes today, partial safety factors make scattered appearances, but a wide adoption of such measures is still some time off. While more general concepts of risk are gaining traction in the marine industry for risk-based approval, direct structural performance simulation remains the exception rather than the rule. Different uncertainty types and underlying data are then presented and reviewed. Initially, basic design parameters and underlying operational data are discussed, and then strength and loading models are presented. The fragmented nature of uncertainty quantification is clear across all these topics – while many key responses are described in stochastic frameworks, an equal number of modeling, fabrication, and operational parameters are subject to a large amount of epistemic uncertainty today. This limits the number of integrated uncertainty quantification computations that can be completed today. Finally, an overview of published uncertainty frameworks to date is presented. While none of these frameworks are all-encompassing, they demonstrate both the practical application of uncertainty quantification to current problems and a foundation for future probabilistic modeling advances.

M. Collette (✉)

Department of Naval Architecture and Marine Engineering, University of Michigan,
Ann Arbor, MI, USA

e-mail: mdcoll@umich.edu

Keywords

Goal-based standards • Common structural rules • Marine • Ship • Ocean • Design code • Fabrication • LRFD • Uncertainty framework • Modeling uncertainty • Modeling idealization

Contents

1	Introduction	1568
2	Current Uncertainty Handling in Design Codes	1571
3	Uncertainty Types and Values	1572
3.1	Types of Uncertainty	1572
3.2	Uncertainty Data Related to Basic Design Parameters	1573
3.3	Uncertainty Data Related to Modeling and Experiments	1578
4	Proposed Frameworks to Date	1583
5	Conclusions	1584
	Cross-References	1585
	References	1585

1 Introduction

The application of uncertainty approaches to ship structural performance remains a paradox. While the ship structure community has been at the forefront of probabilistic model development in the past century, holistic and formal uncertainty quantification is almost unheard of in the ship structure community. Instead, the industry developed a set of piecemeal approaches to handle uncertainty. Different techniques, characterizations, and solution methods are used in different areas of the structural design problem. This fragmented approach allows tailoring of the uncertainty method to the specific subproblem at hand. However, the fragmented approach prevents easy integration of various uncertainty sources into a holistic uncertainty or risk measure. Finally, certain parts of structural analysis and decision making still do not explicitly consider uncertainty. Instead they follow traditional allowable-stress decision formats with a single, integrated safety factor based on experience.

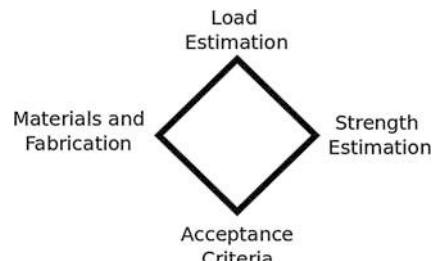
The fragmented uncertainty approach currently in use in the marine industry is a product of several underlying trends. The ocean environment is fundamental stochastic over short time periods, so the need for a probabilistic approach is clear. However, the decentralized growth in analysis methods and regulatory frameworks throughout the long history of the industry has resulted in many independent domains which treat uncertainty in their own way. Movement toward a more integrated uncertainty approach has also been retarded by several unique features of the ship design and analysis problem. Ships are highly flexible assets – inherently capable of moving through different environments across the globe. Furthermore, they are often repurposed or have their original service life extended. Thus, at the outset of their design, their lifespan and future operating conditions are largely unknown. Likewise, regulators are often faced with the need to certify vessels being repurposed whose initial design assumptions and history are largely unknown to

them. Ships are also procured in one-off or short-series production environments. This production environment limits both the time and resources available for custom engineering. For the ship's structure, prototype strength testing and material and production variability surveys are normally cost-prohibitive and not attempted. In this context, existing uncertainty approaches in the marine industry remain fragmented despite past attempts to unify them (e.g., [1]).

The lack of a rigorous uncertainty model has not been without peril in the marine industry. Under the piecemeal approach currently in use, it is possible for new developments in technology to violate the often opaque network of safety factors that keep ship structures safe. This opaque safety factor network arises from assumptions about analysis procedures and construction material and tolerances that are not made explicit in the procedure. Rather, factors of safety are adjusted by experience until satisfactory performance is achieved. But in doing so, load estimation, strength models, material and fabrication standards, and acceptance criteria become tied into a single analysis procedure. It is then no longer possible to substitute a more advanced technique in one area, as the true division of the safety factor and underlying uncertainty between analysis areas is unknown. This is shown in Fig. 46.1 as a diamond, indicating that changes cannot be made to any point of the diamond without impacting the other corners. This concept is based upon work by Sielski [2], where the same idea was represented as an allowable-stress triangle with implied materials and fabrication standards. In Fig. 46.1, the acceptance criteria and strength estimates are more general and the material and fabrication standards are explicit. However, the larger meaning remains – the design approach is a carefully balanced integration of standardized procedures for each point on the diamond. When this balance is disturbed, what appears to be a satisfactory structural design often suffers damage in service.

One example of this pitfall is the introduction of high-strength steel on the trans-Alaskan pipeline service (TAPS) oil tanker design in the 1960s and 1970s. While the high-strength steel raised the ultimate strength of the tanker structure, the fatigue strength of the structural connections used did not increase proportionally. However, fatigue life predictions were not explicitly made for most ship structures at this time – a simple low working stress was being used to handle both strength and fatigue concerns. Combined with higher-than-average wave loading on a route through the Gulf of Alaska and Northern Pacific ocean, numerous ships suffered significant fatigue cracking. This in turn required extensive repairs and an enhanced

Fig. 46.1 Implied balance in current structural design codes and procedures



inspection regime for these vessels to ensure a sufficient level of safety. Barsom and Rolfe present a case study of this problem in Chapter 17 of their fracture mechanics textbook [3].

A second example is the result of the “take tons off sensibly” (TOTS) program in the US Navy. Here, the US Navy reused a smaller hull form from the successful DD-963 class destroyer for the larger CG-47 class cruiser. In an effort to expand the combat capability and increase the weight available for upgrades in the CG-47 class cruiser design, the US Navy and the shipyards involved extensively reviewed the design of the vessel for potential weight savings. By optimizing the structural design, and using higher-strength steel plating, significant weight savings were obtained [4]. Unfortunately, all sources of uncertainty around structural loading and response were not quantified during this optimization. In fact, the hullform is atypical with large bow flare that can dramatically increase impulsive loads from wave impact. Additionally, a new deck-piercing vertical launch system (VLS) for missiles was added to the foredeck near the forward end of the superstructure, creating complex local stress fields. A slight knuckle also exists in the deck forward of the superstructure before the VLS module, leading to an eccentric load path on the deck under compressive loads. While the traditional factors of safety were sufficient to avoid failures in the non-optimized structure without the VLS, the optimized structure suffered frequent deck buckling at the knuckle and fatigue damage in-service [5]. Costly replating and structural retrofits of most vessels in this class were then required. The experience with the TAPS vessels and the CG-47 class highlights the need for better uncertainty management in the marine structural industry.

In addition to in-service failures, the growth of both complex numerical modeling and on-board instrumentation is further highlighting the need for better uncertainty management. The growth in the power of numerical modeling in the past two decades is encouraging engineers to increase the detail of the analysis used in design. This increased power to estimate responses is leading to more aggressive optimization of structures. However, these more advance models also typically require many more inputs of fundamentally stochastic parameters. Without expanding the uncertainty analysis and quantification used in this analysis, a highly precise answer is obtained with unknown variance. Likewise, the rapid growth in in-service monitoring and data recording has led to an increase desire to make structural diagnosis and prognosis decisions for actual structures. However, for most vessels, the as-built and as-repaired condition is rarely fully documented. Such structure-specific analysis also requires an enhanced understanding and quantification of these sources of uncertainty.

The remainder of this chapter summarizes the current state of the art and challenges in uncertainty approaches in ship structural performance. Current approaches in marine industry design codes are presented in Sect. 1. Section 2 discusses the types of uncertainty underlying the ship structural performance problem. Specific focus is given to both uncertainties in the physical system and uncertainties in the modeling and idealization of the structure. Section 3 presents a brief listing and overview of several proposed integrated uncertainty frameworks, and Sect. 4 presents conclusions.

2 Current Uncertainty Handling in Design Codes

All ship structure codes fundamentally balance load capacity and demand to ensure safety in service. Historically, such codes were based on experience and simplified, deterministic calculations that were at best broadly representative of the real loading and structural failure modes occurring in service. Formulations for load magnitudes at specified exceedance probabilities first became possible after a landmark paper in 1953 [6]. This paper extended the electronic filter analogy to a ship operating in waves, allowing the spectrum of the ship's response to be calculated for known wave spectra. Just over a decade later, Dunn [7] advocated a reliability-based approach to the broader problem of design including strength. Dunn framed this argument in terms of stochastic variables following defined distributions, with safety measured probabilistically. Dunn took his inspiration from work with electronics and missile system which were adopting similar formats in the marine industry at this time.

From these initial works, uncertainty rapidly entered the ship structural community via conventional structural reliability analysis (SRA). This development paralleled the rapid advance of reliability in the civil engineering industry. However, unlike civil engineering, reliability did not make the jump to structural design codes via load and resistance factor design (LRFD). While several proposals were made for using reliability, safety indexes, partial safety factors, and LRFD in ship design codes [1, 8, 9], industry-wide adoption of LRFD or partial safety factors has not occurred in the marine industry.

Instead, uncertainty handling via reliability is incorporated into certain areas of design codes. The international association of classification societies (IACS) has developed a standardized code, known as the common structural rules (CSR). The CSR are also a response to the International Maritime Organization (IMO) move toward goal-based standards. Goal-based standards specify the desired safety or performance outcome for a design, instead of specifying prescriptive safety solutions. However, the IMO specification for the goal-based standards only requires that uncertainty be addressed [10] but does not specify how. The current CSR includes the option of using a partial safety factor approach for hull girder collapse limit states. However, other sections of the CSR are still formulated as working stress. IACS has also worked to standardize stochastic descriptions of the ocean environment for design and standardize that extreme loads should be calculated with an exceedance probability of 10^{-8} per wave cycle.

Structural reliability approaches are codified as optional or extra procedures for unusual vessels or situations. These approaches typically go far beyond static LRFD codes. They include direct involvement of the design engineer in building the stochastic uncertainty model. For example, the classification society DNV-GL has issued a classification note on this topic DNV [11]. This note includes guidance on setting up the reliability problem, characterizing the stochastic variables via distributions, computing and evaluating the resulting reliability, and performing inspection updating. However, the application of such approaches remains optional.

The emergence of marine risk-based design methods [12] outside of reliability has also increased the demand for uncertainty modeling. Risk-based design attempts

to quantify risk levels for a variety of events, including many not directly related with structural performance. Much of this work has remained in academic circles to date (e.g. the work of [13] on ship collision probability). Furthermore, performance issues of ship grounding, collision, stability, fire, and evacuation have been investigated in far more detail than structural response. However, a few transitions into codes that address structures have occurred. The ABS classification society has issued a guidance note on approval of novel concepts [14] that codifies how uncertainties are to be handled by a range of quantitative risk assessment procedures beyond structural reliability analysis.

In summary, the current state of the art for handling uncertainties in the marine structural community is fragmented. Certain concepts, such as the wave environment and extreme load exposure levels, are well codified in stochastic models. Other topics such as structural reliability assessment and quantitative risk analysis have been partially codified. However, these tools are typically reserved for novel or special concepts today. Furthermore, no industry-wide standard exists for the implementation of such methods.

3 Uncertainty Types and Values

3.1 Types of Uncertainty

To better understand the nature of uncertainty handling in ship structural design, it is useful to categorize the uncertainties by two different approaches. The first is to separate the uncertainty by its corresponding physical component or mathematical model that it impacts. This section is structured according to this principle. Initially, uncertainties impacting the material and as-fabricated state of the structure are presented. These are followed by uncertainties related to the structure's loading, which includes both environmental descriptions as well as the operational profile associated with the structure. Finally, uncertainties associated with numerical and experimental calculations are addressed.

The second approach in classifying uncertainties is the familiar if philosophically contentious division into *aleatory* and *epistemic* uncertainties. Aleatory uncertainties, those which are fundamentally stochastic properties, primarily impact the basic material and environmental properties for ship structures. In general, statistical data is available for many of these properties. Epistemic uncertainties are those for which there is a general lack of knowledge about the magnitude and distribution of the uncertainty term. For ship structures, operational profiles and behavior, model uncertainty, and the impact of the human engineer comprise the primary epistemic uncertainties. As-fabricated properties, which include the impact of welding residual stresses, misalignment, and similar departures from the ideal structural design, represent a gray zone between the aleatory and epistemic worlds. For some of these topics, limited statistical data exists, while for others there is little generally applicable data available. This division matters in performance prediction,

Table 46.1 Summary of basic material property data uncertainty from Hess et al. [15]

Variable	Mean value	COV	Recommend distribution
Plate thickness, t	1.05 t	0.044	Lognormal
Yield stress, σ_y , ordinary strength steel	1.3 σ_y	0.124	Lognormal
Yield stress, σ_y , high-strength steel	1.19 σ_y	0.083	Lognormal
Ultimate strength, σ_u all steels	1.05 σ_u	0.075	Normal
Elastic modulus, E, all steels	0.987 E	0.076	Normal

as the toolbox for predicting performance with aleatory uncertainties is much richer than that with epistemic uncertainties.

3.2 Uncertainty Data Related to Basic Design Parameters

3.2.1 Structural Geometric and Material Uncertainty

As objects built in either bespoke or short-run production settings, ship structures are primarily assembled from standardized structural components. Hence, statistical data on underlying material and structural shape properties are critical sources of variability. Hess et al. [15] provide a recent listing of material data, including basic material properties such as yield strength and elastic modulus. A summary of this data is shown in Table 46.1. Additionally, deviations from nominal material properties or nominal structural dimensions for structural components are provided. Two-parameter distributions of normal, lognormal, and extreme value distribution are provided for most parameters, with the exception of Poisson's ratio, which is treated as deterministic. While fatigue cracking of built-up structures will be considered later, the fracture toughness of naval steels is an important base material characteristic for crack resistance. Unfortunately, no general distribution of fracture toughness has been proposed to date, with distributions available by steel type, or in the solely in the format of raw data. Notably, Ship Structure Committee report SSC-352 provides extensive Charpy toughness data on a wide range of ship steels from mild steel to high-strength alloys, with about 10,000 individual test results [16]. Sumpter and Kent [17] provide a more recent but more limited investigation into the fracture properties of higher-toughness ship steels.

For the most part, aleatory-type data is available for the underlying material properties and dimensional data of the structural component of ships. Important exceptions to this statement are mainly related to market impact on the structural material. For example, the recent surge in ultra-large containerships (over 10,000 twenty-foot equivalent units (TEU) in capacity) has led to the demand for plates exceeding 50 mm or even 75 mm in thickness, where there is little statistical data to date. Hess et al. [15] also note that contractual ordering practice – paying per pound or per plate piece – can result in bias toward plates that are either at the higher end (paying per pound) or the lower end (paying per piece) of the overall thickness distributions reported.

Through the shipbuilding process, additional stresses and deformations occur in the structure that must be considered for structural performance. The assembly of small structural components into large structures via fusion welding induces residual stresses and geometric imperfections into the structure. As buckling-critical thin shell structures, such deformations and stresses will reduce the ultimate collapse load of the structure. Recent studies have provided data for residual stresses in as-built laboratory specimens (e.g., [18]). Longer-range residual stresses are also known to act on structures. These longer-range stresses result from the subsequent assemblies of smaller structural components into units, blocks, and grand-blocks and finally the finished vessel via welding. Statistical data of as-built finished vessels would be ideal; however, few sources of such data exist.

Smith et al. [19] provides one of the largest databases on such stresses and deformations. Smith et al. summarized field measurements from box-girder bridges and ship structures taken in the 1970s that included both geometric imperfections and residual stresses. Smith et al. then developed slight, average, and severe distortion measurements which they correlated to the mean value and 3% and 97% cumulative probability values of a lognormal distribution fit to their data. The underlying data for this method is now approaching 40 years of age. Its current applicability is uncertain as welding technology has substantially improved over time. Smaller numerical and experimental studies of simple structures have been published since then. These approaches used methods such as FEA analysis of welding operations and taking measurement via the neutron diffraction method (e.g., [20, 21]). However, the community still lacks general predictive statistics that can be run a priori. Therefore, there is both aleatory and epistemic uncertainty present in most deformation and residual stress estimation today.

Unfortunately, the geometric imperfections and residual stresses do not remain constant over a vessel's life. Seaway loading, impact of cargo, tugboats, piers, and assorted floating objects can all increase the amount of geometric distortion present in the structure with time. Firm stochastic information about the impact of these through-life changes is not generally available, though a few studies have begun to shed light on these topics. Jennings et al. [22] provide detailed measurements of in-service structural deformations at 23 locations divided among 13 different ships, including both commercial and military vessels. While deflection data on a grid of points is reported, no statistical distributions are fitted and the total number of samples is rather small considering the variety of structures and locations covered.

Shakedown is another time-varying phenomenon that can reduce the residual stresses in structure over time. In welded plate structures, the peak residual stresses are usually colocated with the welds. These stresses are tensile and result from the need to hold the weld to a roughly constant length as it cools from the solidification point to room temperature. For equilibrium, they must be balanced by compressive stresses away from the weld toe. Typically, the tensile block residual stresses approach the yield strength and strain of the material. The compressive stresses are lower owing to the larger volume they impact and the need for the structure to be in equilibrium. When further tensile loading is applied to the structure in service, the tensile block plastically deforms as it can no longer support higher stress levels.

When this loading is removed, the tensile block unloads elastically to a new, lower, stress level. In this process, applied loads wash out or “shakedown” the residual stresses in the structure.

Much like as-built residual stresses, the community currently lacks widely applicable models to estimate shakedown effects. Gannon et al. [23] reported up to 40% reduction in residual stresses in a stiffened panel with applied loads of roughly 50% of the yield stress. Syahroni and Berge reported on shakedown of peak residual stresses at fatigue notches, including experimental investigation [24]. Similar to Gannon et al. they found applied tensile loads largely reduced the residual stresses at the fatigue notches. The experiments indicated that it is possible to completely eliminate the residual stresses when the tensile overload is 85% of the yield stress of the base metal. While this process has been theoretically explained and experimentally verified, practical uncertainty characterization has not yet been completed. Thus, while basic material properties are well understood, the as-built and in-service shape and stresses in the structure still include epistemic uncertainties.

3.2.2 Structural Degradation Data

In addition to the time-varying changes to deformations and residual stresses, naval structures also degrade over time. Chief among the causes of degradation is corrosion. The marine environment is highly corrosive to most ferrous metals, and corrosion of untreated structural steel is often rapid. Both overall corrosion and pitting corrosion of the structure impact the structure’s ability to function. Overall corrosion impacts structural stress and collapse loads, while pitting corrosion can compromise the hull or tank envelop function of the structure. Further complicating matters, both the environmental and underlying structural stress and deflection characteristics change significantly between ships or even between parts of the same ship. To date, two main approaches have been taken for probabilistic corrosion models. One is based on full-scale measurements from ships in service. For example, Paik et al. [25] fit a shipboard-location-dependent corrosion model to 33,820 oil tanker data points. They adopted a simple stochastic linear corrosion model for the increase of corrosion depth with time and determined the linear coefficient to best follow a Weibull distribution. However, the variability of the data was quite large with underlying population coefficients of variation in excess of 0.8 for most locations and frequently above 1.0. Wang et al. [26] provide similar data using over 110,000 samples, but only provide corrosion amounts at fixed cumulative probabilities, not underlying distribution.

The second corrosion approach taken is to develop a more theoretical model based on scientific studies of smaller coupons and then extend this to the full-scale structure. Melchers [27, 28] provides models for both general and pitting corrosion from this perspective, including coefficient variability and a proposed Frechet distribution for extreme pit depth. However, impacts such as structural cleaning and structural deformations that may disturb corrosion products and recoating are not yet included in such theoretical model. The ideal corrosion model, which can take environmental parameters as a variable and predict probabilistic corrosion over time,

has not yet been formulated; however, reasonable statistical data is now available from these two approaches for common ship structures.

Structural fatigue cracking is another time-varying change to the structure's integrity, load paths, and remaining structural capacity. As one of the few large, weld-fabricated structures that see frequent complete reversal of primary stress fields from tension to compression, naval structures are prone to structural fatigue cracking. In turn, structural fatigue cracking has historically been marked by high variability. Aleatory uncertainties have been shown to be large for structural fatigue crack initiation. It is not uncommon to have otherwise-identical fatigue details fail in laboratory testing with more than one order of magnitude difference in fatigue life under equal loading. Such uncertainty can be recorded in laboratory experiments. However, there are equally significant epistemic uncertainties in fatigue life prediction for most ship structures. Ship structures have historically been fabricated with looser construction tolerances than assembly-line structures such as aerospace structures, and as a result, the weld toe profile, misalignment, and residual stress present in the final as-built structure are often unknown at the design stage. These factors can equally influence the final fatigue life as much as the aleatory uncertainties inherent in the fatigue of large welded structures.

To date, the most common practice is to lump all these uncertainties together and capture aleatory uncertainty measures from laboratory experiments under the assumption that the laboratory specimens represent industrial fabrication tolerances. There have been a few papers focused on the fatigue-related uncertainty in isolation. Wirsching [29] presented the first major study into fatigue crack initiation reliability, including uncertainty in the S-N equation. Collette and Incecik tested simplified reliability formulations against aluminum structural test data and summarized recommended uncertainty measures [30]. Ayyub et al. summarized similar data and proposed LRFD rules for naval ship structures [31], while Folsø et al. performed a similar exercise for commercial ships, including significant data analysis [32]. Depending on the production processes followed, statistical recommendations from the wider engineering community, such as those included in the International Institute of Welding (IIW) publications, may also be applicable to ship structures. For fatigue crack propagation, far fewer studies have been conducted into reasonable values for uncertainty terms in the crack growth laws. In summary, the aleatory aspects of fatigue degradation are relatively well documented for laboratory tests of ship structural components. The epistemic uncertainties have received far less attention to date.

3.2.3 Structural Operational Data

In addition to basic material and structural uncertainty parameters, there are significant uncertainties surrounding the operational condition of the structure. Element such as the operational loading, crew's weather-routing decisions, and the wave climate will all significantly impact the actual loading experienced by the structure. Similar to the basic material data presented above, these uncertainties include both epistemic and aleatory contributions. The state of knowledge varies widely for the

individual topics within the scope of operational data. The operational impact of loading, wave climate, and crew's weather routing will each be reviewed in turn.

The operational weight distribution is the first major contribution to the structural loading of the vessel. Differences in weight and buoyancy along and across the hull lead to large hydrostatic forces and bending moments throughout the structure. While the hydrostatic external loading from the seawater pressure is known with high precision with the vessel at rest, the weight distribution on the vessel is not constant and subject to extreme variability. While the ship's center of gravity is verified after launching, the distribution of mass along the hull is typically not verified, and thus the as-built weight distribution is subject to uncertainty. Once operational, the crew will adjust cargo, fuel, consumables, and ballast water as necessary to maintain adequate trim and stability. These large variable loads are termed still water loads and are typically marked by high uncertainty. Guedes Soares and Moan [33] provide a database of measured still water loads for seven standard ship types and summarized uncertainty measures. The measured data indicated that a normal distribution is appropriate, but that the per-ship coefficient of variation (COV) may be quite high, approaching 0.4.

The ocean wave climate is often represented by engineering idealizations during the assessment of ship structures. The real ocean environment may include multidirectional wave systems (e.g., swell and wind-driven seas from different storm systems) as well as local bathymetric influences. Engineering design idealizes this situation using linear wave theory to reduce the ocean environment into single- or two-parameter unidirectional energy spectra. These spectra are given as functions typically parameterized by wave height and wave period. To represent the different sea states that are likely to occur in a given area, a scatter diagram is created where the relative frequency of specific wave height and period combinations are listed based on observations. The International Association of Classification Societies (IACS) publishes a recommended wave scatter diagram and spectral formulation for worldwide service in Recommendation [34]. Such formulations allow the wave environment to be presented in a framework that includes aleatory uncertainties. Primary epistemic uncertainties include the relevance of the worldwide model to local conditions, the difficulty in predicting extreme values and departure from linear wave theory for extreme wave heights [35], and potential future impacts of climate change on the scatter diagrams [36].

Within the context of the ocean wave environment, the vessel's crew can still decide heading angle and ship's speed with respect to the wave field. Furthermore, using weather forecasts for wide areas of the ocean, the crew can attempt to avoid the worst weather by planning the routing several days in advance. This ability to weather route means that the ocean environment experienced by the vessel may differ from that observed by wave buoys or otherwise stationary observers. Data on crew's decisions with respect to speed and heading can be gathered from two different approaches. Statistical data mining of ship logs can be used to develop operational profiles for vessels, indicating the probability of taking a specific speed and heading given a prevailing sea state. Sikora [37] presents recent data for operational profiles of naval combatants and auxiliaries.

Alternatively, seakeeping calculations for a new design can be made to approximate conditions where the crew would slow down or change heading for comfort or machinery limitation reasons. While many operationally based studies have proposed weather routing algorithms for logistics and fuel savings, few authors have directly addressed the impact of such uncertainties on loading and strength calculations. Sternsson and Björkenstam [38] studied the logs of one type of commercial ship and noted that weather routing significantly reduced the vessel's exposure to wave heights above 7 m, which constitutes most of the extreme hull girder loading. Shu and Moan [39] used numerical simulation to study the impact of crews changing heading or speed based upon the seakeeping performance of the vessel. Their results showed a significant decrease in the vertical bending moment on the ship if such reductions are considered. Most recently, Papanikolaou et al. [40] also examined the uncertainty surrounding on-board decision support systems to assist in such heading and speed choices. In summary, a probabilistic operational profile is now commonplace in ship design, and modification to account for weather routing and crew response is possible. However, the community lacks general statistical data for the uncertainties that go into determining the values in such a profile.

3.3 Uncertainty Data Related to Modeling and Experiments

In addition to the uncertainty surrounding the basic design parameters, further uncertainty in naval structural performance enters from the use of experimental and numerical methods to predict the structure's performance. This uncertainty can also be broken down into aleatory and epistemic categories; however, a more nuanced breakdown is also useful. Unlike basic design variables, modeling and experiments fundamentally involve:

- idealization of the actual responses of interest,
- representation via a potentially imperfectly implemented model of the idealization,
- interaction with a human engineer who may have their own limited understanding of both the idealization and the model.

Experience has shown that such human impacts can be significant, especially for calculations done in a very small community or infrequently, where both the tool reliability and operator knowledge may be less than ideal.

Additionally, compared to basic design parameters, stochastic model of uncertainty in idealizations and models is generally less mature in the marine industry. However, this topic has grown in importance and research activity lately. Several initial studies into modeling and tool uncertainty took place in the 1990s and will be reviewed in this section. A major landmark was the joint International Towing Tank Conference (ITTC) and International Ship and Offshore Structures Congress (ISSC) workshop on stochastic uncertainty modeling for

ship design loads and operational guidance, the papers from which were published in a special issue of *Ocean Engineering* [41].

3.3.1 Uncertainty in Strength Models

Uncertainty in ship structural strength models depends largely on the type of model used to predict stress or strength. Ship structures are investigated in two main approaches, allowable stress and collapse models. Allowable-stress-based approval criteria require linear models to transform the applied loading into stresses. Such stresses are often investigated by a mix of elastic bending theory (e.g., simple beam theory) and linear finite element analysis. Structural deformation and vibration can also be investigated by such approaches. While such an analysis cannot indicate the true margin before collapse, it relies on only limited material and geometric parameters, reducing its exposure to errors in basic design variables that were presented in the previous section. Errors and uncertainty from idealization, modeling, and human error for such models have been investigated, but only limited data is available. A comparison between linear finite element analysis and model results for a containership in the mid-1990s indicated that linear FEA model was able to capture bending and shear stresses within 7% of the true values on average, though the coefficient of variation was relatively high, between 12% and 20%. However, the model struggled much more with torsional-related warping stresses and deformations with errors up to 40% of the true value. In 1996, Östergaard et al. [42] studied a different containership via a series of five different analysis methods, including both FEA and a series of beam analysis approaches. While no full-scale or experimental data was available, the coefficient of variation between the methods was determined to be 9% for the principle stress but 45% for the shear stress. Again, this indicates in part that torsional stresses were harder to capture.

Over the last 20 years since these studies were made, the power of finite element programs has grown. The meshes used for linear analysis today would be much finer than those employed in either study discussed. Regulatory bodies have also made substantial efforts to standardized finite element modeling procedures by issuing detailed modeling guidelines and automating large sections of model creation (e.g., [43, 44]). Such approaches should reduce the variability between different analysts modeling the same structure. However, more recent data on the idealization error of such models has not been published, nor has the role of the human engineering in making modeling decisions been quantified. At the present time, it is difficult to give firm error estimates for the application of FEA approaches.

Idealization and modeling errors also grow in situations where the underlying physics is not clear or is difficult to model via linear methods. Fatigue crack initiation is one such area for ship structures. As most fatigue cracks initiate from notches at weld toes in ship structures, they cannot be directly analyzed by linear FEA. In linear FEA, the local stress approaches infinity at the weld toe owing to the geometric discontinuity at the weld toe. To overcome this, a variety of stress-measuring locations, extrapolation procedures, or geometry model modification procedures have been proposed. Standardized methods have been developed, such as the nominal stress, hot-spot stress, and notch stress approaches. All of these develop

a reference stress related to the stress believed to be causing crack initiation that is calculable with linear finite element analysis. However, while the concept of the reference stress has been standardized, the implementation and modeling do vary, leading to significant modeling uncertainty.

In a 2002 benchmark study, Fricke et al. [45] studied the same weld toe via seven different hot-spot analysis techniques. The predicted fatigue life of the detail varied between 1.8 and 20.7 years, indicating a high level of modeling and idealization uncertainty. A similar study has also been made of notch stress analysis techniques [46], again showing significant variability. While such studies have shown that the variability in fatigue idealizations is high, statistics for a particular given method of analysis on a type of structure are generally not available. In a study investigating aluminum fatigue experimental test results with author-reported hot-spot stress analysis, Collette and Incecik [30] showed that the different test programs were from distinct statistical populations. Thus, a common uncertainty measure was not recommended. Idealization and modeling errors for fatigue analysis remain a topic of investigation for the marine community.

The second major ship structure modeling approach is collapse models where the nonlinear failure of the structure is directly estimated. This is typically done either via simplified semi-analytic models or nonlinear finite element models. As ships are thin-walled shell structures, collapse strength is dependent on far more basic variables than the linear FEA allowable stress method reviewed above. Imperfections, residual stresses, and yield stress variations can all strongly impact the computed collapse strength. Additionally, the modeling is more complex than linear elastic analysis. The uncertainty in such collapse models has been actively investigated for over 30 years.

As would be expected, collapse models for smaller portions of a structure (e.g., an unstiffened plate in isolation) generally perform with less variability than models for more complex assemblies of structural components. This is a result of simpler and better-fitting idealizations at the component level. For a series of stiffened panels tested in compression, results have shown that most simplified analysis methods can predict the strength of the panel within roughly 10% of the true load and the coefficient of variation of the prediction also typically close to 10% [47]. However, these predictions are with fairly complete information about the as-built material yield stresses, imperfections, and residual stresses in the structure – e.g., the underlying material and geometry uncertainties have largely been removed. Modeling such simple structural experiments also present minimal opportunity for human analyst error as the topology, boundary conditions, and relevant failure modes are normally clear by inspection.

For a more complex calculation of the collapse of ship hull, a benchmark calculation between a range of mostly simplified analysis methods was made in 1994 [48] on a 1/3 scale frigate hull that was also tested experimentally. The results showed much greater scatter with most methods having errors between 10% and 20%, with some methods missing by as much as 40%. However, the structure of this test case was much more slender than typically commercial vessels, which may explain some of the struggles of the simplified methods. Recent guidance has

suggested that simplified collapse methods have a bias factor less than 5% and a COV approaching 10% when compared to nonlinear finite element analysis for conventional ships [49].

Errors in nonlinear finite element analysis are not yet clear. In principle, the FEA models are able to capture many more failure modes and mode interactions more successfully than the previous generation of simplified analysis codes. In practice, the number of modeling and idealization decisions that must go into generating and solving such a large model are easily an order of magnitude higher than the simplified analysis codes. Thus, the opportunity for idealization and human analyst errors is higher. Unfortunately, data on FEA model idealization error is quite limited at the moment. Rigo et al. [50] compared six different FEA analyses of an aluminum-stiffened panel with established geometry and boundary conditions. The six analysis runs were done by different authors who used different codes, meshes, and element types. In general, the ultimate strength prediction of the panel fell within a 4% error band, indicating good agreement, though the post-collapse behavior did differ more strikingly.

In 2012, Paik et al. [51] compared ten different overall collapse prediction codes on six different ships. The codes included both nonlinear FEA and various simplified methods. In general, the scatter of the results across all methods was on the order of 10–20%, including the FEA codes. Even within FEA analysis, similar results were seen. For example, the same commercial code (ANSYS) was used by two participants. For most of the vessels, they investigated the results between the two participants differed 10–20%. This indicates the idealization decisions made by the analyst for nonlinear FEA may be a first-order uncertainty in the calculation. Moan et al. [52] provide a framework for more formally assessing such uncertainties, but again without data for the general analyst. Such overall ship collapse analysis with nonlinear finite element analysis has only been tractable for about the past decade with standard FEA codes, so this area is still one in rapid development.

3.3.2 Uncertainty in Loading Models

Similar to strength models, seaway loading models come in various forms. The loading on vessel may be estimated by empirical or semiempirical equations, linear idealizations of the wave-body interaction problem, or nonlinear methods. The nonlinear methods may be further divided into those based on nonviscous potential flow theory and those that included viscosity. Finally, experimental model tests can be run using scale models of the full-scale structure in wave basins or tanks. These five methods form a hierarchy of fidelity, and in general it takes more than an order of magnitude of effort to move up one level. A further complication comes from the need to figure out load combination factors – in general, the different components of load are estimated independently and then need to be combined into design load cases for strength analysis. These load combination factors are also subject to modeling and idealization uncertainty. Loading model uncertainty can also be broken down into aleatory and epistemic categories, as well as categories dealing with modeling or idealization, implementation, and human application errors.

Schellin et al. [53] compared experimental loads on a containership to the predicted load from four different linear, potential-flow analysis tools. These tools included both 2-D and 3-D idealizations of the fluid flow. Extrapolating to design loads with probabilities of 10^{-4} and 10^{-8} showed large variability between the codes, with the 2-D methods somewhat closer and generally above the experimental results. The 3-D method provided predictions that were roughly 75% of the value of the experiments, but why this higher-fidelity method struggled is unclear. A more recent study examined many of the same issues and concluded the bias from the linear seakeeping tools on extreme loads was 0.9 with a COV of 0.1 [54]. However, uncertainties in the nonlinearities that often occur around extreme events were not addressed by either study. In general, such terms remain difficult to estimate today.

Non-extreme load prediction also matters for fatigue assessment of ship structures. Li et al. [55] compared five different hydrodynamic approaches for calculating the fatigue damage on an instrumented containership in oblique seas. The methods ranged from linear 2-D approximations to 3-D nonlinear panel methods. Full-scale stresses were also recorded onboard for comparison. Two different structural models were used to convert the loads to stresses, a beam theory approach, and a finite element model. Many of the methods overestimated the cumulative fatigue damage by a factor of 1.5 to 2 for 20-min records in low sea states, with the highest-fidelity method being the most accurate, generally within about 10% of the true value for one side of the vessel, but off 20–30% on the other side of the vessel. While general design values of bias factors remain elusive, it is clear that the uncertainty associated with load estimation is a first-order component of structural performance assessment.

As the hydrodynamic responses become more non-linear, these uncertainties increase further. A recent study benchmarked a large roll-on/roll-off ferry subject to impulsive slamming loads [56]. Scale model tests from a flexible model were available for this ferry and were used in a blind benchmarking study with six participants. Impulsive loads generally consist of a locally high peak pressure against a small portion of the vessel that excites an overall structural response (termed whipping) in the entire vessel. This study concentrated on the whipping response post-slam, not the local impact pressures. The participants used both 2-D and 3-D FEA models to capture the overall structural response. Notably, the 3-D FEA models struggled to predict the correct modal frequencies and shapes of the experiment more than the 2-D models; this was attributed to the more complex modeling involved. The results for structural fatigue damage from whipping responses were highly scattered with error factors between 2 and 4 common between participants. The fatigue damage mentioned in this paragraph and the previous paragraph is proportional to the predicted stress raised to the third or fourth power. Therefore, the errors in stress analysis are smaller than the raw fatigue damage indicates. However, the limited studies to date make it clear that there is still a high level of variability present in trying to approximate the dynamic response of ship structures. Additionally, estimation of model errors in the local pressure regions has not yet been addressed.

Beyond numerical calculations, scale model experiments are widely used to estimate structural loads. The hydrodynamic experimental community has been at the forefront of uncertainty management in the marine domain. In 2008, the International Towing Tank Conference (ITTC) published recommendations on uncertainty analysis, followed up by a 2011 procedure for uncertainty measurement in seakeeping experiments. This procedure in turn follows the 1995 International Standards Organization ISO-GUM uncertainty approach, in which uncertainties are subdivided into two types, “A” and “B,” with standard uncertainty measures defined. Type “A” uncertainties deal with the repeatability of a given experiment on the experimental apparatus, while Type “B” addressed modeling errors, measuring and calibration errors, and other errors that are likely to be constant throughout an experiment. These values are facility dependent, and thus general values cannot be given. However, a detailed overview of the approach plus sample values for a surface submarine test are given in [57]. Compared to numerical predictions, the uncertainty from such an assessment is clearer and easier to propagate through downstream calculations.

4 Proposed Frameworks to Date

The fragmented nature of uncertainty data and quantification techniques reviewed in the proceeding section have prevented the adoption of a standard uncertainty quantification framework for regulatory approval of ship structures. As reviewed above, hybrid approaches are in place for most regulatory bodies, with mixed uncertainty handling techniques. However, several academic and research studies have proposed the development of more integrated frameworks, which tie multiple uncertainties together in a chained calculation. These studies are largely pointers toward what may be possible in the future, compared to current application. Many of these still only address a fraction of the uncertainties discussed above and generally simplify or ignore the epistemic uncertainties.

Most of frameworks have built off of structural reliability theory from civil engineering applications, though a few have tried to bring in other risk approaches. In general, the early proposals have focused on LRFD design approaches and replacing existing design-stage rules with rules that more consistently treat uncertainty. More recently, authors have focused on the entire through-life problem of understanding, modeling, and predicting adequate structural performance of a degrading ship structure over time. Most of these approaches include structural reliability to include loading and resistance uncertainties, and some include inspection updating as well as uncertainties in degradation models. Recently, some attention has been paid to tying structural performance uncertainty into a more general description of design risk, including consequences of structural failure. Significant bodies of work are highlighted in Table 46.2.

Table 46.2 Recent integrated uncertainty works

Main author	Description	Key references
Mansour	A structural reliability-based design approach for ships, handling material, loading, and model uncertainty. Design stage only. Documented in several Ship Structure Committee reports and journal articles	[8, 9]
Ayyub (1)	A structural reliability-based approach simplified to a load and resistance factor design (LRFD) approach. Design stage only. Published in reports and a special issue of a journal	[1, 15, 31]
Garbatov	A time-varying reliability of a ship's structure including the impact of corrosion, fatigue crack growth, and crew maintenance actions. Uncertainty included in most model and underlying parameters	[58, 59]
Frangopol	A time-varying reliability formulation with redundancy and optimal inspection/maintenance planning adapted from similar civil engineering approaches for bridges. Includes parametric uncertainty in underlying model variables	[60–62]
Faber	A Bayesian-network risk assessment for ship structural failures. Focuses on post-failure performance and consequence based on estimated statistics of structural failure and then optimization of required structural performance for minimal lifecycle costs	[63, 64]
Ayyub (2)	A time-varying structural reliability assessment of ship structures including corrosion and fatigue. Uncertainty included via structural reliability approaches, system-level reliability approaches used to integrate multiple-component reliability measures	[65]
Moan	A structural fatigue reliability including operations in multiple different sea conditions and inspection updating	[66]

5 Conclusions

The inclusion of uncertainty in the analysis of ship structural performance is uneven. While the marine industry has a long history of incorporating probabilistic design concepts into structural analysis and approval, holistic uncertainty approaches are almost entirely lacking. In design codes today, partial safety factors make scattered appearances, but a wide adoption of such measures is still some time off. While more general concepts of risk are gaining traction in the marine industry outside of structures for risk-based approval, direct structural performance simulation remains the exception rather than the rule. In looking toward what is needed for such a transition, the lack of key data required for structural performance analysis under uncertainty is a retarding factor. While some aleatory uncertainties around material properties and environmental conditions are well established, many topics remain in the epistemic realm with little data to guide an analyst today. This is especially true for the in-service structural condition, hydrodynamic load estimation, and the influence of the human analyst on the increasingly complex hydrodynamic and structural

models used. Despite this circumstance, several authors have proposed integrated frameworks that tie multiple pieces of uncertainty together and increasingly include inspection updating and lifecycle performance and risk measures. Such studies point the way to increased use of uncertainty in the future evaluation of ship structural performance.

The author wishes to acknowledge helpful discussion with Dr. Paul Hess of Code 331 at the Office of Naval Research in the USA in framing this chapter.

Cross-References

- [Redundancy of Structures and Fatigue of Bridges and Ships Under Uncertainty](#)
-

References

1. Ayyub, B.M., Assakkaf, I.A., Beach, J.E., Melton, W.M., Nappi, N., Conley, J.A.: Methodology for developing reliability-based load and resistance factor design (LRFD) guidelines for ship structures. *Nav. Eng. J.* **114**(2), 23–42 (2002)
2. Sielski, R.: Aluminum Structure Design and Fabrication Guide. Ship Structure Committee, Washington, DC, SSC-452 (2007)
3. Barsom, J., Rolfe, S.: Fracture and Fatigue Control in Structures. Applications of Fracture Mechanics, 3rd edn. Butterworth-Heinemann, West Conshohocken (1999)
4. Staiman, R.C.: Aegis cruiser weight reduction and control. *Nav. Eng. J.* **99**(3), 190–201 (1987)
5. Keane R. Jr.: Reducing total ownership cost: designing inside-out of the hull. *Nav. Eng. J.* **124**(4), 67–80 (2012)
6. St. Denis, M., Pierson, W.J.: On the motion of ships in confused seas. *Trans. Soc. Nav. Archit. Mar. Eng.* **61**, 280–357 (1953)
7. Dunn, T.W.: Reliability in shipbuilding. *Trans. Soc. Nav. Archit. Mar. Eng.* **72**, 14–40 (1964)
8. Mansour, A.E., Wirsching, P., Lucket, M.D., Plumpton, A.M., Lin, Y.H.: Structural safety of ships. *Trans. Soc. Nav. Archit. Mar. Eng.* **105**, 61–98 (1997)
9. Mansour, A., Wirsching, P., White, G., Ayyub, B.M.: Probability Based Ship Design: Implementation of Design Guidelines. Ship Structure Committee, Washington, DC, SSC-392 (1996)
10. Hoppe, H.: Goal-based standards – a new approach to the international regulation of ship construction. *IMO News Mag.* **2006**(1), 13–17 (2006)
11. DNV, Classification Note 30.6: Structural Reliability Analysis of Marine Structures. Det Norske Veritas, Høvik (1992)
12. Papanikolaou, A.: Risk-Based Ship Design. Springer, Berlin/Heidelberg (2009)
13. Ståhlberg, K., Goerlandt, F., Ehlers, S., Kujala, P.: Impact scenario models for probabilistic risk-based design for ship–ship collision. *Mar. Struct.* **33**, 238–264 (2013)
14. ABS: Guidance Notes on Review and Approval of Novel Concepts. American Bureau of Shipping, Houston (2003)
15. Hess, P.E., Bruchman, D., Assakkaf, I.A., Ayyub, B.M.: Uncertainties in material and geometric strength and load variables. *Nav. Eng. J.* **114**(2), 139–166 (2002)
16. Kaufman, J., Prager, M.: Marine Structural Steel Toughness Data Bank. Ship Structure Committee, Washington, DC, United states, SSC-352 (1990)
17. Sumpter, J.D.G., Kent, J.S.: Fracture toughness of grade D ship steel. *Eng. Fract. Mech.* **73**(10), 1396–1413 (2006)

18. Paik, J., Thayamballi, A.K., Ryu, J., Jang, C., Seo, J., Park, S., Soe, S., Renaud, C., Kim, N.: Mechanical Collapse Testing on Aluminum Stiffened Panels for Marine Applications. *Ship Structure Committee*, Washington, DC, SSC-451 (2007)
19. Smith, C.S., Davidson, P.C., Chapman, J.C., Dowling, P.J.: Strength and stiffness of ship's plating under in-plane compression and tension. *Trans. R. Inst. Nav. Archit.* **130**, 277–296 (1988)
20. Kenno, S.Y., Das, S., Kennedy, J.B., Rogge, R.B., Gharghouri, M.: Residual stress distributions in ship hull specimens. *Mar. Struct.* **23**(3), 263–273 (2010)
21. Gannon, L., Liu, Y., Pegg, N., Smith, M.: Effect of welding sequence on residual stress and distortion in flat-bar stiffened plates. *Mar. Struct.* **23**(3), 385–404 (2010)
22. Jennings, E., Grubbs, K., Zanis, C., Raymond, L.: Inelastic Deformation of Plate Panels. *Ship Structure Committee*, Washington, DC, SSC-364 (1991)
23. Gannon, L.G., Pegg, N.G., Smith, M.J., Liu, Y.: Effect of residual stress shakedown on stiffened plate strength and behaviour. *Ships Offshore Struct.* **8**(6), 638–652 (2013)
24. Syahroni, N., Berge, S.: Fatigue assessment of welded joints taking into account effects of residual stress. *J. Offshore Mech. Arct. Eng.* **134**(2), 021405–021405 (2011)
25. Paik, J.K., Lee, J.M., Hwang, J.S., Park, Y. II: A time-dependent corrosion wastage model for the structures of single- and double-hull tankers and FSOs and FPSOs. *Mar. Technol.* **40**(3), 201–217 (2003)
26. Wang, G., Spencer, J., Sun, H.: Assessment of corrosion risks to aging ships using an experience database. *J. Offshore Mech. Arct. Eng.* **127**(2), 167–174 (2005)
27. Melchers, R.E., Jeffrey, R.J.: Probabilistic models for steel corrosion loss and pitting of marine infrastructure. *Reliab. Eng. Syst. Saf.* **93**(3), 423–432 (2008)
28. Melchers, R.E.: Extreme value statistics and long-term marine pitting corrosion of steel. *Probab. Eng. Mech.* **23**(4), 482–488 (2008)
29. Wirsching, P.: Fatigue reliability for offshore structures. *J. Struct. Eng.* **110**(10), 2340–2356 (1984)
30. Collette, M., Incevik, A.: An approach for reliability-based fatigue design of welded joints on aluminum high-speed vessels. *J. Ship Res.* **50**(1), 85–98 (2006)
31. Ayyub, B.M., Assakkaf, I.A., Kihl, D.P., Siev, M.W.: Reliability-based design guidelines for fatigue of ship structures. *Nav. Eng. J.* **114**(2), 113–138+207 (2002)
32. Folsø, R., Otto, S., Parmentier, G.: Reliability-based calibration of fatigue design guidelines for ship structures. *Mar. Struct.* **15**(6), 627–651 (2002)
33. Soares, C.G., Moan, T.: Statistical analysis of stillwater load effects in ship structures. *Soc. Nav. Archit. Mar. Eng.-Trans.* **96**, 129–156 (1988)
34. IACS: Standard Wave Data. Corrected Nov. 2001. IACS (1992)
35. Jonathan, P., Ewans, K.: Statistical modelling of extreme ocean environments for marine design: a review. *Ocean Eng.* **62**, 91–109 (2013)
36. Bitner-Gregersen, E.M., Eide, L.I., Hørte, T., Skjøng, R.: *Ship and Offshore Structure Design in Climate Change Perspective*. Springer Science & Business Media, Berlin (2013)
37. Sikora, J.P., Michaelson, R.W., Ayyub, B.M.: Assessment of cumulative lifetime seaway loads for ships. *Nav. Eng. J.* **114**(2), 167–180 (2002)
38. Sternsson, M., Björkenstam, U.: Influence of weather routing on encountered wave heights. *Int. Shipbuild. Prog.* **49**(2), 85–94 (2002)
39. Shu, Z., Moan, T.: Effects of avoidance of heavy weather on the wave-induced load on ships. *J. Offshore Mech. Arct. Eng.* **130**(2) (2008)
40. Papanikolaou, A., Alfred Mohammed, E., Hirdaris, S.E.: Stochastic uncertainty modelling for ship design loads and operational guidance. *Ocean Eng.* **86**, 47–57 (2014)
41. Hirdaris, S.: Special issue on uncertainty modelling for ships and offshore structures. *Ocean Eng.* **86**, 1–2 (2014)
42. Östergaard, C., Doglioni, M., Guedes Soares, C., Parmentier, G., Pedersen, P.T.: Measures of model uncertainty in the assessment of primary stresses in ship structures. *Mar. Struct.* **9**(3–4), 427–447 (1996)

43. GL: Rules for Classification and Construction – V Analysis Techniques – Hull Structural Design Analysis – Guidelines for Global Strength Analysis of Container Ships. Germanischer Lloyd SE (2011)
44. DNV: Classification Note 34.1: CSA-Direct Analysis of Ship Structures. Det Norske Veritas (2013)
45. Fricke, W., Cui, W., Kierkegaard, H., Kihl, D., Koval, M., Mikkola, T., Parmentier, G., Toyosada, M., Yoon, J.-H.: Comparative fatigue strength assessment of a structural detail in a containership using various approaches of classification societies. *Mar. Struct.* **15**(1), 1–13 (2002)
46. Fricke, W., Bollero, A., Chirica, I., Garbatov, Y., Jancart, F., Kahl, A., Remes, H., Rizzo, C.M., von Selle, H., Urban, A., Wei, L.: Round robin study on structural hot-spot and effective notch stress analysis. *Ships Offshore Struct.* **3**(4), 335–345 (2008)
47. Hughes, O., Nikolaidis, E., Ayyub, B., White, G., Hess, P.: Uncertainty in Strength Models for Marine Structures. Ship Structure Committee, Washington, DC, SSC-375 (1994)
48. Jensen, J.J.: Ductile Collapse Committee III.1. In: Proceedings of the 12th International Ship and Offshore Structures Congress, St. Johns, pp. 299–388 (1994)
49. Amlashi, H.K.K., Moan, T.: A proposal of reliability-based design formats for ultimate hull girder strength checks for bulk carriers under combined global and local loadings. *J. Mar. Sci. Technol.* **16**(1), 51–67 (2011)
50. Rigo, P., Sarghiuta, R., Estefen, S., Lehmann, E., Otelea, S.C., Pasqualino, I., Simonsen, B.C., Wan, Z., Yao, T.: Sensitivity analysis on ultimate strength of aluminium stiffened panels. *Mar. Struct.* **16**(6), 437–468 (2003)
51. Paik, J.K., Amlashi, H., Boon, B., Branner, K., Cardis, P., Das, P.K., Fujikubo, M., Huang, C.H., Josefson, L., Kaeding, P., Kim, C.W., Parmentier, G., Pasqualino, I., Rizzo, C.M., Vhammane, S., Wang, X., Yang, P.: Committee III.1: Ultimate Strength. In: Proceedings of the 18th International Ship and Offshore Structures Congress, vol. 1, 3 vols., pp. 285–363. Schiffbautechnische Gesellschaft e.V., Hamburg (2012)
52. Moan, T., Dong, G., Amlashi, H.K.K.: Critical assessment of ultimate hull girder capacity of ships from a reliability analysis point of view. In: Maritime Transportation and Exploitation of Ocean and Coastal Resources, Two Volume Set, Volume 1., pp. 477–485. Lisbon, Taylor & Francis (2006)
53. Schellin, T.E., Östergaard, C., Guedes Soares, C.: Uncertainty assessment of low frequency load effects for containerships. *Mar. Struct.* **9**(3–4), 313–332 (1996). SPEC. ISS.
54. Parunov, J., Senjanoviæ, I.: Incorporating model uncertainty in ship reliability analysis. *Trans. Soc. Nav. Archit. Mar. Eng.* **111**, 376–408 (2003)
55. Li, Z., Mao, W., Ringsberg, J.W., Johnson, E., Storhaug, G.: A comparative study of fatigue assessments of container ship structures using various direct calculation approaches. *Ocean Eng.* **82**, 65–74 (2014)
56. Drummen, I., Holtmann, M.: Benchmark study of slamming and whipping. *Ocean Eng.* **86**, 3–10 (2014)
57. Kim, Y., Hermansky, G.: Uncertainties in seakeeping analysis and related loads and response procedures. *Ocean Eng.* **86**, 68–81 (2014)
58. Soares, C.G., Garbatov, Y.: Reliability of maintained ship hull girders subjected to corrosion and fatigue. *Struct. Saf.* **20**(3), 201–219 (1998)
59. Soares, C.G., Garbatov, Y.: Reliability of corrosion protected and maintained ship hulls subjected to corrosion and fatigue. *J. Ship Res.* **43**(2), 65–78 (1999)
60. Deco, A., Frangopol, D.M., Zhu, B.: Reliability and redundancy assessment of ships under different operational conditions. *Eng. Struct.* **42**, 457–471 (2012)
61. Dong, Y., Frangopol, D.M.: Risk-informed life-cycle optimum inspection and maintenance of ship structures considering corrosion and fatigue. *Ocean Eng.* **101**, 161–171 (2015)
62. Frangopol, D.D.M., Bocchini, D.P., Decò, A., Kim, D.S., Kwon, D.K., Okasha, D.N.M., Saydam, D.: Integrated life-cycle framework for maintenance, monitoring, and reliability of naval ship structures. *Nav. Eng. J.* **124**(1), 89–99 (2012)

63. Faber, M.H., Straub, D., Heredia-Zavoni, E., Montes-Iturriaga, R.: Risk assessment for structural design criteria of FPSO systems. Part I: generic models and acceptance criteria. *Mar. Struct.* **28**(1), 120–133 (2012)
64. Heredia-Zavoni, E., Montes-Iturriaga, R., Faber, M.H., Straub, D.: Risk assessment for structural design criteria of FPSO systems. Part II: consequence models and applications to determination of target reliabilities. *Mar. Struct.* **28**(1), 50–66 (2012)
65. Ayyub, B.M., Stambaugh, K.A., McAllister, T.A., de Souza, G.F., Webb, D.: Structural life expectancy of marine vessels: ultimate strength, corrosion, fatigue, fracture, and systems. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B Mech. Eng.* **1**(1), 011001–011001 (2015)
66. Moan, T., Ayala-Uraga, E.: Reliability-based assessment of deteriorating ship structures operating in multiple sea loading climates. *Reliab. Eng. Syst. Saf.* **93**(3), 433–46 (2008)

Uncertainty Quantification's Role in Modeling and Simulation Planning, and Credibility Assessment Through the Predictive Capability Maturity Model

W. J. Rider, W. R. Witkowski, and Vincent A. Mousseau

Abstract

The importance of credible, trustworthy numerical simulations is obvious especially when using the results for making high-consequence decisions. Determining the credibility of such numerical predictions is much more difficult and requires a systematic approach to assessing predictive capability, associated uncertainties and overall confidence in the computational simulation process for the intended use of the model. This process begins with an evaluation of the computational modeling of the identified, important physics of the simulation for its intended use. This is commonly done through a Phenomena Identification Ranking Table (PIRT). Then an assessment of the evidence basis supporting the ability to computationally simulate these physics can be performed using various frameworks such as the Predictive Capability Maturity Model (PCMM). Several critical activities follow in the areas of code and solution verification, validation and uncertainty quantification, which will be described in detail in the following sections. The subject matter is introduced for general applications but specifics are given for the failure prediction project.

The first task that must be completed in the verification & validation procedure is to perform a credibility assessment to fully understand the requirements and limitations of the current computational simulation capability for the specific application intended use. The PIRT and PCMM are tools used at Sandia National Laboratories (SNL) to provide a consistent manner to perform such an assessment. Ideally, all stakeholders should be represented and contribute to perform an accurate credibility assessment. PIRTs and PCMMs are both described in brief detail below and the resulting assessments for an example project are given.

W.J. Rider (✉) • W.R. Witkowski • V.A. Mousseau
Sandia National Laboratories, Albuquerque, NM, USA
e-mail: wjrider@sandia.gov; wrwitko@sandia.gov

Keywords

Application-specific • Foundational • Frameworks • PCMM • PIRT

Contents

1	Introduction to Predictive Capability Maturity Model (PCMM).....	1590
2	Need for Frameworks.....	1593
3	Different Frameworks and Their Vision.....	1596
3.1	The Original PCMM.....	1596
3.2	Variations on a Theme.....	1597
3.3	Foundational PCMM.....	1599
3.4	Application Specific PCMM.....	1600
4	Phenomena Identification Ranking Table (PIRT).....	1601
5	PCMM Example: Qualification of Alternatives to the SPUR Reactor (QASPR)	1603
6	The Role of Frameworks like PCMM and PIRT in Credibility.....	1605
7	Conclusion and Outlook.....	1610
	References.....	1611

1 Introduction to Predictive Capability Maturity Model (PCMM)

The PCMM was developed at SNL for the DOE Advanced Simulation and Computing (ASC) program as a means of assessing completeness of modeling and computational simulation activities for a particular application. There have been 3 generations of PCMM with an emphasis evolving from assessment to evidence inventory [8, 9, 11, 12]. Tables 47.1 and 47.2 show one of the PCMM templates commonly used. There are 6 elements to computational simulation that require investigation with respect to maturity level. Each element has several factors to consider. In the early PCMM versions these were divided into subcategories to be assessed separately but in the newer, current version these are areas to explore, consider and aggregate when determining an overall evaluation of the element. The stakeholders, from analysts to the customers, must determine what the appropriate goals, or required maturity levels, are for each of the elements for the required simulation. The supporting evidence determines the current level of maturity and qualifies the assessment. The difference between the current state and desired maturity level identifies the gaps that need to be addressed. Before the project progresses it must be determined and accepted by all involved whether these gaps are acceptable (and thereby, the required maturity level reduced) or what the mitigation plan is to reduce this gap. Often these gaps cannot be closed due to funding, time or technical constraints. Therefore, it must be determined whether a compromise between the stakeholders is possible and viable.

The PCMM assessment performed by the analysis team for the example is given in Table 47.1 with the specific “grades” highlighted in light green. The programmatic maturity level goal/target was determined to be a level 2 for all

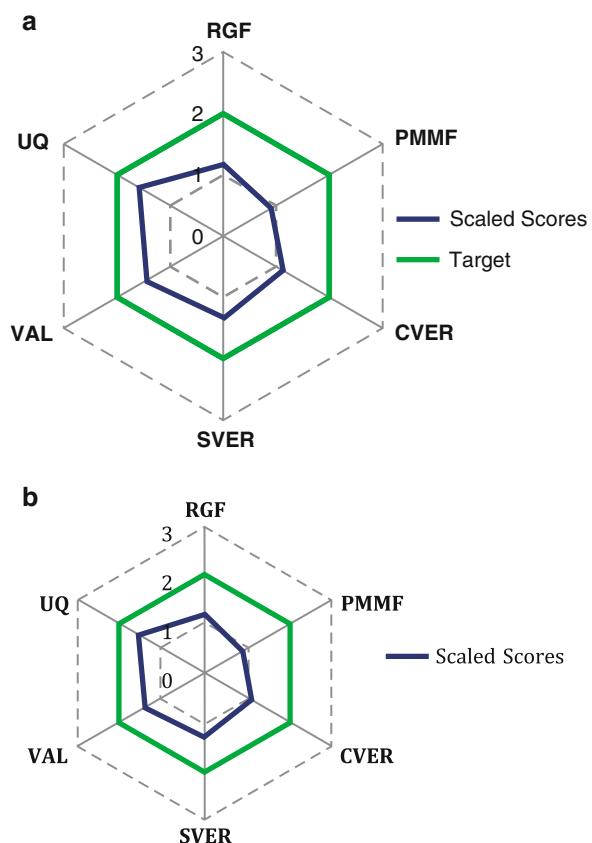
Table 47.1 The commonly seen PCMM Template with current state quality levels in green transparent boxes

MATURITY ELEMENT	Maturity Level 0 Low Consequence, Minimal M&S Impact, e.g. Scoping Studies	Maturity Level 1 Moderate Consequence, Some M&S Impact, e.g. Design Support	Maturity Level 2 High-Consequence, High M&S Impact, e.g. Qualification Support	Maturity Level 3 High-Consequence, Decision-Making Based on M&S, e.g. Qualification or Certification
Representation and Geometric Fidelity <i>What features are neglected because of simplifications or stylizations?</i>	<ul style="list-style-type: none"> Judgment only Little or no representation or geometric fidelity for the system and BCs Geometry or representation of major components is defined 	<ul style="list-style-type: none"> Significant simplification or stylization of the system and BCs Geometry or representation is well defined for major components and some minor components Some peer review conducted 	<ul style="list-style-type: none"> Limited simplification or stylization of major components and BCs Geometry or representation is well defined for major components and some minor components Some peer review conducted 	<ul style="list-style-type: none"> Essentially no simplification or stylization of components in the system and BCs Geometry or representation of all components is at the detail of "as built", e.g., gaps, material interfaces, fasteners Independent peer review conducted
Physics and Material Model Fidelity <i>How fundamental are the physics and material models and what is the level of model calibration?</i>	<ul style="list-style-type: none"> Judgment only Model forms are either unknown or fully empirical Few, if any, physics-informed models No coupling of models 	<ul style="list-style-type: none"> Some models are physics based and are calibrated using data from related systems Minimal or ad hoc coupling of models Some peer review conducted 	<ul style="list-style-type: none"> Physics-based models for all important processes Significant calibration needed using separate effects tests (SETs) and integral effects tests (IETs) One-way coupling of models Some peer review conducted 	<ul style="list-style-type: none"> All models are physics based Minimal need for calibration using SETs and IETs Sound physical basis for extrapolation and coupling of models Full, two-way coupling of models Independent peer review conducted
Code Verification <i>Are algorithm deficiencies, software errors, and poor SOE practices corrupting the simulation results?</i>	<ul style="list-style-type: none"> Judgment only Minimal testing of any software elements Little or no SOE procedures specified or followed 	<ul style="list-style-type: none"> Code is managed by SQE procedures Unit and regression testing conducted Some comparisons made with benchmarks 	<ul style="list-style-type: none"> Some algorithms are tested to determine the observed order of numerical convergence Some features & capabilities (F&C) are tested with benchmark solutions Some peer review conducted 	<ul style="list-style-type: none"> All important algorithms are tested to determine the observed order of numerical convergence All important F&Cs are tested with rigorous benchmark solutions Independent peer review conducted
Solution Verification <i>Are numerical solution errors and human procedural errors corrupting the simulation results?</i>	<ul style="list-style-type: none"> Judgment only Numerical errors have an unknown or large effect on simulation results 	<ul style="list-style-type: none"> Numerical effects on relevant SRQs are qualitatively estimated Input/output (IO) verified only by the analysts 	<ul style="list-style-type: none"> Numerical effects are quantitatively estimated to be small on some SRQs IO independently verified Some peer review conducted 	<ul style="list-style-type: none"> Numerical effects are determined to be small on all important SRQs Important simulations are independently reproduced Independent peer review conducted
Model Validation <i>How carefully is the accuracy of the simulation and experimental results assessed at various tiers in a validation hierarchy?</i>	<ul style="list-style-type: none"> Judgment only Few, if any, comparisons with measurements from similar systems or applications 	<ul style="list-style-type: none"> Quantitative assessment of accuracy of SRQs not directly relevant to the application of interest Large or unknown experimental uncertainties 	<ul style="list-style-type: none"> Quantitative assessment of predictive accuracy for some key SRQs from IETs and SETs Experimental uncertainties are well characterized for most SETs, but poorly known for IETs Some peer review conducted 	<ul style="list-style-type: none"> Quantitative assessment of predictive accuracy for all important SRQs from IETs and SETs at conditions/geometries directly relevant to the application Experimental uncertainties are well characterized for all IETs and SETs Independent peer review conducted
Uncertainty Quantification and Sensitivity Analysis <i>How thoroughly are uncertainties and sensitivities characterized and propagated?</i>	<ul style="list-style-type: none"> Judgment only Only deterministic analyses are conducted Uncertainties and sensitivities are not addressed 	<ul style="list-style-type: none"> Aleatory and epistemic (A&E) uncertainties propagated, but without distinction Informal sensitivity studies conducted Many strong UQ/SA assumptions made 	<ul style="list-style-type: none"> A&E uncertainties segregated, propagated and identified in SRQs Quantitative sensitivity analyses conducted for most parameters Numerical propagation errors are estimated and their effect known Some strong assumptions made Some peer review conducted 	<ul style="list-style-type: none"> A&E uncertainties comprehensively treated and properly interpreted Comprehensive sensitivity analyses conducted for parameters and models Numerical propagation errors are demonstrated to be small No significant UQ/SA assumptions made Independent peer review conducted

Table 47.2 PCMM classification guidance [9]

MATURITY ELEMENT	Maturity Level 0 Low Consequence, Minimal M&S Impact, e.g. Scoping Studies	Maturity Level 1 Moderate Consequence, Some M&S Impact, e.g. Design Support	Maturity Level 2 High-Consequence, High M&S Impact, e.g. Qualification Support	Maturity Level 3 High-Consequence, Decision-Making Based on M&S, e.g. Qualification or Certification
Representation and Geometric Fidelity <i>What features are neglected because of simplifications or stylizations?</i>	<ul style="list-style-type: none"> Judgment only Little or no representation or geometric fidelity for the system and BCs 	<ul style="list-style-type: none"> Significant simplification or stylization of the system and BCs Geometry or representation of major components is defined 	<ul style="list-style-type: none"> Limited simplification or stylization of major components and BCs Geometry or representation is well defined for major components and some minor components Some peer review conducted 	<ul style="list-style-type: none"> Essentially no simplification or stylization of components in the system and BCs Geometry or representation of all components is at the detail of "as built", e.g., gaps, material interfaces, fasteners Independent peer review conducted
Physics and Material Model Fidelity <i>How fundamental are the physics and material models and what is the level of model calibration?</i>	<ul style="list-style-type: none"> Judgment only Model forms are either unknown or fully empirical Few, if any, physics-informed models No coupling of models 	<ul style="list-style-type: none"> Some models are physics based and are calibrated using data from related systems Minimal or ad hoc coupling of models Some peer review conducted 	<ul style="list-style-type: none"> Physics-based models for all important processes Significant calibration needed using separate effects tests (SETs) and integral effects tests (IETs) One-way coupling of models Some peer review conducted 	<ul style="list-style-type: none"> All models are physics based Minimal need for calibration using SETs and IETs Sound physical basis for extrapolation and coupling of models Full, two-way coupling of models Independent peer review conducted
Code Verification <i>Are algorithm deficiencies, software errors, and poor SOE practices corrupting the simulation results?</i>	<ul style="list-style-type: none"> Judgment only Minimal testing of any software elements Little or no SOE procedures specified or followed 	<ul style="list-style-type: none"> Code is managed by SOE procedures Unit and regression testing conducted Some comparisons made with benchmarks 	<ul style="list-style-type: none"> Some algorithms are tested to determine the observed order of numerical convergence Some features & capabilities (F&C) are tested with benchmark solutions Some peer review conducted 	<ul style="list-style-type: none"> All important algorithms are tested to determine the observed order of numerical convergence All important F&Cs are tested with rigorous benchmark solutions Independent peer review conducted
Solution Verification <i>Are numerical solution errors and human procedural errors corrupting the simulation results?</i>	<ul style="list-style-type: none"> Judgment only Numerical errors have an unknown or large effect on simulation results 	<ul style="list-style-type: none"> Numerical effects on relevant SRQs are qualitatively estimated Input/output (IO) verified only by the analysts 	<ul style="list-style-type: none"> Numerical effects are quantitatively estimated to be small on some SRQs IO independently verified Some peer review conducted 	<ul style="list-style-type: none"> Numerical effects are determined to be small on all important SRQs Important simulations are independently reproduced Independent peer review conducted
Model Validation <i>How carefully is the accuracy of the simulation and experimental results assessed at various tiers in a validation hierarchy?</i>	<ul style="list-style-type: none"> Judgment only Few, if any, comparisons with measurements from similar systems or applications 	<ul style="list-style-type: none"> Quantitative assessment of accuracy of SRQs not directly relevant to the application of interest Large or unknown experimental uncertainties 	<ul style="list-style-type: none"> Quantitative assessment of predictive accuracy for some key SRQs from IETs and SETs Experimental uncertainties are well characterized for most SETs, but poorly known for IETs Some peer review conducted 	<ul style="list-style-type: none"> Quantitative assessment of predictive accuracy for all important SRQs from IETs and SETs at conditions/geometries directly relevant to the application Experimental uncertainties are well characterized for all IETs and SETs Independent peer review conducted
Uncertainty Quantification and Sensitivity Analysis <i>How thoroughly are uncertainties and sensitivities characterized and propagated?</i>	<ul style="list-style-type: none"> Judgment only Only deterministic analyses are conducted Uncertainties and sensitivities are not addressed 	<ul style="list-style-type: none"> Aleatory and epistemic (A&E) uncertainties propagated, but without distinction Informal sensitivity studies conducted Many strong UQ/SA assumptions made 	<ul style="list-style-type: none"> A&E uncertainties segregated, propagated and identified in SRQs Quantitative sensitivity analyses conducted for most parameters Numerical propagation errors are estimated and their effect known Some strong assumptions made Some peer review conducted 	<ul style="list-style-type: none"> A&E uncertainties comprehensively treated and properly interpreted Comprehensive sensitivity analyses conducted for parameters and models Numerical propagation errors are demonstrated to be small No significant UQ/SA assumptions made Independent peer review conducted

Fig. 47.1 Kiviat plots for PCMM assessment score display



elements at the beginning of the project. A Kiviat diagram can be used to provide a visual metric of the state of each of the elements as compared to the specified target level. This simple chart quickly shows which element of providing a credible prediction is lacking, and which are meeting the target. For the puncture example, the associated Kiviat diagram is shown in Fig. 47.1 which reveals the state of the PCMM graphically.

The PCMM was defined to allow the assessment of the state of modeling and simulation (M&S) in a manner that balances all aspects of the M&S workflow. PCMM provides a structured breakdown of the component work within an engineering M&S study. Classically it contains six elements, each requiring specific focus when the PCMM is applied. These six elements are geometry/representation fidelity, model fidelity, code verification, solution verification (numerical error estimation), validation and uncertainty quantification/sensitivity analysis. Each of these elements entails significant complexity and contributes to the overall quality.

Part of the original role of PCMM was to organize and structure the achievement of high credibility M&S for the purposes of decision-making. As PCMM was developed at an engineering laboratory, the focus was necessarily centered upon

engineering analysis and thus biased. Over time these biases have become evident upon the application of PCMM to other M&S endeavors. In the process the framework has been modified, extended and its biases lay bare. Each extension has helped the model itself mature and yielded an enhanced understanding regarding its application and form. Among the most important of these extensions is the expansion of the process surrounding PCMM to include pre- and post-conditions and targets and an overall iterative framework for applying the assessment. We have come to view PCMM more as a communication and planning tool than a vehicle for assessment. It is in this broader sense that the PCMM may find its greatest utility in use, the definition of high-credibility M&S for decision making from inception to completion with greater quality. A third way to constructively engage with PCMM is through viewing it as defining the elements of the modeling and simulation workflow. In any workflow the PCMM encapsulates the different issues and topics that must be confronted. Recent work on PCMM has provided a recommended workflow together with the assessment elements.

2 Need for Frameworks

One of the big problems that the entire V&V enterprise has is its sense of imposition on others. Every simulation worth discussing does “V&V” at some level, and almost without exception they have weaknesses. Doing V&V “right” or “well” is not easy or simple. Usually, the proper conduct of V&V will expose numerous problems with a code, model, and/or simulation. It’s kind of like exposing yourself to an annual physical; it’s good for you because you learned something about yourself you didn’t know, but you might have to face some unpleasant realities.

In addition, the activity of V&V is quite broad and something almost always slips between the cracks (or chasms in many cases). To deal with this breadth, the V&V community has developed some frameworks to hold all the details together. Sometimes these frameworks are approached as prescriptions for all the things you must do. Instead we suggest that these frameworks should not be recipes, nor should they be thought of as prescriptions, they are things to be seriously considered. If listed aspects of PCMM are disregarded, it should be justified through rigorous analysis. They are “thou should,” not “thou shalt,” or even “you might.”

Several frameworks exist today and none of them is fit for all purposes, but all of them are instructive on the full range of activities that should be at least considered, if not engaged in.

CSAU – Code Scaling Assessment and Uncertainty [1, 2] developed by the Nuclear Regulatory Committee to manage the quality of analyses done for power plant accidents. It is principally applied to thermal-fluid (i.e. thermal-hydraulic) phenomena that could potentially threaten the ability of nuclear fuel to contain radioactive products. This process led the way and has been updated recently [13]. It includes processes and perspectives that have not been fully replicated in subsequent work. PCMM is attempting to utilize these lessons in improving its completeness.

PCMM – Predictive Capability Maturity Model developed at Sandia National Laboratories for the stockpile stewardship program in the last 10 years. As such it reflects the goals and objectives of this program and Sandia's particular mission space. It was inspired by the CMMI developed by Carnegie Mellon University to measure software process maturity. PCMM was Sandia's response to calls for greater attention to detail in defining the computational input into quantitative margins and uncertainty (QMU), [12] the process for nuclear weapons' certification completed annually.

CAS – Credibility Assessment Scale [3] developed by NASA. They created a similar framework to PCMM for simulation quality in the wake of the shuttle accidents and specifically after Columbia where simulation quality played an unfortunate role. In the process that unfolded with that accident, the practices and approach to modeling and simulation was found to be unsatisfactory. The NASA approach has been adopted by the agency, but does not seem to be enforced. This is a clear problem and potentially important lesson. There is a difference between an enforced standard (i.e., CSAU) and one that comes across as well intentioned, but powerless directives. Analysis should be done with substantial rigor when lives are on the line. Ironically, formally demanding this rigor may not be the most productive way to achieve this end.

PMI – Predictive Maturity Index developed at Los Alamos [5, 19] This framework is substantially more focused upon validation and uncertainty, and it is a bit lax with respect to the code's software and numerical issues. These aspects are necessary to focus upon given advances in the past 25 years since CSAU came into use in the nuclear industry.

MURM – Model Utilization Risk Management was developed by the Applied Physics Laboratory [10] to provide an explicit scoring system for risks associated with using M&S in decision-making. The MURM's intent is to provide a scale for considering the risk-based maturity of the capability and the risk-based assessment of the decision. This should provide the decision-maker with the information necessary to make an informed choice in utilizing M&S in a process.

Computational simulations are increasingly being used in our modern society to replace some degree of expensive or dangerous experiments and tests or where tests can't be conducted. The reliance on computational fluid and solid mechanics are ever more commonplace in modern engineering practice. The challenge of climate change may be another avenue where simulation quality is scrutinized and could benefit from a structured, disciplined approach to quality. Ultimately, these frameworks serve the role of providing greater confidence (faith) in the simulation results and their place in decision-making. Climate modeling is a place where simulation and modeling plays a large role, and the decisions being made are huge.

The question lingers in the mind, "what can these frameworks do for me?" Our answer follows:

1. V&V and UQ are both deep fields with numerous deep subfields. Keeping all of this straight is a massive undertaking beyond the capacity of most professional scientists or engineers. A framework provides a blueprint for

conducting detailed studies and making certain that no aspect of the technical work remains untouched. Such assessments are prone to missing important details associated with topics that may not be within the expertise of those doing the work.

2. Human behavior shows that everyone will default to focusing on where they are strong and comfortable, or interested. For some people it is mesh generation, for others it is modeling, and for yet others it is analysis of results. Such deep focus may not lead (or is not likely to lead) to the right sort of quality. Where quality is needed is dependent upon the problem itself and how the problem's solution is used. The frameworks provide a holistic view of quality where deep focus does not necessarily lead to myopic behavior. It also encourages a more collaborative aspect for the conduct of V&V by highlighting where expertise needs to be sought beyond those conducting the work.
3. These are useful outlines for all of the activities that a modeling and simulation project might consider. Project planning can use the frameworks to develop help set objectives and subtasks and prioritize and schedule work. The frameworks also provide a rubric for the peer review of such efforts and allow for the proper determination of the breadth of capabilities that might be needed for peer review.
4. These are menus of all the sort of things you might do, not all the things you must do. Combining the outcomes from the PIRT and the intended use of the simulation and modeling provides a means of sorting through the menu. It also allows the determination of the activities not conducted to be explicit rather than implicit in the assessment.
5. They provide a sequenced set of activities, prepared in a sequenced rational manner with an eye toward what the modeling and simulation is used for (intended use). Again, the framework provides suggestions and not a straight-jacket. Different sequencing can be executed if reasoned analysis calls for it. The frameworks give a structured manner where the work can be extended or improved in a rational manner if resources become available.
6. They help keep activities in balance. A generally holistic quality mentality is encouraged and deviations from full coverage are explicitly available. The framework if properly utilized will help keep you honest.
7. You will understand what is fit for purpose, when you have put too much effort into a single aspect of quality. Both the impact of neglected aspects of the modeling and simulation effort and steps to remove such neglect can be easily accessed via the framework.
8. V&V and UQ are developing quickly and the frameworks provide a "cheat sheet" for all of the different aspects. It assures that the analysis, assessment and examination remains current. Generally the list of potential activities can never be exhausted given fixed resources most projects operate under. The frameworks provide a means of looking towards and measuring continuous improvement.
9. A framework's flexibility is key and not every application necessarily should focus on every quality aspect, or apply every quality approach in equal

- measure. Again the whole aspect of balancing a holistic view of quality with an appropriate narrowing effort put toward more critical aspects of a particular modeling and simulation effort is necessary.
10. Validation itself is incredibly hard in both breadth and depth. It should be engaged in a structured, thoughtful manner with a strong focus on the end application. Validation is easy to do poorly; the frameworks provide steps where the underlying work for high quality validation is readily available.
 11. The computational science community largely ignores verification of code and calculations. Even when it is done, it is usually done poorly, or insufficiently. The frameworks provide a focus on this aspect of V&V so that it is not entirely ignored by any effort utilizing them.
 12. Error estimation and uncertainty too rarely include the impact of numerical error, and estimate uncertainty primarily through parametric changes in models. These efforts are highlighted in frameworks and lead to higher quality assessments and provide the means for improving many efforts.
 13. Another element of the frameworks is the accounting for numerical errors which is usually much larger than acknowledged. Lots of parametric and model calibration is actually accounting for the numerical error, or providing numerical stability rather than physical modeling.
 14. A framework helps identify gaps and associated risks for each modeling and simulation effort. These gaps can be targeted as additional work to improve quality.
 15. Lastly, a framework can help you incorporate project resource constraints and identify associated risks and consequences in forgoing various V&V activities. They also provide an explicit focus on both what is done, but also what is not done. None of this is an alternative to a hard hitting and technically focused peer review. Bringing in independent experts and fresh perspectives is utterly invaluable to the honesty and integrity of quality assessment.

3 Different Frameworks and Their Vision

3.1 The Original PCMM

As originally constructed (and reported in [9]) the PCMM addressed six elements that were identified to be essential for successful application of modeling and simulation. These elements were:

- Representation and geometric fidelity
- Physics and material model fidelity
- Code verification
- Solution verification
- Model validation
- Uncertainty quantification (UQ) and sensitivity analysis (SA)

In the PCMM process a general set of attributes were identified for each of these elements to permit characterization of each element into one of four maturity levels (0-3) This resulted in defining a matrix of maturity levels verses PCMM elements as illustrated in Table 47.2.

As can be seen from the information contained in Table 47.2, the PCMM maturity levels constitute a hierarchy that represents increasingly greater levels of sophistication and computational fidelity (and expense and resources). The levels contained within this hierarchy can be summarized as follows.

- Level 0: At this level there is little or no assessment of completeness and accuracy and the capabilities for the element being assessed are highly reliant on personal experience and judgment.
- Level 1: For this level an informal assessment of completeness and accuracy has been performed using internal peer review groups.
- Level 2: This level applies a formal process to assess completeness and accuracy of the element being evaluated. At this level use is made of external peer reviews for at least some of these assessments.
- Level 3: Finally, at this level a formal assessment of the element has been completed with the assessments predominantly being conducted by external peer reviews.

We note that the assessment of the levels of maturity does not, by itself, indicate the degree to which the M&S capability will be successful at meeting the requirements identified to address a particular application (e.g. a licensing application or a regulatory requirement, or a design specification). To identify the degree to which such a capability will be present for a particular application, one would compare the assessed maturity level for the PCMM elements with an objective level of maturity that is identified as necessary for the application of interest. Any application will provide specific quantities of interest associated with defining a successful outcome for the intended purpose.

3.2 Variations on a Theme

PCMM is a framework to organize that entire V&V and UQ landscape into neat little boxes. Of course reality is never so neat and tidy, but structure for such a complex and potentially unbounded activity is good. PCMM has gone through numerous revisions, extensions and rearticulations, and the safest conclusion is that the framework will never be complete. While for any given use of PCMM a different and focused version may have utility, any effort to achieve this may be futile. Such a framework can produce better results, but will never be a “silver bullet”.

A close examination of PCMM provides an insight to its intrinsic bias toward a certain class of engineering calculations. This is clearest in the emphasis on geometric fidelity and solution verification, which belie its basis in meshbased

calculations. Nonetheless, peeling back a layer, one should not lose sight of the nature of these entries in the framework. The geometry is really a view of the representation of reality in the overall computational simulation, while the solution verification is the estimate of error in the same. Both aspects are essential to determining the quality and assigning confidence to the simulation results. Other aspects of the PCMM translate across fields more readily. Code verification and software quality are necessary elements in providing reliable computer codes for simulation. Modeling and its credibility are universal in the field. Finally validation provides the tangible and measured connection to reality, and uncertainty quantification is key to decision-making.

Other elements have been added due to their importance and focus when applying the framework. Recent work has taken the validation element and broken it up into four separate activities to be examined. Two elements now involve the quality and acquisition of experimental data; followed by the assessment of the simulation with respect to the data. The validation exercise is divided into two sections, one with data and validation applied to the underlying and low-level models in the code, and the second applying to the application-level or high-level modelling and associated data for comparison. In addition, we considered adding additional elements to PCMM associated with the requirements arising from the simulation customers, and the impact of the code user/analyst/engineer on the results obtained. This area is sensitive and controversial because the human impact of M&S analysis is a “hot-button” issue with many and is tabled for now.

A useful concept in examining complex, difficult problems is the characterization of the problem as “wicked”. A wicked problem has a number of characteristics that make it special and difficult to solve. For example, the problem cannot be fully understood before attempting to solve it. As the problem is tackled, new aspects of the problem are unveiled, and the solution to the problem needs to be rescoped. This is a continual aspect of the problem, and it bedevils those who attempt to apply project plans and complete predictability to the problem. There are even super-wicked problems that are more difficult due to counter-intuitive feedbacks between the problem itself and the solution. Characteristics of super-wicked problems are that those solving the problem are also causing the problem, and future results are irrationally discounted presently among others. V&V with PCMM as an example can probably be characterized as being wicked and perhaps even super-wicked. Thus the activity defies full articulation.

Despite this intrinsic futility we would claim that PCMM is useful. In a real sense the activities within PCMM could be thought of as a menu of activities that one might consider in determining and driving simulation quality. This is as important as the measurement of the quality itself. Being such a complex and potentially unbounded activity means that there is large probability for details to slip from attention. PCMM provides the list of activities for simulation quality improvement and assessment, which can be utilized effectively by code development and application projects to draw upon. A reasonable recommendation is to examine the PCMM for “low hanging fruit” that can easily be incorporated into the simulation process. As we will now describe this process can be further refined by dividing

the PCMM into two pieces; one that applies primarily to code development and the foundation of simulation capability, and a second that is more specific to the application of interest. Another practicality is the potentially overwhelming nature of PCMM. It is therefore rational and reasonable to apply less than the complete framework initially.

Thus a useful observation is that one can decompose the whole computational confidence issue into two relatively neat and tidy pieces: that which is more tool (code)-specific, (or foundational) and that which is problem or application-specific. Thus some work provides the general characterization of the computational tools to be used for analysis, while other work is directly applied to the application of that tool. For example, the general software development approach and documentation can be used over and over for different applications, while the specific options from a code used for a given application are narrow and specific to that application. This principle can be applied repeatedly across the span of activities that comprise the quality pedigree for a given code and its intended application. This principle applies for nearly every area of code quality investigation as laid out below.

3.3 Foundational PCMM

Software quality is one of the obvious lynchpins of the foundation for a code. The practices applied in the development of the code provide the degree of confidence in the code's correctness and stability. High quality code practices provide important tractability to the overall pedigree of the simulation. These practices will apply to every application of the code. We acknowledge that a subset of the software activities will be more directly relevant to a given activity and may be subject to different requirements.

Code verification is another simulation code quality practice that applies across the entire spectrum of potential applications. Code verification in a nutshell provides the evidence and confidence that the solution algorithm in the code is implemented correctly, and the given mathematical description is actually being solved. This is a distinctive to numerical simulation and applies in a complementary manner to the overall software quality approach. Again, some of the code verification will be more specifically applicable to a certain problem attacked by the code.

The decomposition of validation into two sets allows part of the validation activity to be considered foundational. Many models are common to a large number of simulations and comprise the core of the modeling capability of the code. These models must be validated so that their fidelity can be fully assessed away from the intended application. Furthermore, these models can be compared to special purpose experiments that have relatively small errors compared to many application settings. This separation allows some of the modeling capability to be assessed in a manner that should greatly increase confidence in the code. When these errors are convoluted with the integral-large scale data from the applications, the source of discrepancy can become hidden.

Accompanying the basic low-level validation of the code should be assessment of the concomitant uncertainty and sensitivity of the models associated with the code's simulation foundation. This should include the impact of model form and parameters as well as numerical integration effects (i.e., some solution verification). Again, this activity is undertaken to provide a baseline uncertainty and sensitivity away from the convoluted situation offered by the full application setting.

The capability of the code to model circumstances is provided by the user interface. This aspect of the assessment is relatively small in scope, but quite important. In many respects the flexibility offered by the user interface bounds what a code user can achieve. The importance of this activity has increased dramatically in recent years as user interfaces have become codes unto themselves (i.e., the input is itself executable where for example python, or other advanced scripting languages are used).

The foundational aspect of the customer requirements is applied to whomever is providing the resources for the development of the code. This customer can be distinct from the application use of the code, but not necessarily. Nevertheless, the customer has imposed requirements on the code development, and the assessment should provide a check to whether these have been complied with.

3.4 Application Specific PCMM

For any given application there is a computational representation of the problem being solved. This can take many forms including detailed meshes that can be compared with a CAD description. In other cases the representation is simplified as a lumped parameter model and the geometric fidelity is intentionally suppressed.

In every case there is a model of reality that is contained in the computational simulation. This model may be a set of continuously differential equations, integral conservations laws, or algebraic relations. In each case there should be an assessment of the model's capacity to simulate the desired situation in the application. For example one might have a code that contains the incompressible Navier-Stokes equations, and the degree to which incompressibility applies should be examined. Two-phase flow is replete with complexity where for example one should see whether the description in the code is appropriate for the situation (is slip between the phases important, and do the equations appropriately describe the phenomena?).

For software quality and code verification the application specific assessment is bounded. The appropriate question is how much of the foundational aspect of the code's quality is specific to the application. Given the features of the code that the application depends upon are they tested in the software quality or code verification suites? How deep is this coverage and does it provide confidence in the code pedigree in a manner specific to a given application?

Solution verification is quite often overlooked in the practical use of M&S in engineering there. There is no excuse for this. The degree that the model representation and detail impacts solutions must be assessed as part of the overall

uncertainty estimation. Too often the numerical error is simply calibrated for, or muddled with other modeling errors. The key to this step is the clear separation and articulation of this aspect of error and uncertainty apart from other sources. In many cases the solution verification may involve a simple bounding estimate that gives the idea of the magnitude of the effect on the results.

The integral validation aspect of PCMM comes naturally to the nuclear engineer. The unnatural part of the exercise is properly casting the process with all the other elements of PCMM. PCMM is in essence the deconvolution of many effects that often comprise the validation exercise. The foundational aspects of validation, uncertainty and solution verification attempt to peel away this complexity leaving the core of error to be examined. This is the task with integral validation to first understand how well the uncalibrated code models the circumstance, and then calibrate the solution without undoing any of the foundational work. Thus the calibration can be fully exposed to scrutiny and hopefully underlie the capacity to more directly attack the basis of lack of credibility.

Application uncertainty and sensitivity analysis is then quite clearly defined. Again the separation between foundational model validation with requisite uncertainty gives a basis for attacking the specific application uncertainty with clarity. Both aspects of uncertainty impact results and are included in the assessment, but the goal of clearly identifying the application model-specific uncertainties is obtainable through following the structure outlined here.

User qualification has a large impact on the results, yet is rarely assessed. An ideal case would be to have independent models of the same application. This is often not possible despite numerous studies showing that the user effect is large (larger than model differences in cases).

Finally the entity paying for the application work has requirements. These requirements should be assessed for how well the simulation has met these.

4 Phenomena Identification Ranking Table (PIRT)

A PIRT should list all of the important physics phenomena occurring in the physical event being simulated at the specified level of interest [18]. For each individual phenomenon an assessment/ranking must be declared for the “Importance” of this particular phenomenon in the application simulation or, in other words, what is the resulting consequence if the phenomena model is wrong. The rankings are specified as high (“H”), medium (“M”) or low (“L”). Then “Adequacy” determinations must be made with respect to how well the phenomena is represented with a mathematical model, how well it is implemented within the simulation code of choice (Sierra/SM for this project/example) [15], and the level of validation that has been performed for the application space of interest. Once again a ranking is determined for each of these three “Adequacy” areas.

For quick visual adequacy analysis, a color-coding scheme is used to identify areas of inadequacy or gaps. Green signifies that the adequacy is acceptable; yellow indicates that the adequacy is marginal and red indicates that the adequacy level

Table 47.3 PIRT for simulation of interest

Phenomena	Consensus	Adequacy		
	Importance	Math model	Implementation in code	Validation
Phenomena #1	H	H	M	L
Phenomena #2	M	H	L	L
Phenomena #3	M	H	M	L

Table 47.4 Final PIRT for failure prediction example problem

Phenomena	Consensus	Adequacy		
	Importance	Math model	Implementation in code	Validation
Large elastic-plastic deformation of metals	H	H	M	M
Ductile material failure	H	M	M	L
Contact	H	H	M	M
Friction between punch and test item	M	M	M	L
Enforcement of boundary conditions	L	H	H	L
Inertial loads	H	H	H	M

is unacceptable and needs to be addressed. The colors are assigned with respect to the “Importance” ranking. If a phenomena’s “Importance” ranking is low, then any level of adequacy is deemed acceptable. If the “Importance” ranking is medium, then acceptable adequacy rankings are medium and high. An adequacy ranking of low produces a yellow square that indicates a marginal adequacy level. When the “Importance” ranking is high, then the only acceptable adequacy level is “high” earning a green square. For a medium adequacy determination a yellow square is used, however, for a low adequacy assessment a red square is used to indicate an unacceptable adequacy level for that particular category. A sample PIRT with 3 identified physics phenomena is given in Table 47.3 showing various adequacy levels with colored gap indicators.

For the presented project, an assessment of the numerical prediction of important phenomenological aspects of an abnormal fracture problem was evaluated at the start of the project, and presented in Table 47.4. Six different physics phenomena were identified (large elastic-plastic deformation of metals, ductile material failure, contact, friction between punch and test item, enforcement of boundary conditions, inertial loads). All of them were rated of “high” importance except for “friction between punch and test item” which was considered of “medium” importance. Two areas of concern are quickly identified by their “red” coloration that required attention: “Ductile Material Failure” and “Enforcement of Boundary Conditions” with respect to validation adequacy. Both of these were of high importance but ranked low on the validation adequacy scale. Since the intent of the project was to better understand failure mechanics modeling and its limitations, validate its applicability for the application being studied and characterize its reliability

and usability, the fact that “Ductile material failure” was ranked inadequate was worrisome and addressed in the scope of the project.

The boundary condition assumption gap was also identified and investigated. The model enforces a perfect non-slip boundary condition between the plate and the anchoring fixture. In reality it is possible that the test article could slip in the clamping fixture. Everything was done experimentally to minimize the slippage during impact and penetration. Several experiments were performed to assess the level of non-compliance of the non-slip boundary condition and physically interrogated this situation. The experimentalists deemed that the plate was not slipping in the clamping fixture. Therefore, the theoretical boundary condition enforcement was deemed acceptable and the “Importance” ranking of “Enforcement of Boundary Conditions” was downgraded to “L” in the final PIRT assessment (shown in black print in Table 47.4) that elevated the “validation adequacy” coloration to green.

5 PCMM Example: Qualification of Alternatives to the SPUR Reactor (QASPR)

Here we give an example of what an actual capability assessment looks like in a bit of detail. In this case we will be showing the outcomes and process outline for the QASPR project [16] and associated codes, Xyce [4] and Charon [7, 17] (Fig. 47.2). The assessment was conducted at Sandia National Laboratories, Albuquerque by Laboratory staff. QASPR was begun an admittedly ambitious effort to replace the experimental testing of integrated circuits and associated semi-conductor material in a high radiation environment. Sandia sought to replace the expensive and risky SPUR reactor with an extensively validated and verified computational capability. The specific assessment was for the maturity of the III-V Npn model predictability within a defined threat environment for the Xyce and Charon codes. In particular, the device model for the MESA developed using Npn InGap/GaAs.

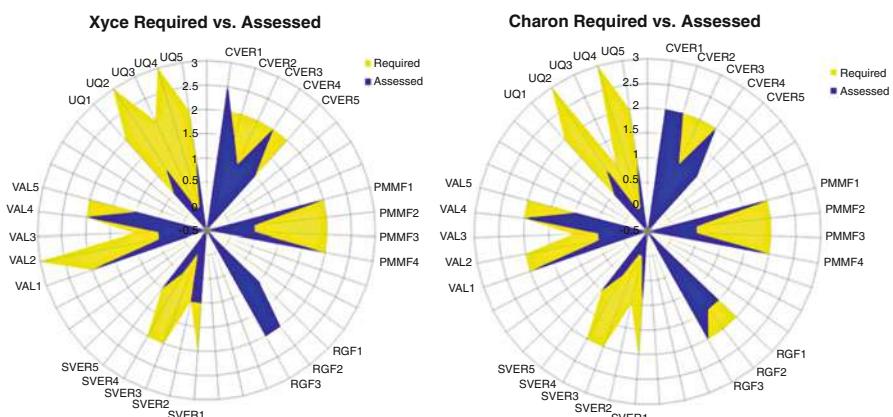


Fig. 47.2 QASPR: PCMM Kiviat Plots showing the assessed status of Xyce and Charon

The first thing to do is pull together a team to conduct the assessment starting with a lead stakeholder. In this case the stakeholder is acting as the customer for the work done by the code. In other cases this may or should be another person altogether. The assessment can take input from any member of the team although each member should not necessarily contribute to each part of the PCMM. Secondly we have a (trained) PCMM assessor who also acts as a moderator for the activity. Next we have a PCMM subject matter expert (SME) whose expertise includes key aspects of V&V. The team is then rounded out by SME's in each major role for the development and use of the code(s). In this case we have SMEs for UQ, V&V analysis, experiments, applied circuit analysis, calibration and code development for both Xyce and Charon (different people).

The first step done was to meet with the entire team to brief them on the process and gain buy-in. In addition the team receives "training" on the PCMM and the assessment process. In the case of QASPR the project had already conducted an extensive PIRT. Rather than repeat this work, the first step was to review and vet the existing PIRT for the assessment that follows. The team then held a set of meetings to conduct the assessment. It is important that the team be well-represented throughout. After a set of meetings the team prepared some focused feedback on the state of the QASPR project's codes as well as the PCMM process itself.

Part of this effort used an Excel spreadsheet as the vehicle for the assessment, and much of the feedback keyed on this. There was feeling that clarification of the language could be done particularly within the UQ section. Some elements seemed to overlap one another. Overall the Excel Tool was judged to be very beneficial and helps capture a great deal of detail, but can be overwhelming at times during the assessment. We might consider creating a reduced version for real time assessment. Overall it is a great organization tool for the assessor. This sort of feedback should always be included to continually improve the process.

A real positive is that the PCMM assessment generated discussion across different teams consisting of analysts, developers, and experimentalists. As an example of the discussion: For qualification, solution verification will be critical (e.g., input/output file verification) and will need to be better formalized – analysts are now aware of this and QASPR will need to better formalize this workflow. We need another assessment iteration to determine the value of some members on the larger team for the purposes of assessment. Experimentalists seemed to be the odd man out, but may be due to QASPR development of Physical Simulation PCMM (the PCMM ideas applied to physical testing or experimentation). The assessment is also captured in some graphical output, and overall spreadsheet content.

At the end of the assessment we crafted a path forward for further assessments.

1. First, review PCMM elements with PCMM SME prior to assessment to insure all the descriptors are well understood. This was done in real time during the assessment which caused delays.
2. The Excel Tool will be used for future assessments.
3. Evidence may be moved and maintained on a SharePoint site.

4. We will do another iteration of the assessment with the larger group.
5. We will continue to schedule review to minimize the time required by participants based on roles.

The next assessment will focus on PnP devices and potentially circuits and an update of the Npn capability should also be done.

6 The Role of Frameworks like PCMM and PIRT in Credibility

“Integrity is telling myself the truth. And honesty is telling the truth to other people.”

– Spencer Johnson

As tools the frameworks of PIRT and PCMM are only as good as how they are used to support high quality work. Using these tools will not necessarily improve your credibility, but rather help you assess it holistically. Ultimately the credibility of your modeling & simulation capability is driven by the honesty and integrity of your assessment. It is easy to discuss what you do well and where you have mastery over a topic. The real key to assessment includes a will and willingness to articulate where an effort is weak and where the fundamental foundational knowledge in a field is the limiting factor in your capacity for solving problems. The goal in credibility assessment is not demonstration of mastery over a topic, but rather a demonstration of the actual state of affairs so that decisions can be made with full knowledge of the weight to place and risks inherent in modeling & simulation (M&S).

A topic like quality is highly subjective and certainly subject to a great deal of relativity where considerations of the problem being solved and the M&S capabilities of the field or fields relevant come into consideration. The frameworks serve to provide a basic rubric and commonly consistent foundation for the consideration of M&S quality. As such, PIRT and PCMM are blueprints for numerical modeling. The craft of executing the M&S within the confines of resources available is the work of the scientists and engineers. These resources include the ability to muster effort towards completing work, but also the knowledge and capability base that you can draw upon. The goal of high quality is to provide an honest and holistic approach to guiding the assessment of M&S quality.

In the assessment of quality the most important task to accomplish is an honest discussion of the technical elements of the work with complete disclosure of faults and shortcomings. There are significant physiological and social factors that lead to a lack of honesty in evaluation and assessment. No framework or system can completely overcome such tendencies, but might act as a hedge against the tendency to overlook critical details that do not reflect well on the assessment. The framework assures that each important category is addressed. The ultimate test for the overall integrity and honesty of an assessment of M&S credibility depends upon deeper technical knowledge than any framework can capture.

Quite often an assessment will avoid dealing with systematic problems for a given capability that have not been solved sufficiently. Several examples of

this phenomenon are useful in demonstrating where this can manifest itself. In fluid dynamics, turbulence remains a largely unsolved problem. Turbulence has intrinsic and irreducible uncertainty associated with it, and no single model or modeling approach is adequate to elucidate the important details. In Lagrangian solid mechanics the technique of element death (where poorly deformed elements are removed to allow the simulation to continue) is pervasively utilized for highly strained flows where fracture and failure occur. It is essential for many simulations and often renders a simulation to be non-convergent under mesh refinement. In both cases the communities dependent upon utilizing M&S with these characteristics tend to under-emphasize the systematic issues associated with both. This produces a systematically higher confidence and credibility than is technically justifiable. The general principle is to be intrinsically wary of unsolved problems in any given technical discipline.

Together the PIRT and PCMM adapted and applied to any M&S activity form part of the delivery of defined credibility of the effort. The PIRT gives context to the modeling efforts and the level of importance and knowledge of each part of the work. It is a structured manner for the experts in a given field to weigh in on the basis for model construction. The actual activities should be strongly reflected in the sort of assessed importance and knowledge basis reflected in the PIRT. Similarly the PCMM can be used for a structured assessment of the specific aspects of the modeling and simulation.

The degree of foundational work providing the basis for confidence for the work is spelled out in the PCMM categories. Included among these are the major areas of emphasis some of which may be drawn from outside the specific effort. Code verification being an exemplar of this where its presence and quality provides a distinct starting point for the specific aspects of the estimation of the numerical error for the specific M&S activity being assessed. Each of the assessed categories forms the starting point for the specific credibility assessment.

One concrete way to facilitate the delivery of results is the consideration of the uncertainty budget for a given modeling activity. Here the delivery of results using PIRT and PCMM is enabled by considering them to be resource guides for the concrete assessment of an analysis and its credibility. This credibility is quantitatively defined by the uncertainty and the intended application's capability to absorb such uncertainties for the sort of questions to be answered or decision to be made. If the application is relatively immune to uncertainty or only needing a qualitative assessment then large uncertainties are not worrisome. If on the other hand an application is operating under tight constraints associated with other considerations (sometimes called a design margin) then the uncertainties need to be carefully considered in making any decisions based on modeling and simulation.

This gets to the topic of how modeling & simulation are being used. Traditionally M&S goes through two distinct phases of use. The first phase is dominated by "what if" modeling efforts where the results are largely qualitative and exploratory in nature. The impact of decisions or options is considered on a qualitative basis and guides decisions in a largely subjective way. Here the standards of quality tend to focus on completeness and high-level issues. As modeling and simulation proves its

worth for these sorts of studies, it begins to have greater quantitative demands placed on it. This forms a transition to a more demanding case for M&S where design or analysis decision is made. In this case the standards for uncertainty become far more taxing. This is the place where these frameworks become vital tools in organizing and managing the assessment of quality.

This is not to say that these tools cannot assist in earlier uses of M&S. In particular the PIRT can be a great tool to engage with in determining modeling requirements for an effort. Similarly, the PCMM can be used to judge the appropriate level of formality and completeness for an effort to engage with. Nonetheless these frameworks are far more important and impactful when utilized for more mature, “engineering” focused modeling and simulation efforts.

Any high level integrated view of credibility is built upon the foundation of the issues exposed in PIRT and PCMM. The problem that often arises in a complex M&S activity is managing the complexity of the overall activity. Invariably gaps, missing efforts and oversights will creep into the execution of the work. The basic modeling activity is informed by the PIRT’s structure. Are there important parts of the model that are missing, or poorly grounded in available knowledge? From PCMM, are the important parts of the model tested adequately? The PIRT becomes a fuel for assessing the quality of the validation, and planning for an appropriate level of activity around important modeling details. Questions regarding the experimental support for the modeling can be explored in a structured and complete manner. While the credibility is not built on the PCMM and PIRT, the ability to manage its assessment is enabled by their mastery of the complexity of modeling and simulation.

In getting to the quantitative basis for assessment of credibility, the definition of the uncertainty budget for a computational simulation activity can be enlightening. While the PCMM and PIRT provide a broadly encompassing view of MS quality from a qualitative point of view, the uncertainty budget is ultimately a quantitative assessment of quality. Forcing the production of numerical values to the quality is immensely useful and provides important focus. For this to be a useful and powerful tool, this budget must be determined with well-defined principles and fairly good disciplined decision-making.

One of the key principles underlying a successful uncertainty budget is the determination of unambiguous categories for assessment. Each of these broad categories can be populated with sub-categories, and finer and finer categorization. Once an effort has committed to a certain level of granularity in defining uncertainty, it is essential that the uncertainty be assessed broadly and holistically. In other words, it is important, if not essential that none of the categories be ignored.

This can be extremely difficult because some areas of uncertainty are truly uncertain, no information may exist to enable a definitive estimation. This is the core of the difficulty for uncertainty estimation, the unknown value and basis for some quantitative uncertainties. Generally speaking, the unknown or poorly known uncertainties are more important to assess than some of the well-known ones. In practice the opposite happens, when something is poorly known the value often adopted in the assessment is implicitly defined quantitatively as “zero”. This is

implicit because the uncertainty is simply ignored, and it is not mentioned, or assigned any value. Again, the availability of the frameworks comes in handy to help the assessment identify major areas of effort.

A reasonable decomposition of the sources of uncertainty can fairly generically be defined at a high level: experimental, modeling and numerical sources. We would suggest that each of these broad areas be populated with a finite uncertainty, and each of the finite values assigned be supported by well-defined technical arguments. Of course, each of these high level areas will have a multitude of finer grained components describing the sources of uncertainty along with routes toward their quantitative assessment. For example, experimental uncertainty has two major components, observational uncertainty and natural variability. Each of these categories can in kind be analyzed by a host of additional detailed aspects. Numerical uncertainty lends itself to many sub-categories: discretization, linear, nonlinear, parallel consistency, and so on.

The key is to provide a quantitative assessment for each category at a high level with a non-zero value for uncertainty and a well-defined technical basis. We note that the technical basis could very well be “expert” judgment as long as this is explicitly defined. This gets to the core of the matter; the assessments should always be explicit and not leave essential content for implicit interpretation. A successful uncertainty budget would define the major sources of uncertainty for all three areas along with a quantitative value for each. In the case where the technical basis for the assessment is weak or non-existent, the uncertainty should be necessarily large to reflect the lack of technical basis. Like statistical sampling, the benefit to doing more work is a reduction in the magnitude of the uncertainty associated with the quantity. Enforcing this principle means that follow-on work that produces larger uncertainties requires the admission that earlier uncertainties were under-estimated. The assessment process and uncertainty budget are inherently learning opportunities for the overall effort. The assessment is simply an encapsulation of the current state of knowledge and understanding.

Too often in modeling and simulation uncertainty, efforts receive a benefit through ignoring important sources of uncertainty. By doing nothing to assess uncertainty they report no uncertainty associated with the quantity. Insult is done to this injury when the effort realizes that doing work to assess the uncertainty then can only increase its value. This sort of dynamic becomes self-sustaining, and more knowledge and information results in more uncertainty. This is a common and often seen impact of uncertainty assessment. Unfortunately this is a pathological issue. The reality is that this indicts the earlier assessment of uncertainty is actually too small. A vital principle is that more work in assessing uncertainty should always reduce uncertainty. If this does not happen the previous assessment of uncertainty was too small. This is an alltocommon occurrence that occurs when a modeling & simulation effort is attempting to convey a too large sense of confidence in their predictive capability. The value of assessed uncertainty should converge to the irreducible core of uncertainty associated with the true lack of knowledge or intrinsic variability of the thing being modeled. In many cases the uncertainty is

interacting with an important design or analysis decision where a performance margin needs to be balanced with the modeling uncertainty.

An ironic aspect to uncertainty estimation is the tendency to estimate large uncertainties where expertise and knowledge are strong, while under-estimating uncertainty in areas where expertise is weak. This is often seen with numerical error. A general trend in modeling & simulation is the tendency to treat computer codes as black boxes. As such, the level of expertise in numerical methods used in modeling & simulation can be quite low. This has the knock-on effect of lowering the estimation of numerical uncertainty, and utilizing the standard methodology for solving the equations numerically. Quite often the numerical error is completely ignored in analysis. In many cases the discretization error should dominate the uncertainty, but aspects of the solution methodology can color this assessment. Key among these issues is the nonlinear error, which can compete with the discretization error if care is not taken.

This problem is compounded by a lack of knowledge associated with the explicit details of the numerical algorithm and the aspects of the solution that can lead to issues. In this case the PCMM can assist greatly in deducing these structural problems. The PCMM provides several key points that allow the work to proceed with greater degrees of transparency with regard to the numerical solution. The code verification category provides a connection to the basis for confidence in any numerical method. Are the basic features and aspects of the numerical solution being adequately tested? The solution verification category asks whether the basic error analysis and uncertainty estimation is being done. Again the frameworks encourage a holistic and complete assessment of important details.

The final aspects to highlight in the definition of credibility are the need for honesty and transparency in the assessment. Too often assessments of M&S lack the fortitude to engage in a fundamental honesty regarding the limitations of the technology and science. If the effort is truly interested in not exposing their flaws, no framework can help. Much of the key value in the assessment is defining where effort can be placed to improve the modeling and simulation. It should help to identify the areas that drive the quality of the current capability.

If the effort is interested in a complete and holistic assessment of its credibility, the frameworks can be invaluable. Their value is key in making certain that important details and areas of focus are not over- or under-valued in the assessment. The areas of strong technical expertise are often focused upon, while areas of weakness can be ignored. This can produce systematic weaknesses in the assessment that may produce wrong conclusions. More perniciously, the assessment ignores systematic shortcomings in a modeling and simulation capability. This can lead to a deep under-estimate in uncertainty while significantly over-estimating confidence and credibility.

For M&S efforts properly focused on an honest and high integrity assessment of their capability, the frameworks of PCMM and PIRT can be an invaluable aid. The assessment can be more focused and complete than it would be in their absence. The principle good of the frameworks is to make the assessment explicit and

intentional, and avoid unintentional oversights. Their use can go great lengths to provide direct evidence of due diligence in the assessment and highlight the quality of the credibility provided to whomever utilizes the results.

“We should not judge people by their peak of excellence; but by the distance they have traveled from the point where they started.”

– Henry Ward Beecher

7 Conclusion and Outlook

We believe it is time for the community to shoulder some of the blame and rethink our approach to engaging other scientists and engineers on the topic of M&S quality. V&V should be an easy sell to the scientific and engineering establishment. It has not been and it has often been resisted at every step. V&V is basically a rearticulation of the scientific method we all learn, use and ultimately love and cherish. Instead, we find a great deal of animosity toward V&V, and outright resistance to including it as part of the M&S product. To some extent it has been successful in growing as a discipline and focus, but too many barriers still exist. Through hard learned lessons we have come to the conclusion that a large part of the reason is the V&V community’s approach. For example, one of the worst ideas the V&V community has ever had is “independent V&V”. In this model V&V comes in independently and renders a judgment on the quality of M&S. It ends up being completely adversarial with the M&S community, and a recipe for disaster. We end up less engaged and hated by those we judge. No lasting V&V legacy is created through the effort. The M&S professionals treat V&V like a disease and spend a lot of time trying to simply ignore or defeat it. This time could be better spent improving the true quality, which ought to be everyone’s actual objective. Archetypical examples of this approach in action are federal regulators (NRC, the Defense Board...for a discussion of PCMM in a regulatory context see the subsequent Chapter by Mousseau and Williams). This idea needs to be modified into something collaborative where the M&S professions end up owning the quality of their work, and V&V engages as a resource to improve quality.

The fact is that everyone doing M&S wants to do the best job they can, but to some degree don’t know how to do everything. In a lot of cases they haven’t even considered some of the issues we can help with. V&V expertise can provide knowledge and capability to improve quality if they are welcome and trusted. One of the main jobs of V&V should be to build trust so that they might provide their knowledge to important work. In sense, the V&V community should be quality “coaches” for M&S. Another way the V&V community can help is to provide appropriately leveled tools for managing quality. PCMM can be such a tool if its flexibility is increased. Most acutely, PCMM needs a simpler version. Most modeling and simulation professionals will do a very good job with some aspects of quality. Other areas of quality fall outside their expertise or interest. In a very real sense, PCMM is a catalog of quality measures that could be taken. Following

the framework helps M&S professionals keep all the aspects of quality in mind and within reach. The V&V community can then provide the necessary expertise to carry out a deeper quality approach.

If V&V allows itself to get into the role of judge and jury on quality, progress will be poor. V&V's job is to ask appropriate questions about quality as partners with M&S professionals interested in improving the quality of their work. By taking this approach we can produce an M&S future where quality continuously improves.

References

1. Boyack, B.E., Lellouche, G.S.: Quantifying reactor safety margins part 1: an overview of the code scaling, applicability, and uncertainty evaluation methodology. *Nucl. Eng. Des.* **119**(1), 1–15 (1990)
2. Boyack, B., et al.: Quantifying reactor safety margins: application of code scaling, applicability, and uncertainty evaluation methodology to a large-break, loss-of-coolant accident. Nuclear Regulatory Commission, Washington, DC (USA). Div. of Systems Research (1989)
3. Babula, M., Bertch, W.J., Green, L.L., Hale, J.P., Mosier, G.E., Steele, M.J., Woods, J.: NASA standard for models and simulations: credibility assessment scale. In: AIAA Proceedings of the 47th AIAA Aerospace Sciences Meeting, AIAA-2009-1011 (2009)
4. Fixel, D., Hennigan, G., Castro, J., Lin, P.: Charon parallel semiconductor device simulator. SAND2010-3905, Sandia National Laboratories, Albuquerque (2010)
5. Hemez, F., Atamurktur, H.S., Unal, C.: Defining predictive maturity for validated numerical simulations. *Comput. Struct.* **88**(7), 497–505 (2010)
6. Hills, R.G., Witkowski, W.R., Rider, W.J., Trucano, T.G., Urbina, A.: Development of a fourth generation predictive capability maturity model. Sandia National Laboratories, Albuquerque (2013)
7. Keiter, E.R., Thornquist, H.K., Hoekstra, R.J., Russo, T.V., Schiek, R.L., Rankin, E.L.: Parallel Transistor-Level Circuit Simulation. *Simulation and Verification of Electronic and Biological Systems*, pp. 1–21. Springer, Dordrecht (2011). http://dx.doi.org/10.1007/978-94-007-0149-6_1
8. Oberkampf, W.L., Roy, C.J.: *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge (2010)
9. Oberkampf, W.L., Pilch, M.M., Trucano, T.G.: Predictive capability maturity model for computational modeling and simulation. *Development* **2** (2007)
10. Pace, D.K.: Comprehensive consideration of uncertainty in simulation use. *J. Def. Model. Simul. Appl. Methodol. Technol.* **10**(4), 367–380 (2013)
11. Pilch, M.M., Trucano, T.G., Helton, J.C.: Ideas underlying quantification of margins and uncertainties (QMU): a white paper. Unlimited Release SAND2006-5001, Sandia National Laboratory, Albuquerque, 87185 (2006)
12. Pilch, M.M.: Predictive capability maturity model (PCMM). No. SAND2007-0102C. Sandia National Laboratories (2007)
13. Regulatory Guide 1.203: U.S. Nuclear Regulatory Commission Office of Nuclear Regulatory Research (2005)
14. Roache, P.J.: *Verification and Validation in Computational Science and Engineering*, Hermosa (1998)
15. Edwards, H.C., Stewart, J.R.: Sierra, a software environment for developing complex multi-physics applications. In: *Computational Fluid and Solid Mechanics*. Proceedings of the First MIT Conference, pp. 1147–1150 (2001)
16. Holmes, S.M.: Tough Enough. *Sandia Res.* **2**(2), 11–15 (2014)
17. Thornquist, H.K., Keiter, E.R., Hoekstra, R.J., Day, D.M., Boman, E.G.: A parallel preconditioning strategy for efficient transistor-level circuit simulation. In: *Proceedings of the 2009*

-
- International Conference on Computer-Aided Design (ICCAD'09), San Jose, pp. 410–417. ACM, New York (2009). <http://doi.acm.org/10.1145/1687399.1687477>
18. Wilson, G.E., Brent, E.B.: The role of the PIRT process in experiments, code development and code applications associated with reactor safety analysis. *Nucl. Eng. Des.* **186**(1), 23–37 (1998)
19. Unal, C., et al.: Improved best estimate plus uncertainty methodology, including advanced validation concepts, to license evolving nuclear reactors. *Nucl. Eng. Des.* **241**(5), 1813–1833 (2011)

Vincent A. Mousseau and Brian J. Williams

Abstract

This chapter describes the use of the Predictive Capability Maturity Model (PCMM) (Oberkampf et al., Predictive capability maturity model for computational modeling and simulation. Technical report, SAND2007-5948, Sandia National Laboratories, 2007) applied to a nuclear reactor simulation. The application and PCMM will be discussed relative to review by the Nuclear Regulatory Commission. In a regulatory environment, one takes on the role of a lawyer presenting evidence to a judge with a prosecuting attorney allowed to cross-examine. In this type of “hostile” environment, a structured process that logically presents the evidence is helpful.

In addition, many simulations are now multi-scale, multi-physics, and multi-code. For this level of complexity, it is easy to get lost in the details. The PCMM method has been adapted for this multi-physics multi-code software. Since the key is to provide the regulator with confidence that the software is capable of predicting the quantity of interest (QoI) with a well-quantified uncertainty, the PCMM approach is a natural solution.

Keywords

Bayesian calibration • Code scaling, applicability, and uncertainty (CSAU) • Dittus-Boelter correlation • Expert opinion distributions • Graphical user interface (GUI) • Joint distributions • Marginal distributions • Markov chain Monte Carlo (MCMC) • Neutronics • Numerical uncertainty • Phenomena

V.A. Mousseau (✉)

Sandia National Laboratories, Albuquerque, NM, USA
e-mail: vamouss@sandia.gov

B.J. Williams

Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM, USA
e-mail: brianw@lanl.gov

Identification and Ranking Table (PIRT) • Predictive Capability Maturity Model (PCMM) • Quantified Parameter Ranking Table (QPRT) • Quantity of interest (QoI) • Reynolds number • Uncertainty quantification (UQ) • Wilks formula

Contents

1	Introduction	1614
2	Uncertainty Quantification Overview	1615
2.1	Total Uncertainty	1616
2.2	Wilks Formula	1618
2.3	Scaling	1620
3	Prerequisite Work	1620
3.1	Top-Down Parameter Exposure	1621
3.2	Bottom-Up Parameter Exposure	1621
3.3	General Parameter Exposure	1622
3.4	As Built	1622
3.5	QoI Extraction	1623
3.6	Parameter Distributions	1623
3.7	User Effect	1630
4	Modifications to PCMM	1631
4.1	Quantified Parameter Ranking Table	1631
4.2	Validation Pyramid	1632
4.3	Partitioning of Code and Application PCMM	1632
5	Sample Application	1634
5.1	PIRT	1634
5.2	Numerical Uncertainty	1635
5.3	QPRT	1635
5.4	PIRT and QPRT Iteration	1636
5.5	Code PCMM	1637
5.6	Application PCMM	1640
5.7	Best Estimate Plus Uncertainty	1640
6	Presentation of the Evidence File	1642
7	Conclusions	1642
Appendix		1643
A.1	Analysis of Wilks Formula	1643
B.2	PIRT	1645
References		1648

1 Introduction

In a regulatory environment, the uncertainty quantification must withstand a strong peer review process. All uncertainty quantification offered must be proven accurate. The Predictive Capability Maturity Model (PCMM) [9] provides an easy-to-understand structure for developing evidence that all of the different modes of uncertainty have been addressed. For more details on PCMM, please see [14]. The PCMM methodology can be considered a modernized version of the code scaling, applicability, and uncertainty (CSAU) [3] which has already been embraced by the Nuclear Regulatory Commission as an appropriate way to define the uncertainty in a simulation tool.

PCMM addresses the major sources of uncertainty:

1. **Code bugs** – documentation and regression testing are two powerful tools that fall under the heading of software quality engineering (SQE) to minimize code bugs.
2. **Code verification** – addresses the uncertainty due to mistakes in implementing numerical methods.
3. **Solution verification** – quantifies the uncertainty due to numerical methods applied in suboptimal conditions like coarse mesh spacing and large time steps required for solution speed.
4. **Validation** – quantifies uncertainty due to the physics models. Verification asks, “Are you solving the equations correctly?” Validation asks, “Are you solving the correct equations?”
5. **Parameter uncertainty** – quantifies the uncertainty due to uncertainty in model parameters.
6. **Calibration** – is the mathematical process by which one reduces uncertainty in model parameters due to the addition of new experimental data.

These six topics are covered by the four PCMM evidence bins, *code verification*, *solution verification*, *model validation*, and *uncertainty quantification and sensitivity analysis*. PCMM provides a structured approach to gather the “evidence” and present it in a logical fashion.

The remainder of this chapter is organized as follows. Section 2 provides an initial overview of uncertainty quantification. This is followed in Sect. 3 by a discussion of the prerequisite work required for conducting an uncertainty quantification study. Sect. 4 introduces changes and improvements to the PCMM process presented in [14]. A sample PCMM analysis is presented in Sect. 5, followed in Sect. 6 by a discussion of the presentation of the evidence file. Conclusions are provided in Sect. 7.

2 Uncertainty Quantification Overview

One often witnesses uncertainty quantification being conducted by directly studying parameter uncertainty. There are three assumptions that are implicitly made following this process that are seldom addressed or even verbalized:

1. The code is relatively bug-free and one knows what is in the code.
2. The equations in the code are solved correctly.
3. The correct equations are solved.

If any of the three previous statements are false, the exercise of conducting a parameter uncertainty quantification study may provide useless information. If any of the three assumptions above do not hold, the potentially substantial perturbations of an uncertainty quantification study are likely to yield incorrect results. So before

one begins an uncertainty quantification process, one needs to gather evidence that the above three assumptions are true.

Statistical methods are sometimes employed by engineers because they sound “too good to be true.” One can treat the simulation tool as a black box, only study the parameters in the input processor, and, if utilizing Wilks formula (see Sect. 2.2 and Appendix A.1), only run the code 59 times for uncertainty quantification. It is important to understand the underlying assumptions of the statistical method and be convinced that it is being applied correctly.

It is normally assumed that the validation data covers the application space. For many applications this is not the case. The experiments at relevant pressures, temperatures, or flow rates are either too expensive or too dangerous to perform. In this case one must use scaling analysis to extrapolate the empirical correlations calibrated to the validation experiments to the range of applicability needed for the application. This creates a new mode of uncertainty that must be understood and addressed.

2.1 Total Uncertainty

It is important to recognize that uncertainty comes from a variety of different sources including numerical, model form, parameter, and many others. It is also important to recognize that these different forms of uncertainty interact with each other in possibly highly nonlinear interactions.

For example, the numerical viscosity induced by the numerical method may be larger than the turbulent viscosity from the turbulence model. Both of these viscosities impact the solution but it is the largest one that dominates. Verification is designed to ensure that the numerical viscosity is not the larger of these two. Validation is then designed to ensure that the model-form uncertainty for a given turbulence model is small. Finally the parameter uncertainty of a specific turbulence model can be addressed upon confirming that the numerical uncertainty and model-form uncertainty are both small.

To highlight key ideas, a framework for addressing total uncertainty is proposed that admittedly ignores important sources of uncertainty like scaling and code bugs and does not correctly capture the nonlinear relationship among different modes of uncertainty. Hopefully future research will fill in or improve this framework for total uncertainty. However, until improvements are proposed, this total uncertainty framework begins to incorporate the complexity of uncertainty quantification beyond parameter uncertainty and accommodates resource allocation arguments via comparisons among coarse metrics representing errors contributed by the individual components of total uncertainty.

The key to total uncertainty quantification is consistency across the different PCMM steps. The quantity of interest (QoI) or QoIs defined by the Phenomena Identification and Ranking Table (PIRT) process discussed further in Sect. 4.1 are the key to analyzing total uncertainty. Using the same QoIs and the same metrics, one can define numerical, model-form, and parametric uncertainty in a consistent fashion.

The first step in the total uncertainty framework is to conduct thorough verification analysis of the code(s) under study. Numerical error is used as a surrogate for “numerical uncertainty” (note that numerical error is usually better quantified as a systematic bias than as a random uncertainty, although numerical error models could incorporate both components). Chapters II and III in [10] provide formal procedures for verification assessments. Numerical errors that are too large compared against desired criteria indicate that additional numerical work is required before moving on to the next step in the total uncertainty framework. For later use, a coarse metric measuring numerical error is calculated by taking the norm of the difference between the computed QoI and the computed QoI on an infinitely refined mesh:

$$\|QoI_{\text{computed}} - QoI_{\Delta x \rightarrow 0}\| = \text{numerical error}. \quad (48.1)$$

The purpose of solution verification is to estimate this numerical error. There is a numerical tool called robust multiple regression (RMR) [13] that aids one in computing this error estimate.

The second step in the total uncertainty framework, once rigorous verification is completed, is to assess model-form uncertainty through its surrogate model-form error. Although formal statistical methods exist that formally integrate model parameter calibration (Sect. 3.6.2) with model-form error assessment (see, e.g., [11]), the total uncertainty framework takes a more traditional approach to quantifying model-form error. Chapter IV in [10] provides formal procedures for validation assessments. Model-form errors that are too large compared against desired criteria indicate that additional physical or empirical modeling work is required before moving on to the final step in the total uncertainty framework. For later use, a coarse metric measuring absolute model-form error is calculated by taking the norm of the difference between the computed QoI and the experimentally measured QoI:

$$\|QoI_{\text{computed}} - QoI_{\text{experimental}}\| = \text{model form error}. \quad (48.2)$$

If the ultimate QoI to uncertainty quantification is unobservable (no existing validation data), one must rigorously justify extrapolation of code calculations to this QoI. For example, scaling arguments may be made (see Sect. 2.3).

The final step in the total uncertainty framework, once rigorous validation is completed, is to assess model parameter uncertainty. This is conducted by formal propagation of model parameter uncertainties through code calculations of the QoIs. Such uncertainties may be obtained through expert opinion (Sect. 3.6.1) or Bayesian calibration (Sect. 3.6.2), for example. The result is a sample of QoIs from their (unknown) probability distributions. Let QoI_{bound} represent a bound from a QoI distribution for regulatory purposes, estimated from the QoI samples (modification of the following discussion is necessary if an interval is of interest instead). Example possibilities include the 95th or 99th percentiles, or fifth or first percentiles, or the bound obtained from Wilks formula. A coarse metric measuring model parameter uncertainty is calculated by taking the norm of the difference between the computed QoI and the QoI distributional bound:

$$\|QoI_{\text{computed}} - QoI_{\text{bound}}\| = \text{model parameter error.} \quad (48.3)$$

Using the coarse metrics of equations (48.1)–(48.3), it is possible to quickly identify which source of error should be prioritized for future reduction: that associated with the largest metric value (always subject to cost considerations), i.e.,

$$UQ_{\max} = \max (\|QoI_{\text{computed}} - QoI_{\Delta x \rightarrow 0}\|, \|QoI_{\text{computed}} - QoI_{\text{experimental}}\|, \\ \|QoI_{\text{computed}} - QoI_{\text{bound}}\|).$$

Although notional, this framework captures an important concept in equation form: uncertainty is a function of numerical methods, physical models, and the parameters in those physical models. The PCMM process recognizes this important concept and provides for equal importance of verification, validation, and uncertainty quantification.

2.2 Wilks Formula

The use of Wilks formula [15] has become popular in recent years in the nuclear power industry. The original work from 1941 was designed for use on assembly lines. For this application, the “parameters” were controls on the assembly line that allowed for fine adjustment. The “distribution” of the parameter was well known since it was uniform and simply the range on the physical control knob. Wilks formula is solid mathematically and a very useful tool when properly applied.

The challenge to properly applying Wilks formula to software instead of an assembly line comes from meeting the implied assumptions. In the four subsections below, the four main assumptions behind the use of Wilks formula in a software context are discussed.

2.2.1 No Code Crashes

Wilks formula must be applied to a random sample of code outputs. If the code crashes during any of the minimum 59 runs that Wilks formula requires for a 95%/95% tolerance-bound calculation, this assumption is likely violated. The sample of parameters and thus outputs is not random if code crashes are functionally related to parameter settings. Therefore one needs a very robust code for successful application of Wilks formula. In addition, as will be discussed later, one needs accurate parameter distribution functions to prevent the code from traversing unphysical parts of state space which can lead to code crashes.

2.2.2 Known Parameter Distributions

There is no requirement on the form of the parameter distribution, but it must be known. Note this technically excludes the use of expert opinion, unless the expert actually knows the parameter distribution. Often with expert opinion, the expert gives a guess at the minimum and maximum parameter values, and a uniform distribution on these bounds is assumed. Although the formula does not mandate that the parameters themselves be statistically independent, it does require that the interdependence of the parameters be explicitly known. This requisite knowledge of

the joint parameter distribution is another difficult requirement for expert opinion-based specifications.

It is interesting to observe that poor choices of parameter ranges and the joint distribution function may be the cause of code crashes. This leads to a nonlinear feedback into the system where the “expert opinion” distribution is then modified to eliminate the code crashes. This process of shrinking the expert opinion-based parameter ranges until code crashes are eliminated is in reality a measure of code robustness and has little to do with uncertainty quantification.

2.2.3 All Parameters are Analyzed

The requirement that all relevant code parameters be exposed to uncertainty quantification is perhaps the hardest assumption to implement in software. There are a very large number of parameters that impact the code calculation of the QoIs. This includes initial conditions and boundary conditions, all discretization parameters, all solver parameters, and all physical model parameters whether they are buried in the code or exposed through the input file. This also includes all of the model modifications that have been made over the years to improve the robustness of the software, such as protection against division by zero, under-relaxation, and “ramps” that connect discontinuous models.

The average code user may only have access to about 10% of the parameters that impact the QoIs. These are precisely the parameters that have been exposed by the code team for study. Finding and exposing the remainder of the parameters for study can be a difficult challenge.

2.2.4 No Biases

The machinery in Wilks formula is not designed to deal with biases. This is a particularly hard obstacle for simulation tools to overcome. This is due to the fact that all numerical errors (inside of the asymptotic convergence range) show up as a bias. That is, as the time step and mesh spacing are reduced, the code approaches the exact solution monotonically. Model-form error also has this same signature in that it introduces a bias to the solution.

2.2.5 Application

Notwithstanding the above, Wilks formula can still be applied if systematic bias adjustments to code output are obtainable through statistical inference. Wilks formula is a valuable tool in statistical analysis. In its intended application to assembly lines, the required knowledge of parameters and their distributions was easy to accommodate. One should use Wilks formula with caution when applying it to software where knowing all of the parameters and their distributions is not practical. Note that there is no provision in Wilks formula to only analyze important parameters; all parameters that impact the QoI must be analyzed.

It should be recognized that many mathematical formulas still provide useful knowledge when applied in settings where all of the assumptions behind them cannot be rigorously verified. Although there are certainly cases where Wilks formula is not conservative, current experience indicates that Wilks formula is accurate or conservative when applied to software. More detail about Wilks formula is contained in Appendix A.1.

2.3 Scaling

Many times due to cost or safety, one cannot perform the validation experiments at the correct power, pressure, or temperature. For single-physics applications, one can often define nondimensional parameters that describe how information can be scaled from one part of state space to another. The Reynolds number in CFD is a good example of how physics can be accurately scaled.

This, however, becomes significantly more complicated for multi-physics applications. There may be a dozen different nondimensional numbers that describe a multi-physics simulation. It is often impossible to match all of the relevant nondimensional numbers. In that case, the experimentalist will attempt to match the important nondimensional numbers and provide justification that the others do not have a significant impact on the solution.

It is not easy to defend applying empirical formulas outside of their range of applicability. The more physics that can be incorporated into the model, the easier it is to justify using an empirical model to extrapolate outside of its validation data range. Well-designed scaling arguments can provide a justification for extrapolation. However, these arguments are easy for a regulator to question.

Whenever possible, respect the validation data range; when this is not possible, ensure that an easily defensible scaling argument justifies extrapolation. It should be noted that CSAU is one of the predecessors of the PCMM process. In CSAU, scaling was considered one of the most important discussions in terms of establishing confidence in software.

3 Prerequisite Work

Determining which physical phenomena are important is the first step in any uncertainty quantification. This is accomplished through the PIRT process. However, this is just the start of a long labor-intensive process. One then needs to find the models in the code that describe the important physical phenomena and expose the parameters in these important models for uncertainty quantification.

In many engineering applications, the models are designed one way in the journal publication but implemented a different way in the software. Testing what the model was supposed to be, based on the journal paper, is not valuable. One needs to test the model “as built” based on how it was implemented in the software.

In addition to the parameters, the QoIs also need to be exposed for study. This can be an error-prone process if the QoIs are computed from the graphics output files instead of directly from the simulation code. The quantity and quality of the data in a graphics file may not support accurate calculation of the QoI. This produces a new mode of uncertainty related simply to errors in computing the QoI.

Assuming all the parameters and QoIs are exposed for study, the next step is to construct parameter distribution functions. Traditionally this is done based on expert opinion. However, Bayesian methods have the ability to construct (not necessarily analytically) the parameter distribution functions that can then be used for uncertainty quantification. This Bayesian process requires large amounts of

high-quality data to construct the parameter distribution functions. The Dakota software for uncertainty quantification analysis from Sandia National Laboratories contains Bayesian calibration capabilities [4–6].

There is an additional form of uncertainty known as user effect. This is caused by different equally qualified engineers making slightly different modeling assumptions. To minimize this user effect, the simulation code teams need to train the code users in best practices and user guidelines. Theoretically, properly trained code users will get very similar answers [12]. There is also a parallel document that describes user guidelines and best practices for Dakota [1].

3.1 Top-Down Parameter Exposure

In legacy multi-physics codes, there can be a large number of physical models, and each of these models can have many parameters. There is a natural way to “prune” the parameter “tree” hierarchy that was initially published in [7]. A key realization is that there are a small number of classes of physical phenomena. One can then determine the sensitivity of each class. If the sensitivity of a class is small, then one does not need to analyze the parameters that make up the models in that class.

For example, wall friction is a class. One can construct a tree of wall friction by first separating it into single-phase and two-phase wall friction. These then decompose into laminar and turbulent wall friction. The laminar and turbulent wall frictions then decompose further into flow regimes. This tree structure can require a large number of lines of code and a large number of parameters to describe.

In the top-down approach, a sensitivity multiplier is applied to the top of the tree, namely, wall friction. If perturbations on wall friction do not impact the QoI, then there is no need to invest any further in uncertainty quantification of the lower-level models. Often a general purpose tool will have a large number of models, and only a small number may be employed for any given application. The top-down approach takes advantage of this idea to minimize the amount of uncertainty quantification work.

It is important to note that if the class wall friction is important, then the same top-down idea can be used to determine important subclasses. For example, one could next determine the importance of single-phase and two-phase wall friction. Recursive application of the top-down approach will continually define smaller and smaller sections of the physical models that need to be studied.

3.2 Bottom-Up Parameter Exposure

Once an important physical phenomenon has been identified from the top-down approach, the bottom-up approach is then applied. In the bottom-up approach, one starts with a single model for an important physical phenomenon. One now determines all of the parameters in that model and their acceptable ranges and distributions. Another important part of the bottom-up approach is to define the range of applicability of the model.

Many of the closure laws are empirical in nature. This means that they are only proven accurate inside of the validation data range used to create the empirical model. In other words empirical correlations work well to interpolate experimental data. However, it is important to know when an empirical model is being extrapolated outside of its range of applicability. At a minimum, the code user needs to be warned that this is happening and ideally the uncertainty should be increased when an empirical correlation is employed outside of its range of applicability. Furthermore, application of an empirical correlation outside its range of applicability may introduce undetectable bias into the calculation that may grow with extrapolation distance.

3.3 General Parameter Exposure

One can only quantify the uncertainty of parameters that have been exposed to study. Often in legacy codes, only a small percentage of parameters are exposed for study. That means that one needs to modify the source code to allow for the parameters to be perturbed. Depending on the complexity and level of documentation, this can be a very labor-intensive process.

It is important to recognize that an uncertainty quantification exercise on the preexisting exposed parameters may provide false information. If only a few of the important parameters are available for study, there is little to no scientific value in a partial uncertainty quantification exercise. One needs to follow a dependable process, like the top-down and bottom-up process defined previously, to demonstrate that all important parameters are studied.

There is a downside to parameter exposure that has to be understood and minimized if possible. Traditionally, uncertainty quantification software works off the code input processor. One perturbs parameters for study by changing their value in the input file. There is a good reason why many important parameters are not exposed for study in the input file. The quality of the software is determined by its physical models. One does not want a code user to be able to modify the code physics. This defeats the idea of software quality engineering that ensures that the code models are correct. If the code user can change the physics through input, then one cannot build a pedigree for the code physics.

Whenever possible, it is best to expose parameters for study through a different mechanism than the input file. One wants a file that is hard to read and hard to modify and ideally the code user does not even know that it exists. One needs to recognize that parameter exposure is important for accurate uncertainty quantification, but one also needs to be cautious how many “tunable” parameters are exposed to the code user.

3.4 As Built

It is an interesting exercise to compare the closure law described in a journal paper (as designed) with what is coded in the software (as built). Sometimes it is difficult

to even recognize the closure law in the software – especially for legacy software. The closure law in the journal paper only has one requirement – the correlation needs to approximate the experimental data.

The closure law in the software has many requirements that include derivatives and correct asymptotic behavior. This is often exacerbated by applying the closure law outside of its range of applicability. Every time the code crashes in the closure law, some code developer has to “bulletproof” the correlation. This “bulletproofing” exercise can make the relation between what is in the code and what is in the journal paper hard to reconcile.

Closure laws are often based on assumptions like steady state and fully developed. This results in discontinuities in time and space in the closure laws. Discontinuities in time are often dealt with by under-relaxation. This process smooths the discontinuous closure law in time but as a consequence modifies the closure law physics. The discontinuities in space are often dealt with by applying “ramps” between closure laws. Here, one smooths the closure laws by taking an average of two adjacent control volumes. This ramp or averaging process also modifies the closure law physics.

Therefore to assess the uncertainty contribution of a closure law, it is important to look at the software and study the closure law as built, not as designed in the journal paper.

3.5 QoI Extraction

One of the challenges for uncertainty quantification (as well as verification and validation) is how to compute the QoI. In an ideal situation, the simulation tool would compute the QoI in a manner that was consistent with the numerical methods, the computational grid, and the physical models of the simulation software. This is not always possible. What is more often the case is that the code exports the required information to an output file and then the uncertainty quantification software must build the QoI from the output file. This process can be error prone because there are often approximations made to minimize the size of the output file. Unless these approximations are made clear, the QoI constructed from the approximated output files may contain significant error. This can lead to large perceived errors in verification and validation when, in reality, the error arose from how the output files were written by the code and how they were interpreted by the uncertainty quantification software.

3.6 Parameter Distributions

The key to uncertainty quantification is to obtain parameter distributions. It is often difficult to estimate parameter distributions for empirical correlations because the parameters often come from fits to experimental data. These parameters seldom have a physical interpretation so it is difficult to estimate their distributions. There are two main methods that can be used to obtain parameter distributions.

3.6.1 Expert Opinion

In expert opinion, someone who is knowledgeable about the field is asked what they expect the parameter distributions to be. Based on their experience, they often provide a uniform parameter distribution with estimates of the minimum and maximum values. Because the parameters define curve fits to experimental data, unless the expert has detailed knowledge of the experiment, this approach is very error prone. That is to say, there is a large uncertainty in the uncertainty quantification process.

Another problem of expert opinion-based parameter distributions is that bad choices for parameter distributions often lead to code crashes. This is caused by combinations of parameters leading to unphysical regions of state space. It is important to understand the relationship among parameters (joint distribution). When expert opinion-based distributions are employed, the parameters are often assumed to be statistically independent. The consequences of this erroneous assumption can be code crashes. There is then a natural tendency to make a second bad assumption to counteract the first bad assumption of parameter independence by “modifying” the expert opinion-based parameter distributions to eliminate the code crashes. In the end, one ends up with parameter distributions based entirely on the robustness of the code.

3.6.2 Bayesian Calibration

In the Bayesian calibration process, parameter distributions are constructed directly from experimental data and a model form. This replaces a very subjective approach, like expert opinion, with a very precise, mathematically sound approach, like Bayesian calibration. This makes defending the parameter distribution easier because it and all implied uncertainties are precisely defined by a process that is repeatable by an uncertainty quantification analyst. First, the basis of the Bayesian calibration theory is discussed, followed by an example for wall heat transfer and wall friction.

Theory

Denote the vector of model parameters to be calibrated by θ . Model and experimental outputs are represented by $f(x, \theta)$ and y , respectively, where x quantifies input settings describing the experimental scenario. With ε denoting observational error, in the absence of systematic model bias, the n experimental and model outputs are related as follows:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} f(x_1, \theta) \\ f(x_2, \theta) \\ \vdots \\ f(x_n, \theta) \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix},$$

which in shorthand notation is written $y = f(\theta) + \varepsilon$. Typically the errors are assumed to be independently distributed as mean zero Gaussian having common

variance σ^2 , $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_n, \sigma^2 \mathbf{I}_n)$, where $\mathbf{0}_n$ and \mathbf{I}_n denote the n -vector of zeros and the identity matrix of size n , respectively. However, this specification may be generalized by adopting less restrictive covariance structures or non-Gaussian form for the errors. In the above error specification, σ^2 joins θ as unknowns to be statistically inferred. Given values for θ and σ^2 , the joint sampling distribution of the experimental data \mathbf{y} is Gaussian having mean $\mathbf{f}(\theta)$ and covariance $\sigma^2 \mathbf{I}_n$. Viewed as a function of the parameters θ and σ^2 , this sampling distribution is referred to as the *likelihood* function and is denoted $\mathcal{L}(\theta, \sigma^2 | \mathbf{y})$.

Bayesian analysis further assumes a joint *prior* distribution for (θ, σ^2) , written $\pi(\theta, \sigma^2)$. This distribution encompasses current knowledge about these parameters and is derived through expert opinion or previously available experimental data. It is most often assumed that θ and σ^2 are a priori independently distributed, allowing the prior to be written in the product form

$$\pi(\theta, \sigma^2) = \pi_1(\theta) \pi_2(\sigma^2).$$

In fact, often the individual components of θ are assumed a priori independent, as in Sect. 3.6.1. In this case, the joint prior distribution of θ has the product form

$$\pi_1(\theta) = \prod_{i=1}^{n_\theta} \pi_1^i(\theta_i),$$

where n_θ is the number of parameters θ and $\pi_1^i(\cdot)$ is the marginal distribution of θ_i , $i = 1, \dots, n_\theta$. The ultimate goal of Bayesian calibration is to determine the *posterior* distribution of parameters θ and σ^2 , denoted $\pi(\theta, \sigma^2 | \mathbf{y})$ and given by

$$\pi(\theta, \sigma^2 | \mathbf{y}) \propto \mathcal{L}(\theta, \sigma^2 | \mathbf{y}) \pi(\theta, \sigma^2).$$

That is, the posterior distribution is proportional to the likelihood function multiplied by the prior distribution. In other words, the experimental data \mathbf{y} is used to update prior knowledge about the parameters θ and σ^2 through the likelihood function $\mathcal{L}(\theta, \sigma^2 | \mathbf{y})$. Relative to the prior distribution, values of θ and σ^2 associated with greater likelihood values (i.e., more likely to have generated the observed data) are more probable.

The posterior distribution of θ and σ^2 is generally not available analytically due to the fact that the normalizing constant

$$c = \int \mathcal{L}(\theta, \sigma^2 | \mathbf{y}) \pi(\theta, \sigma^2) d\theta d\sigma^2$$

is not easily obtained in most realistic applications. Consequently, sampling methods such as Markov chain Monte Carlo (MCMC) are used to generate a sample (say of size M) $\{(\theta_i, \sigma_i^2), i = 1, \dots, M\}$ from this posterior distribution. Given a model $g(\cdot)$ that depends on θ (and perhaps other parameters), uncertainty quantification

about model output is then obtained by summarizing the sample of model outputs $\{g(\theta_i), i = 1, \dots, M\}$. MCMC and predictive inference are infeasible if the model $f(\cdot)$ calculates slowly for each parameter setting θ . In this case, a physical or mathematical surrogate for $f(\cdot)$ must be employed in both Bayesian calibration and uncertainty quantification.

When systematic bias is present in model calculations, the statistical model used to describe the experimental data must be extended. This can be accomplished in several ways depending on the nature of the bias. Here a particular case is considered in which the n -vector of experimental data was collected across ℓ laboratories ($\ell < n$):

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_\ell \end{pmatrix}.$$

Assuming it is reasonable that factors specific to each particular laboratory could modify inference about the distribution of θ , the following sampling model is posited for data from the i -th laboratory, $\mathbf{y}_i = \mathbf{f}_i(\theta + \delta_i) + \boldsymbol{\epsilon}_i$, where $\mathbf{f}_i(\cdot)$ represents model output corresponding to experimental data from the i -th laboratory. In this scenario, θ represents settings of the model parameters common to the population of laboratories contributing data, while δ_i specifies a perturbation of this common parameter that adjusts for biases introduced by the i -th laboratory. The variance of observational errors $\boldsymbol{\epsilon}_i$ corresponding to laboratory i is denoted σ_i^2 . The perturbations δ_i are assumed a priori to be independently distributed from a mean-zero Gaussian distribution having covariance matrix $\Phi(\phi)$ for $i = 1, \dots, \ell$. Here, ϕ is the set of parameters fully describing the covariance structure of the perturbation vectors δ . For example, if the individual components of δ are taken to be independently distributed, ϕ is a n_θ -vector, and $\Phi(\phi)$ is the diagonal matrix having entries ϕ_i for $i = 1, \dots, n_\theta$. In the unrestricted case, ϕ is a $n_\theta(n_\theta + 1)/2$ -vector parameterizing the symmetric, positive definite matrix $\Phi(\phi)$. The covariance parameters ϕ are assigned a joint prior distribution, $\phi \sim \pi_3(\phi)$. Bayesian calibration proceeds as in the no-bias case, resulting in samples of $(\theta, \sigma_1^2, \dots, \sigma_\ell^2, \delta_1, \dots, \delta_\ell, \phi)$ from their joint posterior distribution. Predictive inference then proceeds as in the no-bias case; for example, uncertainty quantification about model output is obtained by summarizing the sample of model outputs $\{g(\theta_j), j = 1, \dots, M\}$.

The first model introduced above in which a common parameter is fit to the available data is referred to as a *fixed* effects model, while the second model discussed above in which a common parameter is randomly perturbed to match data from multiple cohorts is referred to as a *mixed* effects model.

Application

Mixed effects models as described in the previous section were utilized to calibrate three parameters of the Dittus-Boelter correlation for wall heat transfer and two

parameters of the McAdams correlation for wall friction. The Dittus-Boelter correlation has the following form:

$$f_{DB}(x, \theta) = \theta_1 x_1^{\theta_2} x_2^{\theta_3},$$

where x_1 is the Reynolds number and x_2 is the Prandtl number. The nominal value of θ is set to $\theta_0 = (0.023, 0.8, 0.4)$, and the prior distribution of θ is taken to be uniform on the hyperrectangle defined by the Cartesian product of the intervals $[0, 2\theta_{0,i}]$ for $i = 1, 2, 3$. The McAdams correlation has the following form:

$$f_M(x, \theta) = \theta_1 x^{\theta_2},$$

where x is the Reynolds number. The nominal value of θ is set to $\theta_0 = (0.204, -0.2)$, and the prior distribution of θ is taken to be uniform on the square defined by $[0, 2\theta_{0,1}] \times [2\theta_{0,2}, 0]$.

Figure 48.1 presents data from three laboratories used for the Dittus-Boelter calibration and data from 11 laboratories used for the McAdams calibration. Figures 48.2 and 48.3 show posterior samples of common θ and perturbed $\theta + \delta$ by laboratory, for the Dittus-Boelter and McAdams calibrations, respectively. As discussed in the previous section, common θ represents parameter settings pertaining to the entire population of laboratories contributing data to the calibration, while perturbed $\theta + \delta_i$ provides the best fit to data from the i -th laboratory. Posterior samples of common θ represent uncertainty contributions from the Dittus-Boelter and McAdams correlations in the uncertainty quantification results of Sect. 5.

3.6.3 Types of Distributions

In this section the effects of different choices for constructing parameter distribution functions are discussed in an example with two hypothetical parameters shown in Fig. 48.4.

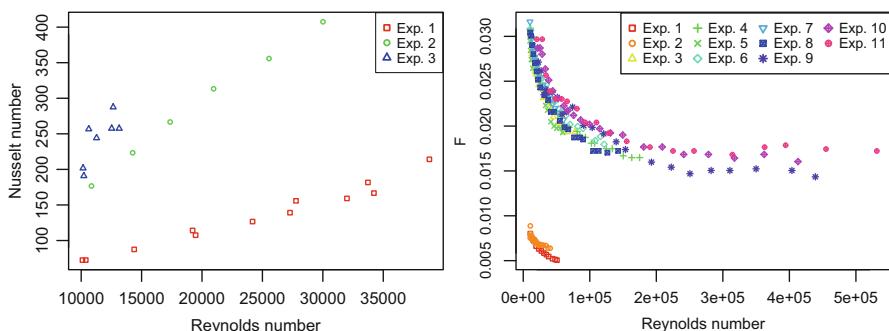


Fig. 48.1 Experimental data used for Dittus-Boelter (left) and McAdams (right) calibrations

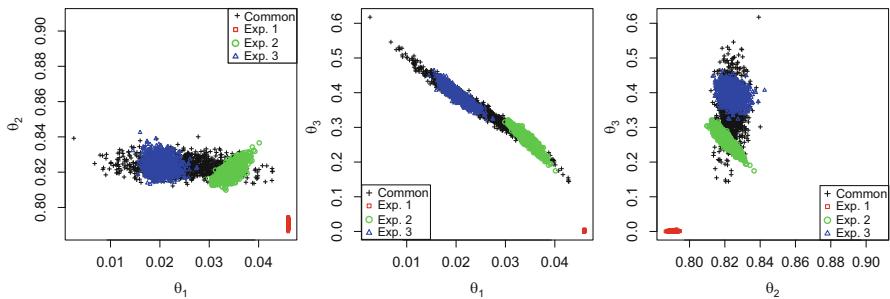


Fig. 48.2 Bivariate projections of posterior samples of common θ (black) and perturbed $\theta + \delta$ corresponding to each laboratory (see legends) from Dittus-Boelter calibration

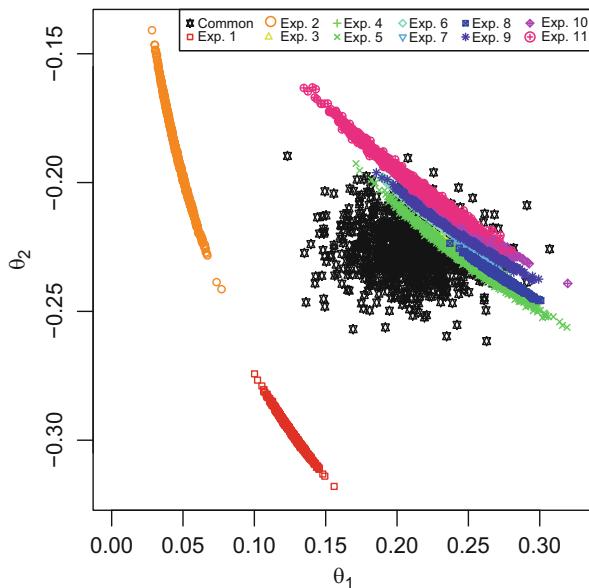


Fig. 48.3 Posterior samples of common θ (black) and perturbed $\theta + \delta$ corresponding to each laboratory (see legend) from McAdams calibration

Expert Opinion Distributions

In Fig. 48.4 the expert opinion is shown as the large blue bounding box. Unlike the Bayesian calibration-based distributions, the expert opinion-based distribution has no or little experimental data basis. Therefore the large range in parameters, which are typically assumed statistically independent and often uniformly distributed, can lead to poor choices and bad simulation results.

Marginal Distributions

Assume the parameters are statistically independent, having marginal distributions obtained from Bayesian calibration to experimental data. The support of these

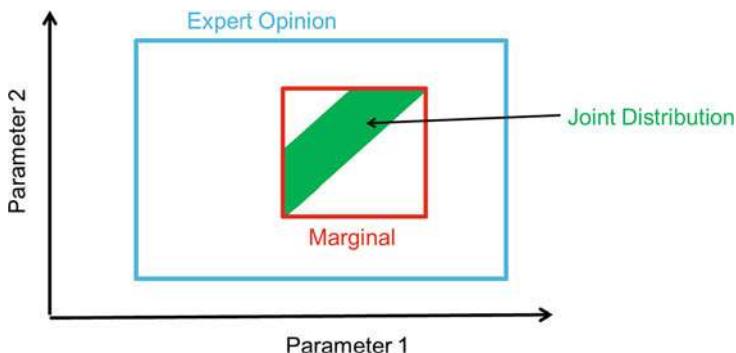


Fig. 48.4 Parameter distributions

distributions is represented as the red bounding box in Fig. 48.4. The experimental-based parameter distributions are now peaked within their ranges, reflecting marginal preferences for parameter values associated with output more consistent with the experimental data. Sampling these marginal distributions thus results in a lower likelihood of bad simulation results relative to the expert opinion case.

Joint Distributions

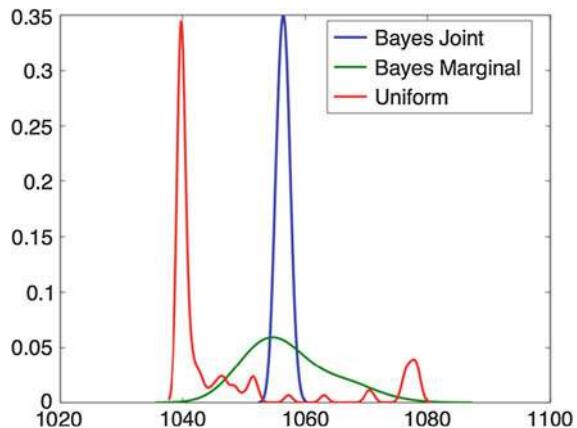
Joint distributions arise from analysis of the interdependency among parameters. The support of the joint distribution is represented as the solid green trapezoid in Fig. 48.4. Here, using experimental data, the relationship among the parameters is recognized. This results in a smaller probabilistic volume to sample from and should eliminate all code crashes due to unphysical parameter choices. It should be stressed that the assumption of parameter independence is essential for expert opinion-based parameter distributions because it is impossible, without detailed mathematical analysis of the experimental data, for an expert to intuit the relationship among parameters.

3.6.4 Effects of Different Types of Distributions

To illuminate the impact of these choices for parameter distribution function construction, an example is provided based on thermal hydraulic wall heat transfer. The uncertainty in a calculation of maximum fuel temperature is explained in terms of parameter distributions. The starting point was an expert opinion-based parameter distribution that was uniform between zero and twice the nominal value for each parameter. Random sampling of this uniform distribution resulted in many code crashes and a confusing distribution for maximum fuel temperature shown in red in Fig. 48.5. Then the expert opinion-based distribution was replaced by parameter distributions derived from Bayesian calibration of experimental data. Specifically, the Dittus-Boelter calibration results of Sect. 3.6.2 were used, noting the expert opinion-based distribution was selected as the parameter prior distribution.

In the first analysis using the Bayesian calibration results, each parameter was sampled from its marginal distribution, and assuming statistical independence, these

Fig. 48.5 Parameter distribution effects on uncertainty quantification



samples were combined across parameters to form a joint sample. The resulting distribution for maximum fuel temperature is shown in green in Fig. 48.5. A maximum fuel temperature distribution is seen that looks right skewed as also seen from expert opinion, but now unimodal due to much lower probability assigned to unphysical parameter settings. It is interesting to note that the range of the maximum fuel temperature distribution is not much different than the expert opinion-based distribution shown in red.

In the second analysis using the Bayesian calibration results, the maximum fuel temperature distribution was computed based on the joint parameter distribution. This is shown in blue in Fig. 48.5. A very tight distribution function is now seen with a nominally Gaussian shape. All sampled parameter settings in this case are probabilistically consistent with the experimental data.

It is important to note that a significant reduction in uncertainty, and a resulting larger margin, was accomplished simply through improved parameter distributions. The lower uncertainty resulting from the joint parameter distribution is easier to defend to a regulator than the larger expert opinion-based uniform distribution. As long as the expert opinion-based independent uniform parameter distribution is “larger” than the “real” parameter distribution based on experimental data, it will typically (under weak assumptions of code input-output behavior) result in a more conservative estimate of the output uncertainty.

3.7 User Effect

There is recognition in the nuclear power industry that there is a mode of uncertainty that comes from the code user called the user effect [12]. Based on the training of the code user and the level of code documentation for user guidelines and best practices, two different code users can get two different answers given the same simulation problem. This difference in answers is called the user effect.

To minimize user effect, one needs to eliminate, as much as possible, “tunable parameters.” Tunable parameters allow the code user to “dial in” whatever answer

they expect. They can force the simulation answer to behave the way the code user expects. This overriding of the code physics minimizes the work on code quality that provided the pedigree asserting the code physics was correct. If parameters are put into the input file for uncertainty quantification, a large number of tunable parameters can be created.

If ten people, given the same problem definition and the same software, get ten different answers, this typifies user effect. Code documentation that discusses best practices and user guidelines is the key to minimizing user effect. This can also be minimized by expanded input processing such as a graphical user interface (GUI). The GUI can have best practices and user guidelines built into the default values which reduce the number of parameters code users can change.

4 Modifications to PCMM

The PCMM process is evolving and can be modified to better suit the application. The following is a list of modifications, changes, and improvements to the PCMM process for this application:

1. Quantified Parameter Ranking Table (QPRT) – the QPRT augments the information from the PIRT and provides a more complete picture of the importance of physical models.
2. Validation Pyramid – a validation pyramid decomposes the QoI into its individual components.
3. Partitioning of Code and Application PCMM – PCMM applied to an application drives PCMM analysis of individual codes.

These will be discussed in the following subsections.

4.1 Quantified Parameter Ranking Table

The PIRT plays a key role in analyzing the predictability of software. The PIRT process consists of a team of experts defining which physical phenomena are important to correctly model for a given application and the set of QoIs. The experts list the important phenomena and then rank them in terms of each phenomenon's impact on each QoI (high, medium, or low) and then on available knowledge to model that phenomenon (high, medium, or low). Resources for modeling and simulation are then focused on phenomena that have high importance and low knowledge.

The challenge of a PIRT is that it is very subjective and there can be a large uncertainty in the PIRT process based on one's choice of experts. Therefore the PIRT process is augmented with a QPRT. The QPRT is a sensitivity analysis based on the parameters in the software. This is a quantitative process that is repeatable by different investigators (the expert bias has been removed). One can numerically

measure the impact of any given parameter on each QoI. These parameters can then be ranked ordinally from most important to least important. Given a working knowledge of the simulation code, each of these parameters can then be mapped back to a model of a physical phenomenon. In this way one can construct a list of phenomena that the “software” believes is important.

One is then left with the task of making the PIRT and QPRT agree. It is possible that an important physical phenomenon is ranked high in the PIRT but does not show up in the QPRT because a model for that phenomenon was not included in the code. This means that new models have to be added to the code. Sometimes the phenomena defined by the experts are not relevant to the simulation. For example, an expert may believe that interfacial friction is a key phenomenon; however, if the flow is single phase, then two-phase phenomena are not important.

There is another process that can be used to assess the quality of the PIRT and QPRT, and that is sensitivity experiments. In a sensitivity experiment, one makes small perturbations to the nominal experiment based on a physical phenomenon. One then experimentally confirms that perturbations in the phenomenon have a large or small effect on the QoI measured in the experiment.

If the PIRT, QPRT, and experimental sensitivity study all agree on what phenomena are important and are not important, then one can proceed with confidence that the important phenomena have been correctly identified. When the theoretical expert opinion-based PIRT and the mathematical and computer science-based QPRT agree with the experimental-based sensitivity studies, it is an easy process to convince a regulator that the important phenomena have been captured.

4.2 Validation Pyramid

The validation pyramid shown in Fig. 48.6 is constructed by starting with the QoI at the top and then decomposing the physics into its component parts. In the example given here, the QoI is the fuel temperature. The fuel temperature is computed based on the fuel code, the neutronic code which provides the heating, and the thermal hydraulic code which provides the cooling. In addition this process has a variety of code coupling parameters and some nonlinear feedback mechanisms. The heat source from the neutronics is a nonlinear function of the coolant density and the fuel temperature. The neutronics generates the heat which is passed into the fuel which then conducts it to the clad which is then removed by the fluid through convective heating and boiling. The validation pyramid pictorially describes all of the physical processes that need to be validated and all of the coupling and nonlinear feedback in the multi-physics code.

4.3 Partitioning of Code and Application PCMM

For a given application, one can divide the PCMM work between the top of the validation pyramid, which is specific to the application and the code coupling, and the bottom of the validation pyramid which is foundational and is independent of

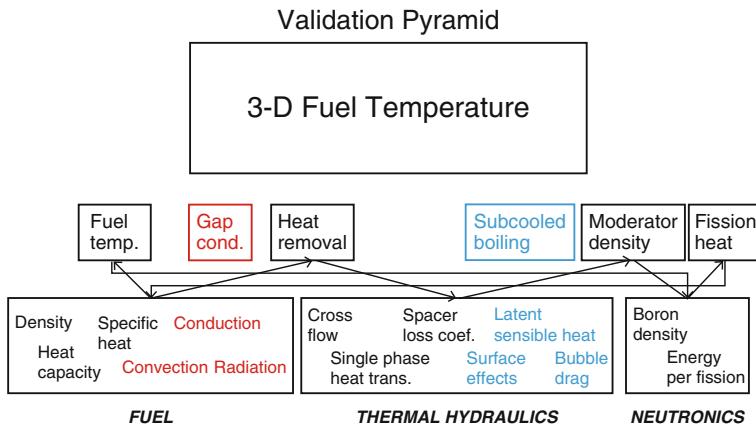


Fig. 48.6 Simplified validation pyramid

the application. This way one can partition PCMM work between the separate codes that make up the multi-physics application – in this case, fuels, thermal hydraulics, and neutronics – and the application itself which is computed from coupled codes. For a single QoI and a single application, this appears to be a somewhat arbitrary division. However, when one considers multiple QoIs and multiple applications, the value becomes more clear.

Each application and subsequent QoI clearly define a set of requirements of the individual codes. The validation pyramid precisely defines which physics are required by each separate code to construct the QoI for a given application. The individual codes can then build a foundational PCMM analysis that establishes the pedigree for the low-level capability required for the application. If there are multiple QoIs and multiple applications, then the code-level requirements are constructed from the union of all of the individual application and QoI requirements. This builds a set of application-independent requirements that have their maturity assessed by the code PCMM. The code PCMM is at a low level so validation is based on separate effects experiments.

Given a set of codes that have a PCMM pedigree for their application and QoI requirements, one is ready to then build the application and QoI PCMM pedigree. The pedigree of the coupled code utilized in the application is established. The verification and validation are now applied to the coupled application based on the solid foundation provided by the individual code PCMM. Validation at the application level is at a higher level and is therefore based on integral effects experiments.

This process provides a rapid way to move to new applications. The basic assumption is that the code PCMM will begin to saturate and all of the low-level requirements will be pedigreed by an earlier application. This way additional applications will focus mainly on integral effects experiments and coupled code applications due to the fact that the amount of work for code PCMM will diminish because of the redundancy in code-level requirements.

5 Sample Application

The sample application is for a single assembly with 17x17 fuel pins shown in Fig. 48.7. This is the simulation of a pressurized water reactor at hot full power. It includes thermal hydraulic feedback and is run at 100% of nominal power with a boron concentration of 1300 parts per million. More details of this study are presented in a technical report [8]. This test problem was chosen because it is small enough to do hundreds of runs easily, but complicated enough that it includes code coupling and key nuclear reactor phenomena. It should be noted that there are no validation data for this problem so the application validation effort will not be presented. The code validation effort was implemented with separate effects data used to calibrate some of the model parameters.

This is a steady-state problem at 100% of nominal power, implying single phase without subcooled boiling (the subcooled boiling occurs in the hot assemblies that are at higher than 100% nominal power). Due to this only a small fraction of the thermal hydraulic capability is utilized.

5.1 PIRT

The PIRT is presented in its entirety in Appendix B.2. The result of the PIRT is summarized in the following list of high-importance phenomena:

1. (H,M) Subcooled boiling (minimal effect due to void, possible effect due to improved heat transfer, but subcooled boiling occurs where the power is low)
2. (H,H) Single-phase heat transfer (Dittus-Boelter affects fuel temperature)

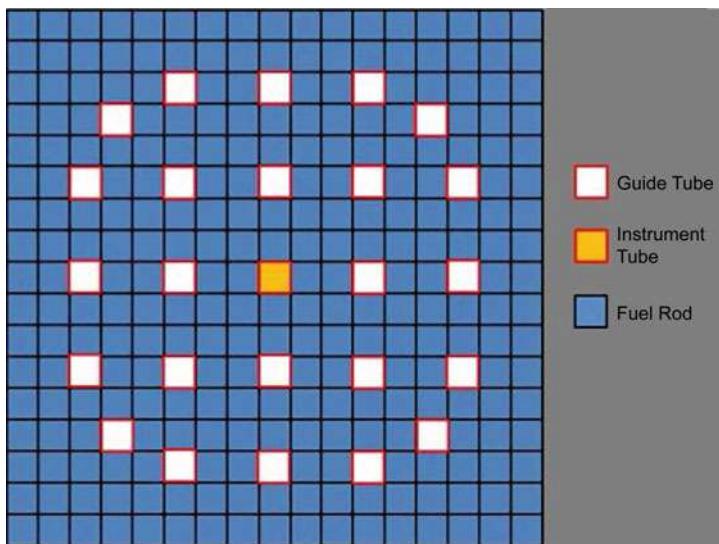


Fig. 48.7 Test problem

3. (H,L) Cross-flow models (important near guide tubes)
4. (H,M) Spacer grid loss coefficient
5. (H,M) Gap conduction model (pure He filled)
6. (H,H) Clad heat capacity
7. (H,H) Heat addition from neutronics
8. (H,H) Heat removal from thermal hydraulics
9. (H,H) Neutron cross sections
10. (H,H) Energy released per fission
11. (H,M) Boron density in the coolant
12. (H,H) Moderator density
13. (H,H) Fuel temperature (Doppler feedback)

It should be noted that some phenomena from the PIRT in this list have been collapsed into more general categories.

5.2 Numerical Uncertainty

Our first step is to use solution verification to estimate the numerical uncertainty for each of our three QoIs. The QoIs can sometimes be estimated by the stand-alone codes (CTF) and the coupled codes (VERA-CS). The verification results are presented in Table 48.1 which quantifies estimates of the numerical uncertainty.

5.3 QPRT

The code is used to estimate the important phenomena based on the software. Note that due to the complexity of the interrelationships among the cross sections, they were not simply “perturbed.” Instead, a set of 100 consistent cross-sectional libraries was constructed and each was run separately. For all other parameters, the multiplier of the nominal parameter value is listed as well as the resulting change in the QoI. It should be noted that some parameters received a larger perturbation due to the larger uncertainty in that parameter (gap conductivity and fuel conductivity in the fuel model are noted).

Table 48.2 provides sensitivities of the different parameters that represent the different phenomena in the code. The numerical uncertainty is presented in italics.

Table 48.1 Verification

Code	QoI	Numerical percent error estimate
VERA-CS	k-eff	0.002
VERA-CS	Maximum pin power	0.250
VERA-CS	Maximum fuel temperature	0.050
CTF	Maximum fuel temperature	0.015

Table 48.2 Reactivity

Variable description	Multiplier	Percent difference	Physics
Cross sections	NA	1.283	Neutronics
Cross sections	NA	-1.163	Neutronics
Gap conductivity	0.5	-0.422	Fuel
Gap conductivity	1.5	0.152	Fuel
Fuel temperature	0.95	0.081	Coupling
Fuel temperature	1.05	-0.080	Coupling
Fuel conductivity	0.9	-0.073	Fuel
Fission heat	1.05	-0.062	Coupling
Fission heat	0.95	0.061	Coupling
Fuel conductivity	1.1	0.059	Fuel
Moderator density	1.05	-0.035	Coupling
<i>Mesh spacing</i>	NA	0.020	Numerical
Moderator density	0.95	-0.017	Coupling
Wall heat transfer	0.95	-0.013	Thermal hydraulics
Moderator temperature	1.05	-0.010	Coupling
Moderator temperature	0.95	0.009	Coupling
Wall heat transfer	1.05	0.009	Thermal hydraulics

The numerical uncertainty sets the floor for future parameter work. Utilizing the concept of total uncertainty in Sect. 2.1, focus centers on the largest uncertainties among numerical, validation, and parameter. Since validation data are not available for this problem, only numerical and parameter uncertainties remain. Therefore interest in parameter uncertainties smaller than the numerical uncertainty is merely academic. As expected, the sensitivities associated with cross sections dominate the reactivity QoI. This is followed by fuel model phenomena and code coupling.

Table 48.3 provides the QPRT for the pin power QoI. As opposed to reactivity, which is a single global number, for this simulation pin power is different in every fuel pin and every axial level. As a local variable, it is a stronger function of the fuel temperature and the resulting feedback than the cross sections.

Table 48.4 provides the QPRT for the maximum fuel temperature QoI. This table is dominated by the fuel model which is expected for this QoI.

5.4 PIRT and QPRT Iteration

Table 48.5 compares the PIRT results with the QPRT results. The PIRT panel had a bias to members with a thermal hydraulic background. This resulted in thermal hydraulic phenomena being given a higher ranking. However, in the problem studied and for the QoIs selected, the neutronic and fuel models were more important than thermal hydraulics. It should also be noted that some entries in the PIRT were redundant since the lists were constructed separately for fuels, thermal hydraulics,

Table 48.3 Pin power

Variable description	Multiplier	Percent difference	Physics
Gap conductivity	0.5	-2.150	Fuel
Gap conductivity	1.5	0.905	Fuel
Cross sections	NA	0.815	Neutronics
Fuel conductivity	0.9	-0.493	Fuel
Fuel conductivity	1.1	0.411	Fuel
Fission heat	0.95	0.327	Coupling
Fission heat	1.05	-0.315	Coupling
Fuel temperature	0.95	0.280	Coupling
Fuel temperature	1.05	-0.260	Coupling
<i>Mesh spacing</i>	NA	0.250	<i>Numerical</i>
Cross sections	NA	-0.228	Neutronics
Moderator density	1.05	-0.073	Coupling
Wall heat transfer	0.95	-0.072	Thermal hydraulics
Wall heat transfer	1.05	0.053	Thermal hydraulics
Moderator density	0.95	0.052	Coupling
Moderator temperature	0.95	0.038	Coupling
Moderator temperature	1.05	-0.035	Coupling
Turbulent mixing of liquid mass	1.05	0.006	Thermal hydraulics
Turbulent mixing of liquid mass	0.95	0.005	Thermal hydraulics
Turbulent mixing of liquid momentum	1.05	0.002	Thermal hydraulics
Turbulent mixing of liquid energy	1.05	0.002	Thermal hydraulics
Turbulent mixing of liquid energy	0.95	0.001	Thermal hydraulics

and neutronics. The code coupling parameters (fission heat and fuel temperature) play a major role in this application.

5.5 Code PCMM

The next step in the process is to establish the initial PCMM scores for the separate codes. The expected PCMM values, referred to as “goals,” are also established. This first step provides an initial external assessment (outside of the code team) which identifies areas of improvement for the code team. It should be noted that the PCMM goals reflect the intended use of the application results. In this example, the application was simply a progression problem used to monitor code development progress. Since no important decisions are based on the results, the goal PCMM scores are low.

The second step is to come back later and remeasure the PCMM score relative to the goal. Once the software has achieved its PCMM goal, additional work is of lower priority.

Table 48.4 Maximum fuel temperature

Variable description	Multiplier	Percent difference	Physics
Gap conductivity	0.5	21.423	Fuel
Gap conductivity	1.5	-7.445	Fuel
Fuel conductivity	0.9	6.807	Fuel
Fuel conductivity	1.1	-5.405	Fuel
Fission heat	1.05	4.436	Coupling
Fission heat	0.95	-4.354	Coupling
Cross sections	NA	0.739	Neutronics
Wall heat transfer	0.95	0.610	Thermal hydraulics
Wall heat transfer	1.05	-0.495	Thermal hydraulics
Fuel temperature	0.95	0.254	Coupling
Fuel temperature	1.05	-0.236	Coupling
Cross sections	NA	-0.198	Neutronics
Moderator density	1.05	-0.106	Coupling
Moderator density	0.95	0.085	Coupling
<i>Mesh spacing</i>	NA	0.050	Numerical
Moderator temperature	0.95	0.034	Coupling
Moderator temperature	1.05	-0.032	Coupling
Turbulent mixing of liquid mass	0.95	0.005	Thermal hydraulics
Turbulent mixing of liquid momentum	1.05	0.002	Thermal hydraulics
Turbulent mixing of liquid mass	1.05	-0.001	Thermal hydraulics
Turbulent mixing of liquid energy	0.95	0.001	Thermal hydraulics

Table 48.5 Comparison of PIRT and QPRT

Phenomenon	PIRT	QPRT	Resolution
Subcooled boiling	High	No effect	No subcooled boiling occurred in the simulation studied
Dittus-Boelter	High	Low	Minor effect
Cross-flow models	High	Very low	Less than numerical uncertainty
Grid spacer loss coefficient	High	No effect	Minor impact on density
<i>Gap conduction model</i>	<i>High</i>	<i>High</i>	<i>Major contributor</i>
Clad heat capacity	High	Low	Overwhelmed by fuel heat capacity
<i>Fission heat</i>	<i>High</i>	<i>High</i>	<i>Major contributor</i>
Heat removal from thermal hydraulics	High	Low	Already accounted for by Dittus-Boelter
<i>Neutron cross sections</i>	<i>High</i>	<i>High</i>	<i>Major contributor</i>
Energy released from fission	High	High	Already accounted for by fission heat
Boron density	High	No effect	Boron density was held constant
Moderator density	High	Low	Minor impact
<i>Fuel temperature</i>	<i>High</i>	<i>High</i>	<i>Major contributor</i>

Table 48.6 Neutronic PCMM scores

Element\score	Goal	Initial	Current
Geometry (Geom)	1.0	0.0	1.5
Physics (Phys)	2.0	1.0	2.0
Code verification (Code Ver.)	1.0	1.0	1.0
Solution verification (Sol. Ver.)	1.0	0.0	1.0
Validation (Val.)	1.0	0.5	1.5
Uncertainty quantification (UQ)	1.0	0.0	1.0

Fig. 48.8 Neutronic Kiviat diagram

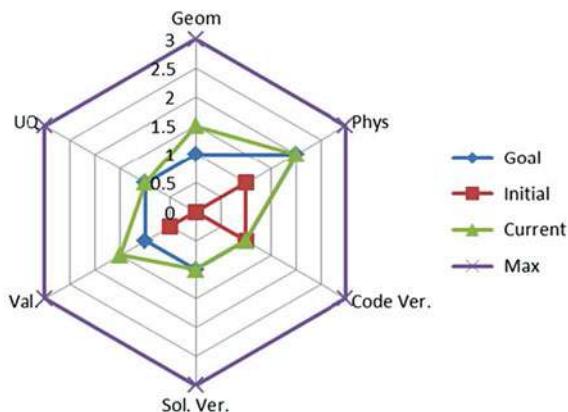


Table 48.7 Thermal hydraulic PCMM scores

Element\score	Goal	Initial	Current
Geometry (Geom)	2.0	2.0	2.0
Physics (Phys)	1.0	1.0	1.0
Code verification (Code Ver.)	2.0	1.0	1.25
Solution verification (Sol. Ver.)	1.5	1.5	1.5
Validation (Val.)	2.0	0.5	1.5
Uncertainty quantification (UQ)	1.0	0.0	1.0

5.5.1 Neutronics

Table 48.6 provides the initial PCMM scores, goals, and current scores for neutronics. In all cases, the current score exceeds or meets the goal. Note that during this study, it was decided that the neutronic code would be replaced. This resulted in relatively low goal PCMM values.

The results of Table 48.6 can be quickly summarized by the Kiviat diagram shown in Fig. 48.8. The Kiviat diagram provides a quick reference for where additional work is needed, if anywhere.

5.5.2 Thermal Hydraulics

Table 48.7 provides the PCMM scores for thermal hydraulics. In this case, the thermal hydraulic code was older and therefore was appropriately measured as more mature software.

Fig. 48.9 Thermal hydraulic Kiviat diagram

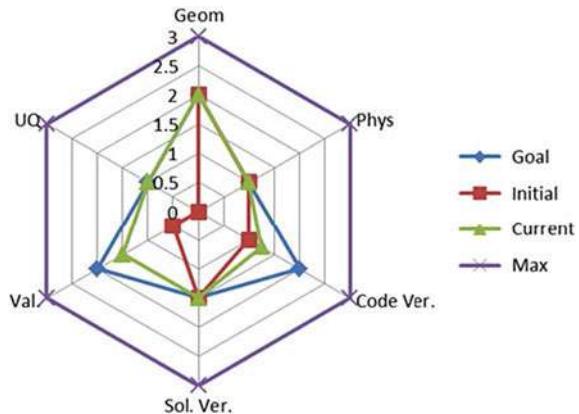


Table 48.8 Application PCMM scores

Element\score	Goal	Current
Geometry (Geom)	1.0	1.0
Physics (Phys)	1.0	1.0
Code verification (Code Ver.)	1.0	0.5
Solution verification (Sol. Ver.)	1.5	1.5
Validation (Val.)	0.0	0.0
Uncertainty quantification (UQ)	1.0	1.0

Figure 48.9 shows the pictorial summary of Table 48.7. This diagram shows that additional work is needed in code verification and validation. It should be noted that this work was completed after this study was conducted.

5.6 Application PCMM

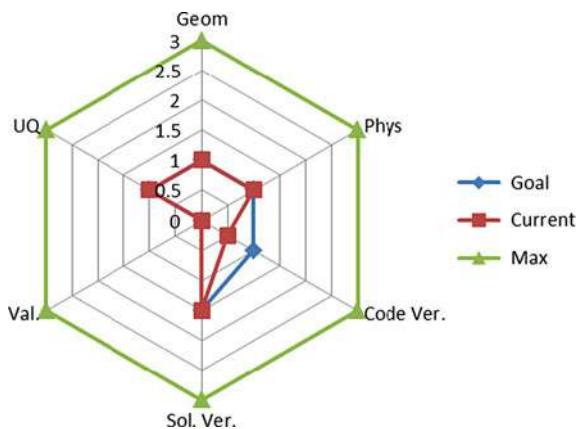
Table 48.8 provides the PCMM scores for the coupled code application. It is sometimes challenging to separate the code PCMM from the application PCMM since there is a certain amount of overlap. Note that since this application does not have any validation experimental data, the goal for validation is zero.

Figure 48.10 shows the summary of Table 48.8. It is seen that all the goal values have been met except for application code verification. It should be noted that whereas solution verification is easy for an application, code verification often requires additional software development (e.g., method of manufactured solutions).

5.7 Best Estimate Plus Uncertainty

Based on the parameter downselect from the sensitivity study, an uncertainty quantification study was conducted using a small subset of the code parameters focusing on the ones that had significant impact on the QoIs. These UQ studies

Fig. 48.10 Application Kiviat diagram



included a variety of approaches. For neutronics, all of the cross sections were represented as a single parameter having uncertainty captured by perturbed cross-sectional libraries. Some of the parameters had detailed parameter distributions constructed from Bayesian calibration of separate effects tests as described in Sect. 3.6.2. Some parameters used expert opinion which is always the fallback position. The result was the construction of a two-standard-deviation uncertainty range for each of the QoIs.

5.7.1 Reactivity

The reactivity uncertainty, although small on a percentage basis, was higher than expected. Future efforts will be applied to understand this large uncertainty and make attempts to reduce it:

$$\text{Reactivity (nondimensional)} = 1.16415 \pm 0.01242 \quad (48.4)$$

5.7.2 Maximum Rod Power

The uncertainty in the maximum rod power was small and would have little impact on the decision-making process:

$$\text{Power} \left(\frac{\text{Watts}}{\text{cm}^2} \right) = 269.445 \pm 4.60 \quad (48.5)$$

5.7.3 Maximum Fuel Temperature

The uncertainty in the maximum fuel temperature was the result of large uncertainty in the gap conductance model and the fuel conductivity model. Work is currently underway to employ a high-fidelity fuel performance code to reduce the uncertainty in the low-fidelity fuel model employed in this study:

$$\text{Temperature (C)} = 1991.859 \pm 338.0 \quad (48.6)$$

6 Presentation of the Evidence File

When the best estimate plus uncertainty results are being presented in a hostile environment where a regulator is reviewing the information, it is important they be presented in a logical fashion that is easy to follow. The “story” should be presented in a manner that demonstrates a high-level understanding of both the application physical phenomena and a deep understanding of the code results. The evidence files for each of the six PCMM elements should be available for a “deep dive” if details are needed to support conclusions. However, these details can be summarized quickly in a Kiviat diagram, and there is no need to “step through” all of the evidence.

It is important to recognize the major sources of uncertainty as numerical, model-form (validation), and parameter uncertainty. It is important to present a succinct summary that all important topics have been covered in a logical fashion. For parameter uncertainty, it is desirable to establish which parameters were studied and which parameters were not with a logical explanation of why parameters were excluded. When possible parameter distributions based on Bayesian calibration of separate effects tests should be employed, but expert opinion is always the fallback position.

The key to presenting uncertainty quantification results in a regulatory environment is to prepare as much as possible for whatever reasonable questions could be asked. If questions are answered quickly and concisely with backup evidence available, the regulators will quickly gain confidence in the presenter. Because of the high level of stress induced by a regulatory review, “thinking off the top of one’s head” can be dangerous. It is sometimes preferable to simply admit ignorance and ask for more time to study than to give an incorrect answer.

7 Conclusions

There is a strong pull to use “black box” uncertainty quantification methods. However, accurate application of these methods requires detailed understanding of both the UQ method and the simulation code being studied. In that sense there are really no “black box” methods that work for all codes all of the time. This is particularly true when presenting results in a regulatory environment:

1. One may be reviewed by a statistician who has a detailed knowledge of the UQ methods and their underlying assumptions.
2. One may be reviewed by a numerical method expert that clearly understands all of the shortcomings of the numerical approaches employed by the code.
3. One may be reviewed by an experimentalist having detailed understanding of the physics being modeled and the flaws in the physical models employed by the code.

4. Finally one may be reviewed by someone who has intimate knowledge of the simulation software and knows what model parameters are exposed for study and which model parameters are hidden in the code.

If the regulatory decision is important (e.g., related to nuclear reactor safety), one will most likely encounter experts in the review process from all four fields above. In a regulatory environment, black box approaches are only useful if the regulator does not know what is in the box.

It is important to understand all of the different sources of uncertainty and to address all of them clearly and logically. The PCMM process provides a structure that ensures that all important topics are accounted for. It is important for the regulator to be able to clearly understand the pedigree behind the best estimate value of the QoI and the quantified uncertainty in the QoI. It is not important to present all of the details, but it is important to have all of the details readily available when questions are asked. The PCMM Kiviat diagrams provide a very succinct summary of the work, and the PCMM evidence files provide the detailed information needed to answer specific questions.

Appendix

A.1 Analysis of Wilks Formula

In 1941 Wilks [15] introduced a nonparametric method for setting tolerance limits. This method assumes that the population cumulative distribution function $G(\cdot)$ of the QoI admits a density function $g(\cdot)$ with respect to Lebesgue measure, i.e., $\frac{d}{dy}G(y) = g(y)$. Calculation of Wilks tolerance limits requires the capability to randomly sample the distribution $G(\cdot)$. In the applications of interest here, the distribution $G(y)$ is defined as the probability of the set $\{\theta : f(\theta) \leq y\}$, where $f(\theta)$ denotes the results of a computational model evaluated at parameter setting θ . In this scenario, the random sampling requirement is met by (a) possessing an ability to sample θ from its correct distribution, even if this distribution is not available analytically (cf. Sect. 2.2.2), and (b) confirming that code crashes are independent of parameter values θ (cf. Sect. 2.2.1). The density $g(y)$ will not correctly represent the true output distribution, and therefore Wilks tolerance limits will be biased, if (a) the assumed parameter vector θ excludes additional inputs that if varied would induce changes in the QoI (cf. Sect. 2.2.3) or (b) $f(\theta)$ produces biased predictions of the QoI for θ values belonging to a set of positive probability (cf. Sect. 2.2.4).

Given an independent and identically distributed sample (Y_1, \dots, Y_n) from $g(\cdot)$, form the *order statistics* $(Y_{(1)}, \dots, Y_{(n)})$, where $Y_{(1)} \leq Y_{(2)} \leq \dots \leq Y_{(n)}$. Upper [lower] one-sided tolerance bounds of the form $(-\infty, U_t) = (-\infty, Y_{(t)})$ [$(L_t, \infty) = (Y_{(s)}, \infty)$] are considered and tolerance intervals of the form $(L_t, U_t) = (Y_{(s)}, Y_{(t)})$, $s < t$. Define the following random variables,

$$\begin{aligned} P_U &= 1 - G(Y_{(t)}) = \int_{Y_{(t)}}^{\infty} g(y) dy, \\ P_L &= G(Y_{(s)}) = \int_{-\infty}^{Y_{(s)}} g(y) dy, \text{ and} \\ P_C &= G(Y_{(t)}) - G(Y_{(s)}) = \int_{Y_{(s)}}^{Y_{(t)}} g(y) dy. \end{aligned}$$

The upper one-sided tolerance bound $(-\infty, Y_{(t)})$ has (random) coverage probability $1 - P_U$, the lower one-sided tolerance bound $(Y_{(s)}, \infty)$ has coverage probability $1 - P_L$, and the tolerance interval $(Y_{(s)}, Y_{(t)})$ has coverage probability P_C . Starting from the joint probability distribution of $(Y_{(s)}, Y_{(t)})$ (cf. [2]), these coverage probabilities each have beta distributions:

$$\begin{aligned} P_U &\sim \text{Beta}(n - t + 1, t), \\ P_L &\sim \text{Beta}(s, n - s + 1), \text{ and} \\ P_C &\sim \text{Beta}(t - s, n - t + s + 1). \end{aligned}$$

The cumulative distribution function of the beta(a, b) distribution evaluated at p is the regularized incomplete beta function denoted $I_p(a, b)$. The upper one-sided tolerance bound $(-\infty, Y_{(t)})$ is $(100 \times (1 - \alpha))/100 \times P\%$ if

$$\Pr[1 - P_U \geq P] \geq 1 - \alpha,$$

or equivalently $I_{1-P}(n - t + 1, t) \geq 1 - \alpha$. In similar fashion, the lower one-sided tolerance bound $(Y_{(s)}, \infty)$ is $(100 \times (1 - \alpha))/100 \times P\%$ if $I_{1-P}(s, n - s + 1) \geq 1 - \alpha$, and the tolerance interval $(Y_{(s)}, Y_{(t)})$ is $(100 \times (1 - \alpha))/100 \times P\%$ if $I_P(t - s, n - t + s + 1) \leq \alpha$. The Wilks method involves selecting the smallest sample size n for which the relevant condition involving the beta cumulative distribution function is satisfied, thus guaranteeing the corresponding tolerance bound/interval is $(100 \times (1 - \alpha))/100 \times P\%$. This method is nonparametric because no knowledge of $g(\cdot)$ is required other than its existence. This is advantageous in the many applications for which $g(\cdot)$ is unknown, although this approach can be highly inefficient for known $g(\cdot)$.

Often, the Wilks method is applied to a symmetric interval, obtained by taking $s = r$ for r a positive integer, and $t = n - r + 1$. Therefore, the upper one-sided tolerance bound $(-\infty, Y_{(n-r+1)})$, or the lower one-sided tolerance bound $(Y_{(r)}, \infty)$, are $(100 \times (1 - \alpha))/100 \times P\%$ by selecting the smallest n satisfying

$$I_{1-P}(r, n - r + 1) \geq 1 - \alpha.$$

Table 48.9 Minimum sample size n required for 95%/95% Wilks tolerance bounds/intervals

Tolerance bound		Tolerance interval	
r	n	r	n
1	59	1	93
2	93	2	153
3	124	3	208
4	153	4	260
5	181	5	311
6	208	6	361
7	234	7	410
8	260	8	458
9	286	9	506
10	311	10	554

For $r = 1$, this condition is equivalent to selecting the smallest n satisfying $P^n \leq \alpha$. The tolerance interval $(Y_{(r)}, Y_{(n-r+1)})$ is $(100 \times (1 - \alpha)\%) / (100 \times P\%)$ by selecting the smallest n satisfying

$$I_P(n - 2r + 1, 2r) \leq \alpha.$$

Table 48.9 provides the minimum sample sizes n required to obtain 95%/95% Wilks tolerance bounds/intervals for $r \in \{1, 2, \dots, 10\}$, calculated from the recipe of the previous paragraph. Note that increasing r corresponds to using more centrally located order statistics to define the tolerance bounds/intervals. It is seen that larger sample sizes n are required to guarantee that tolerance bounds/intervals using increasingly centralized order statistics will achieve the required 95% coverage of the distribution $G(\cdot)$ at the 95% confidence level.

B.2 PIRT

This simulation models a single assembly in a nuclear reactor with neutronic, thermal hydraulic, and a fuel model.

B.2.1 The Basic Physics

1. Compute a neutron flux that produces energy from fission (deposited in the fuel and the coolant).
2. Conduct the energy in the fuel radially out from the center, across the gap, and through the clad.
3. Remove the heat from the clad to the coolant and advect it out of the core.

B.2.2 Quantity of Interest

1. Reactivity (eigenvalue)
2. 3D pin power distribution
3. 3D fuel temperature

B.2.3 Phenomena

Phenomena are listed for each of the three physics areas. Phenomena will be ranked as an ordered pair (importance, knowledge) immediately after each phenomenon number.

Thermal Hydraulics

Here the fluid flow over the fueled portion of the pin is considered:

1. (H,M) Subcooled boiling (minimal effect due to void, possible effect due to improved heat transfer, but subcooled boiling occurs where the power is low).
2. (M,H) Water density (10% change over core, uncertainty in steam table ASME 1967 vs. ASME 1998).
3. (H,H) Single-phase heat transfer (Dittus-Boelter affects fuel temperature).
4. (L,M) Gamma heating (2% of total power directly into the coolant).
5. (L,L) Dimensional changes due to thermal expansion (may be smaller than other uncertainties in thermal hydraulics).
6. (M,M) Inlet mass flow rate (no feedback from loop so pressure drops sum to zero).
7. (M,M) Inlet temperature (no feedback from loop so changes in temperature sum to zero).
8. (M,H) Wall friction.
9. (H,H) Since the figure of merit is reactivity, power can be fixed for an eigenvalue search (cannot fix eigenvalue = fix reactivity and do a power search). Until transient capability is available, eigenvalue is the only logical search variable.
10. (H,L) Cross-flow models (important near guide tubes).
11. (H,M) Spacer grid model:
 - (a) (H,M) Loss coefficient is steady state.
 - (b) (M,M) Mixing term.
 - (c) (M,M) Enhanced heat transfer.

Fuel Model

Here the pellet, the gap, and the clad are considered. Energy is transferred from the fuel across the gap to the clad. The fuel model in CTF is currently being employed; later this will be replaced by Bison-CASL. Note that only fresh fuel is being considered so there are no burnup effects:

1. (M,M) Density of the fuel pellet
2. (M,M) Thermal conductivity of the fuel pellet
3. (M,M) Heat capacity of the fuel pellet
4. (H,M) Gap conduction model (pure He filled)
5. (L,L) Dimensional changes due to thermal expansion
6. (H,H) Clad density
7. (H,H) Clad thermal conductivity
8. (H,H) Clad heat capacity

9. (M,L) Model error due to assumptions in channel code
10. (H,H) Heat addition from neutronics
11. (H,H) Heat removal from thermal hydraulics

Neutronics

The challenge here is figuring out what to do with the large number of cross sections, which include fission, absorption, and scattering. The following list of materials pertain to this application:

1. U235
2. U238
3. U234
4. U236
5. Inconel
6. Zirconium
7. Oxygen
8. Hydrogen
9. Helium
10. Boron
11. Steel (nozzles and core plates)

There are 23 energy groups in the SPn calculation and 252 energy groups in the pin cell computations. It is important to note that the neutronic domain is larger than the thermal hydraulic domain. SCALE may provide uncertainty quantification information. It is necessary to conduct a literature search to look for cross-sectional prioritization work. The important phenomena for the QoIs (eigenvalue and 3D pin power directly or 3D fuel temperature is coupled through 3D power and cross-sectional feedback) are:

1. (H,H) Neutron fission (nuclear data would be the main component of this).
2. (H,H) Neutron/gamma absorption.
3. (H,H) Neutron/gamma scattering.
4. (H,H) Neutrons released per fission.
5. (M,H) Other neutron reactions ($n,2n$), ($n,3n$).
6. (H,H) Energy released per fission.
7. (L,H) Energy released per capture.
8. (L,M) Gamma energy deposition (in fuel, clad, and moderator).
9. (H,M) Boron density in the coolant.
10. (H,H) Moderator density.
11. (M,H) Moderator temperature.
12. (H,H) Fuel temperature (Doppler feedback).
13. (L,H) Clad temperature.
14. (M,M) Fuel density.
15. (L,L) Manufacturing uncertainties (mass and dimensions in fuel/clad/grids/structure).

16. (M,M) Neutron/gamma transport (includes things like SPn order, number of groups, homogenization, etc.).
17. (L,M) Bowing (only effects of thermal expansion since depletion is not included).
18. (L,M) Thermal expansion.
19. (L,L) What are the neutronic effects of layer of bubbles on fuel rod surface?

References

1. Adams, B.M., Hooper, R.W., Lewis, A., McMahan, J.A. Jr., Smith, R., Swiler, L.P., Williams, B.J.: User guidelines and best practices for CASL VUQ analysis using Dakota. Technical report, CASL-U-2014-0038-000, Consortium for Advanced Simulation of LWRs (2014)
2. Arnold, B.C., Balakrishnan, N., Nagaraja, H.N.: A First Course in Order Statistics. SIAM, Philadelphia (2008)
3. Boyack, B., Duffey, R., Griffith, P., Lellouche, G., Levy, S., Rohatgi, U., Wilson, G., Wulff, W., Zuber, N.: Quantifying reactor safety margins: application of code scaling, applicability, and uncertainty evaluation methodology to a large-break, loss-of-coolant accident. Technical report, NUREG/CR-5249, Nuclear Regulatory Commission (1989)
4. Adams, B.M. et al.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.2 reference manual. Technical report, SAND2014-5015, Sandia National Laboratories (2015)
5. Adams, B.M. et al.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.2 theory manual. Technical report, SAND2014-4253, Sandia National Laboratories (2015)
6. Adams, B.M. et al.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.2 user's manual. Technical report, SAND2014-4633, Sandia National Laboratories (2015)
7. Martin, R.P., O'Dell, L.D.: AREVA's realistic large break LOCA analysis methodology. *Nucl. Eng. Des.* **235**, 1713–1725 (2005)
8. Mousseau, V.A., Dinh, N., Rider, W., Abdel-Khalik, H., Williams, B., Smith, R., Adams, B., Belcourt, N., Hooper, R., Coppers, K.: Demonstration of integrated DA/UQ for VERA-CS on a core physics progression problem. Technical report, CASL-I-2014-0158-000, Consortium for Advanced Simulation of LWRs (2014)
9. Oberkampf, W.L., Pilch, M., Trucano, T.G.: Predictive capability maturity model for computational modeling and simulation. Technical report, SAND2007-5948, Sandia National Laboratories (2007)
10. Oberkampf, W.L., Roy, C.J.: Verification and Validation in Scientific Computing. Cambridge University Press, Cambridge (2010)
11. Oliver, T.A., Terejanu, G., Simmons, C.S., Moser, R.D.: Validating predictions of unobserved quantities. *Comput. Methods Appl. Eng.* **283**, 1310–1335 (2015)
12. Petruzzelli, A., D'Auria, F., Bajs, T., Reventos, F., Hassan, Y.: International course to support nuclear licensing by user training in the areas of scaling, uncertainty, and 3D thermal-hydraulics/neutron-kinetics coupled codes: 3D S.UN.COP seminars. *Sci. Technol. Nucl. Install.* **2008**(874023), 1–16 (2008)
13. Rider, W., Witkowski, W., Kamm, J.R., Wildey, T.: Robust verification analysis. *J. Comput. Phys.* **307**, 146–163 (2016)
14. Rider, W., Witkowski, W., Mousseau, V.: UQs Role in Modeling and Simulation Planning, Credibility and Assessment Through the Predictive Capability Maturity Model. Springer, Switzerland (2015)
15. Wilks, S.S.: Determination of sample sizes for setting tolerance limits. *Ann. Math. Stat.* **12**(1), 91–96 (1941)

Part VII

**Introduction to Software for Uncertainty
Quantification**

Dakota: Bridging Advanced Scalable Uncertainty Quantification Algorithms with Production Deployment

49

Laura P. Swiler, Michael S. Eldred, and Brian M. Adams

Abstract

This chapter highlights uncertainty quantification (UQ) methods in Sandia National Laboratories' Dakota software. The UQ methods primarily focus on forward propagation of uncertainty, but inverse propagation with Bayesian calibration is also discussed. The chapter begins with a brief Dakota history and mechanics of licensing, software and documentation acquisition, and getting started, including interfacing simulations to Dakota. Early sections are devoted to core sampling, stochastic expansion, reliability, and epistemic methods, while subsequent sections discuss more advanced capabilities such as mixed epistemic-aleatory UQ, multifidelity UQ, optimization under uncertainty, and Bayesian calibration. The chapter concludes with usage guidelines and a discussion of future directions.

Keywords

Dakota software • Open-source software • Black box • Parallel computing • Surrogate models • Sampling • Reliability • Polynomial chaos expansions • Stochastic collocation • Epistemic UQ • Interval estimation • Multifidelity • Stochastic design • Bayesian calibration • Adaptive methods • Sensitivity analysis • Optimization • Calibration • Importance sampling

L.P. Swiler (✉) • M.S. Eldred • B.M. Adams

Optimization and Uncertainty Quantification Department, Sandia National Laboratories,
Albuquerque, NM, USA

e-mail: lpswile@sandia.gov; mseldre@sandia.gov; briadam@sandia.gov

Contents

1	Introduction	1652
1.1	History and Capabilities	1653
1.2	Obtaining the Software	1654
1.3	Getting Started Resources	1655
1.4	Interfacing to Dakota	1655
1.5	Characterizing Uncertain Variables	1658
2	Sampling Methods	1658
2.1	Latin Hypercube Sampling Example	1660
2.2	Importance Sampling	1662
3	Stochastic Expansion Methods	1662
3.1	Polynomial Chaos Expansion	1663
3.2	Stochastic Collocation	1665
3.3	PCE Example	1665
4	Reliability Methods	1666
4.1	Local Reliability Methods	1670
4.2	Global Reliability Methods	1670
4.3	Local Reliability Example	1672
5	Epistemic Methods	1673
5.1	Interval Methods for Epistemic Analysis	1674
5.2	Dempster-Shafer Theory of Evidence	1675
6	Advanced Capabilities	1675
6.1	Mixed Aleatory-Epistemic UQ	1675
6.2	Multifidelity UQ	1677
6.3	Optimization Under Uncertainty (OUU)	1682
6.4	Bayesian Methods	1683
7	Usage Guidelines	1688
7.1	Sampling Methods	1688
7.2	Reliability Methods	1689
7.3	Stochastic Expansions Methods	1690
7.4	Epistemic Uncertainty Quantification Methods	1690
7.5	Mixed Aleatory-Epistemic UQ Methods	1690
7.6	Bayesian Methods	1690
8	Conclusion	1691
	References	1691

1 Introduction

Sandia National Laboratories' Dakota software (available at <http://dakota.sandia.gov>) supports advanced exploration of computational models using a variety of algorithms, including uncertainty quantification methods. In general, Dakota manages and analyzes ensembles of simulations to provide broader and deeper perspective for analysts and decision makers. This enables them to enhance their understanding of risk, improve products, and assess simulation credibility. In its simplest mode, Dakota can automate parameter variation studies through a generic interface to a computational model. However, it also includes advanced parametric analysis techniques enabling global sensitivity analysis, design exploration, optimization, model calibration, risk analysis, and quantification of margins and uncertainty. These often support verification and validation activities. The Dakota project

includes significant algorithm research and development to make these analyses practical for complex science and engineering models.

1.1 History and Capabilities

Dakota was initiated as an internal R&D project at Sandia National Laboratories in 1994 and has continually evolved since then. Over 30 people, including external collaborators, have contributed over the years; see <http://dakota.sandia.gov/contributors.html>. Early development focused on design optimization, both unifying popular algorithmic approaches and developing new optimization methods to improve performance on computational engineering problems. This enabled designers to easily exercise parameter sweeps and various optimizers on a problem of interest.

Dakota grew to include many iterative analysis approaches which fit in the general workflow shown in Fig. 49.1, where Dakota iteratively executes a simulation code. Based on the characterization of input variables and method chosen, e.g., parameter study or uncertainty quantification, Dakota generates one or more parameter sets at which to evaluate the model. Here, each parameter set is a combination of simulation input values, also known as design points in optimization or sample points in UQ. For each parameter set, Dakota executes the computational simulation and any specified postprocessing and reads the resulting response metrics (derived quantities of interest). Dakota will repeat this process a number of times, depending on the method and the number of simulation evaluations it requires. Some methods, such as a design of experiments, may generate all the sample points in a single pass, while others such as optimization or adaptive UQ utilize feedback to generate new batches of parameter sets as they work to achieve their analysis goal. A key strength of the framework is the ability to easily switch among these different types of studies as a user explores different aspects of a computational model, from identifying influential parameters to calibrating them and from characterizing the effect of uncertainties to design optimization in their presence.

Over time, Dakota expanded to generate and perform these analyses on various types of multivariate surrogate models, e.g., polynomials, splines, neural networks, or Gaussian processes, to address computational cost and smoothness of parameter

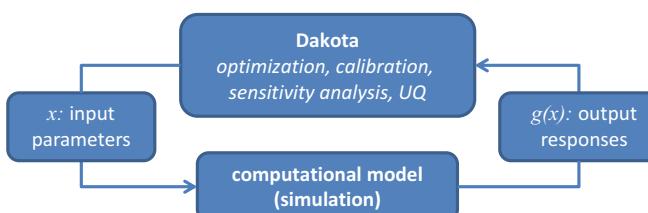


Fig. 49.1 Overall interaction between Dakota and a parameterized simulation

to response mappings. Uncertainty quantification methods evolved from Monte Carlo sampling through reliability and polynomial chaos to the plethora discussed in this chapter. Dakota can couple algorithms together in sequential or nested approaches, leading to such capabilities as hybrid optimization, surrogate-based optimization, optimization under uncertainty, reliability-based design optimization, and mixed epistemic-aleatory uncertainty analyses.

Throughout its existence, Dakota has supported parallel computing from the desktop to high-performance computers. It has an extensive set of multilevel parallel capabilities to handle parallel concurrency at the algorithm level and/or at the function evaluation level. These enable greater throughput of simulation runs in the course of iterative analyses.

A key goal of the Dakota project is to support development activities spanning from research to production, in support of Department of Energy (DOE) mission areas. The software is extensively used by analysts solving practical science and engineering optimization, sensitivity analysis, and uncertainty problems. These production users most commonly use well-established, tested methods such as gradient-based optimization or Monte Carlo sampling. However the development team strives to routinely deploy leading-edge research capabilities to address challenges analyzing expensive simulation codes. For example, recent work has focused on adaptive sparse grid methods to generate sample points for stochastic expansions such as in polynomial chaos and compressive sensing to optimally select bases used in such expansions. These methods help reduce the number of expensive simulations required to obtain a certain level of accuracy in an uncertainty estimate, for example, the estimate of the mean or 95th percentile of a stochastic response quantity.

1.2 Obtaining the Software

Dakota is open source and distributed under the GNU Lesser General Public License (LGPL): <http://dakota.sandia.gov/license.html>. The core source code is written in C++. Dakota distributions include required third-party libraries, written in C, C++, Fortran 77, and Fortran 90, with their own LGPL-compatible licenses. Given Dakota's research to development spectrum, release notes on the webpage indicate which capabilities are more recent and whether they are in a prototype stage.

More than 13,000 users worldwide have downloaded Dakota, and it has hundreds of users within the Department of Energy laboratory complex. It is supported on numerous Linux, Mac, and Windows operating system variants, including high-performance computing platforms. Software downloads (source and binary), system requirements, and installation information are available at <http://dakota.sandia.gov/download.html>. When feasible, users should download binary packages of Dakota. However, the website includes information on satisfying system prerequisites including compilers, MPI, Boost, and linear algebra libraries and compiling the source code with CMake.

1.3 Getting Started Resources

Extensive Dakota documentation is available at <http://dakota.sandia.gov/documentation.html>, including the following four primary manuals:

- **Users:** The Users Manual [1] addresses the various Dakota method categories such as parameter studies, design of experiments, optimization, and nonlinear least squares. Chapter 5 (Uncertainty Quantification Capabilities) addresses uncertainty quantification. This manual also covers surrogate and other types of models, how to define input variables and output responses, creating an interface between Dakota and the simulation code, and parallelism options.
- **Reference:** The Reference Manual [2] describes all valid keywords available to specify a Dakota study. For example, when using a genetic algorithm for optimization, one needs to define the population size, the mutation rate, the crossover rate, etc. This manual details the keywords and associated arguments that qualify them.
- **Theory:** The Theory Manual [3] supplements the Users Manual with advanced topics and more theoretical depth. For example, it covers surrogate-based optimization and optimization under uncertainty. In terms of uncertainty quantification, the Theory Manual has comprehensive descriptions of reliability methods, stochastic expansion methods, and evidence theory.
- **Developers:** The Developers Manual describes Dakota’s architecture, interfaces, development practices, and C++ class details.

To get started with Dakota, we recommend downloading and installing Dakota, reading Chapters 1 (Introduction) and 2 (Tutorial) of the Users Manual, and trying some of the examples (see the QuickStart page: <http://dakota.sandia.gov/quickstart.html>). The tutorial demonstrates Dakota use for a few simple studies, including a parameter study, an optimization study, and a Monte Carlo sampling study. After working through it, users will understand how to run Dakota, what inputs it expects, and what outputs it will produce.

The manuals are supplemented by examples included in the Dakota distribution in `examples/` and training materials from previous training sessions available at <http://dakota.sandia.gov/training/>. Dakota-related publications at <http://dakota.sandia.gov/publications.html> offer more in-depth discussion of many of the algorithmic approaches in Dakota.

1.4 Interfacing to Dakota

Dakota generally views the simulation model that it is interfaced to as a “black box.” In this mode, Dakota need not know details about nor have access to its internal state. Rather, when using the simulation in an optimization or uncertainty study, Dakota must be able to send input parameter values at which it would like

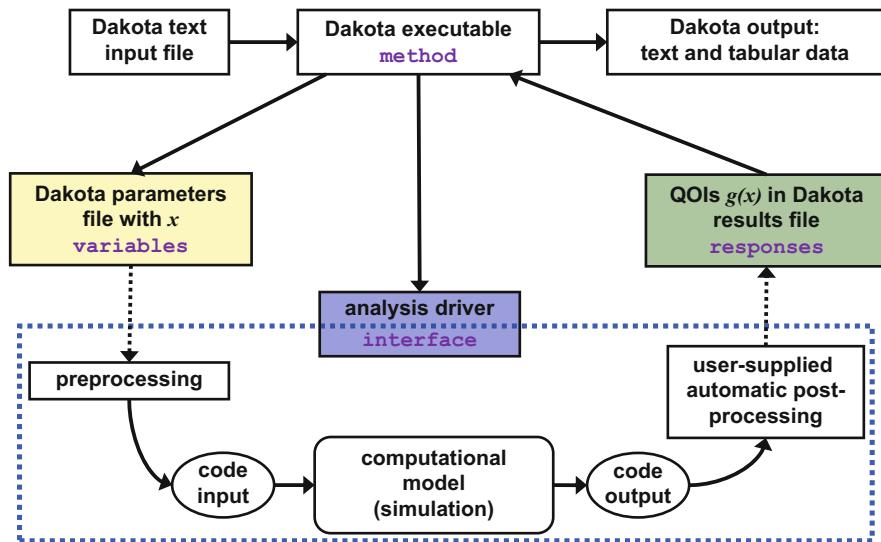


Fig. 49.2 Details of Dakota execution and information flow, including user-provided analysis driver

output responses, reliably execute the simulation in a non-interactive manner, and obtain the response data after the simulation has executed. While Dakota can be integrated into applications as a C++ library, it is most commonly used as separate executable, with an “analysis driver” script that manages the simulation workflow for each variable to response mapping. Dakota documentation and distributions include many examples of interfacing to simulations in this loosely coupled manner.

Figure 49.2 details an example of this information flow during Dakota execution. Dakota is invoked with a text-based input file, by running the command `dakota -i inputfile.in`, where `inputfile.in` is a text input file which can have any name. The input file tells Dakota what type of method will be executed and describes the variables, the responses, and the interface. Dakota then runs the user-specified method. For each model evaluation, Dakota generates a file containing parameter values and runs the analysis driver specified in the interface section of the input file. The analysis driver typically is a script that the user provides: it may be written in shell script, Python, Perl, or another language of the user’s choice. It can also be a program, such as a C++ executable program.

The analysis driver is required to accept a Dakota parameter file and preprocess as necessary to generate the proper parameterized input for the simulation code. Typically, this involves substituting the Dakota parameters into the simulation input file, which Dakota’s “dprepro” Perl-based preprocessor can help with. Once the simulation input is prepared for this sample point, the script runs the simulation and extracts the results of interest from the code output to send back to Dakota. This may be done by simple Unix commands such as searching and cutting particular

```

# Dakota Input File: textbook_uq_sampling.in
environment
    tabular_graphics_data
    tabular_graphics_file = 'textbook_uq_sampling.dat'

method
    sampling
        samples = 10
        seed = 98765  rng rnum2
        response_levels = 0.1 0.2 0.6
            0.1 0.2 0.6
            0.1 0.2 0.6
    sample_type lhs
    distribution cumulative

variables
    uniform_uncertain = 2
    lower_bounds = 0. 0.
    upper_bounds = 1. 1.
    descriptors = 'x1' 'x2'

interface
    fork asynch evaluation_concurrency = 5
    analysis_driver = 'text_book'

responses
    response_functions = 3
    no_gradients
    no_hessians

```

Fig. 49.3 Dakota input file for UQ example using Latin hypercube sampling

fields from an output file, or it may involve more significant manipulation of a binary output file. The script returns the responses to Dakota, and then Dakota starts another iteration of the simulation based on the method. At the end of all of the simulation evaluations, Dakota outputs the results both in text form and in a tabular data file.

Figure 49.3 provides an example of a basic Dakota input file. This example demonstrates an input file for sampling-based UQ. The method section indicates that the method is `sampling`, of type `lhs`, for Latin hypercube sampling (as opposed to pure Monte Carlo or importance sampling), and there will be 10 samples with a seed of 98765. The keyword `response_levels` indicates that Dakota should compute the cumulative distribution function (CDF) value at the specified response levels 0.1, 0.2, and 0.6, i.e., the probability that the simulation response is less than each of these values. Dakota will calculate sample statistics based on the ten samples; external postprocessing is not necessary.

In the variables section, there are two input variables x_1 and x_2 , characterized as uniformly distributed on the interval $[0, 1]$. The interface for this example is implemented in an analysis driver called “`text_book`,” an executable included with

Dakota that calculates the value of three algebraic “textbook” response_functions, polynomials that are a function of the input parameters x_1 and x_2 . Since this is a simple sampling study, gradients and Hessian information are not returned from the simulation to Dakota. The environment block at the top specifies output generation. For example, in this input file, the samples and the results are written to a tabular file called `textbook_uq_sampling.dat`. Another Dakota input block `model` is not shown in this example, implying the default single model. Models can be used for more complex variable to response mappings, including surrogate or nested models.

1.5 Characterizing Uncertain Variables

With the exception of deterministic moment matching and Bayesian calibration, which can characterize input variable uncertainty from data, Dakota largely focuses on forward uncertainty propagation. In doing so, it supports a number of characterizations of the uncertain input variables x .

Aleatory uncertain variable types include 11 standard distribution types: normal, log-normal, uniform, log-uniform, triangular, exponential, beta, gamma, Gumbel, Frechet, and Weibull, as well as an empirical continuous bin-based histogram. Five standard discrete distribution types are included: Poisson, binomial, negative binomial, geometric, and hypergeometric, as well as a point-wise histogram type supporting discrete integer-, string-, or real-valued variables. Correlations among these variable types can be specified as well. To characterize epistemic uncertainty, Dakota supports continuous and discrete interval types (e.g., to use in Dempster-Shafer theory of evidence approaches), as well as discrete sets of integer, string, and real values.

These various variable types will be used in subsequent sections which discuss the various Dakota uncertainty quantification methods.

2 Sampling Methods

A common method of propagating uncertainty through computational simulations is to assume particular distributions on the uncertain input variables, randomly sample from those distributions, run the model for each of the sampled inputs, and then assemble the set of output results to build up a distribution of the output values. This is Monte Carlo (MC) propagation of uncertainty. The output values can be analyzed to determine characteristics of the response, e.g., the maximum and minimum response values, the mean and the variance of the response, the probability of exceeding response thresholds, etc. For example, pure MC methods use a random number generator to draw samples from the specified distributions. The samples are unstructured; each “random draw” yields a realization unaware of its relation to samples drawn before or after it. Pure MC sampling produces unbiased estimates for the means, variances, and percentiles of the outputs.

In general, an advantage of random sampling is that the accuracy and computational burden is independent of the number of uncertain parameters, in contrast to many other methods in Dakota. It may be preferred when a sufficiently large number of samples are affordable. A disadvantage is that a large number of samples are required to accurately estimate the output statistics. The accuracy of the mean estimate obtained from a set of random samples exhibits $1/\sqrt{N}$ convergence, meaning that on average one needs to quadruple the number of sample points N to halve the error. To resolve small probabilities regarding predicted responses, hundreds of thousands of sample points may be required to obtain the desired accuracy.

A good alternative to pure random sampling is Latin hypercube sampling (LHS) [25, 34]. LHS is a stratified random sampling method where the distribution of each variable is divided into strata or bins. Each stratum is chosen to be equally probable, so that the strata are of equal length for uniform distributions but of unequal length for normal distributions (shorter strata near the center; longer strata near the tails). To create a total sample of size N using LHS, an individual sample value is chosen from each of the N equally probable strata for each input variable. This stratification approach forces better sampling across the entirety of each distribution, reducing clustering often seen in pure random sampling.

For multidimensional sampling, LHS also achieves good mixing of sample values from different input variables. For example, one would not want the sample in strata 1 from input A to be paired with the sample in strata 1 from input B. Instead, the pairing of the strata is performed in such a way [19] to generate multidimensional samples that are “well mixed” or randomized. These pairing algorithms allow the generation of independent inputs, but can also generate samples which honor a user-specified correlation structure. For example, one might want input C to have a correlation coefficient of 0.8 with input D.

Finally, LHS is more efficient than pure Monte Carlo; it requires fewer samples to achieve the same accuracy in statistics (e.g., standard error of the computed mean). LHS gives an estimator for a function mean that has lower variance than MC for any function having finite second moment [28, 31]. Further, the convergence behavior of LHS improves if the function is additively separable, meaning it can be decomposed into additive functions of the individual input parameters.

Dakota sampling techniques are selected using the `sampling` method selection. Currently, both traditional Monte Carlo (MC) and Latin hypercube sampling (LHS) are supported by specifying `sample_type` as `random` or `lhs`, respectively. The sampling method generates sets of samples according to the probability distributions of the uncertain variables and maps them into corresponding sets of response functions, where the number of samples is specified by the `samples` integer specification. The response function means, standard deviations, skewness, and kurtosis values are computed. The probability that a response is less than or greater than a particular value (cumulative distribution function, CDF or complementary cumulative distribution function, CCDF) may be computed for `response_levels` specifications. Similarly, the response levels may be computed by specifying either `probability_levels` or `reliability_levels`. The Dakota Reference Manual contains more information about the specification of these keywords.

Dakota's sampling implementation (MC and LHS) supports the full range of continuous and discrete distributions mentioned in the introduction. In addition, LHS accounts for correlations among the variables, which can be used to accommodate a user-supplied correlation matrix or to minimize correlation when a correlation matrix is not supplied.

2.1 Latin Hypercube Sampling Example

The input file previously shown in Fig. 49.3 demonstrates the use of Latin hypercube Monte Carlo sampling for assessing probability of failure as measured by specified response levels. The textbook example problem has two uniform variables on $[0, 1]$. The number of samples to perform is controlled with the `samples` specification, the type of sampling algorithm to use is controlled with the `sample_type` specification, the levels used for computing statistics on the response functions are specified with the `response_levels` input, and the `seed` specification controls the sequence of the pseudorandom numbers used by the sampling algorithms.

The response function statistics generated by Dakota when running this input file are shown in Fig. 49.4. The first two blocks of output specify the response sample moments and the confidence intervals for the mean and standard deviation.

Statistics based on 10 samples:

```
Moment-based statistics for each response function:
      Mean          Std Dev          Skewness          Kurtosis
response_fn_1  3.8383990322e-01  4.0281539886e-01  1.2404952971e+00  6.5529797327e-01
response_fn_2  7.4798705803e-02  3.4686110941e-01  4.5716015887e-01  -5.8418924529e-01
response_fn_3  7.0946176558e-02  3.4153246532e-01  5.2851897926e-01  -8.2527332042e-01

95% confidence intervals for each response function:
      LowerCI_Mean      UpperCI_Mean      LowerCI_StdDev      UpperCI_StdDev
response_fn_1  9.5683125821e-02  6.7199668063e-01  2.7707061315e-01  7.3538389383e-01
response_fn_2  -1.7333078422e-01  3.2292819583e-01  2.3858328290e-01  6.3323317325e-01
response_fn_3  -1.7337143113e-01  3.1526378424e-01  2.3491805390e-01  6.2350514636e-01

Probabilities for each response function:
Cumulative Distribution Function (CDF) for response_fn_1:
      Response Level      Probability Level      Reliability Index
      -----  -----
      1.0000000000e-01    3.0000000000e-01
      2.0000000000e-01    5.0000000000e-01
      6.0000000000e-01    7.0000000000e-01
Cumulative Distribution Function (CDF) for response_fn_2:
      Response Level      Probability Level      Reliability Index
      -----  -----
      1.0000000000e-01    5.0000000000e-01
      2.0000000000e-01    7.0000000000e-01
      6.0000000000e-01    9.0000000000e-01
Cumulative Distribution Function (CDF) for response_fn_3:
      Response Level      Probability Level      Reliability Index
      -----  -----
      1.0000000000e-01    6.0000000000e-01
      2.0000000000e-01    6.0000000000e-01
      6.0000000000e-01    9.0000000000e-01
```

Fig. 49.4 Dakota response function statistics from UQ sampling example

The last section of the output defines CDF pairs (distribution cumulative was specified) for the response functions by presenting the probability levels corresponding to the specified response levels (response_levels were set and the default compute probabilities was used). Alternatively, Dakota could have provided CCDF pairings, reliability levels corresponding to prescribed response levels, or response levels corresponding to prescribed probability or reliability levels.

In addition to obtaining statistical summary information of the type shown in Fig. 49.4, the results of LHS sampling also include correlations. These can be helpful in understanding the influence of parameters on responses for ranking or screening. Four types of correlations are output: simple and partial “raw” correlations and simple and partial “rank” correlations. Raw correlations refer to correlations performed on the values of the input and output data. Rank correlations are instead computed on the ranks of the data. Ranks are obtained by replacing the data by their ranked (ascending) values. The simple correlation is calculated as Pearson’s correlation coefficient.

Figure 49.5 shows an example of the correlation output provided by Dakota for the input file in Fig. 49.3. Note that the simple correlations among the inputs are nearly zero. This is due to the default “restricted pairing” method in the LHS routine which forces near-zero correlation among uncorrelated inputs. In this particular example, the first response function is strongly negatively correlated with both input variables, meaning that the function decreases as the input values increase from lower to upper bounds. In contrast, the second response function is positively correlated with the first variable.

```
Simple Correlation Matrix between input and output:
      x1          x2 response_fn_1 response_fn_2 response_fn_3
x1  1.00000e+00
x2 -7.22482e-02  1.00000e+00
response_fn_1 -7.04965e-01 -6.27351e-01  1.00000e+00
response_fn_2  8.61628e-01 -5.31298e-01 -2.60486e-01  1.00000e+00
response_fn_3 -5.83075e-01  8.33989e-01 -1.23374e-01 -8.92771e-01  1.00000e+00

Partial Correlation Matrix between input and output:
      response_fn_1 response_fn_2 response_fn_3
x1 -9.65994e-01  9.74285e-01 -9.49997e-01
x2 -9.58854e-01 -9.26578e-01  9.77252e-01

Simple Rank Correlation Matrix between input and output:
      x1          x2 response_fn_1 response_fn_2 response_fn_3
x1  1.00000e+00
x2 -6.66667e-02  1.00000e+00
response_fn_1 -6.60606e-01 -5.27273e-01  1.00000e+00
response_fn_2  8.18182e-01 -6.00000e-01 -2.36364e-01  1.00000e+00
response_fn_3 -6.24242e-01  7.93939e-01 -5.45455e-02 -9.27273e-01  1.00000e+00

Partial Rank Correlation Matrix between input and output:
      response_fn_1 response_fn_2 response_fn_3
x1 -8.20657e-01  9.74896e-01 -9.41760e-01
x2 -7.62704e-01 -9.50799e-01  9.65145e-01
```

Fig. 49.5 Correlation results using LHS sampling

2.2 Importance Sampling

Importance sampling preferentially samples important regions in the parameter domain, e.g., in or near a failure region or other user-defined region of interest, and then appropriately weights the samples to obtain an unbiased estimate of the failure probability [30]. In this approach, samples are generated from an “importance density,” which differs from the original probability density of the input variables. To be accurate, the importance density should be centered near the region of interest. For black-box simulations such as those commonly interfaced with Dakota, it is difficult to specify the importance density *a priori*; the user often does not know where the failure region lies, especially in a high-dimensional space [33].

One Dakota importance sampling method is based on ideas from reliability modeling. An initial Latin hypercube sample is generated and evaluated. The initial samples are augmented with samples from an importance density as follows. The variables are transformed to standard normal space. In the transformed space, the importance density is a set of normal densities centered around points which are in the failure region. This is similar in spirit to reliability methods (discussed later), in which importance sampling is centered around a most probable point (MPP).

This method is specified with the keyword `importance_sampling` with options as follows. The keyword `import` centers a sampling density at one of the initial LHS samples identified in the failure region. It then generates the importance samples, weights them by their probability of occurrence given the original density, and calculates the required probability. The `adapt_import` option is similar, but is performed iteratively (adapts) until the failure probability estimate converges. Finally, `mm_adapt_import` takes all of the failure region samples to build a multimodal sampling density. First, it uses a small number of samples around each of the initial samples in the failure region. These samples are allocated to the different points based on their relative probabilities of occurrence: more probable points get more samples. This early part of the approach is done to search for “representative” points. Once these are located, the multimodal sampling density is set and the multimodal sampling proceeds until convergence.

3 Stochastic Expansion Methods

Stochastic expansion UQ methods approximate the functional dependence of the simulation response on uncertain model parameters by expansion in a polynomial basis. The polynomials used are tailored to the characterization of the uncertain variables. Dakota includes two stochastic expansion methods: polynomial chaos expansion (PCE) and stochastic collocation (SC). Polynomial chaos expansion is based on a multidimensional orthogonal polynomial approximation, while stochastic collocation is based on a multidimensional interpolation polynomial approximation, both formed in terms of standardized random variables. A distinguishing feature of these two expansion methods is that the final solution is expressed as a functional mapping, not merely as a set of statistics, as in many nondeterministic

methodologies. This makes expansion techniques particularly attractive in multi-physics applications.

One advantage of PCE methods is their convergence rate. For smooth functions (i.e., analytic, infinitely differentiable) in ℓ_2 (i.e., possessing finite variance), exponential convergence rates can be obtained under order refinement for integrated statistical quantities of interest such as mean, variance, and probability. Another advantage of stochastic expansion methods is that the moments of the expansion (e.g., mean or variance of the response) can be written analytically, along with analytic formulations of the derivatives of these moments with respect to the uncertain variables. This property can be exploited in design optimization under uncertainty or epistemic uncertainty problems [11]. A disadvantage of polynomial chaos, as for all global approximation-based methods, is that they may not scale well to high dimensions. Recent research in adaptive refinement and sparse recovery methods strives to address this limitation.

Variance-based decomposition, which explains how output variance relates to the variance of each input variable, may also be calculated analytically from a stochastic expansion. This is a powerful capability for global sensitivity analysis, where influential input parameters can be identified and rank ordered. In particular, Dakota can generate Sobol' indices for main, interaction, and total effects. A larger value of the sensitivity index, S_i , means that the uncertainty in the input variable i has a larger effect on the variance of the output. Analytic dependence on expansion coefficients makes computing Sobol' indices essentially free. In contrast, estimating Sobol' indices with LHS can be extremely expensive, since repeat multidimensional integrations must be performed.

Stochastic expansion methods also support automated refinement using uniform or adaptive approaches. Adaptive approaches currently include dimension-adaptive refinement or generalized sparse grid refinement. In the former, anisotropic dimension weightings for tensor or sparse grids are updated based on either the total Sobol' indices or the spectral coefficient decay rates computed for the current iteration. For generalized sparse grid refinement, a greedy approach is used in which candidate index sets are evaluated for their impact on the statistical quantities of interest (QoI), the most influential sets are selected and used to generate additional candidates, and the index set frontier of a sparse grid is evolved in an unstructured and goal-oriented manner. Since this approach evaluates fine-grained effects from individual index sets, precision in QoI increments (e.g., from hierarchical interpolation) becomes increasingly important.

3.1 Polynomial Chaos Expansion

In PCE, the output response is modeled as a function of the input random variables using a carefully chosen set of polynomials. For example, PCE employs Hermite polynomials to model Gaussian random variables, as originally employed by Wiener [39]. Dakota implements the generalized PCE approach using the Wiener-Askey scheme [40], in which Hermite, Legendre, Laguerre, Jacobi, and generalized

Laguerre orthogonal polynomials are used for modeling the effect of continuous random variables described by normal, uniform, exponential, beta, and gamma probability distributions, respectively. These orthogonal polynomial selections are optimal for these distribution types since the inner product weighting function corresponds to the probability density functions for these continuous distributions.

Dakota allows the selection of three types of basis functions for PCE: Wiener, Askey, and Extended. The Wiener option uses a Hermite orthogonal polynomial basis for all random variables. It employs the Nataf variable transformation to transform non-normal, correlated distributions to normal, independent distributions. The Askey option, however, employs an extended basis of Hermite, Legendre, Laguerre, Jacobi, and generalized Laguerre orthogonal polynomials. The Extended option avoids the use of any nonlinear variable transformations by augmenting the Askey approach with numerically generated orthogonal polynomials for non-Askey probability density functions (e.g., bounded normal, Weibull, histogram, etc.).

To propagate input uncertainty through a model using PCE, Dakota performs the following steps: (1) input uncertainties are transformed to a set of uncorrelated random variables, (2) a basis such as Hermite polynomials is selected, and (3) the parameters of the functional approximation are determined. The general polynomial chaos expansion for a response g has the form

$$g(\mathbf{x}) \approx \sum_{j=0}^P \alpha_j \Psi_j(\mathbf{x}) \quad (49.1)$$

where each multivariate basis polynomial $\Psi_j(\mathbf{x})$ involves products of univariate polynomials that are tailored to the individual random variables. If a total-order polynomial basis is used (e.g., a total order of 2 would involve terms whose exponents are less than or equal to 2, such as x_1^2 , x_2^2 , and x_1x_2 but not $x_1^2x_2^2$), the total number of terms N in a polynomial chaos expansion of arbitrary order p for a response function involving n uncertain input variables is given by $(n + p)!/n!p!$. If on the other hand, an isotropic tensor product expansion is used with order p in each dimension, the number of terms is $(p + 1)^n$. If the order p of the expansion captures the behavior of the true function, polynomial chaos methods will give very accurate results for the output statistics of the response.

In nonintrusive PCE, as in Dakota, simulations are used as black boxes and the calculation of the expansion coefficients α_j for response metrics of interest is based on a set of simulation response evaluations. To calculate these response PCE coefficients, two primary classes of approaches are used: spectral projection and regression. The spectral projection approach projects the response against each basis function $\Psi_j(\mathbf{x})$ using inner products and employs the polynomial orthogonality properties to extract each coefficient. Each inner product involves a multidimensional integral over the support range of the weighting function, which can be evaluated numerically using sampling, tensor product quadrature, Smolyak sparse grid [29], or cubature [32] approaches. The regression approach finds a set of PCE coefficients α_j which best match a set of response values obtained from either

a design of computer experiments (“point collocation” [38]) or from subsampling a set of tensor Gauss points (“probabilistic collocation” [36]). Various methods can be used to solve the resulting linear system, including least squares methods for overdetermined systems and compressed sensing methods for under-determined systems. Details of these methods are documented in the Linear Regression section of the Dakota Theory Manual [3].

3.2 Stochastic Collocation

Stochastic collocation (SC) is closely related to PCE, but replaces projection or regression of spectral coefficients with interpolation. An SC interpolant takes the form

$$g(\mathbf{x}) \approx \sum_{j=1}^{N_p} r_j L_j(x) \quad (49.2)$$

where N_p is the number of unique collocation points in the multidimensional grid and r_j is the response at those points. Since the expansion coefficients are just the response values at the sampled points, the core of the method is formation of the multidimensional interpolation polynomials used as the basis. This basis can be formulated to be nodal or hierarchical and either value based (Lagrange interpolation polynomials) or gradient enhanced (Hermite interpolation polynomials). Point sets are generally defined using either tensor or sparse grid constructions, where the latter lead to sparse interpolants formed from linear combinations of low-order tensor interpolants. Relative to PCE, SC can be a good selection when either local error estimates (formed from hierarchical surpluses) or explicit interpolation of derivatives (using Hermite interpolants) is important.

3.3 PCE Example

In the simplest case, if one wants to use a polynomial chaos expansion in Dakota with isotropic tensor product expansion and five quadrature points for each input variable, the method section of the Dakota input would look like the following:

```
method
  polynomial_chaos
  quadrature_order = 5
```

If there were two variables, this would involve $5 \times 5 = 25$ function evaluations. If instead, one wanted to use a sparse grid at level 1, the method section of the Dakota input would look like the following:

```
method
  polynomial_chaos
  sparse_grid_level = 1
```

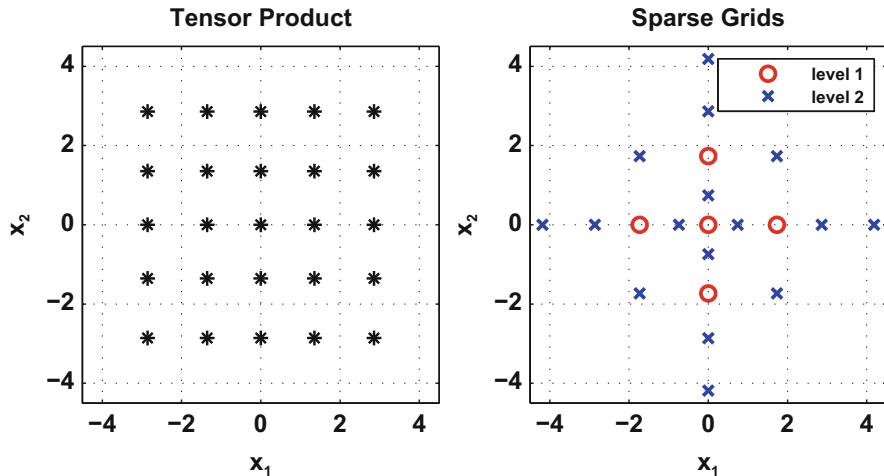


Fig. 49.6 Comparison of tensor product (*left*) and sparse grids (*right*) used in Dakota PCE construction

Figure 49.6 contrasts the resulting sample designs for two standard normal variables. The full tensor grid with quadrature order 5 is shown on the left. On the right, the level 1 sparse grid requires evaluation of the center point and three-point quadrature rules for each variable ($2n + 1$ total points for symmetric weakly-nested rules). The level 2 sparse grid requires 21 points, including those in the level 1 sparse grid due to the default use of nested rules (Genz-Keister for Hermite). The sparse grid points have a bigger range than the quadrature points in this particular example.

Using a PCE method in Dakota results in the moments of each response function, the covariance matrix for the response functions, and the local and global sensitivities, as shown in Fig. 49.7. The local sensitivities of each response function are calculated as the derivative of the response function evaluated at the means of the uncertain variables. The global sensitivities are calculated by variance-based decomposition, and the main, interaction, and total effects are reported. In this particular example, the responses are separable in the two inputs, so the interaction sensitivities are negligible. As with sampling, one may also request information about the probability density of the response functions in terms of histogram information and about the CDF values. This is shown in Fig. 49.8.

Dakota's stochastic expansions support many options. In lieu of listing them here, the reader is encouraged to explore the Dakota Reference Manual [2]. Table 49.1 summarizes key Dakota PCE features; similar options are available for SC.

4 Reliability Methods

Reliability UQ methods are a popular probabilistic approach which can be less computationally demanding than sampling, particularly when quantifying uncertainties with respect to a particular response or probability level. The prototypical

```

Statistics derived analytically from polynomial expansion:
Moment-based statistics for each response function:
      Mean          Std Dev        Skewness       Kurtosis
response_fn_1
  expansion: 2.0000000000e+01 3.6441734317e+01
  integration: 2.0000000000e+01 3.6441734317e+01 4.4464443753e+00 2.4003402163e+01
response_fn_2
  expansion: 1.0000000000e+00 1.5000000000e+00
  integration: 1.0000000000e+00 1.5000000000e+00 2.3703703704e+00 9.4814814815e+00
response_fn_3
  expansion: 1.0000000000e+00 1.5000000000e+00
  integration: 1.0000000000e+00 1.5000000000e+00 2.3703703704e+00 9.4814814815e+00

Covariance matrix for response functions:
[[ 1.3280000000e+03 3.2000000000e+01 3.2000000000e+01
   3.2000000000e+01 2.2500000000e+00 3.5735303605e-16
   3.2000000000e+01 3.5735303605e-16 2.2500000000e+00 ]]

Local sensitivities for each response function evaluated at uncertain variable means:
response_fn_1:
[ -4.0000000000e+00 -4.0000000000e+00 ]
response_fn_2:
[ 2.6888213878e-17 -5.0000000000e-01 ]
response_fn_3:
[ -5.0000000000e-01 -1.3877787808e-16 ]

Global sensitivity indices for each response function:
response_fn_1 Sobol' indices:
      Main          Total
5.0000000000e-01 5.0000000000e-01 TF1ln
5.0000000000e-01 5.0000000000e-01 TF2ln
      Interaction
3.0417793027e-32 TF1ln TF2ln
response_fn_2 Sobol' indices:
      Main          Total
8.888888889e-01 8.888888889e-01 TF1ln
1.111111111e-01 1.111111111e-01 TF2ln
      Interaction
4.736509654ie-32 TF1ln TF2ln
response_fn_3 Sobol' indices:
      Main          Total
1.111111111e-01 1.111111111e-01 TF1ln
8.888888889e-01 8.888888889e-01 TF2ln
      Interaction
5.0301248072e-32 TF1ln TF2ln

```

Fig. 49.7 Statistical output generated from a polynomial chaos expansion in Dakota

application of reliability methods is to estimate failure probabilities, i.e., the probability that an output response exceeds or falls below a threshold value. These approaches preferentially evaluate points around the area of interest, namely, the failure region, and may use successive approximations to refine failure estimates. Reliability methods can generally be used to compute response mean, response standard deviation, and CDF/CCDF values; however, their strength is they can be more efficient than Monte Carlo sampling when computing statistics in the tails of the response distributions (low probability events). The interested reader is encouraged to start with [15] for an excellent overview of local reliability methods.

```

Probability Density Function (PDF) histograms for each response function:
PDF for response_fn_1:
      Bin Lower      Bin Upper      Density Value
      -----      -----
1.5102874507e-10 4.0000000000e-01 3.4700000013e-01
4.0000000000e-01 5.0000000000e-01 1.4200000000e-01
5.0000000000e-01 5.5000000000e-01 1.9000000000e-01
5.5000000000e-01 6.0000000000e-01 1.5000000000e-01
6.0000000000e-01 6.5000000000e-01 1.3800000000e-01
6.5000000000e-01 7.0000000000e-01 1.3200000000e-01
7.0000000000e-01 7.5000000000e-01 1.3400000000e-01
7.5000000000e-01 8.0000000000e-01 1.2200000000e-01
8.0000000000e-01 5.5911038456e+02 1.4395218542e-03
...
Level mappings for each response function:
Cumulative Distribution Function (CDF) for response_fn_1:
      Response Level      Probability Level      Reliability Index      General Rel Index
      -----      -----
4.0000000000e-01 1.3880000000e-01
5.0000000000e-01 1.5300000000e-01
5.5000000000e-01 1.6250000000e-01
6.0000000000e-01 1.7000000000e-01
6.5000000000e-01 1.7690000000e-01
7.0000000000e-01 1.8350000000e-01
7.5000000000e-01 1.9020000000e-01
8.0000000000e-01 1.9630000000e-01

```

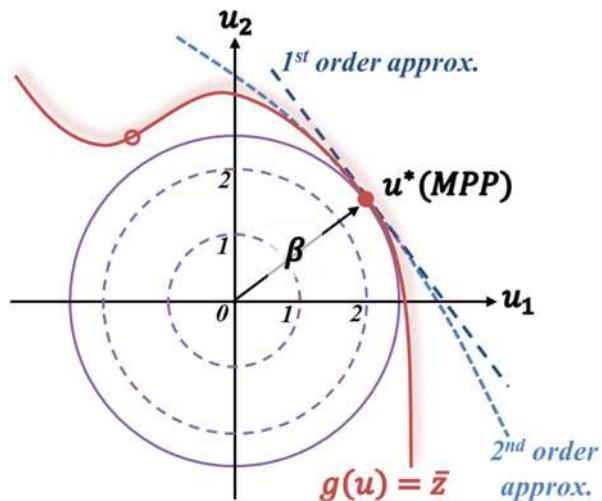
Fig. 49.8 PDF and CDF information from results of a polynomial chaos expansion in Dakota

Table 49.1 Major options available in Dakota's PCE capability

Capability	Option
Orthogonal polynomial basis	Askey, Wiener, or extended
Spectral projection approach to estimate PCE coefficients	Sampling, tensor product quadrature, Smolyak sparse grid, or cubature
Regression approach to estimate PCE coefficients	Least squares, basis pursuit, basis pursuit denoising, orthogonal matching pursuit (OMP), least absolute shrinkage (LASSO), least angle regression (LARS), and orthogonal least interpolation
Automated refinement	Uniform, anisotropic dimension adaptivity (Sobol, decay), or generalized sparse grid
Variance-based decomposition	Limit reported indices using interaction order and drop tolerance

More specifically, reliability methods address the fundamental question: “Given a set of uncertain input variables x with a probability density $\rho(x)$, and a scalar response function, $g(x)$, what is the probability that the response function is below (or above) a certain level, \bar{z} ? ” The former can be written as $P[g(x) \leq \bar{z}] = F_g(\bar{z})$ where $F_g(\bar{z})$ is the cumulative distribution function (CDF) of the uncertain response $g(x)$, evaluated at the target response level. The latter can be written as $P[g(x) > \bar{z}]$ and defines the complementary cumulative distribution function (CCDF). This is depicted in Fig. 49.9, where a contour of the response function corresponding to

Fig. 49.9 Reliability methods: failure boundary $g(u) = \bar{z}$ in uncorrelated standard normal probability space, with most probable point of failure indicated



the level of interest \bar{z} is shown in solid red. The probability calculation involves a probability-weighted multidimensional integral of the implicit simulation mapping $g(x)$ over the failure domain, for example, outside the red curve, as indicated by the shading.

Figure 49.9 depicts a standard normal probability space u , in which probability calculations can be more tractable. In Dakota, the Nataf transformation [24] can be used to automatically map the user-specified uncertain variables, x , with probability density function, $\rho(x_1, x_2)$, which can be nonnormal and correlated, to a space of independent Gaussian random variables, (u_1, u_2) , each with mean zero and unit variance. In this transformed space, probability contours are circular, instead of the rotated hyper-ellipses of correlated variables, or arbitrary level curves of a general joint probability density function.

In u -space, the multidimensional integrals that define the failure probability can be approximated by simple functions of a single parameter, β , called the reliability index. β is the minimum Euclidean distance from the origin in the transformed space to the failure boundary, also known as the limit state surface. This closest point is known as the most probable point (MPP) of failure. This nomenclature is due to the origin of these methods within the disciplines of structural safety and reliability; however, the methodology is equally applicable for computation of probabilities unrelated to failure.

Within the class of reliability methods, there are local and global algorithms. The most well-known are local methods, which locate a single MPP using a local, often gradient-based, optimization search method and then utilize an approximation centered around this point (see the notional linear and quadratic approximations in Fig. 49.9). In contrast, global methods generate approximations over the full random variable space and can find multiple MPPs if they exist, for example, the second most likely point, shown as an open red circle in Fig. 49.9. In both cases, a primary

strength of the methods lies in the fact that the computational expense is generally unrelated to the probability level; the cost of evaluating a probability in the far tails is no more than that of evaluating near the means.

4.1 Local Reliability Methods

The Dakota Theory Manual [3] provides the algorithmic details for the local reliability methods as well as references to related research activities. Local methods include first- and second-order versions of the mean value method (MVFOSM and MVSOSM) and a variety of most probable point (MPP) search methods, including the advanced mean value method (AMV and AMV²), the iterated advanced mean value method (AMV+ and AMV²+), the two-point adaptive nonlinearity approximation method (TANA-3), and the traditional first-order and second-order reliability methods (FORM and SORM) [15]. The MPP search methods may be used in forward (reliability index approach (RIA)) or inverse (performance measure approach (PMA)) modes, as dictated by the type of level mappings. Each of the MPP search techniques solve local optimization problems in order to locate the MPP, which is then used as the point about which approximate probabilities are integrated (using first- or second-order integrations in combination with refinements based on importance sampling).

Given variants of limit state approximation, approximation order, integration approach, and MPP search type, the number of algorithmic combinations is significant. Table 49.2 provides a succinct mapping for some of these combinations to common method names from the reliability literature.

4.2 Global Reliability Methods

Global reliability methods are designed to handle nonsmooth, multimodal, and highly nonlinear failure surfaces by creating global approximations and adaptively refining them in the vicinity of a particular response threshold. Three variants are available: EGRA, GPAIS, and POFDarts.

Table 49.2 Mapping from Dakota options to standard reliability methods

MPP search	Order of approximation and integration	
	First order	Second order
None	MVFOSM	MVSOSM
x_taylor_mean	AMV	AMV ²
u_taylor_mean	u-space AMV	u-space AMV ²
x_taylor_mpp	AMV+	AMV ² +
u_taylor_mpp	u-space AMV+	u-space AMV ² +
x_two_point	TANA	
u_two_point	u-space TANA	
no_approx	FORM	SORM

4.2.1 EGRA

As the name implies, efficient global reliability analysis (EGRA) [6] has its roots in efficient global optimization (EGO) [18, 21]. The main idea in EGO-type optimization methods is that a global approximation is made of the underlying function. This Gaussian process model approximation is used to guide the search by finding points which maximize the expected improvement function (EIF). The EIF is used to select the location at which a new training point should be added to the Gaussian process model by maximizing the amount of improvement in the objective function that can be expected by adding that point. A point could be expected to produce an improvement in the objective function if its predicted value is better than the current best solution, or if the uncertainty in its prediction is such that the probability of it producing a better solution is high. Because the uncertainty is higher in regions of the design space with fewer observations, this provides a balance between exploiting areas of the design space that predict good solutions and exploring areas where more information is needed.

In the case of reliability analysis, specified with `global_reliability`, the goal of optimizing an objective function is replaced with the goal of resolving the failure boundary. In this case, the Gaussian process model is adaptively refined to accurately resolve a particular contour of a response function using a variation of the EIF known as the expected feasibility function. Then, the failure probabilities are estimated on this adaptively refined Gaussian process model using multimodal adaptive importance sampling.

4.2.2 GPAIS

Gaussian process adaptive importance sampling (GPAIS) [7] starts with an initial set of LHS samples and adds samples one at a time, with the goal of adaptively improving the estimate of the ideal importance density during the process. The approach uses a mixture of component densities. An iterative process is used to construct the sequence of improving component densities. The GPs are not used to directly calculate the failure probability; they are only used to approximate the importance density. Thus, GPAIS overcomes limitations involving using a potentially inaccurate surrogate model directly in importance sampling calculations.

This method is specified with the keyword `gpaiss`. There are three main controls which govern the behavior of the algorithm. `samples` specifies the initial number of Latin hypercube samples which are used to create the initial Gaussian process surrogate. `emulator_samples` specifies the number of samples taken on the latest Gaussian process model each iteration of the algorithm. The third control is `max_iterations`, which controls the number of iterations of the algorithm after the initial LHS samples.

4.2.3 Adaptive Sampling with Dart Throwing

Probability of failure darts, specified in Dakota as `pof_darts`, estimates the probability of failure based on random sphere packing. Random spheres are sampled from the domain with the constraint that each new sphere center has to be outside prior disks [8]. The radius of each sphere is chosen such that the entire sphere lies

either in the failure or the non-failure region. This radius depends on the function evaluation at the disk center, the failure threshold, and an estimate of the function gradient at the disk center.

After exhausting the sampling budget specified by `samples`, which is the number of spheres per failure threshold, the domain is decomposed into two regions. These regions correspond to failure and non-failure, each represented by the union of the spheres of each type. After sphere construction, a surrogate model is built and extensively sampled to estimate the probability of failure for each threshold.

The surrogate model can either be a global surrogate or an ensemble of local surrogates. The local option leverages a piecewise surrogate model, called a Voronoi piecewise surrogate (VPS). The VPS model is used to construct high-dimensional surrogates fitting a few data points, allowing the user to estimate high-dimensional function values with cheap polynomial evaluations. The core idea in the VPS is to naturally decompose a high-dimensional domain into its Voronoi tessellation, with the few given data points as Voronoi seeds. Each Voronoi cell then builds its own surrogate. The surrogate used is a polynomial that passes through the cell's seed and optimally fits its local neighbors by minimizing their error in the least squares sense. Therefore, a function evaluation of a new point requires finding the closest seed and using its particular polynomial coefficients to find the function value estimate.

4.3 Local Reliability Example

Figure 49.10 shows the Dakota input file for an example problem that demonstrates the simplest reliability method, called the mean value method (also referred to as the

```
# Dakota Input File: textbook_uq_meanvalue.in
method
    local_reliability

interface
    fork asynch
        analysis_driver = 'text_book'

variables
    lognormal_uncertain = 2
        means                  = 1.          1.
        std_deviations         = 0.5       0.5
        descriptors           = 'TF1ln' 'TF2ln'

responses
    response_functions = 3
    numerical_gradients
        method_source dakota
        interval_type central
        fd_gradient_step_size = 1.e-4
    no_hessians
```

Fig. 49.10 Mean value reliability method: the Dakota input file

```
MV Statistics for response_fn_1:
  Approximate Mean Response           =  0.0000000000e+00
  Approximate Standard Deviation of Response =  0.0000000000e+00
  Importance Factors not available.

MV Statistics for response_fn_2:
  Approximate Mean Response           =  5.0000000000e-01
  Approximate Standard Deviation of Response =  1.0307764064e+00
  Importance Factor for variable TF1ln      =  9.4117647059e-01
  Importance Factor for variable TF2ln      =  5.8823529412e-02

MV Statistics for response_fn_3:
  Approximate Mean Response           =  5.0000000000e-01
  Approximate Standard Deviation of Response =  1.0307764064e+00
  Importance Factor for variable TF1ln      =  5.8823529412e-02
  Importance Factor for variable TF2ln      =  9.4117647059e-01
```

Fig. 49.11 Results of the mean value method on the textbook function

mean value first-order, second-moment, or MVFOSM method). It is specified with method keyword `local_reliability`. This method calculates the mean and variance of the response function based on information about the mean and variance of the inputs and gradient information at the mean of the inputs. The mean value method is very inexpensive (only five runs were required for the textbook function based on a central finite difference for two inputs), but can be quite inaccurate, especially for nonlinear problems and/or problems with uncertain inputs that are significantly non-normal. More detail on the mean value method can be found in the Local Reliability Methods section of the Dakota Theory Manual [3].

Example output from the mean value method is displayed in Fig. 49.11. The textbook objective function is given by $f = (x_1 - 1)^4 + (x_2 - 1)^4$. Since the mean of both inputs is 1, the mean value of the output for response 1 is zero. However, the mean values of the constraints, given by $c_1 = x_1^2 - \frac{x_2}{2} \leq 0$ and $c_2 = x_2^2 - \frac{x_1}{2} \leq 0$, are both 0.5. The mean value results indicate that variable x_1 is more important in constraint 1 while x_2 is more important in constraint 2.

5 Epistemic Methods

Uncertainty quantification is often used for assessing the risk, reliability, and safety of engineered systems. In these contexts, uncertainty is increasingly separated into two categories for analysis purposes: aleatory and epistemic uncertainty [16, 27]. Aleatory uncertainty is also referred to as variability, irreducible or inherent uncertainty, or uncertainty due to chance. Examples of aleatory uncertainty include the height of individuals in a population, or the temperature in a processing environment. Aleatory uncertainty is usually modeled with probability distributions. In contrast, epistemic uncertainty refers to lack of knowledge or lack of information

about a particular aspect of the simulation model, including the system and environment being modeled. An increase in knowledge or information relating to epistemic uncertainty will lead to a reduction in the predicted uncertainty of the system response or performance. Epistemic uncertainty is referred to as subjective, reducible, or lack of knowledge uncertainty. Examples of epistemic uncertainty include little or no experimental data for a fixed but unknown physical parameter, incomplete understanding of complex physical phenomena, uncertainty about the correct model form to use, etc.

There are many approaches which have been developed to model epistemic uncertainty, including fuzzy set theory, possibility theory, and evidence theory. There are three approaches to treat epistemic uncertainties in Dakota: interval analysis, evidence theory, and subjective probability. In the case of subjective probability, the same probabilistic methods for sampling, reliability, or stochastic expansion may be used, albeit with a different subjective interpretation of the statistical results. We describe the interval analysis and evidence theory capabilities in the following sections.

5.1 Interval Methods for Epistemic Analysis

In interval analysis, one assumes only that the value of each epistemic uncertain variable lies somewhere within a specified interval. It is not assumed that the value has a uniform probability over the interval. Instead, the interpretation is that any value within the interval is a possible value or a potential realization of that variable. In interval analysis, the uncertainty quantification problem translates to determining bounds on the output (defining the output interval), given interval bounds on the inputs. Again, any output response that falls within the output interval is a possible output with no frequency information assigned to it.

Dakota supports interval analysis using either global (`global_interval_est`) or local (`local_interval_est`) approaches. The global approach uses either optimization or sampling to estimate the bounds, with options for acceleration with surrogates. Specifying the keyword `lhs` performs Latin hypercube sampling and takes the minimum and maximum of the samples as the bounds (no optimization is performed), while optimization approaches are specified via `ego`, `sbo`, or `ea`. In the case of `ego`, the efficient global optimization method adaptively refines a Gaussian process surrogate to calculate bounds. The latter two (`sbo` for surrogate-based optimization and `ea` for evolutionary algorithm) support mixed-integer nonlinear programming (MINLP), enabling the inclusion of discrete epistemic parameters such as model form selections. If the problem is continuous and is not expected to contain multiple extrema, then one can use local gradient-based optimization methods (`sqp` for sequential quadratic programming or `nip` for a nonlinear interior point) to calculate epistemic bounds. Local methods may scale better with the number of epistemic variables, though care must be exercised when potentially working with a multimodal response.

5.2 Dempster-Shafer Theory of Evidence

Evidence theory, also referred to as Dempster-Shafer theory or the theory of random sets [27], has found favor at Sandia for modeling epistemic uncertainty, in part because evidence theory is a generalization of probability theory. In this framework, there are two complementary measures of uncertainty: belief and plausibility. Together, belief and plausibility can be thought of as defining lower and upper bounds, respectively, on probability values consistent with the evidence.

In Dempster-Shafer evidence theory, the uncertain input variables are modeled as sets of intervals. The user assigns a basic probability assignment (BPA) to each interval, indicating how likely it is that the uncertain input falls within the interval. The BPAs for a particular uncertain input variable must sum to one and may be overlapping, contiguous, or have gaps. In Dakota, an interval uncertain variable is specified as `interval_uncertain`. When one defines an interval type variable in Dakota, it is also necessary to specify the number of intervals defined for each variable with `num_intervals` as well the basic probability assignments per interval, `interval_probs`, and the associated bounds per each interval, `interval_bounds`.

Once the intervals, the BPAs, and the interval bounds are defined, the user can run an epistemic analysis by specifying the method as either `global_evidence` or `local_evidence` in the Dakota input file. Epistemic analysis is then performed using either global or local methods, using the same algorithm approaches described previously for interval analysis. The primary difference from interval analysis is the number of solves that must be performed, as each unique input BPA bound defines new cells requiring separate minimum and maximum response values. This ensemble of cell minima and maxima are used to define cumulative distribution functions on belief and plausibility.

6 Advanced Capabilities

The following sections describe advanced capabilities that build on the core foundation of sampling, stochastic expansion, reliability, and epistemic methods. Dakota allows for flexible combination of core components to create “meta-algorithms” which may use facilities for nesting, recasting, surrogate modeling, and multilevel parallel scheduling.

6.1 Mixed Aleatory-Epistemic UQ

Mixed UQ approaches employ Dakota nested models to embed one uncertainty quantification (UQ) within another, as depicted in Fig. 49.12. The outer level UQ is commonly linked to epistemic uncertainties and the inner UQ is commonly linked to aleatory uncertainties. The outer level generates sets of realizations of the epistemic

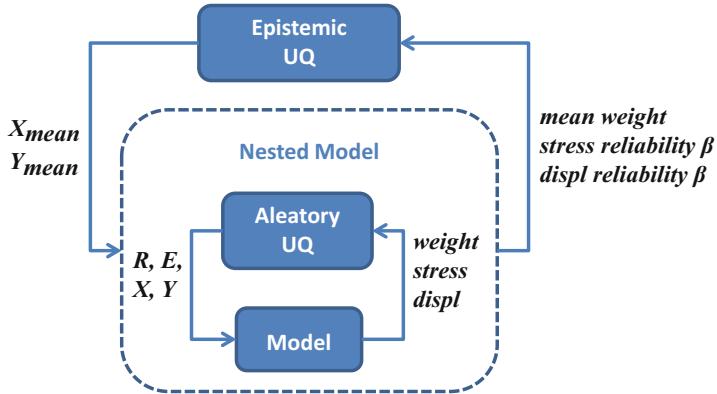


Fig. 49.12 Dakota nested model for mixed UQ analysis

parameters, and each set of these epistemic parameters is used within a separate inner loop probabilistic analysis over the aleatory random variables. In this manner, ensembles of aleatory statistics are generated, one set for each realization of the epistemic parameters. Each level may flexibly use any relevant Dakota UQ method.

Dakota supports three approaches for mixed UQ: interval-valued probability (IVP), Dempster-Shafer theory of evidence (DSTE), and second-order probability (SOP). These three approaches differ in how they treat the outer loop epistemic variables: they are treated as intervals in IVP, as belief structures in DSTE, and as subjective probability distributions in SOP. This yields a spectrum of assumed epistemic structure, from strongest assumptions in SOP to weakest in IVP. IVP (also known as probability bounds analysis [5, 12, 22]), employs an outer loop interval estimation in combination with an aleatory inner loop to generate bounds on aleatory statistics. Sampling-based outer loop approaches yield an ensemble of cumulative distribution functions or CCDFs, one CDF/CCDF result for each aleatory analysis. Plotting an entire ensemble of CDFs or CCDFs in a “horsetail” plot allows one to visualize the upper and lower bounds on the family of distributions (see Fig. 49.13). Given that the ensemble arises from realizations of the epistemic uncertain variables, the interpretation is that each CDF/CCDF instance has no relative probability of occurrence, only that each instance is possible. For prescribed response levels on the CDF/CCDF, an interval on the probability is computed based on the bounds of the ensemble at that level and vice versa for prescribed probability levels. For example, in Fig. 49.13, intervals on the response levels corresponding to probabilities of 0.05 and 0.95 are emphasized. Once again, this interval-valued statistic is interpreted simply as a possible range, where the statistic could take any of the possible values in the range.

The example input file shown in Fig. 49.14 is complex compared to previous examples, as one must specify two entire UQ problems, together with a nested model to coordinate them. The depiction in Fig. 49.12 should help considerably in understanding it. Here, the outer epistemic variables X_{mean} , Y_{mean} are characterized

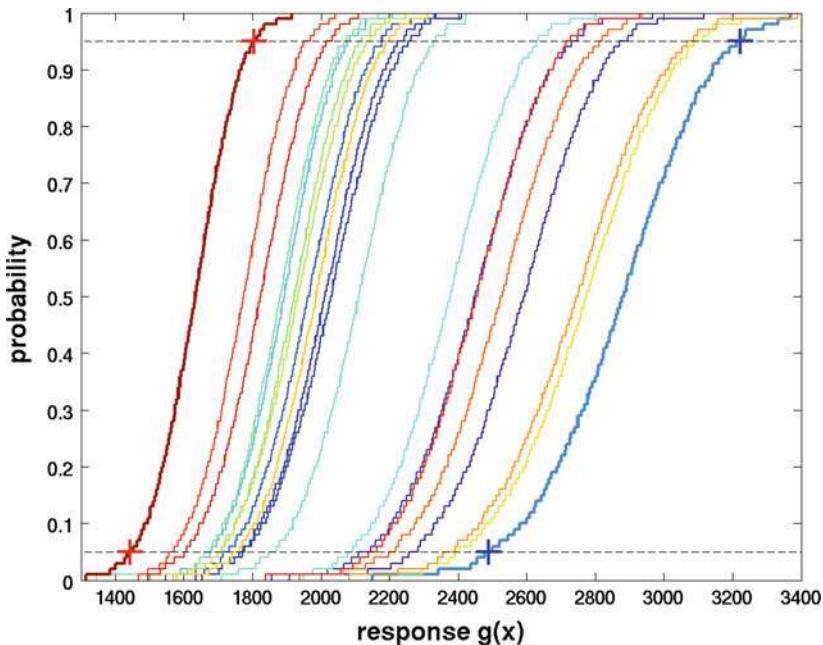


Fig. 49.13 Example CDF ensemble. Commonly referred to as a “horsetail” plot

as intervals. Each sample generated from these intervals will define the mean for uncertain variables X and Y employed in an entire inner level reliability analysis. The analysis problem studied is an algebraic model of a cantilevered beam, subject to loading. The inner reliability analysis holds the width w and thickness t fixed, while assessing the effect of uncertainty in residual stress R , Young’s modulus E , horizontal load X , and vertical load Y on responses weight, stress, and displacement.

Figure 49.15 shows excerpts from the resulting output. In this particular example, the outer loop generates 50 possible realizations of epistemic variables, each of which are communicated to the inner loop to calculate statistics such as the mean weight and cumulative distribution function for the stress and displacement reliability indices. Thus, the outer loop has 50 possible values for the mean weight but there is no distribution structure on these 50 samples. So, only the minimum and maximum values are reported. Similarly, the minimum and maximum values of the CCDF for the stress and displacement reliability indices are reported.

6.2 Multifidelity UQ

Multifidelity UQ approaches use a predictive low-fidelity model to reduce the number of high-fidelity model evaluations required to compute high-fidelity statistics

```

# Dakota Input File: cantilever_uq_sop_rel.in
environment
  method_pointer = 'EPISTEMIC'

method
  id_method = 'EPISTEMIC'
  model_pointer = 'EPIST_M'
  sampling
    samples = 50 seed = 12347

model
  id_model = 'EPIST_M'
  nested
    variables_pointer = 'EPIST_V'
    sub_method_pointer = 'ALEATORY'
    responses_pointer = 'EPIST_R'
    primary_variable_mapping = 'X' 'Y'
    secondary_variable_mapping = 'mean' 'mean'
    primary_response_mapping = 1. 0. 0. 0. 0. 0. 0. 0.
                                0. 0. 0. 0. 1. 0. 0. 0.
                                0. 0. 0. 0. 0. 0. 0. 1.

variables
  id_variables = 'EPIST_V'
  continuous_interval_uncertain =2
  num_intervals = 1 1
  interval_probs = 1.0 1.0
  lower_bounds = 400. 800.
  upper_bounds = 600. 1200.
  descriptors 'X_mean' 'Y_mean'

responses
  id_responses = 'EPIST_R'
  response_functions = 3
  response_descriptors = 'mean_wt' 'ccdf_beta_s' 'ccdf_beta_d'
  no_gradients
  no_hessians

method
  id_method = 'ALEATORY'
  model_pointer = 'ALEAT_M'
  local_reliability
    mpp_search no_approx
    num_response_levels = 0 1 1
    response_levels = 0.0 0.0
    compute reliabilities
    complementary distribution

model
  id_model = 'ALEAT_M'
  single
    variables_pointer = 'ALEAT_V'
    interface_pointer = 'ALEAT_I'
    responses_pointer = 'ALEAT_R'

```

Fig. 49.14 (continued)

Fig. 49.14 Dakota input file for the interval-valued probability example

```

variables
    id_variables = 'ALEAT_V'
    continuous_design = 2
        initial_point    2.4522 3.8826
        descriptors 'w''t'
    normal_uncertain = 4
        means            = 40000. 29.E+6 500. 1000.
        std_deviations   = 2000. 1.45E+6 100. 100.
        descriptors      = 'R' 'E' 'X' 'Y'

interface
    id_interface = 'ALEAT_I'
    direct
        analysis_driver = 'cantilever'
        deactivate evaluation_cache restart_file

responses
    id_responses = 'ALEAT_R'
    response_functions = 3
    response_descriptors = 'weight' 'stress' 'displ'
    analytic_gradients
    no_hessians

Statistics based on 50 samples:

Min and Max values for each response function:
mean_wt: Min = 9.5209117200e+00 Max = 9.5209117200e+00
ccdf_beta_s: Min = 1.7627715524e+00 Max = 4.2949468386e+00
ccdf_beta_d: Min = 2.0125192955e+00 Max = 3.9385559339e+00

```

Fig. 49.15 Interval-valued statistics for cantilever beam reliability indices

to a specified precision. When a low-fidelity model captures useful trends of the high-fidelity model, the model discrepancy may have either lower complexity, lower variance, or greater sparsity, requiring less computational effort to resolve its functional form than that required for the original high-fidelity model [26]. In the case of multifidelity polynomial chaos expansions, this reduction in computational effort can often be linked to a more rapid decay in the coefficient spectrum of the model discrepancy relative to the decay of the high-fidelity coefficient spectrum.

Dakota capabilities for multifidelity UQ are currently implemented using stochastic expansion methods. To enable the goal of the low-fidelity model informing the high-fidelity statistics, an expansion is formed for the model discrepancy (the difference between high- and low-fidelity responses). An additive or multiplicative discrepancy may be used:

$$A(\mathbf{x}) = g_{hi}(\mathbf{x}) - g_{lo}(\mathbf{x}) \quad (49.3)$$

$$B(\mathbf{x}) = \frac{g_{hi}(\mathbf{x})}{g_{lo}(\mathbf{x})} \quad (49.4)$$

Approximating the high-fidelity response functions using approximations of these discrepancy functions then involves

$$\hat{g}_{hi_A}(\mathbf{x}) = g_{lo}(\mathbf{x}) + \hat{A}(\mathbf{x}) \quad (49.5)$$

$$\hat{g}_{hi_B}(\mathbf{x}) = g_{lo}(\mathbf{x}) \hat{B}(\mathbf{x}) \quad (49.6)$$

where $\hat{A}(\mathbf{x})$ and $\hat{B}(\mathbf{x})$ are stochastic expansion approximations to the exact correction functions:

$$A(\mathbf{x}) \approx \hat{A}(\mathbf{x}) = \sum_{j=0}^{P_{hi}} \alpha_j \Psi_j(\mathbf{x}) \quad \text{or} \quad \sum_{j=1}^{N_{hi}} a_j \mathbf{L}_j(\mathbf{x}) \quad (49.7)$$

$$B(\mathbf{x}) \approx \hat{B}(\mathbf{x}) = \sum_{j=0}^{P_{hi}} \beta_j \Psi_j(\mathbf{x}) \quad \text{or} \quad \sum_{j=1}^{N_{hi}} b_j \mathbf{L}_j(\mathbf{x}) \quad (49.8)$$

where α_j and β_j are the spectral coefficients for a polynomial chaos expansion and a_j and b_j are the interpolation coefficients for stochastic collocation.

In addition to the stochastic expansion model for the discrepancy term, Dakota also forms a stochastic expansion for the low-fidelity surrogate model, where the intent is for the level of stochastic resolution to be higher than that required to resolve the discrepancy ($P_{lo} \gg P_{hi}$ or $N_{lo} \gg N_{hi}$), such that the low-fidelity expansion accurately captures the primary trends of the response using less expensive simulations:

$$g_{lo}(\mathbf{x}) \approx \sum_{j=0}^{P_{lo}} \gamma_j \Psi_j(\mathbf{x}) \quad \text{or} \quad \sum_{j=1}^{N_{lo}} r_{lo,j} \mathbf{L}_j(\mathbf{x}) \quad (49.9)$$

This separation in resolution can be enforced statically through order/level selections or automatically through adaptive refinement. The use of adaptive refinement strategies is conceptually appealing: we wish to rely on the low-fidelity model for parameter ranges where it is predictive and focus effort on refining the discrepancy model in regions where the low-fidelity model starts to break down. A multifidelity generalized sparse grid algorithm has been developed for this purpose [26]. After both expansions are formed, the two expansions are combined (added or multiplied) into a new expansion that approximates the high-fidelity model, from which the final set of statistics are generated. For more detail on the corrections and multifidelity approaches in general, see the Dakota Theory Manual [3].

Figure 49.16 shows an example of the Dakota input specification for an additive correction approach. Note that the two sparse grid levels refer to the sparse grid level used to construct the discrepancy term (in this case, 1) and the level used to approximate the low-fidelity model (in this case, 3). The model is defined as a hierarchical surrogate model, with pointers to the low- and high-fidelity models. The

Fig. 49.16 Multifidelity UQ example with additive discrepancy term: the Dakota input file

```

environment,
graphics tabular_graphics_data
method_pointer = 'SBUQ'

method,
id_method = 'SBUQ'
model_pointer = 'SURROGATE'
polynomial_chaos
    sparse_grid_level = 1 3
    variance_based_decomp

model,
id_model = 'SURROGATE'
surrogate hierarchical
    low_fidelity_model = 'LOFI'
    high_fidelity_model = 'HIFI'
    correction additive zeroth_order

variables,
normal_uncertain = 2
    means           = 0.   0.
    std_deviations = 1.   1.
descriptors      = 'x1' 'x2'

responses,
response_functions = 1
no_gradients
no_hessians

model,
id_model = 'LOFI'
single
    interface_pointer = 'LOFI_FN'

interface,
id_interface = 'LOFI_FN'
direct
    analysis_driver = 'lf_rosenbrock'
    deactivate restart_file

model,
id_model = 'HIFI'
single
    interface_pointer = 'HIFI_FN'

interface,
id_interface = 'HIFI_FN'
direct
    analysis_driver = 'rosenbrock'
    deactivate restart_file

```

correction term is additive and the zeroth_order specification indicates that the high-fidelity approximation given by (49.5) interpolates the true high-fidelity values at each of the high-fidelity simulation points, but not higher-order derivative information.

6.3 Optimization Under Uncertainty (OUU)

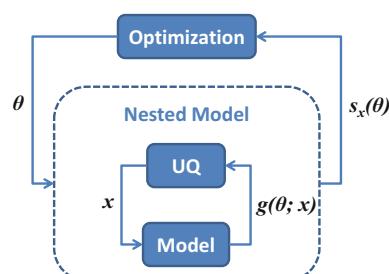
Another common hybrid method scenario is to nest uncertainty analysis within an optimization process to perform optimization under uncertainty (OUU). This allows design processes to account for the effect of input parameter uncertainties by seeking robust or reliable designs. In the former case, one might seek to minimize the variability of a critical performance metric in the presence of uncertainties, and in the latter case, one might constrain the probability of failure of a structure to be less than some allowable level, given uncertainty in applied loads and/or material properties of the structural components.

In OUU, a nondeterministic (UQ) method is used to evaluate the effect of uncertain variable distributions on response functions of interest. Statistics on these response functions are then included in the objective and constraint functions of a nonlinear optimization problem. Different UQ methods can have very different features from an optimization perspective, leading to the tailoring of optimization under uncertainty approaches to particular underlying UQ methodologies.

If the UQ method is sampling based, then three approaches are currently supported: nested OUU, surrogate-based OUU, and trust-region surrogate-based OUU. Within the nested OUU approach, the outer loop seeks to optimize a nondeterministic quantity (e.g., minimize probability of failure) over a set of design parameters, while the inner loop performs UQ to evaluate this nondeterministic quantity for a particular point in design space. Figure 49.17 depicts this case, where θ are the design variables, x are the uncertain variables characterized by probability distributions, $g(\theta, x)$ are the response functions from the simulation, and $s_x(\theta)$ are the statistics generated from the uncertainty quantification on these response functions. Surrogate-based OUU methods extend the nested approach by creating a surrogate over the design variables and/or the uncertain variables, and trust-region surrogate-based OUU manages the approximation models to ensure that the process converges to the optimum of the original problem. Additional details and computational results are provided in [9].

A second class of OUU algorithms is called reliability-based design optimization (RBDO). RBDO methods are used to perform design optimization accounting for reliability metrics. Dakota's reliability analysis capabilities provide a rich foundation for exploring a variety of RBDO formulations, and analytic sensitivities of

Fig. 49.17 Advanced capability example: nested optimization under uncertainty



reliability metrics can be used to accelerate gradient-based optimization algorithms. The simplest and most direct RBDO approach is the bi-level approach in which a full reliability analysis is performed for every optimization function evaluation, as in Fig. 49.17. An alternative RBDO approach is the sequential approach, in which additional efficiency is sought through breaking the nested relationship of the MPP and design searches through the use of local surrogate models (first- and second-order Taylor series approximations).

A third class of OUU algorithms performs OUU with stochastic expansions. Similar to RBDO, an advantage of using stochastic expansions for UQ is that analytic design sensitivities can be exploited, using either first-order expansions of response sensitivities or zeroth-order combined expansions over both design and uncertain parameter spaces [10]. Armed with analytic moments and their design sensitivities, a variety of bi-level, sequential, and multifidelity formulations can be explored; see the Dakota Theory Manual [3] for details.

Finally, when employing epistemic or mixed aleatory-epistemic UQ, the OUU formulation will involve optimizing metrics related to the epistemic interval (a worst case interval bound in the case of reliability and the width of the interval in the case of robustness). This enables the design process to directly account for lack of knowledge.

Configuring a Dakota study for any of these OUU approaches is similar to the earlier mixed UQ example. Figure 49.18 shows an outer gradient-based optimization method seeking values of width w and thickness t to minimize the mean weight of a cantilevered beam, subject to nonlinear inequality constraints that enforce stress and displacement reliability of at least 3.0. For each set of design variables selected by the optimizer, the nested model uses a polynomial chaos method to evaluate statistics on the weight, stress, and displacement responses; see Fig. 49.19. The nested model specification includes `primary_response_mapping` and `secondary_response_mapping`, which indicate which inner method statistics map to objectives and constraints, respectively.

6.4 Bayesian Methods

The UQ approaches presented thus far, e.g., sampling, reliability, stochastic expansions, and interval analysis, are *forward UQ* methods; they propagate information from uncertain input parameters through a computational model to inform response uncertainty. Dakota also includes *inverse UQ* methods for which the uncertainty in a response (typically given through physical observations or experiments) is propagated backward to obtain corresponding uncertainties on inputs consistent with the observed response uncertainty. Thus inverse UQ methods can be classified as parameter estimation or calibration methods that result in a richer characterization of input variables than point estimation.

Traditional optimization and least-squares calibration methods can be used for backward uncertainty propagation, when the input uncertainty characterizations are parameterized. For example, a log-normal distribution might be assumed, and the

```

environment
    method_pointer = 'OPTIM'

method
    id_method = 'OPTIM'
    model_pointer = 'OPTIM_M'
    npsol_sqp
        convergence_tolerance = 1.e-6

model
    id_model = 'OPTIM_M'
    nested
        variables_pointer = 'OPTIM_V'
        sub_method_pointer = 'UQ'
        responses_pointer = 'OPTIM_R'
        primary_response_mapping = 1. 0. 0. 0. 0. 0. 0. 0.
        secondary_response_mapping = 0. 0. 0. 0. 1. 0. 0. 0.
                                            0. 0. 0. 0. 0. 0. 1.

variables
    id_variables = 'OPTIM_V'
    continuous_design = 2
        initial_point    2.5    2.5
        upper_bounds     10.0   10.0
        lower_bounds     1.0    1.0
        descriptors      'w'    't'

responses
    id_responses = 'OPTIM_R'
    objective_functions = 1
    nonlinear_inequality_constraints = 2
        nonlinear_inequality_lower_bounds = 3. 3.
        nonlinear_inequality_upper_bounds = 1.e+50 1.e+50
    analytic_gradients
    no_hessians

```

Fig. 49.18 OUU: portion of Dakota input file showing outer gradient-based optimizer minimizing mean weight subject to reliability constraints

optimization procedure works to find the input mean and variance most consistent with the provided response data. A typical goal is to minimize the difference between statistics of the simulation response (as computed by forward UQ) and the experiment response, such as its mean, variance, and/or percentiles. This approach is sometimes called “moment matching” or “backward propagation of variance.” In Dakota, this would be expressed as nested OUU formulation, where a least squares objective over UQ statistics is defined on the outer loop [35].

Bayesian methods offer another means to characterize uncertainties on input distributions given observational data, and providing effective options in Dakota for this approach is a current point of emphasis. Bayesian calibration theory is well described elsewhere (e.g., [23]), so only a brief summary is provided in the following.

Fig. 49.19 OUU: portion of Dakota input file showing inner polynomial chaos UQ method

```

method
    id_method = 'UQ'
    model_pointer = 'UQ_M'
    polynomial_chaos
        expansion_order = 2
        collocation_ratio = 2 seed = 12347 rng rnum2
        num_response_levels = 0 1 1
        response_levels = 0.0 0.0
        compute reliabilities
        complementary distribution

model
    id_model = 'UQ_M'
    single
        variables_pointer = 'UQ_V'
        interface_pointer = 'UQ_I'
        responses_pointer = 'UQ_R'

variables
    id_variables = 'UQ_V'
    continuous_design = 2
    normal_uncertain = 4
        means           = 40000. 29.E+6 500. 1000.
        std_deviations   = 2000. 1.45E+6 100. 100.
        descriptors     = 'R' 'E' 'X' 'Y'

interface
    id_interface = 'UQ_I'
    direct
        analysis_driver = 'mod_cantilever'

responses
    id_responses = 'UQ_R'
    response_functions = 3
        response_descriptors = 'weight' 'stress' 'displ'
    analytic_gradients
    no_hessians

```

In Bayesian approaches, uncertain parameters are characterized by probability density functions. During the calibration process, a “prior distribution” (the probability density function that describes knowledge before the incorporation of data, $f_{\Theta}(\theta)$) is assumed for each input parameter. This prior is then iteratively updated through a Bayesian framework involving experimental data and a likelihood function. The likelihood function describes how well each parameter value is supported by the provided data. Bayes Theorem [20], shown in (49.10), is used for inference: to derive the plausible parameter values based on the prior probability density and the data y . The result is the posterior parameter density of the parameters $f_{\Theta|Y}(\theta|y)$. It is interpreted the same way as the prior, but includes the information derived from the data.

$$f_{\Theta|Y}(\theta|y) = \frac{f_{\Theta}(\theta) \mathcal{L}(\theta; y)}{f_Y(y)} \quad (49.10)$$

The likelihood function is used to describe how well a model's predictions are supported by the data. The likelihood function can be written generally as

$$\mathcal{L}(\theta; y) = f(g(\theta) - y) \quad (49.11)$$

where θ are the parameters of model g . The particular form for the function f can have significant influence on the results; Dakota employs likelihoods based on Gaussian probability density functions. These assume that the mismatch between the model (e.g., computer simulation) and the experimental observations (errors) is Gaussian:

$$y_i = g(\theta) + \epsilon_i, \quad (49.12)$$

where ϵ_i is a random variable that can encompass both measurement errors on y_i and modeling errors associated with the simulation $g(\theta)$. By further assuming that all n observations from experiments are independent, the probabilistic model defined by (49.12) results in a likelihood function for θ that is the product of n normal probability density functions:

$$\mathcal{L}(\theta; y) = \prod_{i=1}^n \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{(y_i - g(\theta))^2}{2\sigma^2} \right]. \quad (49.13)$$

Markov Chain Monte Carlo (MCMC) is the standard method used to compute posterior parameter densities, given the observational data and the priors. There are many references that describe the basic Metropolis algorithm [13]. One variation used in Dakota is DRAM: Delayed Rejection and Adaptive Metropolis [14]. Note that MCMC algorithms typically take tens or hundreds of thousands of steps to converge. Since each iteration involves an evaluation of the model $g(\theta)$, surrogate models of the simulation response are typically employed, allowing the MCMC process to quickly generate thousands of samples on the emulator. One can specify a Gaussian process, a polynomial chaos expansion, or a stochastic collocation expansion as the emulator for the Bayesian calibration methods. The specification details for these are listed in the Reference Manual [2].

There are two implementations of Bayesian calibration in Dakota, specified with `bayes_calibration` followed by `queso` or `dream`. The QUESO method uses components from the QUESO library (Quantification of Uncertainty for Estimation, Simulation, and Optimization) developed at The University of Texas at Austin. DREAM is based on the DiffeRential Evolution Adaptive Metropolis approach which runs multiple different chains simultaneously for global exploration and automatically tunes the proposal covariance during the process by a self-adaptive randomized subspace sampling [37]. Dakota also has an experimental implementation of the Los Alamos National Laboratory-developed GPMSA (Gaussian Process Models for Simulation Analysis) approach [17]. It is currently being reimplemented in the QUESO framework for better integration with Dakota and is

```

method
    bayes_calibration queso
    samples = 1000 seed = 348
    dram # or delayed_rejection, adaptive_metropolis, metropolis_hastings
    proposal_covariance
        diagonal values 1.0e6 1.0e-1

variables
    uniform_uncertain 2
    upper_bounds 1.e8 10.0
    lower_bounds 1.e6 0.1
    initial_point 2.85e7 2.5
    descriptors 'E' 'w'
    continuous_state 4
    initial_state 3 40000 500 1000
    descriptors 't' 'R' 'X' 'Y'

interface
    system
        analysis_driver = 'cantilever3'

responses
    calibration_terms = 2
    calibration_data_file = 'dakota_cantilever_queso.withsigma.dat'
    freeform
    num_experiments = 10
    variance_type = 'scalar'
    descriptors = 'stress''displacement'
    no_gradients
    no_hessians

```

Fig. 49.20 Sample Dakota input file for Bayesian calibration

not ready for general use as of this writing. For some detailed application examples with Dakota’s Bayesian capabilities, see [4].

One can also specify various configuration options for these different algorithm selections. For the QUESO MCMC, one can select the standard Metropolis-Hastings algorithm or an adaptive Metropolis in which the covariance of the proposal density is updated adaptively. There is also a setting to use delayed rejection. For the DREAM method, one can define the number of chains used, as well as the number of chains randomly selected to be used in crossover. There are also other settings governing the convergence and other features of DREAM. For more details about these parameters, see [37]. While DREAM is distributed and built with Dakota by default, the QUESO-based DRAM requires custom compilation at the user site.

Figure 49.20 shows a sample Dakota input file that gives a uniform prior distribution on parameters E (Young’s modulus) and w (width) of the cantilever beam. The QUESO Bayesian calibration method will use the cantilever beam simulation together with the ten user-provided experimental data points with corresponding measurement error to draw samples from the joint posterior distribution of E and w . The other parameters specified with `continuous_state` will be held fixed at their nominal values.

The choice of proposal covariance is important as it governs the size of the proposed steps in the MCMC chain. Dakota has a variety of options for the `proposal_covariance` keyword: one can specify the diagonals of the covariance matrix (i.e., the variance of the steps in each input parameter direction) or a full covariance matrix, one can use the prior to calculate a proposal covariance, or one can automatically use any available simulation derivative information to estimate it. The experimental data points in this example are provided in a file named `dakota_cantilever_queso.withsigma.dat`. The ten data points are provided along with estimates of their variance, as specified by the scalar variance type. Dakota supports several types of measurement error variances: constant measurement error for all of the data, a distinct scalar variance estimate per data point, and a full covariance matrix of measurement errors. These measurement errors are used in the calculation of the likelihood function.

To summarize, the DRAM and DREAM MCMC capabilities in Dakota can use data to generate posterior distributions on parameters. These capabilities have a number of method controls and can utilize Dakota's surrogate models so that the MCMC can be performed on an emulator in place of expensive simulations. Current research efforts in Dakota are adding new Bayesian capabilities and options. For example, analytic derivatives of emulator models such as PCE give rise to an inverse Hessian of the likelihood function to use as a proposal covariance, preconditioning the MCMC. Bayesian methods will support adaptive refinement of the posterior with emulator models and use various discrepancy functions δ to model the difference between the simulation and the observational data. Finally, recent capabilities allow Bayesian methods to handle field, or functional, data, where the output depends on independent coordinates such as time or space.

7 Usage Guidelines

The choice of uncertainty quantification method depends on how the input uncertainty is characterized, the computational budget, the nonlinearity and smoothness of the input/output mapping, and the desired output accuracy. Some recommendations for Dakota UQ methods are summarized in Table 49.3 and discussed in this section.

7.1 Sampling Methods

Sampling-based methods are the most robust uncertainty techniques available, are applicable to almost all simulations, and possess rigorous error bounds; consequently, they should be used whenever the function is relatively inexpensive to compute and adequate sampling can be performed. In the case of expensive computational simulations, however, the number of function evaluations required by traditional techniques such as Monte Carlo and Latin hypercube sampling (LHS)

Table 49.3 Guidelines for UQ method selection

Method classification	Desired problem characteristics	Applicable methods
Sampling	Nonsmooth, multimodal response functions; Larger sets of random variables; Response evaluations are relatively inexpensive	Sampling (Monte Carlo or LHS) Importance sampling
Local reliability	Smooth, unimodal response functions; Larger sets of random variables; Estimation of tail probabilities	Local reliability (MV, AMV/AMV ² , AMV+/AMV ²⁺ , TANA, FORM/SORM)
Global reliability	Smooth or limited nonsmooth response; Multimodal response; low dimensional; Estimation of tail probabilities	Global reliability, GPAIS, POFDarts
Stochastic expansions	Smooth or limited nonsmooth response; Multimodal response; low dimensional; Estimation of moments or moment-based metrics	Polynomial chaos, stochastic collocation
Epistemic	Uncertainties are poorly characterized	Interval: local/global interval estimation, sampling; BPA: local/global evidence
Mixed UQ	Some uncertainties are poorly characterized	Nested UQ (IVP, SOP, DSTE) with epistemic outer loop and aleatory inner loop, sampling
Bayesian calibration	Calibration of prior densities with data resulting in a posterior	Bayesian calibration with QUESO (DRAM), DREAM

quickly becomes prohibitive, especially if tail statistics are needed. Importance sampling is one goal-oriented approach that can reduce the number of samples needed.

7.2 Reliability Methods

Local reliability methods (e.g., MV, AMV/AMV², AMV+/AMV²⁺, TANA, and FORM/SORM) are more computationally efficient in general than sampling methods and are effective when applied to reasonably well-behaved response functions, i.e., functions that are smooth, unimodal, and only mildly nonlinear. When confronted with nonsmooth, multimodal, and/or highly nonlinear response functions, global reliability methods (efficient global reliability analysis (EGRA), Gaussian process adaptive importance sampling (GPAIS), and probability of failure darts (POFDarts)) should be used. These techniques employ adaptive point selection and refinement of surrogate models to accurately resolve the failure domain. For relatively low-dimensional problems (typically less than ten variables), global reliability methods display efficiency similar to local reliability methods, but with the accuracy of exhaustive sampling.

7.3 Stochastic Expansions Methods

Stochastic expansion methods (polynomial chaos and stochastic collocation) are general-purpose techniques provided that the response functions possess finite second-order moments. Further, these methods capture the underlying functional relationship between a key response metric and its random variables. The current challenge in the development of these methods, as for other global surrogate-based methods, is effective scaling for large numbers of random variables. Recent advances in adaptive collocation and sparsity detection methods address some of the scaling issues, with successful deployments approaching 100 random dimensions.

7.4 Epistemic Uncertainty Quantification Methods

Epistemic uncertainty methods in Dakota focus on uncertainties resulting from a lack of knowledge. In these problems, the assignment of input probability distributions when data is sparse can be somewhat suspect. One approach to handling epistemic uncertainties is interval analysis (`local_interval_est` and `global_interval_est`), where a set of intervals on inputs, one interval for each input variable, is mapped to a set of intervals on outputs. To perform this process efficiently, global or local optimization methods can be used. Another related technique is Dempster-Shafer theory of evidence (Dakota methods `local_evidence` and `global_evidence`), where multiple intervals per input variable (which can be overlapping, contiguous, or disjoint) are propagated, again potentially using optimization methods.

7.5 Mixed Aleatory-Epistemic UQ Methods

For problems with a mixture of epistemic and aleatoric uncertainties, it is desirable to separate the two uncertainty types within a nested analysis, segregating the reducible components of the uncertainty for purposes of clarifying the interpretation of the statistical results. In this nested approach, an outer epistemic level selects realizations of epistemic parameters, and for each epistemic realization, a probabilistic analysis is performed on the inner aleatory level. In the case where the outer loop involves propagation of subjective probability, the nested approach is known as second-order probability. In the case where the outer loop is an interval propagation, the nested approach is known as interval-valued probability. Between these two extremes lies the case where the outer loop is an evidence-based approach, for which belief and plausibility bounds are generated on aleatory statistics.

7.6 Bayesian Methods

Dakota has two MCMC approaches for calculating posterior distributions on model parameters using Bayesian calibration. These posterior distributions are based on

the prior parameter distributions and informed and updated with observational data. Currently, the MCMC approaches are based on DREAM or DRAM algorithms.

8 Conclusion

The freely available Dakota software delivers a suite of uncertainty quantification algorithms to address challenges associated with simulation-based science and engineering analyses. It allows the hybridization of UQ with optimization and the use of surrogates and multifidelity models with both analyses. Smart adaptive methods that can mitigate the curse of dimensionality for stochastic expansions and sampling are a current research emphasis. Bayesian calibration methods are also under active development, specifically focusing on surrogate modeling, discrepancy, postprocessing, and multi-model extensions.

References

1. Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Ebeida, M.S., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.4 users manual. Technical report SAND2010-2183, Sandia National Laboratories, Albuquerque (Updated Nov 2013). Available online from <http://dakota.sandia.gov/documentation.html>
2. Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Ebeida, M.S., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.4 reference manual. Technical report SAND2010-2184, Sandia National Laboratories, Albuquerque (Updated Nov 2013). Available online from <http://dakota.sandia.gov/documentation.html>
3. Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Ebeida, M.S., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.4 theory manual. Technical report SAND2011-9106, Sandia National Laboratories, Albuquerque (Updated Nov. 2013). Available online from <http://dakota.sandia.gov/documentation.html>
4. Adams, B.M., Hopper, R.W., Lewis, A., McMahan, J.A., Smith, R.C., Swiler, L.P., Williams, B.J.: User guidelines and best practices for casl vuq analysis using Dakota. Technical report SAND2014-2864, Sandia National Laboratories, Albuquerque (Mar 2014). Available online from <http://dakota.sandia.gov/documentation.html>
5. Aughenbaugh, J.M., Paredis, C.J.J.: Probability bounds analysis as a general approach to sensitivity analysis in decision making under uncertainty. In: SAE World Congress and Exposition, SAE, Detroit, SAE-2007-01-1480 (2007)
6. Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M.: Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA J.* **46**(10), 2459–2468 (2008)
7. Dalbey, K., Swiler, L.P.: Gaussian process adaptive importance sampling. *Int. J. Uncertain. Quantif.* **4**(2), 133–149 (2014)
8. Ebeida, M., Mitchell, S., Swiler, L., Romero, V.: Pof-darts: Geometric adaptive sampling for probability of failure. *SIAM J. Uncertain. Quantif.* (2014, submitted)

9. Eldred, M.S., Giunta, A.A., Wojtkiewicz, S.F., Jr., Trucano, T.G.: Formulations for surrogate-based optimization under uncertainty. In: Proceedings of 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, AIAA-2002-5585 (2002)
10. Eldred, M.S., Webster, C.G., Constantine, P.: Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos. In: Proceedings of the 10th AIAA Non-deterministic Approaches Conference, Schaumburg, AIAA-2008-1892 (2008)
11. Eldred, M.S., Swiler, L.P., Tang, G.: Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation. *Reliab. Eng. Syst. Saf.* **96**(9), 1092–1113 (2011)
12. Ferson, S., Tucker, W.T.: Sensitivity analysis using probability bounding. *Reliab. Eng. Syst. Saf.* **91**, 1435–1442 (2006)
13. Gilks, W., Richardson, S., Spiegelhalter, D.: *Markov Chain Monte Carlo in Practice*. Chapman & Hall, Boca Raton (1998)
14. Haario, H., Laine, M., Mira, A., Saksman, E.: Dram: efficient adaptive MCMC. *Stat. Comput.* **16**, 339–354 (2006). <http://dx.doi.org/10.1007/s11222-006-9438-0>
15. Haldar, A., Mahadevan, S.: *Probability, Reliability, and Statistical Methods in Engineering Design*. Wiley, New York (2000)
16. Helton, J.C., Johnson, J.D., Oberkampf, W.L., Storlie, C.B.: A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory. *Comput. Methods Appl. Mech. Eng.* **196**, 3980–3998 (2007)
17. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
18. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. *J. Glob. Optim.* **34**, 441–466 (2006)
19. Iman, R.L., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat.: Simul. Comput.* **B11**(3), 311–334 (1982)
20. Jaynes, E.T., Brethorst, G.L.: *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge/New York (2003)
21. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **13**, 455–492 (1998)
22. Karanki, D.R., Kishwaha, H.S., Verma, A.K., Ajit, S.: Uncertainty analysis based on probability bounds (p-box) approach in probabilistic safety assessment. *Risk Anal.* **29**, 662–675 (2009)
23. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. *J. R. Stat. Soc.* **63**, 425–464 (2001)
24. Der Kiureghian, A., Liu, P.L.: Structural reliability under incomplete information. *J. Eng. Mech. ASCE* **112**(EM-1), 85–104 (1986)
25. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
26. Ng, L.W.T., Eldred, M.S.: Multifidelity uncertainty quantification using nonintrusive polynomial chaos and stochastic collocation. In: Proceedings of the 14th AIAA Non-deterministic Approaches Conference, Honolulu, AIAA-2012-1852 (2012)
27. Oberkampf, W.L., Helton, J.C.: Evidence theory for engineering applications. Technical report SAND2003-3559P, Sandia National Laboratories, Albuquerque (2003)
28. Owen, A.: A central limit theorem for Latin hypercube sampling. *J. R. Stat. Soc. Ser. B (Methodol.)* **54**(2), 541–551 (1992)
29. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad Nauk SSSR* **4**, 240–243 (1963)
30. Srinivasan, R.: *Importance Sampling*. Springer, Berlin/New York (2002)
31. Stein, M.: Large sample properties of simulations using Latin hypercube sampling. *Technometrics* **29**(2), 143–151 (1987)
32. Stroud, A.: *Approximate Calculation of Multiple Integrals*. Prentice Hall, Englewood Cliffs (1971)

33. Swiler, L.P., West, N.J.: Importance sampling: promises and limitations. In: Proceedings of the 12th AIAA Non-deterministic Approaches Conference, Orlando, AIAA-2010-2850 (2010)
34. Swiler, L.P., Wyss, G.D.: A user's guide to Sandia's Latin hypercube sampling software: LHS UNIX library and standalone version. Technical report, SAND04-2439, Sandia National Laboratories, Albuquerque (2004)
35. Swiler, L.P., Adams, B.M., Eldred, M.S.: Model calibration under uncertainty: matching distribution information. In: Proceedings of 12th AIAA/ISSMO multidisciplinary analysis optimization conference, Victoria, AIAA-2008-5944 (2008)
36. Tatang, M.: Direct incorporation of uncertainty in chemical and environmental engineering systems. Ph.D. thesis, MIT (1995)
37. Vrugt, J.A., ter Braak, C.J.F., Diks, C.G.H., Robinson, B.A., Hyman, J.M., Higdon, D.: Accelerating Markov chain Monte Carlo simulation by self-adaptive differential evolution with randomized subspace sampling. *Int. J. Nonlinear Sci. Numer. Simul.* **10**(3), 273–290 (2009)
38. Walters, R.W.: Towards stochastic fluid mechanics via polynomial chaos. In: Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, AIAA-2003-0413 (2003)
39. Wiener, N.: The homogeneuous chaos. *Am. J. Math.* **60**, 897–936 (1938)
40. Xiu, D., Karniadakis, G.M.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)

Problem Solving Environment for Uncertainty Analysis and Design Exploration

50

Charles Tong

Abstract

This chapter gives an overview of the capabilities of the PSUADE (acronym for Problem Solving Environment for Uncertainty Analysis and Design Exploration) software package, which has been developed to support the many operations involved in a typical nonintrusive (i.e., simulation codes are to be treated as “black boxes”) uncertainty quantification (UQ) study, such as sample generation, ensemble simulations, and analysis of simulation results. Specifically, the software enables users to perform detailed UQ analysis such as uncertainty analysis (for computing statistical moments and probability distributions), sensitivity analysis (e.g., variance decomposition), parameter screening or down-selection, response surface analysis, statistical inferences, and optimization under uncertainty. In addition to a rich suite of UQ capabilities accessible via either batch or command line processing, PSUADE also provides many tools for data manipulation and visualization, which may be useful for more immersive data analysis. PSUADE is a public domain software that has been released under the LGPL license since 2007.

Keywords

Uncertainty quantification • Global sensitivity analysis • Bayesian inference • Response surface methodology • Dimension reduction • Numerical optimization • Optimization under uncertainty • Markov chain Monte Carlo • Discrepancy modeling • Cross validation • Aleatory-epistemic uncertainty

Contents

1	Introduction	1696
1.1	PSUADE Installation	1696
1.2	A General UQ Process	1697

C. Tong (✉)

Computation Directorate, Lawrence Livermore National Laboratory, Livermore, CA, USA
e-mail: tong10@llnl.gov

1.3	PSUADE Basics.....	1700
1.4	Running PSUADE.....	1702
2	UQ Methods and Tools in PSUADE.....	1703
2.1	Uncertain Parameter Screening/Dimension Reduction.....	1703
2.2	Response Surface Analysis.....	1707
2.3	Uncertainty Analysis.....	1714
2.4	Quantitative Sensitivity Analysis.....	1717
2.5	Parameter Inference.....	1720
2.6	Optimization Under Uncertainty.....	1728
2.7	Other PSUADE Capabilities.....	1730
3	Conclusion.....	1731
	References.....	1731

1 Introduction

UQ is defined as “the process of quantifying uncertainties associated with model calculations of true, physical quantities of interest, with the goals of accounting for all sources of uncertainty and quantifying the contributions of specific sources to the overall uncertainty” [1]. In this chapter a presentation will be given on how the different elements of this process can be performed using the mathematical, statistical, and computer science tools available from PSUADE.

At a glance, PSUADE contains a comprehensive suite of capabilities for UQ tasks such as:

1. Dimension reduction (parameter screening/down-selection)
2. Response surface (including adaptive) analysis
3. Basic uncertainty analysis (sampling- or response surface-based)
4. Mixed aleatory-epistemic uncertainty analysis
5. Global sensitivity analysis (sampling- or response surface-based)
6. Calibration/Bayesian inference with model form correction
7. Deterministic optimization and optimization under uncertainty

Each of these capabilities is equipped with many optional features (for expert users) and diagnostics (including Matlab-based graphics) for immersive analysis of simulation or UQ results.

The rest of this chapter is organized as follows: after giving a brief tutorial on PSUADE installation, a systematic exploration of PSUADE’s capabilities will be described via following the steps of a general process for performing UQ on a simulation model.

1.1 PSUADE Installation

PSUADE is a free software and its source code can be downloaded from

http://computation.llnl.gov/casc/uncertainty_quantification.

Successful build of PSUADE requires system tools such as C, C++, and Fortran compilers, as well as “ccmake” utilities. Installing PSUADE on Linux or Mac OS-based systems is straightforward by following the instructions below:

1. `tar xvfz PSUADE_vx.x.x.tar.gz`
2. `cd PSUADE_vx.x.x`
3. `mkdir build`
4. `cd build`
5. `ccmake ..`
 - Enter “c” to display the package information.
 - To install PSUADE other than locally in your home directory, set the “install directory.”
 - Check the default compilers and turn on/off other options as desired.
 - Enter “c” again to update the selection.
 - Finally, enter “g” to save and exit.
6. Now run `make` to create the “psuade” executable and the associated libraries in the “build/bin” and “build/lib” directories. Alternatively, run `make install` to build the “psuade” executable in your designated install directory.
7. To verify correct installation, run an example (on Linux) by:
 - `cd PSUADE_vx.x.x/Examples/Bungee`
 - `cc -o simulator simulator.c -lm`
 - `../../build/bin/psuade psuade.in`
 - If it runs successfully, some summary statistics (e.g., mean ≈18 and standard deviation ≈9) will be displayed.

Generally, this installation and testing procedure should take no more than a few minutes. Details of building and installing PSUADE on Windows-based platform can be found in the user manual that comes with the source code download.

1.2 A General UQ Process

Before embarking on a serious UQ study of complex simulation models, a UQ process (or plan) should be developed to guide the completion of the study. Without a well-thought-out plan, computational resources may be wasted and the results may not be defensible. A typical UQ process comprises the following steps (which may be refined depending on the needs of individual applications); for a given application (simulation model):

1. Define objectives of the UQ study.
2. Identify all relevant sources of uncertainties.
3. Characterize the identified sources of uncertainties.
4. Propagate the characterized uncertainties through computer simulations.
5. Analyze uncertainties of the model outputs of interest.

Each of these steps may take hours to months. In the following, a brief discussion of these steps is given to show how to set up PSUADE input files to capture these steps.

1.2.1 Defining UQ Objectives

This step entails a full description of the objectives of the UQ study as well as assumptions about the simulation models under study. In addition, it includes a full specification of all the essential components of the simulation model so that sufficient information are gathered to enable replication of the same UQ study at a later date. This is a crucial step in a UQ process, since conclusions drawn from the study are valid only under the assumptions made in this step.

Some of the UQ objectives are:

- (a) To quantify the prediction uncertainties of a given simulation model (uncertainty analysis)
- (b) To quantify contributions from major sources of uncertainty in the simulation model (sensitivity analysis)
- (c) To calibrate model parameters to match noisy data from physical experiments (parameter inference)

The objectives of the planned UQ study may affect which UQ methods are appropriate. For example, if the objective is to help prioritize research by assessing and comparing the impact of the various uncertain sources on the design, then sensitivity analysis will be valuable. Furthermore, if simulations are themselves computationally expensive, response surfaces can be used as inexpensive surrogates in place of ensemble model simulations in UQ analysis. PSUADE provides a spectrum of methods to achieve these UQ objectives.

1.2.2 Identification of Uncertainty Sources

A complex simulation model may contain hundreds of internal parameters that are tunable or uncertain. Due diligence should be exercised in compiling this comprehensive list of tunable parameters. If the parameter space (number of uncertain parameters) identified initially is too large (e.g., more than a few hundreds) for a comprehensive analysis, expert judgment may be needed to shorten the list. Other forms of simulation model uncertainty, such as model form uncertainties due to missing physics or model simplification, may also be identified at this stage. In addition, any available experimental data that may be helpful in reducing model prediction uncertainties should be included in this step.

1.2.3 Characterization of Uncertainty Sources

This step corresponds to compiling credible initial ranges and/or probability distributions for the uncertain parameters and is one of the most important and time-consuming tasks. Extreme caution should be taken in this step because the credibility of the final UQ results depends critically on the choice of ranges and

distributions. Therefore, every choice of ranges and distributions should be carefully made and justified.

Before performing the task of uncertainty characterization, it is useful to classify the identified sources of uncertainty. One such classification is based on the nature of uncertainty for which uncertainty is classified as either *aleatory* or *epistemic* (e.g., [12]). For all practical purposes, the definition of aleatory uncertainty can be confined to be uncertainty that can be characterized by a known probability distribution and the definition of epistemic uncertainty to be uncertainty that can be characterized by an interval (or a number of disjoint intervals) with unknown probability distribution. Furthermore, epistemic uncertainty includes systematic bias of model output caused by missing or simplified physics. This distinction clarifies the different approaches to quantifying uncertainties for the two types.

For uncertain variables (will be used interchangeably with uncertain parameters), other useful classifications are (1) continuous variables, (2) ordinal variables, and (3) categorical variables. Continuous variables can be set to any value within a given interval (or a set of intervals), and thus, they have infinite number of distinct settings. Discrete variables, on the other hand, can only take on a limited number values within their ranges. Finally, categorical variables do not have meaningful numerical values. They are used for the purpose of distinguishing different options, such as different material fracture models, that may have little functional relationship with each other.

After the uncertain sources have been classified, their ranges or probability distributions are to be prescribed. Information about reasonable ranges and distributions can be obtained from theory, literature, or local subject matter experts. In summary, the product of this step is a comprehensive list of relevant uncertain sources appended with their classifications and probability distributions (or intervals).

1.2.4 Propagation of Uncertainties

Uncertainties can be propagated via intrusive or nonintrusive approaches. Due to their practicality, PSUADE provides primarily nonintrusive methods. Propagation involves generating samples and running a possibly large number of simulations. A key element to uncertainty propagation is sampling strategy (also called design of computer experiments). Proper selection of sampling strategies depends on the UQ objectives as well as certain properties of the simulation models. For example, if simulation cost is high and response surfaces are desired, “space-filling” sampling strategies such as quasi-Monte Carlo or specialized Latin hypercube should be selected. Sampling strategies available in PSUADE will be given later. In summary, the execution of this step will yield a sample (consisting of a number of sample points drawn from the parameter probability distributions) as well as the corresponding simulation results. Propagation of uncertainties often involves large ensemble simulations that may benefit from compatible job schedulers on high-performance computer systems.

1.2.5 Analysis of Uncertainties

UQ objectives guide the selection of relevant sampling strategies and analysis methods. In the following, several analyses commonly performed during a UQ study are listed:

- (a) Quantify the means and standard deviations of selected model outputs.
- (b) Rank a large number of uncertain parameters in order of their importance in affecting the model outputs.
- (c) Construct response surfaces relating uncertain parameters to the model outputs.
- (d) Quantify the global sensitivities of uncertain parameters.
- (e) Compute the best parameter settings that match observational data.

1.3 PSUADE Basics

In this section an example is given on how to develop a simple UQ process and how to use PSUADE to help execute the steps. Specifically, given a simulation model called “simulator” which has 100 uncertain parameters, the UQ objective is to identify the 10 most sensitive parameters and quantify their individual contribution to the overall uncertainty of a certain simulation output of interest. In addition, suppose the simulator takes hours to run and little knowledge is provided on how the output of interest varies with the uncertain inputs (linearly or nonlinearly). With this initial specification, a possible UQ workflow is as follows:

1. Define clearly the UQ objective.
2. Identify and prescribe feasible ranges for the 100 uncertain parameters.
3. Perform parameter screening to identify the ten most sensitive parameters (this step is needed because the simulation cost is relatively expensive).
4. Build a response surface for the ten sensitive parameters.
5. Use the response surface to compute uncertainty and quantify individual contributions via global sensitivity analysis.

The user information required to perform Step (4) can be captured in the following PSUADE input file (the other steps can be set up and run in similar fashions).

```

PSUADE
INPUT
# There are 10 inputs with ranges in [-3,3].
# Probability information is not needed for
# response surface construction
dimension = 10
variable 1 X1 = -3.0 3.0
...
variable 10 X10 = -3.0 3.0
END

```

```
OUTPUT
# Output of interest
dimension = 1
variable 1 Y
END
METHOD
# Sampling design = randomized Latin hypercube
sampling = LH
num_samples = 100
random_seed = 129932931
END
APPLICATION
# The simulation code (executable) is 'simulator'
driver = ./simulator
max_parallel_jobs = 1
END
ANALYSIS
# analysis will be performed in command line mode
printlevel = 1
END
END
```

This file should have **PSUADE** as the first line, and it should normally contain five sections (but at the very least should have the **INPUT**, **OUTPUT**, and **METHOD** sections) followed by the keyword **END** in the last line.

1.3.1 The Input Section

The **INPUT** section allows users to specify the number of inputs, input names, input ranges, and input probability distributions, which are enclosed in an **INPUT** block. In this example, the number of inputs is ten with names X1, X2, and so on. Input probability distributions (PDF) are not needed at this stage (they are needed in the quantitative sensitivity analysis).

PSUADE currently supports primarily continuous uncertain variables. The available PDFs are the popular ones such as uniform, normal, lognormal, gamma, beta, exponential, triangle, and Weibull distributions. In addition, PSUADE also provides a distribution type called “S” (meaning user-provided distribution), which allows user to prescribe, via a user-provided sample, general non-parametrized PDFs or distributions for discrete variables.

1.3.2 The Output Section

The **OUTPUT** section is similar to but simpler than the **INPUT** section. Here only the output dimension and the names of the output variables are to be specified.

1.3.3 The Method Section

The **METHOD** section specifies the selected sampling method and additional information on sampling. In this example, the sampling method is Latin hypercube (LH) with sample size 100. Other available sampling methods will be given later. Also, the internal random number seed can also be provided. This is useful if repeatability is desired, as randomness is often introduced in various sampling strategies. Other advanced features include options in setting up uniform or adaptive sample refinements.

1.3.4 The Application Section

The **APPLICATION** section sets up the user-provided simulation executable and other runtime parameters. In this example, **driver** points to the simulation code **simulator**. The simulation code can be just a simple program or a complex superscript performing preprocessing, actual model evaluation, and postprocessing. The **driver** can also be a PSUADE data file, which is used internally within PSUADE to construct surrogate models.

max_parallel_jobs directs PSUADE to launch the desired number of jobs simultaneously. If it is set to larger than 1, the asynchronous job scheduling mode will be turned on using the Linux fork-join mechanism. Other available features include alternative methods for job control as well as support for fault detection and recovery.

1.3.5 The Analysis Section

The **ANALYSIS** section specifies the desired UQ analysis. For example, if the selected analysis method is **Moment**, PSUADE will run and sample and computes the first few statistical moments based on the sample design specified in the **METHOD** section. The **ANALYSIS** section also provides many options for fine-tuning the various statistical analysis (including numerical optimization) as well as setting diagnostics output levels (e.g., **printlevel** in the example PSUADE input file given above).

1.4 Running PSUADE

PSUADE can be run in either batch or command line mode. Batch mode is intended primarily for creating and running samples. Command line mode provides interactive tools for UQ analysis, sample data manipulation, and creation of plots for visualization. In the following, the two modes are discussed in detail.

1.4.1 PSUADE Batch Mode

In this mode the input file (say, **psuade.in**) described previously should be run via

```
[Linux prompt] psuade psuade.in
```

If the **driver** variable (pointing to the executable simulator code) is defined in the **METHOD** section, PSUADE will first generate the desired sample and then call the

driver code repeatedly, each with a different sample point, until all sample points have been evaluated. Batch mode is useful when simulation run time is relatively short. More often, when simulation time is long or when the simulation platform has its own job control system, it may be more preferable to have PSUADE generate the sample points only and let application users handle the ensemble simulation runs. This latter scenario can be facilitated by running PSUADE without defining driver. In this case PSUADE will create a sample file only. Users can take the sample points in this file and run the ensemble independently. Upon completion of all simulation runs, the results can be compiled into a PSUADE data file and analyzed in the command line mode.

1.4.2 PSUADE Command Line Mode

For PSUADE to run in command line mode, simply run “psuade” without any argument. A PSUADE prompt will be displayed and PSUADE is ready to accept user commands. A list of available command categories will be displayed by entering “help,” and details of how to use a specific command are available by issuing the command with “-h.” Command categories include reading/writing from/to files with different formats, uncertainty and sensitivity analyses, response surface fitting and validation, parameter estimation, sample data manipulation, and creation of Matlab graphics for visualization. A full list of commands can be found in the PSUADE reference manual released together with the source code.

2 UQ Methods and Tools in PSUADE

As discussed above, many different types of analysis may be needed during a UQ study. In this section details of these capabilities are given to show how to use them via the PSUADE batch or command line mode.

2.1 Uncertain Parameter Screening/Dimension Reduction

When the number of independent uncertain parameters is large (say, >100) and the outputs of interest exhibit nonlinear responses to parameter variation with possibly significant parameter interactions, it may not be feasible to tackle this high-dimensional problem directly (the notorious “curse of dimensionality”) when the objective is to accurately quantify uncertainties or sensitivities. Instead, it may be preferable to precede detailed quantitative analyses with a parameter screening step that seeks to qualitatively assess parameter importance via a relatively inexpensive “coarse” sampling of the parameter space. Parameter screening is also known as “variable selection” or “subset selection,” which is one of the most pervasive subject areas in statistical and machine learning applications. One basic assumption in applying screening methods is that most of the output variations are driven by a small subset of uncertain parameters (the so-called Pareto effect that has been

observed in many physical phenomena), and the goal is to identify this subset (hopefully with <20 independent parameters) for more detailed UQ analysis.

2.1.1 Parameter Screening Methods in PSUADE

In selecting screening methods, many practitioners resort to simple mathematical or statistical techniques that make major assumptions about linearity, smoothness, and interactions (e.g., derivative or gradient-based local sensitivity analysis and the classical correlation analysis). Violation of these assumptions, for example, as in many multi-physics simulation models, can impact the correctness of the analysis results. As such, it is pertinent that suitable “screening” methods should be used, based on prior knowledge about the simulation models. For example, known linearity and additivity (in the form of the function that maps uncertain parameters to the model output of interest) should make the computationally inexpensive linear sensitivity analysis method (or Plackett-Burman method [10]) adequate, while a general nonlinear input-output relationship may require more exotic methods such as the Morris method [8] or Bayesian screening [9].

In using coarse sampling strategy to reduce the overall computational cost for screening, errors may be made in identifying important parameters. There are two types of errors: (1) Type I error or when unimportant parameters are identified as important and (2) Type II error or when important parameters are identified as unimportant. Type I error is benign resulting only in higher computational cost for subsequent analysis. Type II is much more serious and safeguard against this error should be employed, for example, by using several methods (including expert judgment and visual inspection) to confirm the final selection.

Parameter screening can be conducted in PSUADE using one of the following methods:

- Local sensitivity method for linear models (“LSA”)
- Fractional factorial methods (“FFx” where $x = 4$ or 5)
- Morris method for general nonlinear models (“MOAT”)

Here “LSA” is the least expensive method requiring only $m + 1$ simulations (where m is the number of uncertain parameters), but they also have the least discriminating power, since their effectiveness hinges on the assumption that the model output is approximately a linear function of the uncertain inputs within their ranges. “FFx” also assumes linear models, but it accommodates an additional pairwise interaction. Finally, the “MOAT” method does not make assumptions about linearity or additivity. However, because of its coarse sampling nature, it is not effective in detecting small-scale changes (e.g., spikes in the parameter space) inside the parameter space.

There are other parameter screening methods in PSUADE that are less common. For example, if the number of parameters is not too large (say, <100) and a large random or quasi-random sample has already been generated and run, then the sample may be analyzed by the Delta test [4].

2.1.2 How to Perform Screening Methods in PSUADE

All screening methods in PSUADE are set up and performed in similar fashion; hence it suffices to simply show one of them. To perform MOAT screening, for example, the steps are as follows:

1. Create a PSUADE input file specifying the uncertain parameters, the sampling method to be MOAT with the appropriate sample size, and the executable simulation code (simulator). The simulation code reads a parameter file for the 20 input values (this example is available as one of PSUADE's test examples called **Morris20**), uses them in the simulation, and returns the simulation results in the output file). An example of PSUADE input file is (the number of sample points is $10(m + 1)$ where m is the number of inputs):

```

PSUADE
INPUT
# There are 20 inputs and with ranges in [0,1].
dimension = 20
variable 1 X1 = 0.0 1.0
...
variable 9 X9 = 0.0 1.0
variable 10 XA = 0.0 1.0
...
variable 20 XK = 0.0 1.0
END
OUTPUT
dimension = 1
variable 1 Y
END
METHOD
sampling = MOAT
num_samples = 210
END
APPLICATION
driver = simulator
END
END

```

2. Execute the input file to generate and run the sample as shown below, and then rename the sample output file from the default name **psuadeData** to, for example, **psdata**. (Alternatively, export the sample points to be run separately and compile the outputs in PSUADE data file format for analysis.)

```

[Linux prompt] psuade psuade.in
...
[Linux prompt] mv psuadeData psdata

```

3. Run screening analysis in command line mode (note: the text after `psuade>` are user inputs):

```
[Linux prompt] psuade
*****
*      Welcome to PSUADE (version 1.7.4)
*****
PSUADE - A Problem Solving environment for
          Uncertainty Analysis and Design
          Exploration (1.7.4)
(for help, enter <help>)
psuade> load psdata
psuade> moat
... results will be displayed ...
psuade> quit
[Linux prompt]
```

4. The screening results can also be inspected by visualizing an optionally generated Matlab file. An example is given in Fig. 50.1 for a 20-parameter problem where the first 10 parameters are important (screening results are generated using 210 simulations).
5. There are other diagnostics features in the MOAT screening methods for more immersive analysis, such as visualization tools for outlier identification and parameter interaction analysis.

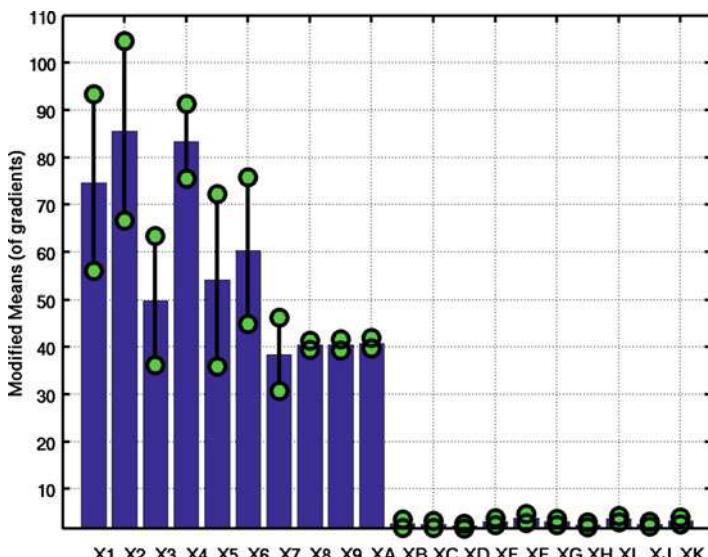


Fig. 50.1 MOAT screening plot for the 20-parameter model

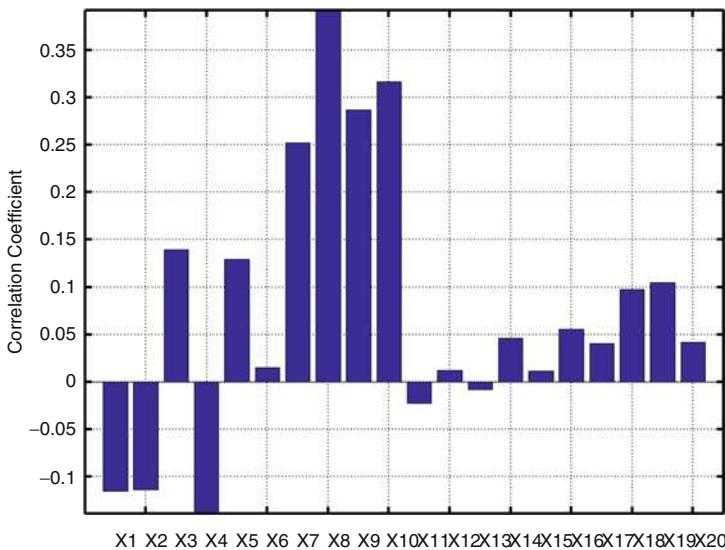


Fig. 50.2 Pearson screening plot for the 20-parameter model

6. To show the effectiveness of the MOAT method for nonlinear models, the same 20-parameter problem was analyzed with the Pearson correlation analysis on a Latin hypercube sample of size 500. Figure 50.2 shows the Pearson results, which do not reflect the correct ranking because the Pearson analysis is intended for near-linear models.

2.2 Response Surface Analysis

Response surface (also called surrogate models, emulators, meta-models, or reduced models) methods are a collection of statistical and mathematical techniques useful for reducing the computational cost in UQ analysis provided that the model output of interest is sufficiently smooth with respect to the uncertain parameters (so that the input-output relationship can be expressed by a “smooth” or piecewise “smooth” function). Response surface analysis should be performed on models with small to moderate number of parameters (up to 15–20). For models with large number of parameters, it is best to precede this step with parameter screening.

The construction of a response surface requires (1) a good set of “space-filling” sample points to best capture the model behavior in the parameter space, (2) a surface fitting method to construct a mathematical/statistical expression to fit the sample points, and (3) a mathematical technique to assess the “goodness-of-fit.”

2.2.1 Response Surface Methods in PSUADE

The response surface (RS) library in PSUADE provides a rich set of tools to facilitate the construction and validation of response surfaces. The library has four major components: (1) a collection of sampling designs and curve-fitting tools, (2) several methods for validating response surfaces, (3) a number of Matlab-based response surface visualization capabilities, and (4) software tools to create stand-alone response surface predictors (or interpolators) that can readily be used as surrogates in user codes.

To help capture dominant output behaviors in the parameter space, PSUADE provides many “space-filling” sampling methods for response surfaces. Some of them are:

- Monte Carlo sampling
- Quasi-Monte Carlo sampling (e.g., LP- τ)
- Latin hypercube sampling
- Orthogonal array sampling
- Orthogonal array-based Latin hypercube sampling
- Factorial and fractional factorial sampling methods
- Box-Behnken, Plackett-Burman, and central composite designs
- Sampling based on multidimensional domain decomposition
- Sparse grid sampling

Once generated, the sample points and their corresponding outputs (via simulation runs) are ready for “training” a response surface. The choice of response surface methods for a given simulation model depends on existing knowledge about the model itself. Response surface methods available in PSUADE are:

- MARS – multivariate regression splines by Friedman [5]
- MARS with bootstrapped aggregation
- Regression – linear, quadratic, cubic, Legendre polynomials
- Regression with user-provided kernel functions
- Derivative-based Legendre regression methods
- Support vector machine (SVM)
- Gaussian process model by MacKay ([6], to be downloaded separately)
- Universal Kriging method
- Radial basis function
- Sum-of-trees method
- K-nearest neighbor method
- 2- and 3-dimensional splines
- Sparse grids

Some of these methods may require special sampling method (e.g., splines require the use of the full factorial designs).

Since there is no one-size-fits-all approach to response surface analysis, the key is to select the best representation via rigorous validation. There are several RS validation methods available: (1) adjusted R-squared and resubstitution test

(error analysis on the training set), (2) hold-out sample test, and (3) K-fold cross validation. After rigorous validation, the final product should be a “good” response surface that can be used for further UQ analysis. A special feature in PSUADE is that users have the option to create a stand-alone code (in C and Python) that can be used to estimate the outputs of other locations in the parameter space (i.e., the code can be used as a surrogate for the simulation model). The quality of the response surface fit can also be examined visually by displaying the Matlab script created after response surface validation is completed.

2.2.2 How to Perform Response Surface Analysis in PSUADE

To illustrate how to build a response surface, the following Ishigami function is used:

$$Y = \sin(X_1) + 7 \sin^2(X_2) + 0.1X_3^4 \sin(X_1).$$

Once this function is implemented and compiled (into the executable “simulator”), the next steps are:

1. Create a PSUADE input file (e.g., psuadeRS.in) specifying the uncertain parameters, the sampling method (selected from the list above), and the sample size (should be chosen judiciously based on the available computing resources and the target curve fitting method). In the following example, the LP- τ quasi-random method is selected with an initial sample size of 100.

```

PSUADE
INPUT
    dimension = 3
    variable 1 X1    =    -3.1416    3.1416
    variable 2 X2    =    -3.1416    3.1416
    variable 3 X3    =    -3.1416    3.1416
END
OUTPUT
    dimension = 1
    variable 1 Y
END
METHOD
    sampling = LPTAU
    num_samples = 100
END
APPLICATION
    driver = simulator
END
END

```

2. Execute the input file to generate and run the sample as shown below and rename the sample output file from **psuadeData** to, for example, **psdata**. (If the simulation is expensive, this step can be replaced by exporting the sample points to be run separately and then re-assembling the simulation outputs in PSUADE data file format.)

```
[Linux prompt] psuade psuadeRS.in
...
[Linux prompt] mv psuadeData psdata
```

3. Perform response surface validation using **rscheck** in command line mode:

```
[Linux prompt] psuade
...
psuade> load psdata
load complete : nSamples = 100
            nInputs = 3
            nOutputs = 1
psuade> rscheck
# When prompted for RS type, choose MARS
...
Perform cross validation ? (y or n) y
Enter the number of groups to validate :
(2 - 100) 10
...
RSA: L 1:cross validation (CV) completed.
CV error file is RSFA_CV_err.m
psuade> quit
[Linux prompt]
```

In particular, after cross validation, summary statistics will be displayed on the screen. In addition, the cross-validation results can be visually inspected by running the Matlab file called **RSFA_CV_err.m**. An example is shown in Fig. 50.3 where the left plot displays the distribution of cross-validation errors and the right plot shows the one-to-one matching between the estimated and the actual sample outputs in cross validation. A perfect fit will give a “spike” at zero (error mean and standard deviations are both zero) on the left plot and all the “*” marks will be on the diagonal of the right plot. Observe from Fig. 50.3 that MARS does not give a good fit for a sample size of 100. In fact, for this sample size, none of the curve-fitting methods in PSUADE gives adequate results.

To improve the quality of the response surface, more sample points were added. Figure 50.4 shows the response surface validation results from using 500 sample points with Kriging, which gives a much better fit.

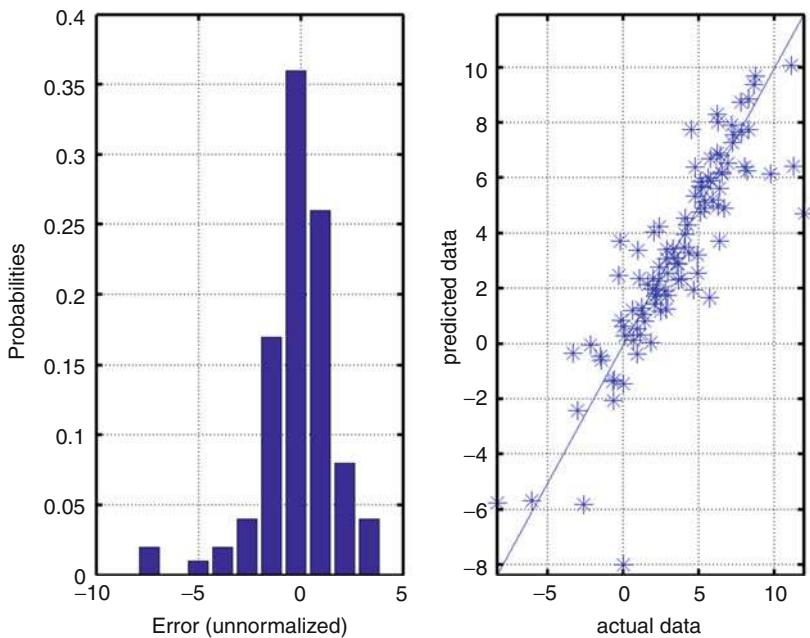


Fig. 50.3 MARS response surface validation for the Ishigami function

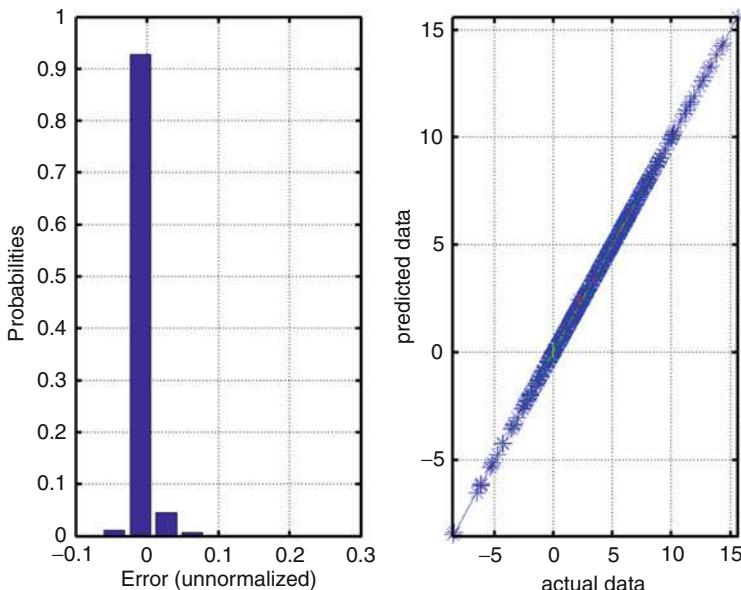


Fig. 50.4 Kriging response surface validation for the Ishigami function

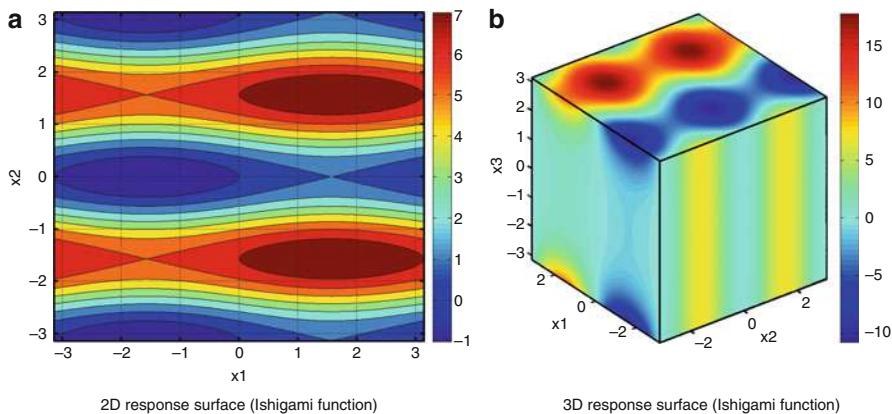


Fig. 50.5 Response surface visualization of the Ishigami function: 2D (*left*), 3D (*right*)

- Once the response surface validation has been completed, the response surface can be visualized with Matlab. For example, to generate a two-dimensional response surface plot using X_1 and X_2 in the Ishigami function, do:

```
[Linux prompt] psuade
...
psuade> load psdata
load complete : nSamples = 500
            nInputs = 3
            nOutputs = 1
psuade> rs2
Grid resolution ? (32 - 256) 256
# When prompted for the RS type, select kriging
...
matlabrs2.m is now available for response surface
and contour plots
```

Upon completion, `matlabrs2.m` can be run in Matlab for viewing the response surface. Three-dimensional plots can be generated in a similar manner using `rs3`. Examples of two- and three-dimensional plots are given in the Figure 50.5.

2.2.3 Adaptive Response Surface Analysis in PSUADE

PSUADE also provides computational tools for adaptive response surface analysis. Adaptive analysis can help reduce the simulation cost when “significant” variations in the simulation outputs are localized in some small regions in the parameter space. Adaptive analysis begins with the generation of a small (e.g., 50) initial space-filling sample. This sample is run on the simulation model and subsequently analyzed for

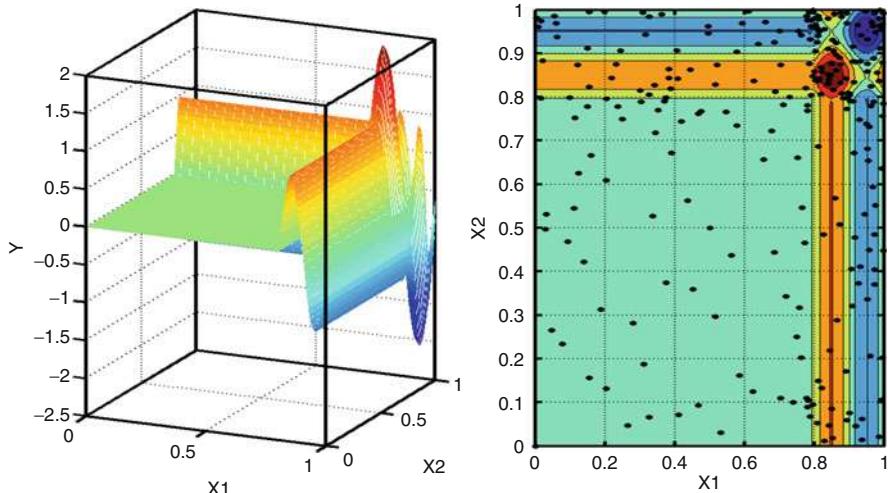


Fig. 50.6 An example of adaptive response surface analysis (dots denote sample points)

a good response surface fit. If more sample points are needed, adaptive sampling refinement can be performed using the `a_refine` command:

```
[Linux prompt] psuade
...
psuade> load <name of your sample file>
psuade> a_refine
...
How many sample points to add? <enter a number>
...
psuade> write <new sample file>
psuade> quit
[Linux prompt]
```

The steps above may be repeated multiple times to manually increment the sample size until the response surface errors are acceptable. This process can also be automated via PSUADE's batch processing mode (an example is provided in the software package). Figure 50.6 shows the result of applying adaptive response surface analysis to the following function (observe the higher density of sample points, represented by “*dots*,” in the highly oscillatory regions):

$$Y = \begin{cases} 0 & \text{if } X_1 \leq 0.8, X_2 \leq 0.8 \\ \sin(10\pi(X_1 - 0.8)) & \text{if } X_1 > 0.8, X_2 \leq 0.8 \\ \sin(10\pi(X_2 - 0.8)) & \text{if } X_1 \leq 0.8, X_2 > 0.8 \\ \sin(10\pi(X_1 - 0.8)) + \sin(10\pi(X_2 - 0.8)) & \text{otherwise} \end{cases}$$

In this case the initial sample size is 50 and 10 refinements are performed each with an additional 20 sample points, so the total number of sample points is 250. Subsequently, the MARS response surface is used to produce the response surface plot.

2.3 Uncertainty Analysis

The term “uncertainty analysis” is distinguished from “uncertainty quantification” in the sense that the former pertains narrowly to the estimation of the statistical moments and other metrics for characterizing model output uncertainties. For example, aleatory uncertainties can be described by basic statistics such as mean, standard deviation, skewness, and kurtosis. In addition, confidence of the calculated mean can be assessed by computing the standard error of mean. These quantities can then be used to certify or to quantify the uncertainty of a design under study. On the other hand, epistemic uncertainties should be quantified by lower and upper bounds. As for mixed uncertainty types, more complex mechanisms, such as probability bounds analysis, are more appropriate.

2.3.1 Uncertainty Analysis Methods in PSUADE

There are three types of uncertainty analysis in PSUADE:

1. Uncertainty analysis performed directly on large samples
2. Uncertainty analysis performed on response surfaces
3. Aleatory-epistemic uncertainty analysis performed on response surfaces

Since the accuracy of the estimated uncertainty metrics depends largely on the sample size (tens of thousands) and simulation costs are generally expensive, response surface-based uncertainty analysis is to be preferred in most cases. The first two types above are for uncertain parameters with known probability distribution functions. When there are also uncertain parameters that are epistemic – meaning that their probability distributions are not known – the third type is more suitable. In this mixed case, the epistemic parameters are described by their lower and upper bounds only, and ensemble uncertainty analysis is performed in a double inner-outer iterative loop. For each outer iteration, a sample point is drawn from the epistemic variables and uncertainty analysis is performed on the aleatory variables. The final results will be an ensemble of cumulative distribution functions (CDFs). The lower and upper bounds of these ensemble CDF plots provide useful information for, for example, probability bound analysis in the presence of epistemic uncertainty.

2.3.2 How to Perform Uncertainty Analysis in PSUADE

The steps to perform response surface-based uncertainty analysis using PSUADE are as follows:

1. Create a response surface for the simulation model by following the steps in the response surface analysis section. Upon completion, a sample file (say, `psdata`) and a suitable curve-fitting method will be available for the next step.
2. This step creates a large sample to be propagated through the response surface. Take the response surface created previously for the Ishigami function; first create a PSUADE input file (say, `psuadeRSUA.in`) as follows (note the absence of a driver since the objective is to create a sample only):

```

PSUADE
INPUT
    dimension = 3
    variable 1 X1   = -3.1416  3.1416
    variable 2 X2   = -3.1416  3.1416
    variable 3 X3   = -3.1416  3.1416
    PDF  1  N  0  0.6
    PDF  2  T  0  3.1416
    PDF  3  N  0  0.6
    COR  1  3  0.1
END
OUTPUT
    dimension = 1
    variable 1 Y
END
METHOD
    sampling = MC
    num_samples = 100000
END
END

```

The large sample will be drawn from a multivariate probability distribution function specified in the **INPUT** section where inputs 1 and 3 are normal distributions with zero means, standard deviations of 0.6, and a correlation of 0.1 (correlation is allowed only for normal distributions), and input 2 has a triangular probability distribution centered at 0 with a spread of π on each side. Other distribution types available in PSUADE are:

- (a) Lognormal parametrized by a $\log(\mu)$ and a standard deviation
- (b) Beta distribution parametrized by $\alpha > 0$ and $\beta > 0$
- (c) Weibull distribution parametrized by $\lambda > 0$ and $k > 0$
- (d) Gamma distribution parametrized by $\alpha > 0$ and $\beta > 0$
- (e) Exponential distribution parametrized by $\lambda > 0$
- (f) F distribution parametrized by $d_1 > 0$ and $d_2 > 0$
- (g) User-specified distribution provided in a sample file

3. Run this PSUADE input file to create a large sample in the file `sample`:

```
[Linux prompt] psuade psuadeRSUA.in
...
[Linux prompt] mv psuadeData sample
```

4. Perform uncertainty analysis using **rsua** or **rsuab** in command line mode:

```
[Linux prompt] psuade
...
psuade> load psdata
load complete : nSamples = 500
            nInputs = 3
            nOutputs = 1
psuade> rsua
Please enter your choice (response surface
type) : 18 <Kriging>
...
Output distribution plots are in matlabrsua.m.
psuade> quit
```

Upon completion, **matlabrsua.m** can be run in Matlab for viewing the probability distribution function. Alternately, one can use **rsuab** (**rsua** with bootstrapping) to also account for response surface uncertainties. An example Matlab plot is given in Fig. 50.7.

5. If the set of uncertain parameters contains both aleatory and epistemic parameters, a mixed uncertainty analysis can be performed using the **aeua** command. A PSUADE session to perform this analysis on the Ishigami function with X_2

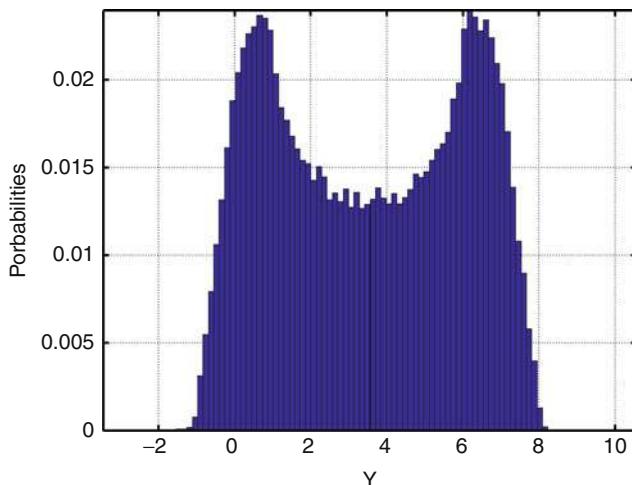


Fig. 50.7 Output probability distribution for the Ishigami function

as an epistemic variable is given below. In this analysis X_2 was re-ranged to $[-0.2, 0.2]$ and both X_1 and X_3 were given a normal distribution with zero mean and a standard deviation of 0.6. (Note: these re-ranged settings are to be provided in the sample file psdata. In addition, the response surface to be used in this analysis should also be set in this file.)

```
[Linux prompt] psuade
....
psuade> load psdata
load complete : nSamples = 500
            nInputs = 3
            nOutputs = 1
psuade> aeua
....
Step 1: select aleatory and epistemic parameters
Select epistemic parameters (1 - 3, 0 if
done) : 2
Select epistemic parameters (1 - 3, 0 if
done) : 0
....
Plot file for aleatory-epistemic analysis is now
in matlabaeua.m.
psuade> quit
```

Upon completion, `matlabaeua.m` can be run in Matlab for viewing the ensemble cumulative distribution function (CDF) plots. An example plot is given in Fig. 50.8:

2.4 Quantitative Sensitivity Analysis

Sensitivity analysis (SA) is the study of how the output variation of a model describing a certain static or dynamic process can be accounted for by variations in the model uncertain parameters. Only global sensitivity analysis, which studies the effects of the variations of uncertain parameters on the model outputs in the entire allowable ranges of the parameter space, is considered here. Saltelli et al. [13, 14] defines global methods by two properties:

1. The inclusion of influence of scales and shapes of the probability density functions for all inputs.
2. The sensitivity estimates of individual inputs are evaluated while varying all other inputs.

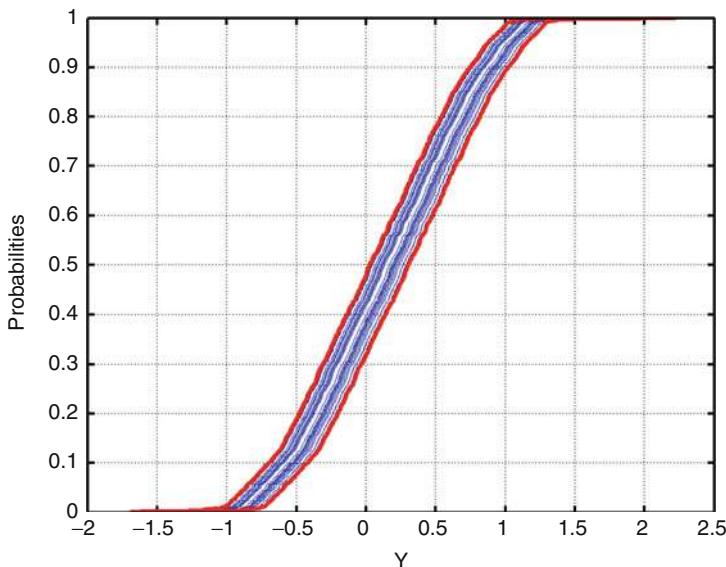


Fig. 50.8 Aleatory-epistemic CDF plot for the Ishigami function

One popular metric for quantitative sensitivity analysis is based on variance decomposition. This is a suitable metric if the objective is to quantify the contribution of each uncertain parameter toward the total output variance (i.e., metric that quantifies the percentages of the output variance from individual parameters).

2.4.1 Sensitivity Analysis Methods in PSUADE

There are three common types of variance-based sensitivity analysis:

- Main effects (first-order sensitivities: `rssobol1`, `rssobol1b`)
- Pairwise effects (second-order sensitivities: `rssobol2`, `rssobol2b`)
- Total order sensitivities (`rssoboltsi`, `rssoboltsib`)

where, for example, `rssobol1` is the PSUADE command for main effect analysis and `rssobol1b` is its bootstrapped version, which also provides uncertainty bounds for the sensitivity measures.

Since variance-based sensitivity analysis requires large samples to achieve sufficient accuracy, it is best performed on response surfaces that have been validated, although tools for computing sensitivity metrics directly from the samples are also available. Variance-based sensitivity analysis tools are available in both batch and command line modes. An example is given below for command line mode only.

The steps to perform uncertainty analysis using PSUADE are as follows:

1. Create a response surface for the simulation model by following the steps in the response surface analysis section. Upon completion, a sample file (say, `psdata`) and a suitable curve-fitting method will be available for the next step.
2. Launch and run PSUADE's command line interpreter as follows:

```
[Linux prompt] psuade
psuade> load psdata <This is a sample for the
Ishigami function)
...
psuade> rssobol1b
....
How many bootstrapped samples to use (10 - 300):10
...
Sobol1 Statistics (based on 10 replications):
Input 1: mean = 4.17199676e-01, std = 4.63387907e-02
Input 2: mean = 4.79228971e-04, std = 1.51545507e-03
Input 3: mean = 5.82298908e-01, std = 4.68131716e-02
rssobol1b plot file = matlabrssobol1b.m
...
psuade> quit
```

3. View parameter sensitivities by running `matlabrssobol1b.m` in Matlab. Figure 50.9 shows an example of both the first- and second-order (with first-order) sensitivity analysis results (second-order sensitivity analysis produces a Matlab file called `matlabrssobol2b.m`). Comparing the first- and second-order sensitivity plots, the second-order sensitivity (which includes first-order sensitivity) plot shows that there is some nontrivial interaction between X_1 and X_3 , which is clear from inspecting the Ishigami function. Similarly, total sensitivity analysis can be performed using the same procedure.

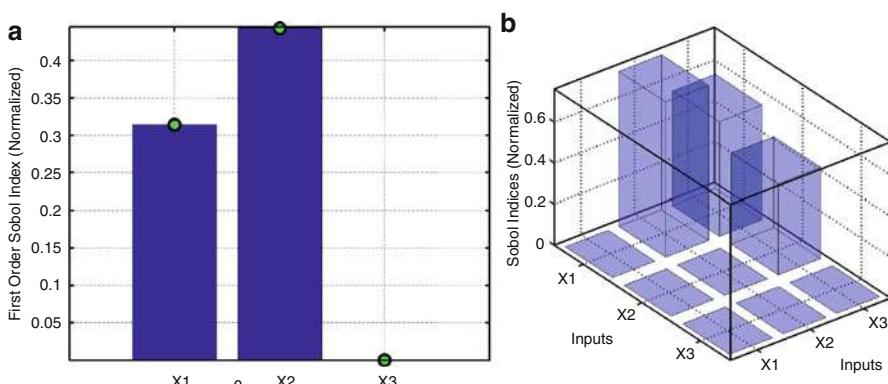


Fig. 50.9 Sensitivity analysis results. (a) First-order sensitivities. (b) First- and second-order sensitivities

Main effect analysis also supports the use of parameter distributions (which should be specified in the sample file, `psdata`). In addition, the `rssobolg` command enables group (dividing the parameter space into groups of parameters) sensitivity analysis.

2.5 Parameter Inference

A basic element of model validation is the comparison of data generated by a simulation model with experimental data. Due to the many uncertainties and unknowns in creating a simulation model, perfect match between simulation outputs and experimental data can rarely be achieved without some parameter tuning or model calibration. Calibration is essentially an optimization problem: given a set of parameters and their ranges, search for the parameter values that best match the data. Two optimization approaches can be used for model calibration, namely, deterministic numerical optimization and Bayesian optimization.

2.5.1 Deterministic Optimization Methods in PSUADE

In deterministic optimization, an objective function (e.g., sum-of-squares error) is first constructed from simulation model outputs and experimental data. In addition, the optimization (or design) parameters are to be specified with their initial guesses and their bounds. PSUADE currently only provides several derivative-free global optimization methods for bound-constrained problems: e.g.,

- `bobyqa`: a bound-constrained optimizer by Powell [11]
- `SCE`: a shuffled complex evolution method by Duan [2]
- `MM`: a two-level optimizer by Echeverria [3]

An example PSUADE input file (say, `psuadeBobyqa.in` to set up the optimization is given below.

```

PSUADE
INPUT
dimension = 5
variable 1 X1 = -2.0    2.0
variable 2 X2 = -2.0    2.0
variable 3 X3 = -2.0    2.0
variable 4 X4 = -2.0    2.0
variable 5 X5 = -2.0    2.0
END
OUTPUT
dimension = 1
variable 1 Y
END
METHOD

```

```

    sampling = MC
    num_samples = 1
END
APPLICATION
    opt_driver = simulator
END
ANALYSIS
    optimization method = bobyqa
    optimization tolerance = 1.000000e-06
END
END

```

This example has 5 optimization parameters with lower and upper bound -2 and 2 , respectively. The optimizer is **bobyqa** and the initial guess is a point drawn using a Monte Carlo sampling scheme (it is also possible to specify a user-generated initial guess).

Suppose the simulation model **simulator** is a 5-parameter Rosenbrock function compiled from:

```

#include <math.h>
#include <stdio.h>
main(int argc, char **argv) {
    int n, i;
    double X[5], Y=0.0;
    FILE *fileIn = fopen(argv[1], "r"), *fileOut;
    fscanf(fileIn, "%d", &n);
    for (i = 0; i < n; i++) fscanf(fileIn, "%lg", &X[i]);
    for (i = 0; i < n-1; i++)
        Y += (pow(1 - X[i], 2.0) + 100 * pow(X[i+1]
        - X[i]*X[i], 2.0));
    fileOut = fopen(argv[2], "w");
    fprintf(fileOut, "%24.16e\n", Y);
    fclose(fileIn); fclose(fOut);
}

```

The optimal solution will be obtained by launching PSUADE with

```

[Linux prompt] psuade psuadeBobyqa.in
....
PSUADE OPTIMIZATION : CURRENT GLOBAL MINIMUM -
    X      1 = 1.00000014e+00
    X      2 = 1.00000028e+00
    X      3 = 1.00000058e+00
    X      4 = 1.00000115e+00
    X      5 = 1.00000231e+00
                    Ymin = 1.80113178e-12

```

There are other optimization options and parameters that can be set to fine-tune the process (e.g., convergence tolerance and maximum number of function evaluations).

2.5.2 Bayesian Inference Methods in PSUADE

To describe the complex operations in inference, a more formal presentation is given below.

Let the simulation model be given by $Y = M(X, \theta)$ where X is some design parameter (assumed to be scalar for reason of simplification) and θ is an uncertain parameter that needs to be calibrated, and let $D(X^*)$ be the measurement data (also assumed scalar) at some design configuration X^* . The Bayes' formula relates the best θ values that match $D(X^*)$ via

$$\pi(\theta|D(X^*)) \propto p(D(X^*)|\theta)p(\theta)$$

where $p(\theta)$ (called the “prior” distribution) represents the initial knowledge of the likely values of θ ; $p(D(X^*)|\theta)$ is the likelihood function (probability that a specific setting of θ exactly produces $D(X^*)$ from the simulator; and $p(\theta|D(X^*))$ (called the “posterior”) represents the revised knowledge of the likely values of θ based on matching simulation results against measurements $D(X^*)$.

A popular likelihood function is the exponentiation of some error metric such as the weighted sum-of-squares errors:

$$p(D(X)|\theta) = C \exp\left(-0.5 \frac{(D(X^*) - M(X^*, \theta))^2}{\sigma_D^2}\right).$$

where σ_D is the standard deviation of $D(X^*)$ (measurement noise) and C is some normalization constant. A popular method for computing the posteriors is the Markov Chain Monte Carlo (MCMC) method using Gibbs or Metropolis-Hastings samplings.

Suppose an imperfect simulation model is given, which results in the presence of systematic errors between the model outputs and measurements. A set of measurement data $D(X_i^*)$, $i = 1, \dots, m_e$ is also given to mitigate the model imperfection. It has been shown [9] that accounting for systematic errors may have significant effect for robust calibration. One approach to model the systematic errors is to introduce a discrepancy model $\delta(X)$ built from the simulation outputs and measurement data:

$$\delta(X_i^*) = D(X_i^*) - M(X_i^*, \theta^*) \quad i = 1, \dots, m_e$$

where θ^* is a preselected set of values for the uncertain parameters (based on best prior knowledge about the calibration parameter). This preselection is needed to alleviate the “identifiability” problem). This discrepancy data set is used to construct the corresponding discrepancy function $\delta(X)$ using any regression technique

(but preferably global methods such as polynomial regression or Gaussian process). With this additional discrepancy model, the calibration process proceeds on the following modified simulation model

$$\tilde{M}(X, \theta) = M(X, \theta) + \delta(X)$$

in place of $M(X, \theta)$ in the Bayes' formulas.

2.5.3 An Example for Bayesian Inference Using PSUADE

To illustrate Bayesian inference with discrepancy modeling, consider the following simulation model:

$$Y = \theta X$$

where X and θ are the design and calibration parameters, respectively.

To perform Bayesian calibration, follow the steps below:

1. Create a response surface for the simulation model using the following PSUADE input file:

```
PSUADE
INPUT
    dimension = 2
    variable   1 X      = 0.2      4.0
    variable   2 B      = 0.35     0.75
END
OUTPUT
    dimension = 1
    variable 1 Y
END
METHOD
    sampling = LH
    num_samples = 50
END
APPLICATION
    driver = simulator
END
END
```

where `simulator` is compiled from

```
#include <stdio.h>
main(int argc, char **argv) {
    int i, n;
    double X[2], Y;
```

```

FILE *fileIn = fopen(argv[1], "r"),
*fileOut;
fscanf(fileIn, "%d", &n);
for (i = 0; i < n; i++) fscanf(fileIn, "%lg", &X[i]);
Y = X[0] * X[1];
fileOut = fopen(argv[2], "w");
fprintf(fileOut, "%24.16e\n", Y);
fclose(fileIn); fclose(fOut);
}

```

After running PSUADE on the above input file, rename the sample output file to **simdata**. To make sure that this sample set is adequate, perform a response surface validation using, for example, the quadratic regression method.

2. Prepare the experimental data for use in constructing the likelihood function. Suppose the true model (which is generally not known) is of the following form:

$$\delta(X) = \theta^* X / (1 + 0.05X)$$

where $\theta^* = 0.65$. This true model is run to produce the following set of experimental data:

```

PSUADE_BEGIN
11 1 1 1
1 2.00e-01 1.2871287128712872e-01 0.1
2 5.80e-01 3.6637512147716234e-01 0.1
3 9.60e-01 5.9541984732824427e-01 0.1
4 1.34e+00 8.1630740393627010e-01 0.1
5 1.72e+00 1.0294659300184161e+00 0.1
6 2.10e+00 1.2352941176470591e+00 0.1
7 2.48e+00 1.4341637010676156e+00 0.1
8 2.86e+00 1.6264216972878389e+00 0.1
9 3.24e+00 1.8123924268502585e+00 0.1
10 3.62e+00 1.9923793395427605e+00 0.1
11 4.00e+00 2.166666666666670e+00 0.1
PSUADE_END

```

This data set above has been put into a special format understood by PSUADE. Specifically, the first and last lines of this file contain the keywords **PSUADE_BEGIN** and **PSUADE_END**, respectively. The second line contains four integers in this example (but in general contains 3 or more integers) denoting the number of measurements (11), the number of outputs of interest (1), the number of design parameters (1), and the input parameter index of the design parameters (1 in this example). Subsequent lines contain measurement data, with each line having the experiment number, the design value(s), the data

mean(s), and the data standard deviation(s) for each output. This data set should be presented to PSUADE as a file, say, `expdata`.

3. Launch PSUADE in command line mode and run the `rsmcmc` command, which uses the response surfaces created from the sample file (`psdata`) and also `expdata` to compute the likelihood values. The MCMC implementation in PSUADE uses Gibbs sampling with multiple Markov chains and discrepancy modeling. An example PSUADE session for this analysis is as follows:

```
[Linux prompt] psuade
...
psuade> load psdata
psuade> ana_expert    # (optional) to turn on more
MCMC options
psuade> rsmcmc
...
# When prompted for a specification file, enter
'expdata'
# When prompted for model response surface,
answer 'quadratic'
# When prompted for discrepancy option, answer 'y'
# When prompted for discrepancy response surface,
answer 'Kriging'
# When prompted for posterior sample option,
answer 'y'
# When prompted for posterior plot, select input
2 for plot
# At the end, three files will have been created
# 1. a posterior sample file ('MCMCPostSample')
# 2. a Matlab file for plotting the posteriors
('matlabmcmc2.m')
# 3. a discrepancy sample file ('psDiscrepancyModel')
```

To generate the posterior plot, launch Matlab and run `matlabmcmc2.m`. The posterior plot for parameter 2 (θ) is given in Fig. 50.10. The value of θ with peak probability is at around $\theta = 0.55$, which has been set earlier in constructing the discrepancy model. A general posterior plot is given in Fig. 50.11. The histogram plots in the figure correspond to the posterior distributions of individual parameters, while the heatmap plot corresponds to joint posterior for the of parameters.

4. Examine the discrepancy model: if discrepancy modeling is activated (as in this example), a discrepancy sample file will have been created at the end of inference. This file is in the standard PSUADE data format, which can be loaded and analyzed as described above. Specifically, it can be verified via the `rscheck` command that Kriging is a suitable response surface. The corresponding Matlab plot is shown in Fig. 50.12.

Recall that the exact discrepancy function is

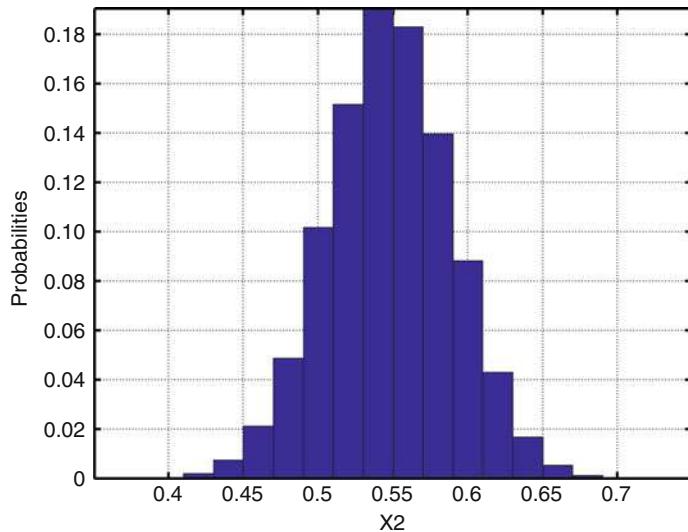


Fig. 50.10 The MCMC posterior distribution plot for this example

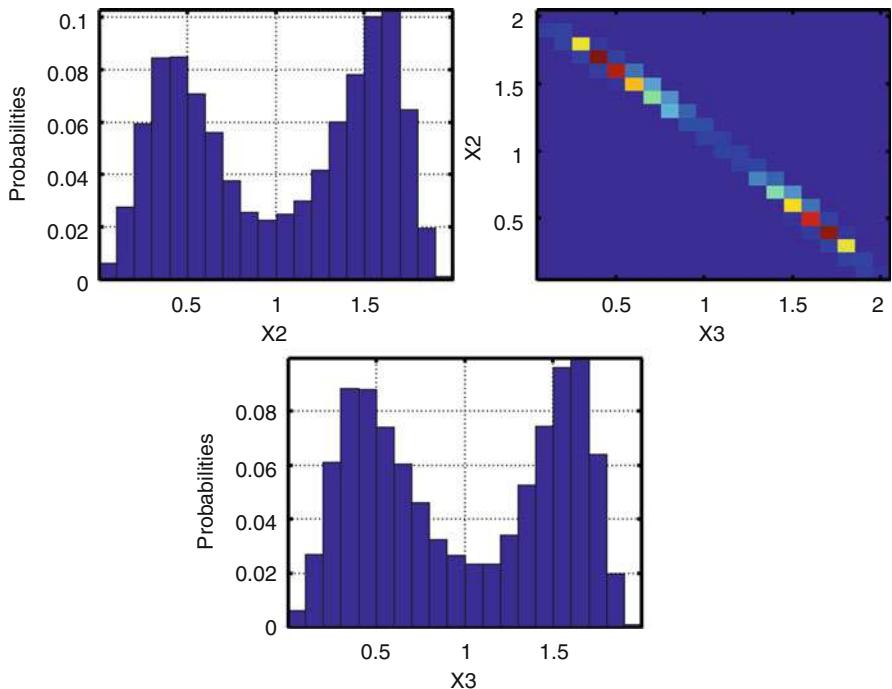


Fig. 50.11 Another example of MCMC posterior distribution plots

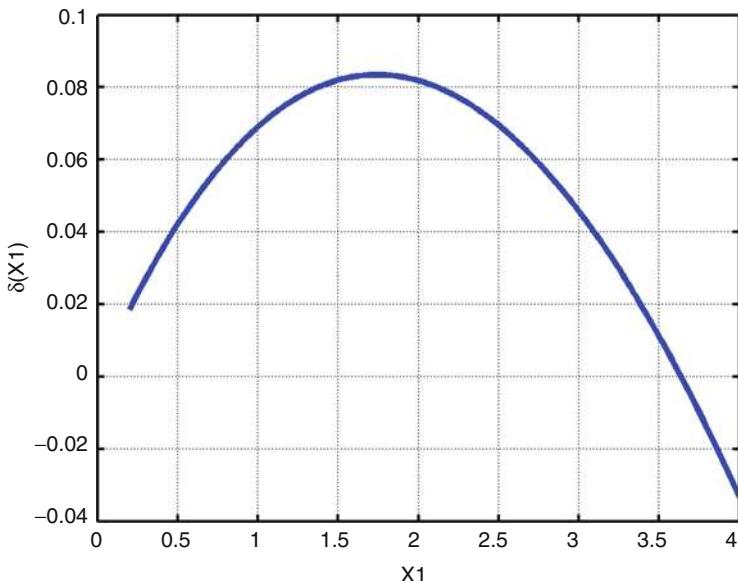


Fig. 50.12 Kriging response surface of the discrepancy model

$$\delta(X) = Y_e(X, \theta = 0.65) - Y(X, \theta = 0.55) = 0.65X/(1 + 0.05X) - 0.55 * X.$$

Observe that the discrepancy plot agrees well with this functional form.

5. Finally, the corrected model $Y^*(X, \theta) = Y(X, \theta) + \delta(X)$ can be used to predict other untested designs (other X 's). For example, to predict the model response at $X = 0.8$, do the following:

```
[Linux prompt] psuade
...
psuade> iread MCMCPostSample      # read the
          posterior sample
psuade> ireset                      # reset X = 0.8
...
psuade> write sample                # a sample around
          X = 0.8
psuade> load simdata               # load the
          simulation data
psuade> rsua                        # response surface
          -based analysis
# When prompted for discrepancy model, answer
# 'y', enter
# 'psDiscrepancyModel', and select Kriging.
# When prompted for the 'UA sample', enter
```

```

'sample'
# When prompted for model response surface, enter
'quadratic'
Output distribution plots are in matlabrsua.m.
psuade>

```

Upon completion, the prediction distribution at $X = 0.8$ can be visualized in Matlab, as given in Fig. 50.13. Observe that the prediction mean is around 0.5 consistent with the corrected model. (Note: calibration without the use of discrepancy modeling estimates the prediction mean to be 0.45) The prediction standard deviation is around 0.033, which has been induced by uncertainties in the posterior of θ .

2.6 Optimization Under Uncertainty

Consider again the simulation model $Y = M(X, \theta)$ where X is a design parameter and θ is an uncertain parameter. Suppose the objective is to minimize Y with respect to X . To include the effect of θ in the optimal solution, one approach is to minimize some functional Φ of Y with respect to θ (e.g., the mean of Y):

$$\min_X \Phi_\theta(M(X, \theta))$$

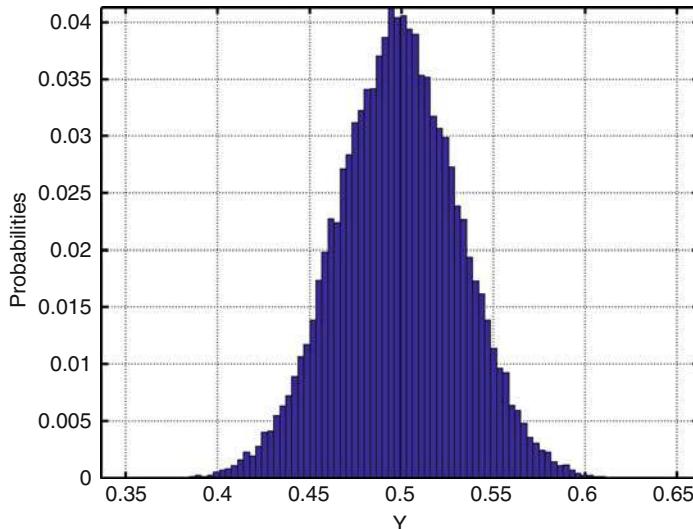


Fig. 50.13 Predicted distribution of the corrected model at $X = 0.8$

subject to some constraints on the inputs (e.g., bound, equality, or inequality constraints). A popular variant of this formulation (by computing the minimum value in the functional Φ on a discrete set of θ) is the scenario optimization problem:

$$\min_X \frac{1}{N} \sum_{i=1,\dots,N} M(X, \theta_i) p(\theta_i)$$

subject to some constraints on the inputs, where N is the number of scenarios drawn from θ and $p(\theta_i)$ is the probability of the i -th scenario.

Here N should be chosen to estimate the statistical quantities with sufficient accuracy. As such, a large sample may be needed. PSUADE provides the option to compute these estimates via response surfaces.

2.6.1 Performing Optimization Under Uncertainty in PSUADE

To solve the above problems in PSUADE, follow the steps below:

1. Create a PSUADE input file (sf psuadeOUU.in) specifying both the design (D_1, D_2) and uncertain parameters (W_1, W_2) in the INPUT section, set the optimization driver (opt_driver) to point to the simulation code ($M(X, \theta)$ or optdriver), and turn on the ouu optimization method. An example of this input file is

```

PSUADE
INPUT
    dimension = 4
    variable 1 D1 = -5      5
    variable 2 D2 = -5      5
    variable 3 W1 = -5      5
    variable 4 W2 = -5      5
END
OUTPUT
    dimension = 1
    variable 1 Y
END
METHOD
    sampling = MC
    num_samples = 1
END
APPLICATION
    opt_driver = optdriver
END
ANALYSIS
    optimization method = ouu
END
END

```

2. Run optimization in batch mode. Additional information will be requested in setting other options.

```
[Linux prompt] psuade psuadeOUU.in
# When prompted, enter the number of design and
# uncertain parameters
# When prompted, enter the sample file for the
# scenarios, or have PSUADE generate one for you
# Set other options
...
# Upon completion, the optimal values will be
# displayed.
```

PSUADE provides other more complex optimization under uncertainty solvers not covered in this chapter.

2.7 Other PSUADE Capabilities

There are other notable features in PSUADE that have been designed to assist users in improving the efficiency of certain computational analyses, diagnosing sample anomalies, and handling more complex data manipulations. Some of these features are:

1. Parallel analysis for
 - (a) The k-fold cross validation in response surface analysis
 - (b) The Kriging response surface method
 - (c) The bootstrapped quantitative sensitivity analysis methods
 - (d) The Bayesian inference using multiple chainsTo be able to use these parallel tools, PSUADE will have to be compiled with the MPI (message passing interface) library.
2. Support for a number of probability distributions such as multivariate Gaussian, lognormal, gamma, beta, Weibull, triangle, exponential, and nonanalytical distributions that are characterized by a user-provided sample (this last option is useful for the advanced hierarchical UQ analysis).
3. Read/write from/to files of different data formats
4. Sample data manipulation such as adding/deleting an input or output, input-output filtering, and sample splitting
5. Sample visualization such as input and output scatter plots (useful for detecting outliers)

In addition, a graphical user interface is currently under development to provide a more user-friendly environment for using these capabilities [7].

3 Conclusion

PSUADE comprises a rich set of mathematical and statistical tools for quantifying uncertainties and sensitivities. In addition to the analysis tools, it also provides many tools to generate Matlab files for visualizing response surfaces and uncertainties, automating the launching of ensemble simulations, as well as examining and manipulating sample data. This toolkit is suitable for quantifying uncertainties of both simple and complex simulation models.

References

1. Committee on Mathematical Foundations of Verification, Validation, Uncertainty Quantification, National Research Council: Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification. National Academies (2012). ISBN:978-0-309-25634-6
2. Duan, Q., Sorooshian, S., Gupta, V.: Optimal use of the SCE-UA global optimization method for calibrating watershed model. *J. Hydrol.* **158**, 265–284 (1994)
3. Echeverría, D., Hemker, P.W.: Manifold mapping: a two-level optimization technique. *Comput. Vis. Sci.* **11**, 193–206 (2008)
4. Eirola, E., Liitiainen, E., Lendasse, A., Corona, F., Verleysen, M.: Using the delta test for variable selection. In: European Symposium on Artificial Neural Network, Burges, 23–25 Apr 2008
5. Friedman, J.H.: Multivariate adaptive regression splines. *Ann. Stat.* **19**(1), 1–141 (1991)
6. MacKay, D.: Introduction to Gaussian processes. In: Bishop, C.M. (ed.) *Neural Networks and Machine Learning*. Springer, Berlin/New York (1998)
7. Miller, D., Ng, B., Eslick, J., Tong, C.: Advanced computational tools for optimization and uncertainty quantification of carbon capture processes. In: Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design, Cle Elum, 13–17 July 2014
8. Morris, M.D.: Factorial sampling plans for preliminary computational experiments. *Technometrics* **21**(2), 239–245 (1991)
9. Oakley, J.E., O'Hagan, A.: Probabilistic sensitivity analysis of complex models: a Bayesian approach. *J. R. Stat. Soc. B* **66**(Part 3), 751–769 (2004)
10. Plackett, R.L., Burman, J.P.: The design of optimum multifactorial experiments. *Biometrika* **33**, 305–325 (1946)
11. Powell, M.J.D.: The BOBYQA algorithm for bound constrained optimization without derivatives. Report No. DAMTP 2009/NA06, Centre for Mathematical Sciences, University of Cambridge
12. Roy, C., Oberkampf, W.L.: A complete framework for verification, validation, and uncertainty quantification in scientific computing. In: AIAA 2010-124, 48th AIAA Aerospace Sciences Meeting, Orlando, Jan 2010
13. Saltelli, A., Chan, K., Scott, E.M.: *Sensitivity Analysis*. Wiley, Chichester/New York (2000)
14. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, Chichester/Hoboken (2004)

Probabilistic Analysis Using NESSUS (Numerical Evaluation of Stochastic Structures Under Stress)

51

John M. McFarland and David S. Riha

Abstract

NESSUS is a general-purpose software program for probabilistic analysis with original development beginning in the 1980s as part of a 10-year NASA project focused on predicting risk and reliability of space shuttle main engine components. NESSUS now includes a streamlined graphical interface, 16 different reliability methods, interfaces to many third-party commercial analysis codes, and the ability to define custom interfaces to computational models. Recent developments have added powerful tools for global sensitivity analysis and have expanded the response surface methods to include Gaussian process modeling. This chapter highlights NESSUS's core capabilities and gives an overview of setting up and solving a probabilistic analysis problem using the software. A detailed tutorial is used to illustrate how to use NESSUS to perform a probabilistic analysis on a finite element model. A turbine blade case study is then presented to illustrate a practical solution strategy for assessing model uncertainties for computationally intensive models. The strategy demonstrates the use of response surface models and efficient probabilistic methods to make the best use of model evaluations to iteratively improve the uncertainty assessments.

Keywords

Probabilistic analysis • NESSUS • Software • Graphical user interface • AMV • EGRA • Finite element analysis • Response surface • Surrogate model • Gaussian process

Contents

1	Introduction.....	1734
2	Capabilities Overview.....	1735

J.M. McFarland (✉) • D.S. Riha

Mechanical Engineering Division, Southwest Research Institute, San Antonio, TX, USA
e-mail: john.mcfarland@swri.org; david.riha@swri.org

3	Problem Formulation and Solution.....	1739
3.1	Problem Statement Definition.....	1739
3.2	Random Variable Input.....	1740
3.3	Response Model Definition.....	1741
3.4	Deterministic Parameter Variation.....	1743
3.5	Probabilistic Analysis Definition.....	1743
3.6	Results Visualization.....	1744
4	Tutorial.....	1745
5	Solution Strategy Example.....	1754
6	Conclusions.....	1763
	References.....	1763

1 Introduction

NESSUS (Numerical Evaluation of Stochastic Structures Under Stress) is a general-purpose software program for probabilistic analysis. It was originally created by a team led by Southwest Research Institute (SwRI) as part of a 10-year NASA project started in 1984 to develop a probabilistic design tool for the space shuttle main engine with a focus on probabilistic finite element analysis. The methods and capabilities in NESSUS were designed to support predicting the probabilistic response and/or probability of failure for computational intensive models. The input variations were modeled using probability density functions and propagated using traditional and newly developed probabilistic algorithms. In 1999, SwRI was contracted by Los Alamos National Laboratory to adapt NESSUS for application to extremely large and complex weapon reliability and uncertainty problems in support of its Stockpile Stewardship program. In 2002, SwRI was contracted by the NASA Glenn Research Center to further enhance NESSUS for application to large-scale aero-propulsion system problems. The end result of these large research programs was a completely redesigned software tool that includes a sophisticated graphical user interface (GUI), capabilities for performing design of experiments and sensitivity analysis, a probabilistic input database, a geometric uncertainty modeling tool for perturbing geometry in existing finite element models, and state-of-the-art interfaces to many third-party codes such as Abaqus, ANSYS, LS-DYNA, MSC.NASTRAN, and NASGRO.

NESSUS has seen continuous improvement and application since its beginnings in the late 1980s. Most recently, the CENTAUR (Collection of ENgineering Tools for Analyzing Uncertainty and Reliability) software library, which contains an array of methods for solving various types of problems with an emphasis on nondeterministic analysis, was developed to support NESSUS. CENTAUR provides methods for reliability analysis, distribution fitting, Bayesian uncertainty quantification, numerical optimization, and more. CENTAUR has been under active development since 2008 and provides several of the reliability analysis methods used by NESSUS.

NESSUS has been applied to a diverse range of problems to support decisions in such areas as aerospace structures, automotive structures, biomechanics, gas turbine engines, geomechanics, offshore structures, pipelines, pressure vessels, space

systems, and weapon systems. These probabilistic and uncertainty quantification analyses have supported decisions in design, failure analysis, and model verification and validation (V&V). The ability to quantify uncertainties in model predictions enables decisions to be made in light of known limitations in the models and data. Probabilistic approaches are also powerful for design analysis to evaluate the impact of variations in new processes, environments, and other uncertainties, especially early in the design process when material and process data are limited.

Uncertainty quantification is a key component of model V&V, which is a framework for collecting evidence and building credibility to substantiate that a model is adequate for its intended use. A successful V&V program includes assessments of the uncertainties in both experimental and simulation results, as well as sensitivity analysis to identify important variables. NESSUS provides a suite of uncertainty propagation methods to support V&V through quantification of uncertainty in the model output. In addition, NESSUS can be used to conduct various types of sensitivity studies, which may be used to guide model development and experimental activities, or to substantiate model assumptions/simplifications.

Commercial and academic licenses for NESSUS may be obtained directly from Southwest Research Institute. More information as well as demonstration licenses can be obtained through the website at www.nessus.swri.org. NESSUS is available for the Windows, Mac, and Linux platforms. NESSUS is installed by simply downloading and running the installer file, and the provided license key is then installed using the GUI.

This chapter gives an overview of the NESSUS software and its capabilities, followed by a description of the steps involved in setting up and executing a probabilistic analysis using NESSUS. Next, a detailed tutorial is presented that demonstrates a probabilistic analysis for a simple finite element model of a plate with a hole. Lastly, a case study involving a turbine blade is presented to illustrate a practical solution strategy to answer questions about the importance of potential uncertainties in the model. The strategy demonstrates the use of response surface models and efficient probabilistic methods to make the best use of model evaluations to iteratively improve the uncertainty assessments.

2 Capabilities Overview

NESSUS allows users to perform probabilistic analysis with analytical models, external computer programs such as commercial finite element codes, and general combinations of the two. NESSUS includes a graphical user interface that facilitates formulation of the problem and visualization of results.

While NESSUS was originally developed for reliability analysis by propagation of random variables through one or more performance models, it now contains a variety of related capabilities for working with models. These include generation of designs of experiments (including space-filling Latin hypercube designs), various methods for sensitivity analysis (including global sensitivity analysis, also known as variance decomposition), and sophisticated response surface modeling capabilities.

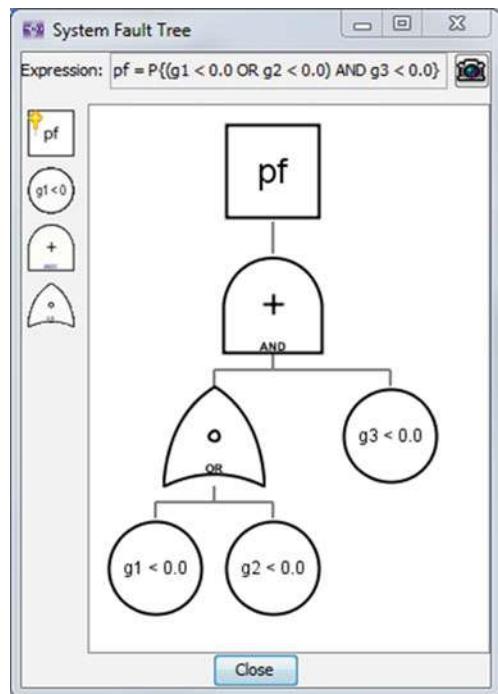
The core functionality of NESSUS is to compute the cumulative distribution function (CDF) of model responses using methods that can accurately and efficiently compute tail probabilities. The generated CDF is used to quantify the variability or uncertainty in the model predictions and to support probability of failure and reliability analyses. In NESSUS, a probabilistic analysis problem is formulated in terms of a series of equations that establish a performance function, g , which relates the input random variables, X , to the response variable, Z :

$$Z = g(\mathbf{X})$$

Next, the user may define one or more “specified performance values,” which NESSUS uses to compute probabilities. By convention, NESSUS computes the probability that the performance is less than each specified performance value. This is the traditional forward reliability analysis problem. NESSUS also allows the user to set up an inverse reliability analysis problem, in which case the user specifies the probabilities and NESSUS computes the corresponding performance levels.

NESSUS also allows users to define system-level reliability analysis problems using a graphical fault tree editor (Fig. 51.1). An individual limit-state equation is defined for each failure mode, and the system-level probability of failure is formulated in terms of these using any general combination of “AND” and “OR”

Fig. 51.1 NESSUS fault tree editor for system reliability analysis



gates. The probabilistic fault tree formulation in NESSUS correctly accounts for common variables in the failure events [1].

NESSUS includes a variety of capabilities for working with response surface models. The user may provide a set of training data and have NESSUS fit a response surface model using either polynomial or Gaussian process (GP) regression. The GP model supports several options for noise in the response data, including noise-free data (resulting in a response surface that exactly interpolates the training data), a user-defined noise level, or automatic estimation of the noise level (ideal when modeling data from physical experiments). NESSUS includes goodness-of-fit information based on leave-one-out cross-validation assessment [2] as part of the output. Response surface models created from user-provided training data can be interrogated using Monte Carlo simulation or any of the other deterministic or probabilistic analysis methods available in NESSUS.

In addition, NESSUS includes several tailored probabilistic methods that use response surface models internally. When using these methods, NESSUS automates several steps that the user would normally do, including generation of the DOE, analysis of the true performance model at each design point, creation of the response surface model, and finally random sampling using the response surface to compute a probability. The efficient global reliability analysis (EGRA) method uses an adaptive procedure designed to target response surface accuracy near the limit state to achieve an accurate and efficient probability estimate.

NESSUS has tailored capabilities for the design of computer experiments using Latin hypercube sampling. This includes the ability to search for optimal designs based on a minimum distance criterion, correlation reduction using the Iman-Conover algorithm [3], and the capability to augment designs while maintaining the Latin hypercube property.

NESSUS includes 16 different reliability methods, which include sampling methods, analytical methods, and response surface-based methods. These are listed in Table 51.1. All probabilistic methods support forward reliability analysis with a single limit state, but only a subset of the methods support inverse reliability analysis and system reliability analysis problems.

One of the original and most powerful reliability methods in NESSUS is the advanced mean value plus (AMV+) method [4]. The algorithm was developed to efficiently predict the probabilistic response for computationally intensive models. The algorithm iteratively replaces the actual model with a linear response surface to estimate the cumulative distribution function (CDF) over a range of response values. The algorithm is useful for incrementally improving CDF accuracy as the model runs are completed. AMV+ has been demonstrated to successfully identify multiple failure regions in the design space [4] and obtain solutions for noisy response functions [5], two issues that are problematic for most gradient-based probabilistic algorithms.

One of the newest methods in NESSUS is the efficient global reliability analysis (EGRA) method [6]. EGRA uses an algorithm that employs Gaussian process modeling to iteratively select points in the design space targeted at maximizing the

Table 51.1 Probabilistic analysis methods in NESSUS

Probabilistic method	Inverse analysis	System analysis
Monte Carlo sampling	X	X
Latin hypercube sampling	X	
First-order reliability method (FORM)	X	
Second-order reliability method (SORM)		
Mean value (MV)	X	
Advanced mean value (AMV)	X	
Advanced mean value with iterations (AMV+)	X	
Advanced mean value with adaptive importance sampling		
Importance sampling with radius reduction factor		X
Importance sampling with user-defined radius		
Importance sampling at user-defined MPP		
Plane-based adaptive importance sampling		
Curvature-based adaptive importance sampling		X
Efficient global reliability analysis (EGRA)	X	
Response surface method	X	
Gaussian process response surface method	X	

accuracy in the vicinity of the limit state. This targeting produces a locally accurate model sufficient for reliability analysis with fewer points than would be required to construct a globally accurate model.

Each of the reliability methods in NESSUS computes probabilistic sensitivity results, which can be used to identify important random variables. These include derivatives of the probability of failure, p , with respect to input random variable distribution parameters, μ and σ :

$$\frac{\partial p}{\partial \mu_i} \frac{\sigma_i}{p} \text{ and } \frac{\partial p}{\partial \sigma_i} \frac{\sigma_i}{p}$$

In addition, methods that employ the most probable point (MPP) concept also compute the associated importance factors associated with the random variables.

Recent work has been devoted to expanding NESSUS's sensitivity analysis capabilities to include methods for global sensitivity analysis using variance decomposition. NESSUS supports variance decomposition using "structured" Monte Carlo sampling, Fourier amplitude sensitivity test (FAST), and analytical solution using a Gaussian process response surface. The structured Monte Carlo sampling method uses an efficient scheme to compute both main and total effects using a single Monte Carlo loop [7]. Options for the use of low-discrepancy Sobol sequences, as well as random Monte Carlo or Latin hypercube sampling, are provided.

3 Problem Formulation and Solution

The process of setting up and solving a probabilistic or uncertainty analysis problem in NESSUS is broken into the following steps:

1. Declaration of variables and equations
2. Definition of basic random variables
3. Definition of numerical models
4. Probabilistic analysis type and method selection
5. Analysis and results visualization

3.1 Problem Statement Definition

The first step when setting up a new problem in NESSUS is to declare the variables and their basic relationships using the “Define Problem Statement” window. This window provides a text field for entry of one or more equations using a simple algebraic syntax. Multiple hierarchical equations can be defined on separate lines and are evaluated from the bottom to the top. For example in Fig. 51.2, first the value of “stress” is computed, and then this result is used in a subsequent equation to compute the value of “life.” For the purpose of probabilistic analysis, the response variable on the left-hand side of the top equation is treated as the overall response variable for the model (i.e., “life” in this example).

For models that have simple algebraic form, the NESSUS problem statement editor may be sufficient to completely define the functional relationships among the variables. For more complex models, the problem statement allows the use of a function notation syntax, which creates placeholders for numerical models to be defined separately. Any combination of numerical and analytical models may be used in the problem statement.

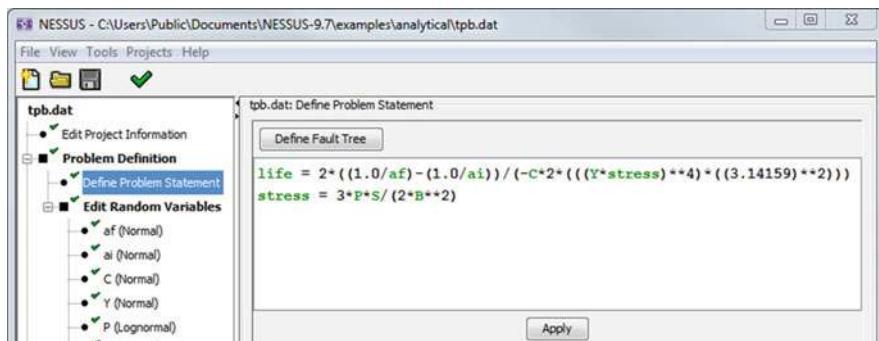


Fig. 51.2 NESSUS problem statement

3.2 Random Variable Input

Once the problem statement is successfully defined, NESSUS automatically identifies the independent variables and the computed variables. The next step for the user is to assign probability distributions to the independent variables. As shown in Fig. 51.3, NESSUS provides a graphical editor for defining the random variables. NESSUS supports 16 different continuous probability distribution models, including the beta distribution, truncated normal and Weibull distributions, and three-parameter generalized extreme value distributions. Where applicable, NESSUS allows for definition of distribution parameters using either natural parameters or mean and standard deviation. Linear (Pearson) correlations between variables are supported for many of the distribution types. In addition, independent variables in the problem statement may also be assigned fixed values and treated as deterministic variables.

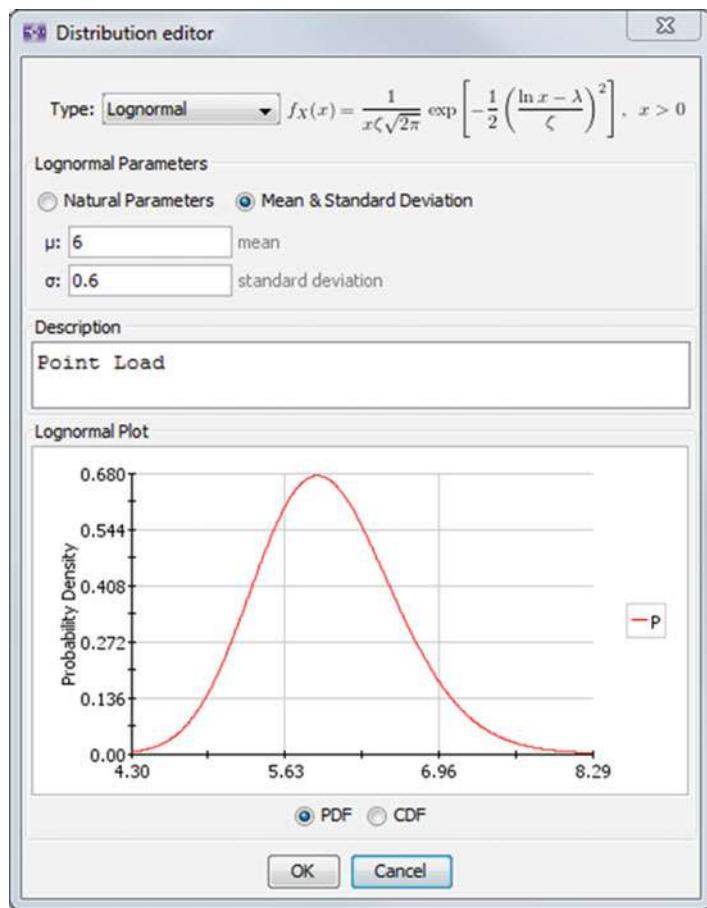


Fig. 51.3 NESSUS probability distribution editor

3.3 Response Model Definition

The next step is to define any response models that have been included in the problem statement using function notation. NESSUS provides several options for how these models may be defined, including “numerical,” “regression,” and “dynamically linked” model types. Numerical models involve the execution of external programs such as commercial finite element codes or scripts created by the user. The regression (response surface) model option includes a Gaussian process model as well as first- or second-order polynomial models. Dynamically linked models allow the user to write C, C++, or Fortran functions for model evaluation and compile the code into a shared library such as a Windows DLL, which is loaded by NESSUS at runtime.

When the “numerical” model type is selected, the “application” menu is used to define the numerical model interface. This menu provides 12 options, which include various commercial codes as well as a “user-defined” option. With the exception of the MATLAB interface, which involves the use of the shared MATLAB runtime libraries, all of the numerical model interfaces involve the use of system calls to external programs. Input and output data are passed between NESSUS and the external programs using file i/o. Input files are typically text-based files with numerical fields that NESSUS can modify. The mechanism for specifying output data is application dependent (e.g., for Abaqus, NESSUS reads directly from the binary ODB results database). The user-defined application uses plain text files for output data. Using these capabilities, it is possible for NESSUS to interface with virtually any computational model.

NESSUS provides two external model execution options: interactive and batch. For interactive operation, NESSUS will wait for each simulation to complete before starting the next analysis. This option works well for relatively fast-running models. In batch mode, NESSUS will create the analysis code input files (e.g., input files for 100 Monte Carlo samples or the initial steps in an iterative method such as AMV+) as well as a script with the execution commands for each required model analysis. However, in batch mode, the analyses are not executed. This provides the user greater flexibility in managing multiple analyses and can be used to run jobs in parallel or distribute jobs to multiple computers.

The NESSUS graphical interface facilitates the process of defining how input variables are mapped into numerical model input files. The process is very flexible and allows each model input variable to be mapped into one or more locations in one or more input files. Two mapping types are supported: “replace” and “vector scaling.” The “replace” mapping simply instructs NESSUS to overwrite a specified section of the input file with the current value of a variable. The “vector scaling” mapping can be used to control multiple fields based on the value of a single variable. For example, vector scaling can be used to define how nodal coordinates need to change based on the value of a geometric variable. This facility can be used to “parameterize” certain parts of a model, such as tabular material property definitions or geometric variables [8]. Figure 51.4 shows an example of the NESSUS editor for visually selecting fields within a numerical model input file.

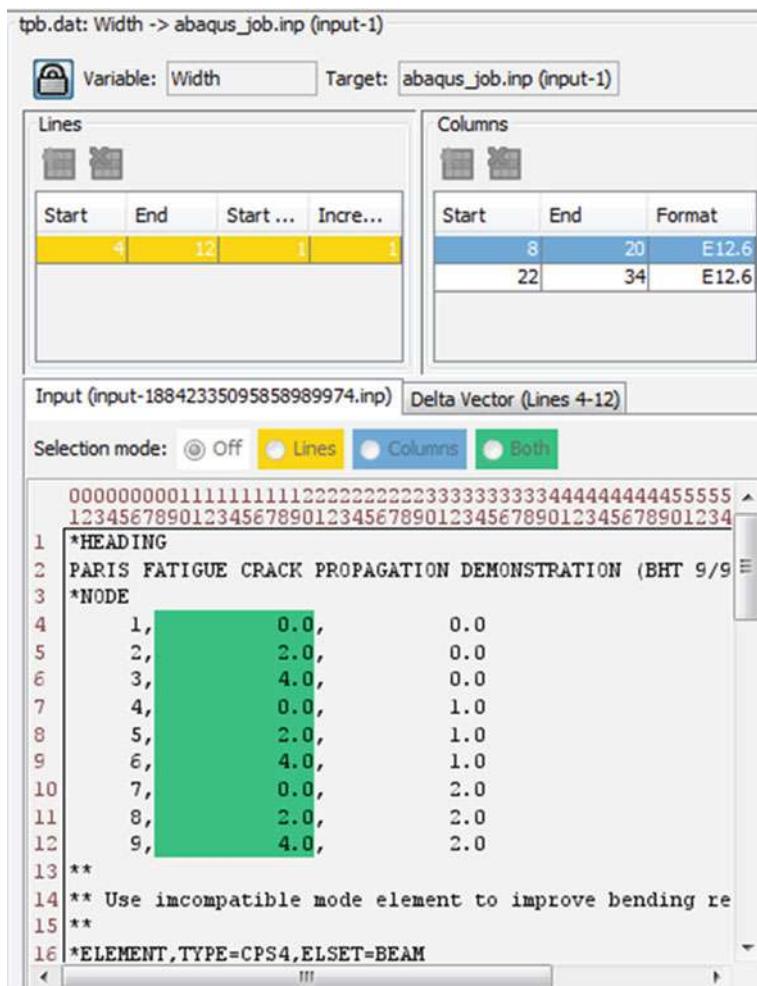
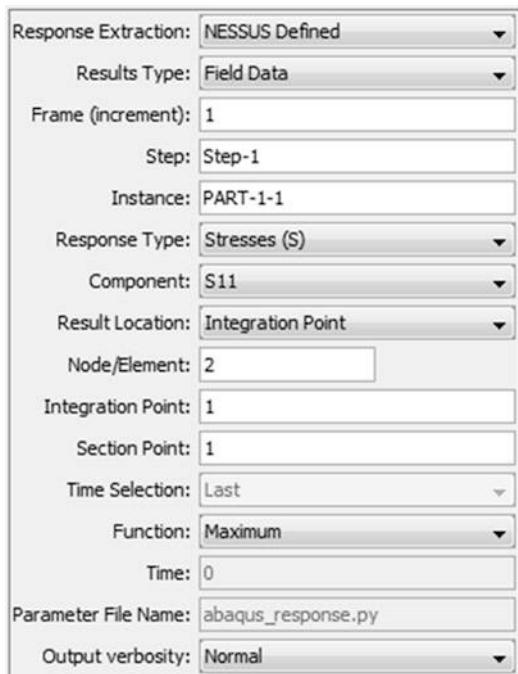


Fig. 51.4 NESSUS visual editor for definition of vector scaling mapping in numerical model input file

The options for defining the model output depend on the particular application. Customized dialogs are available for several of the commercial code interfaces. Figure 51.5 shows the menu for Abaqus, which includes options for response type and location, as well as functions for retrieving the maximum or minimum response over a specified set of nodes or elements.

When working with external analysis codes, NESSUS maintains a directory structure for each individual model run, including all input and output files, which facilitates debugging and verification of individual analysis cases. NESSUS also maintains a restart database, so that existing results are used when possible, as opposed to rerunning potentially time-consuming model evaluations.

Fig. 51.5 Definition of output variable for Abaqus response model



3.4 Deterministic Parameter Variation

The deterministic analysis section in NESSUS allows the user to exercise the models in the problem statement using a specified set of values for the variables. These values may be user defined (either manually entered into the table or imported from a file) or automatically generated based on built-in designs. The deterministic analysis capability can be used to manage or automate a series of external numerical models (e.g., finite element analyses), perform simple sensitivity studies (e.g., by perturbing each of the input variables one at a time), verify the NESSUS problem setup and model behavior (e.g., to verify that increasing the applied load results in a corresponding increase in stress), and more.

3.5 Probabilistic Analysis Definition

In NESSUS, the probabilistic analysis section defines four categories, referred to as “analysis types”:

- Specified probability levels
- Specified performance levels
- Full cumulative distribution
- Global sensitivity

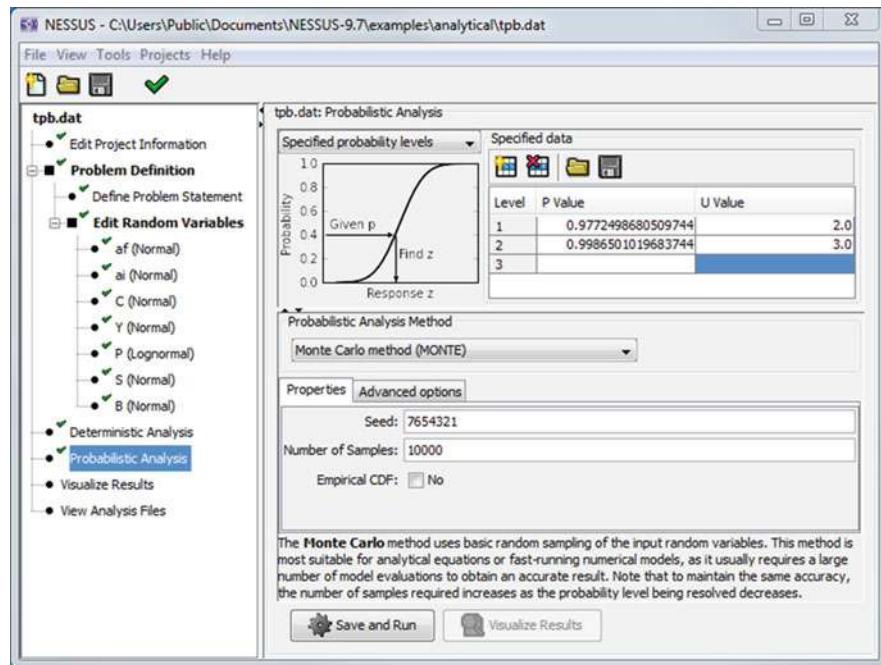


Fig. 51.6 NESSUS probabilistic analysis type and method selections

The specified probability levels and specified performance levels analysis types correspond to forward and inverse reliability analysis. In each case, the user may specify one or more “levels” at which to perform the analysis. For the full cumulative distribution analysis type, NESSUS automatically chooses a range of probability levels and performs an inverse reliability analysis at each one, in order to construct the CDF of the response variable. Global sensitivity analysis is provided as a separate analysis type, in which the objective is to compute the main and total effect indices associated with the basic random variables.

Lastly, the user selects the desired probabilistic method and configures the method-specific options, such as sample size (Fig. 51.6).

3.6 Results Visualization

The NESSUS graphical interface provides capabilities for visualizing results, which include:

- Deterministic response curves for one-variable-at-a-time studies
- Cumulative distribution function
- Derivative-based probabilistic sensitivities as a function of response level

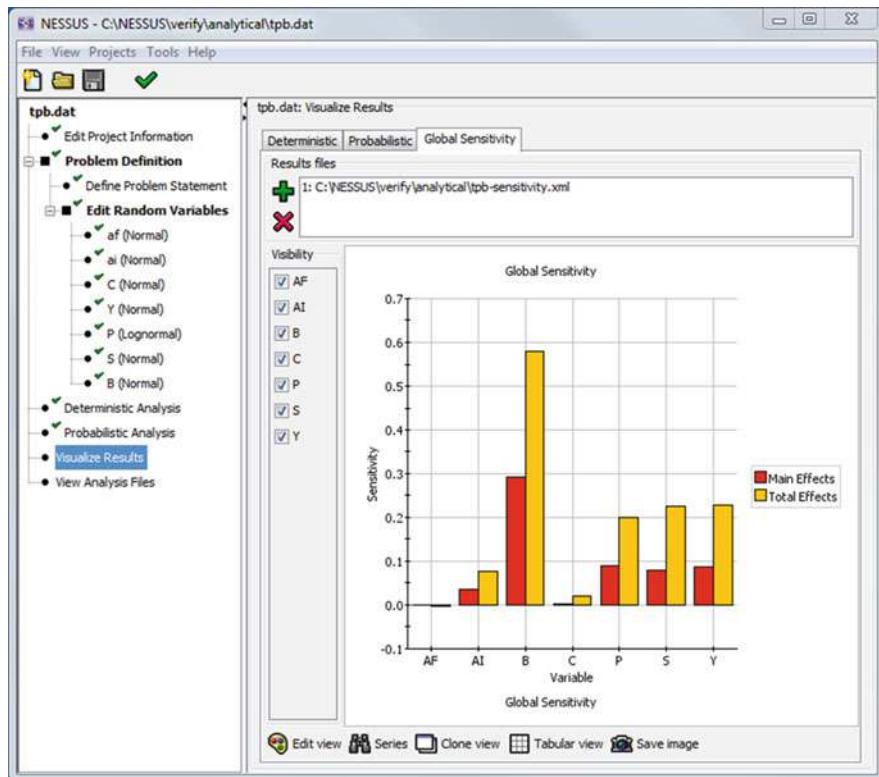


Fig. 51.7 Visualization of global sensitivity analysis results

- Probabilistic importance factors as a function of response level
- Bar charts for main and total effects from global sensitivity analysis
- Comparison of multiple results from separate analyses on the same plot

An example of the results visualization screen for global sensitivity analysis results is shown in Fig. 51.7. The graph shows the main and total effect sensitivity indices for each of the random variables defined in the problem statement.

4 Tutorial

The NESSUS problem formulation, analysis, and results visualization are demonstrated in this tutorial using a finite element model of a plate with a hole. The model predicts the stress concentration at a hole in a long plate of finite width due to end plate loads. The plate geometry and finite element model are shown in Fig. 51.8. The random variables are defined in Table 51.2. The far-field applied point force P is applied as an edge pressure to more accurately model the long plate condition.

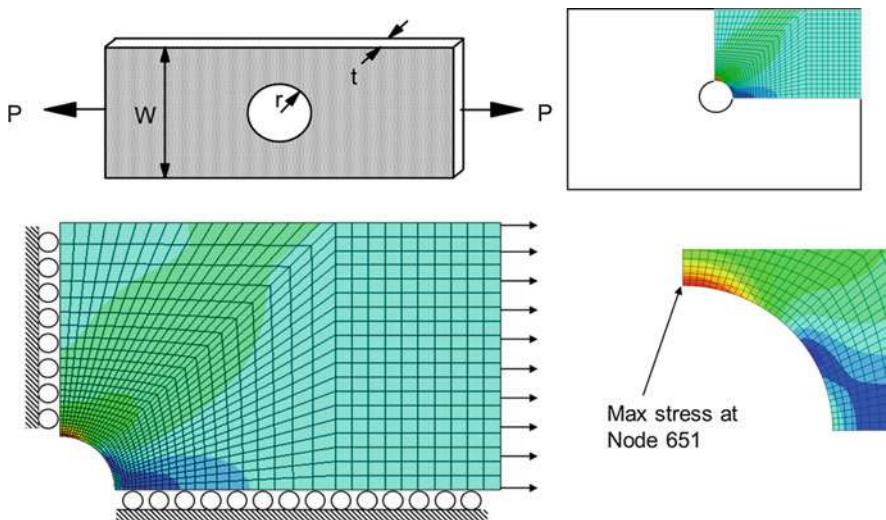


Fig. 51.8 Geometry and variables of a plate with a hole under uniform loading

Table 51.2 Random variable definitions for the plate with a hole probabilistic study

Variable	Description	Type	Mean	Standard deviation	Distribution
t	Thickness (in.)	Geometry	0.1	0.005	Lognormal
w_full	Width (in.)	Geometry	5.0	0.25	Lognormal
r	Radius (in.)	Geometry	0.5	0.025	Lognormal
P	Applied force (lbs)	Loading	7000	700	Normal
sigy	Yield stress (psi)	Material	80,000	4000	Normal

The model uses a linear elastic material definition ($E = 300,000$ psi, $\nu = 0.3$). An Abaqus CAE parametric model was developed that can create plate models for different values of the random variables. The geometry parameters in the Abaqus CAE model are used in the point and line definitions to create the plate geometry. A portion of the CAE script is shown in (Fig. 51.9) where the radius variable r is used in point definitions to create an arc for the hole in the plate. The script also shows the conversion of the applied force to edge pressure used for the loading.

For this example, NESSUS will be used to predict the variability in peak stress caused by the random variables defined in Table 51.2 and the probability of failure caused by exceeding the material's yield strength. Once the deterministic Abaqus CAE model has been developed, the first step is to define the problem statement in NESSUS, as shown in Fig. 51.10. In this example, the problem statement specifies the relationship between the response quantity (stress) and the basic random variables (plate geometry and loading). In this case, two equations are used, which are evaluated from the bottom to the top. The first equation divides the full width of the plate (w_{full}) by 2 to account for the use of symmetry in the finite element model. This second equation declares a function called "fe," which accepts the plate

```

.
.
.

# Model parameters
r = 0.5000      # Plate Radius
w = 2.5000      # Plate Width
F = 7000.000    # Load
t = 0.1000      # Plate Thickness
P = -F/(2*w*t) # Converts LOAD to pressure
L = 1.500       # Plate Length = L+W
.
.

mdb.models['Modell'].sketches['__profile__'].ArcByCenterEnds(center=(0.0,
0.0), direction=COUNTERCLOCKWISE, point1=(r, 0.0), point2=(0.0, r))

```

Fig. 51.9 Portion of the Abaqus CAE script for the plate with a hole problem showing the relationship of the parametric values and point and line definitions in the model

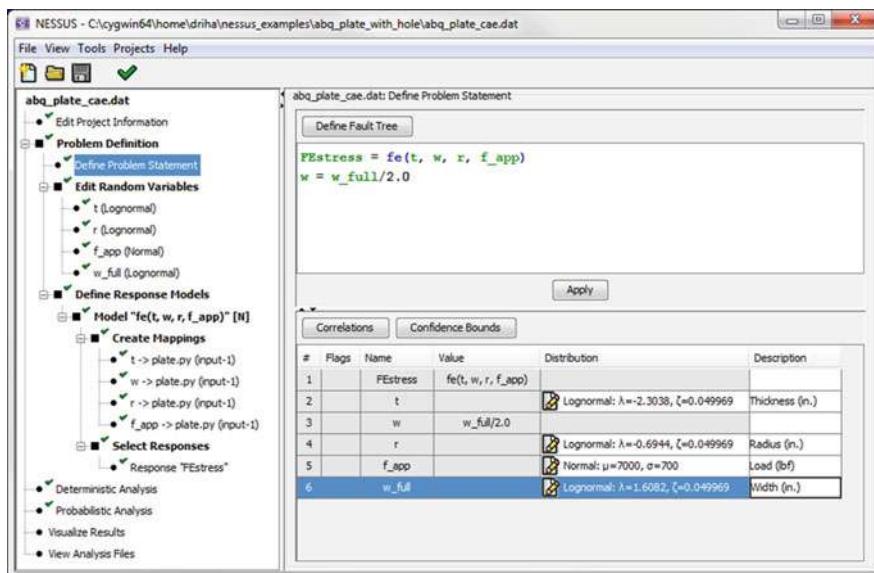


Fig. 51.10 NESSUS problem statement for the plate with a hole probabilistic study

half-width (w) as well as three additional inputs as arguments. Any name for the function can be chosen by the user, with the exception of the intrinsic functions such as \sin , \cos , and \exp . After clicking the Apply button, NESSUS recognizes the user-defined function and creates an entry for it in the outline for later definition.

Probability density functions are defined for the independent variables either in the Problem Statement section (Fig. 51.10) or in the Edit Random Variables section (Fig. 51.11) of the GUI. To edit a random variable directly from either of these screens, double-click on the pen icon under the “Distribution” column

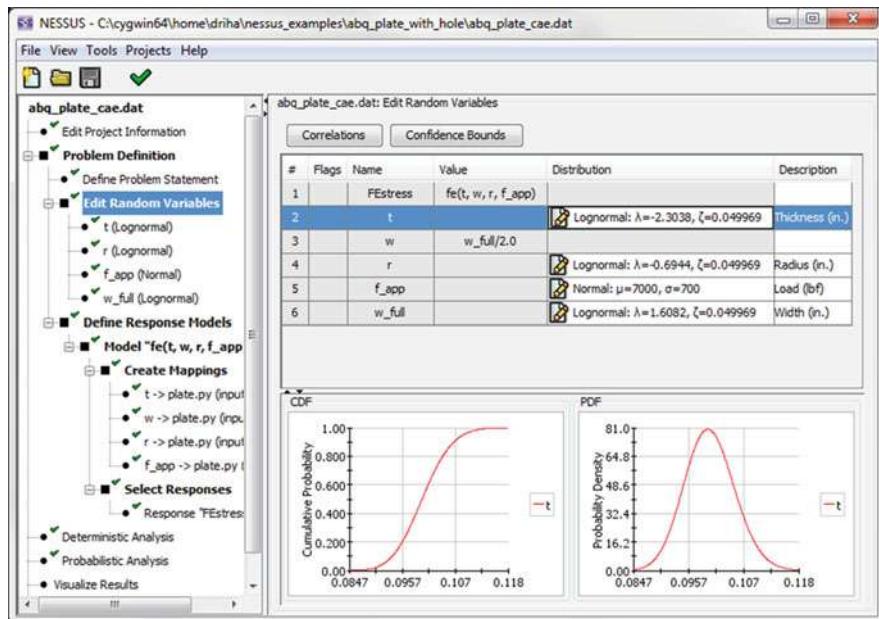


Fig. 51.11 Random variable definitions for the plate with a hole probabilistic study

in the table. This brings up an editor, as shown in Fig. 51.3. Using the editor, select the probability distribution type (e.g., normal, lognormal, etc.) and define the distribution parameters. Most probability distributions in NESSUS can be defined using either the mean and standard deviation or the natural parameters (e.g., shape and scale for a Weibull distribution). Each random variable is assigned its own probability distribution function. Correlations between any of the random variables may also be defined by clicking on the “Correlations” button.

Next, the function fe is assigned to the Abaqus numerical model interface capability in NESSUS as shown in Fig. 51.12. This screen defines the required input and output files and how to execute Abaqus. The deterministic Abaqus CAE script is defined in the Input files section under Source. NESSUS will modify this file based on variable mapping definitions (described in the next step) and write the modified file to the Destination file in a unique directory. NESSUS will then execute the commands in the Execution command window. The execution command for this problem requires two steps. First, Abaqus CAE is used to create the finite element model followed by executing the finite element analysis using Abaqus standard. Note that consistency is required between the execution commands and the Destination file and Output file names. For this problem, the destination filename plate.py is used in the first command. When run by Abaqus CAE, the plate.py script creates the abaqus_job.inp file that is used in the second command to run the finite element analysis. This analysis creates the abaqus_job.odb results file from which NESSUS will read the stress results.

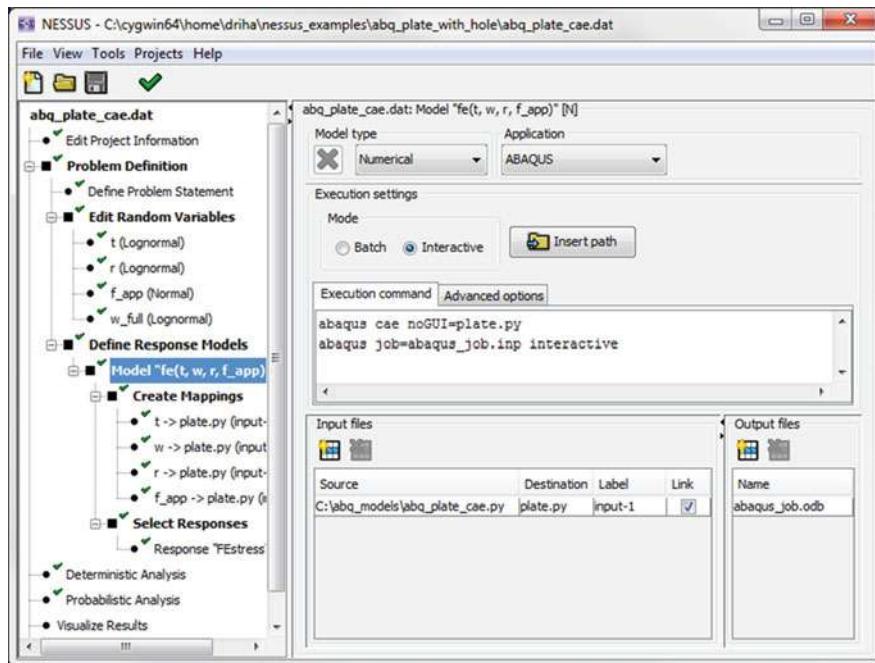


Fig. 51.12 Abaqus model definition for function fe

The next step is to map the variables defined in the fe function argument list to the Abaqus CAE script. Mappings are defined in the Create Mappings section under the Model fe (Fig. 51.13). The file and type of mapping for each variable are defined under Actions in this section of the GUI. Variables can be mapped multiple times into multiple files providing flexibility in relating random variables to model inputs. The “replace” type mapping is used for each variable in this model since the geometry has been parameterized in the Abaqus CAE script in terms of the random variables. NESSUS provides a graphical capability to map the variables to lines and columns in the source file. The mapping for the thickness variable t is shown in Fig. 51.14. The lines and columns in the tables are automatically populated when the lines and columns are graphically selected in the source file. These line and column numbers can also be inserted manually. Replace mappings can be applied to multiple lines and columns. Additional line blocks and columns can be added as needed. Line blocks are used to map to different line sections within the same file and mapping.

It is also necessary to define the numerical format to be used by NESSUS when writing values into the input file. This must be consistent with available field width in the input file, and the user must take care to provide sufficient precision so that changes made to the value of the variable by the NESSUS probabilistic analysis algorithms are correctly reflected in the numerical model input file. In this example,

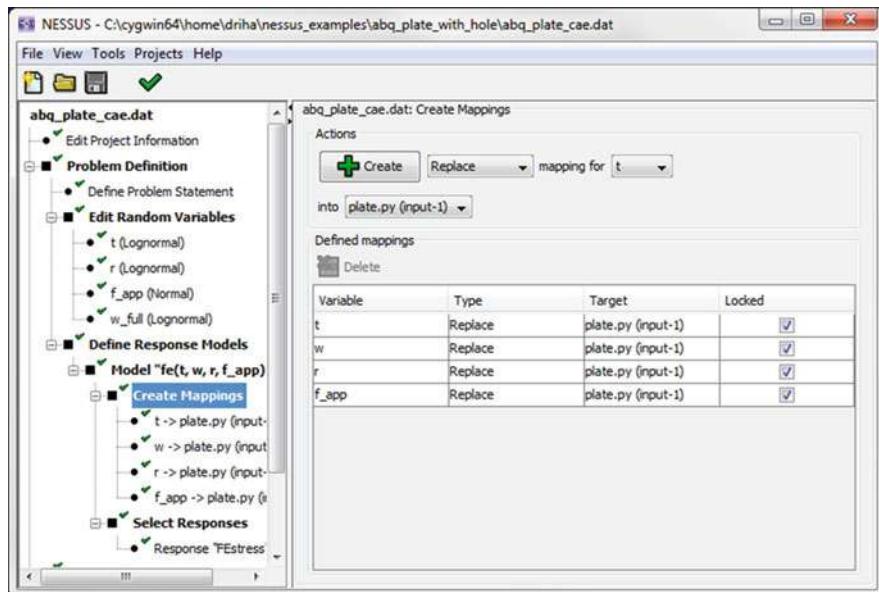


Fig. 51.13 Variable mapping definition

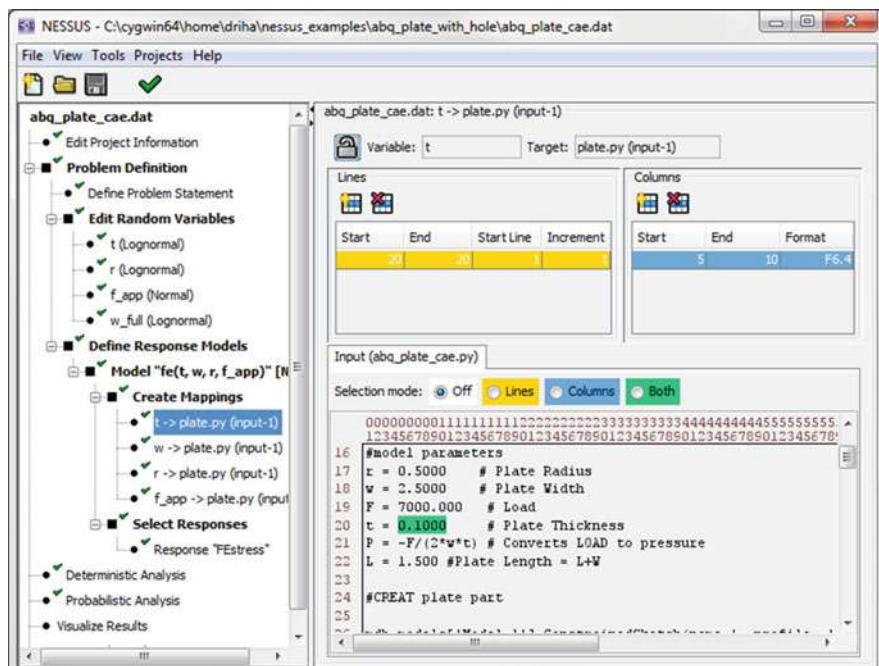


Fig. 51.14 Graphical mapping of the thickness variable to the Abaqus CAE input file.

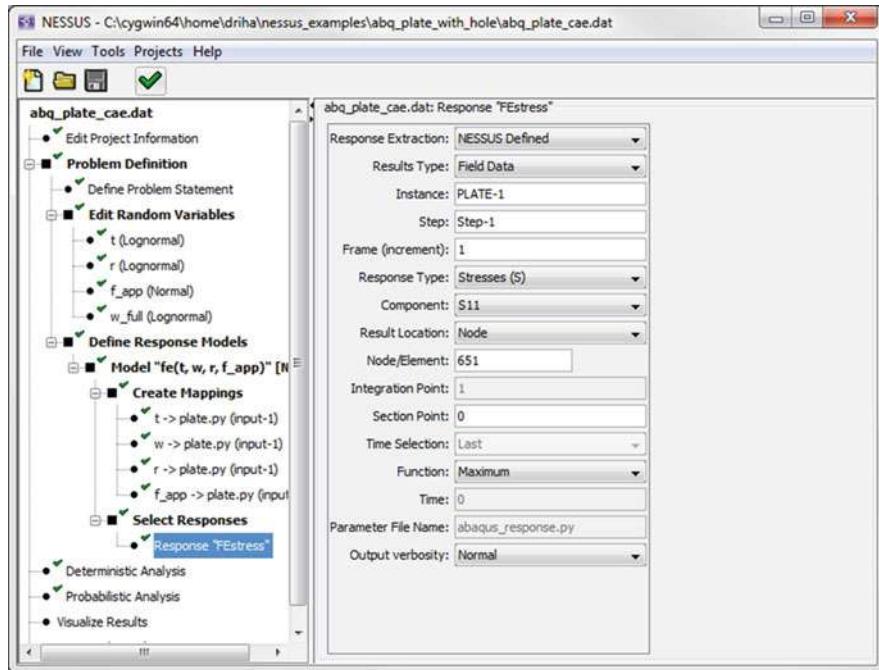


Fig. 51.15 Abaqus finite element response definition screen

the format specifies a field width of six characters, with four digits after the decimal place. This means changes to the thickness variable occurring beyond the fourth decimal place will not be captured in the numerical model input file. This can be an important consideration when using methods that make use of finite difference approximations.

The final step to define the model is to assign the computed variable FESTress in the problem statement. This definition is found in Select Responses under the model definition in the outline. The response selection screen is unique for the results and analysis capabilities for each predefined external analysis code interface in NESSUS. NESSUS is able to extract many responses of interest to engineering analysis from the Abaqus results file (Fig. 51.15). Results from both implicit and explicit analyses are supported for many nodal and element results of interest. Responses are identified at nodal or element locations for specified instances and load steps/increments. Several useful capabilities include extracting the maximum or minimum response for a list of nodes or elements and identifying responses in time series such as the last time or maximum or minimum over time. NESSUS also provides a mechanism for the user to define their own Abaqus CAE script to extract responses not supported by NESSUS.

The problem statement in NESSUS provides a concise and visual representation of the relationships between the models, variables, computed variables, and the

numerical model inputs (Figs. 51.10 and 51.14). These capabilities have evolved as means to better communicate the relationships between the variables and models and to reduce error in the problem definition. The model definition and mapping capabilities are flexible to non-intrusively define random variables in simple and complex numerical analysis models, execute the analyses, and extract results.

Once the model and variables are defined, any of the deterministic and probabilistic methods in NESSUS can be used. Continuing the tutorial, NESSUS is first used to predict the variation in the peak stress in the plate using the AMV+ method to compute the CDF. The AMV+ method is based on locating the MPP using successive linear approximations to the response function and then estimating the probability using FORM. The AMV+ method is useful to iteratively compute an increasingly accurate CDF for computationally intensive models. The first step in the AMV+ method is the MV solution, which is based on replacing the actual model with a first-order Taylor series expansion. The derivatives are approximated by finite difference using only $n+1$ model evaluations where n is the number of random variables. The MV method definition and step size for the finite difference are shown in Fig. 51.16. Additional step sizes for each variable can be added to the table for the MV, AMV, and AMV+ methods, which allows for a linear regression approximation to the derivative. For example, central difference or larger/smaller step sizes may provide a better representation of the derivative for highly nonlinear or noisy responses [5].

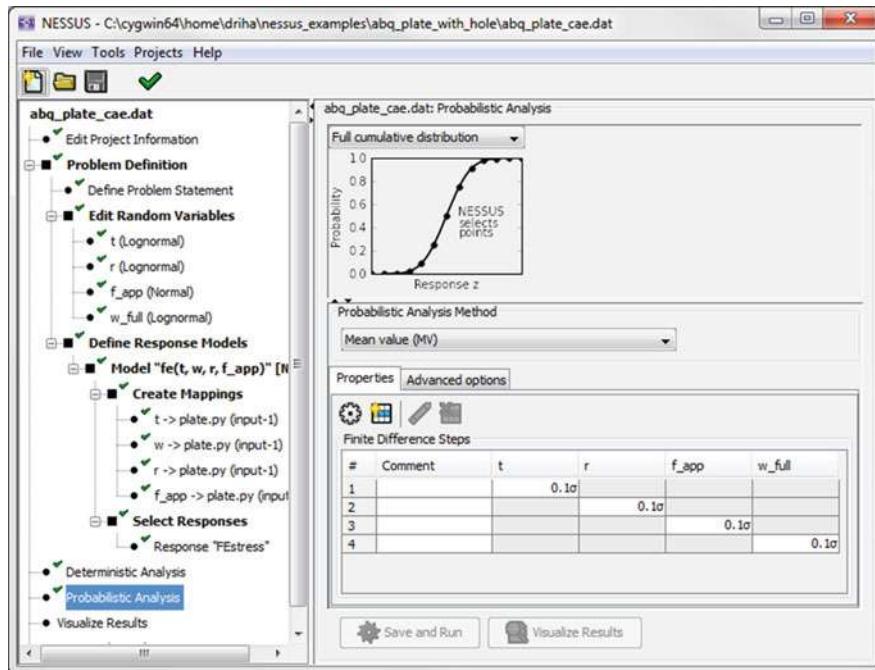


Fig. 51.16 Mean value probabilistic analysis method definition

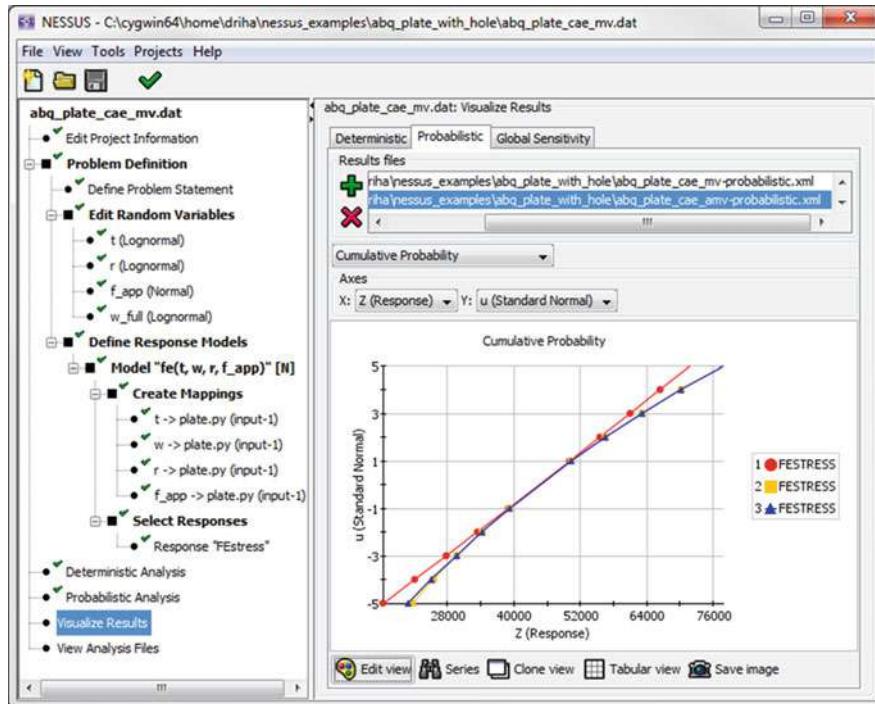


Fig. 51.17 CDF of peak stress using the MV, AMV, and AMV+ methods

The MV solution is shown by the red line in Fig. 51.17 and required four finite element solutions to estimate the full range of the CDF. The AMV solution (yellow line) requires one additional finite element analysis for each point on the CDF (10 for this analysis). The AMV+ method (blue line) iteratively recreates the Taylor series expansion at each MPP until a user specified convergence tolerance on the response value is reached. A 10% tolerance was used for this example, which required one additional Taylor series expansion for the two points defining the left tail of the CDF. The AMV solution met the defined tolerance for all other points in the CDF, without requiring additional iterations. The restart capability in NESSUS makes this solution approach very efficient. NESSUS identifies the finite element solutions already computed in each previous analysis step so no finite element solution is ever repeated for any previously computed set of variable values. For example, the AMV analysis reuses the existing results obtained from the MV analysis, without needing to repeat the finite element solutions.

Next, a reliability problem is formulated to compute the probability that the stress in the plate exceeds the material yield strength. The probability of failure, p_f , is defined as

$$p_f = P[\text{sigy} < \text{FEstress}] = P[\text{sigy} - \text{FEstress} < 0] = P\left[\frac{\text{sigy}}{\text{FEstress}} - 1 < 0\right]$$

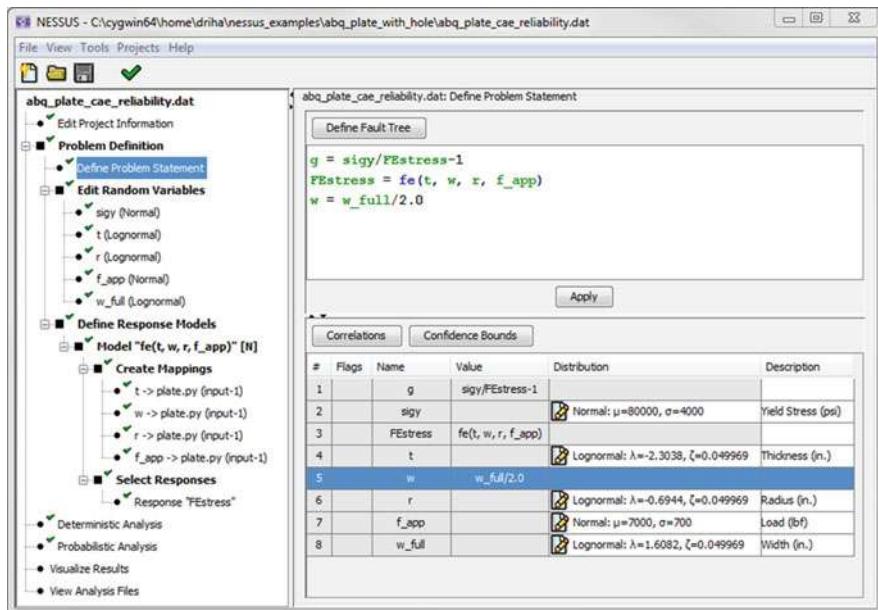


Fig. 51.18 NESSUS problem statement for reliability analysis

Each of these probabilities is equivalent. The last formulation normalizes the stress values and improves the convergence for the AMV+ method. This limit-state formulation is defined by g in the NESSUS problem statement (Fig. 51.18) and corresponds to the left side of the inequality in the probability definition. The analysis type is set to specified performance levels, and the performance level is set to zero based on the right side of the inequality, as shown in Fig. 51.19. The AMV+ solution required a total of 12 finite element analyses to determine a $pf = 0.000002$ using a FORM approximation to the limit state. The probabilistic importance levels are shown in Fig. 51.20 for this probability of failure. These give an indication of the relative importance of the contribution of each random variable to the probability of failure.

5 Solution Strategy Example

NESSUS was used to support a model verification and validation effort for a large-scale multi-physics system where full-scale testing was not practical. A solution strategy was developed to support model V&V and answer questions about the importance of potential uncertainties in the model. The model was expected to run for 16–18 h, and decisions using the model predictions were needed within a relatively short time frame. Therefore, efficient probabilistic methods and solution

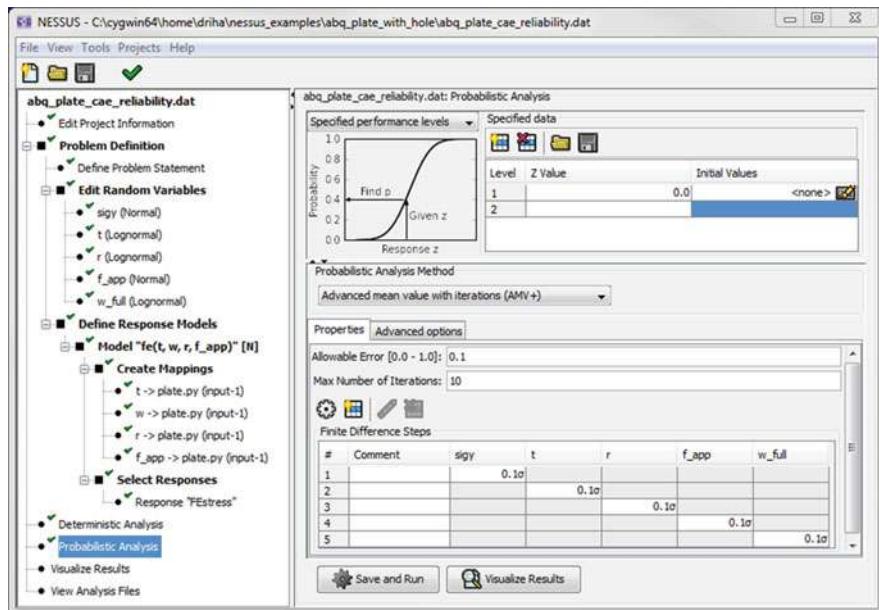


Fig. 51.19 NESSUS analysis type and probabilistic method for reliability analysis example

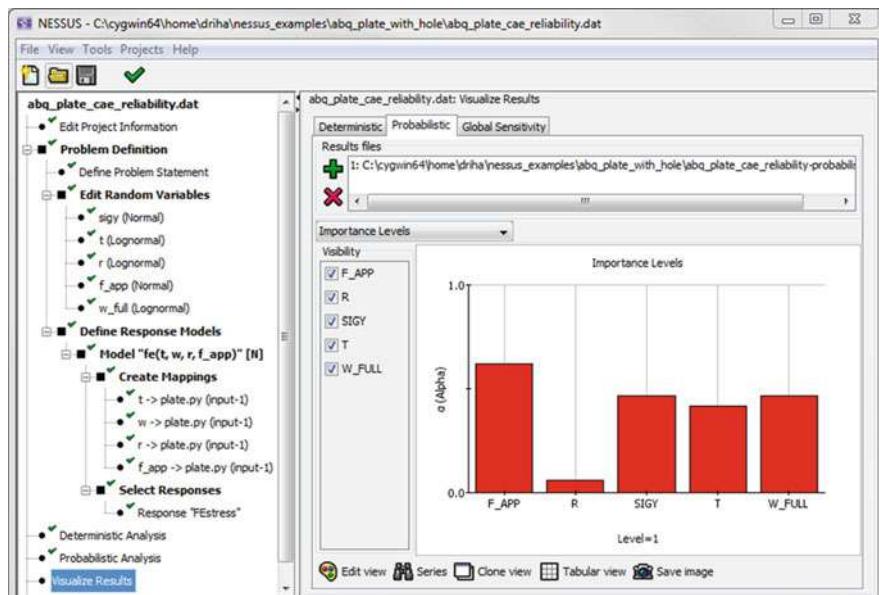
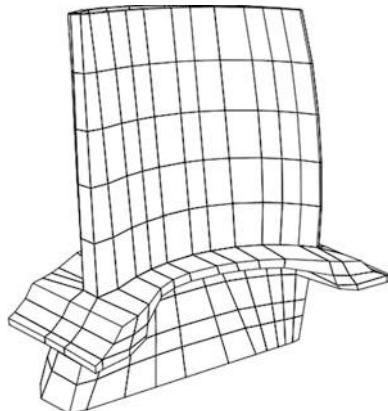


Fig. 51.20 Probabilistic importance levels for plate reliability analysis

Fig. 51.21 Space shuttle main engine turbopump turbine blade finite element model



strategies that could provide initial assessments for uncertainty and support for verification early in the process were required. Due to the sensitive nature and complexity of the actual model, a simplified model in terms of the number of variables and runtimes is used to demonstrate the developed strategy and how NESSUS can be used to perform these studies.

The demonstration uses a hypothetical design of a turbine blade used in NASA's Space Shuttle main engine turbopumps, which is represented by the finite element model shown in Fig. 51.21 [9]. A key characteristic of this approximately 2-inch blade is that it uses a directionally solidified single-crystal metal, which results in direction-dependent material properties. This anisotropic material is defined by material orientation angles relative to the model coordinate system. An actual blade in service would experience large centrifugal loads, vibrations, and extreme and varying temperatures and pressures. For this example study, uncertainties and variations in strain under a constant centrifugal load are computed using finite element analysis for a hypothetical comparison to a relatively simple laboratory experiment. Several questions are posed about the model and inputs that can be assessed using NESSUS:

- Does the model predict correct strains and trends?
- How important are variations in material properties to predicting strain?
- Are variations in material orientation important for strain?
- Are correlations in material properties important?

Uncertainties and variations are defined using normally distributed random variables based on engineering judgment as listed in Table 51.3.

The solution strategy was developed to answer specific questions about uncertainties in the model using a limited number of model evaluations. Deterministic and probabilistic assessments were selected to make the best use of the model runs to improve the solutions incrementally to increase understanding and correct models and data through the analysis process. Table 51.4 lists the analyses,

Table 51.3 Random variable definitions for turbine blade model

Variable	Description	Mean	Standard Deviation
rpm1	Angular velocity	3000 revolutions/min	150 revolutions/min
density	Material density	0.805E-3 lbm/in ³	0.805E-4 lbm/in ³
xangle	Material axis about x	0.05236 radians	0.05 radians
yangle	Material axis about y	-0.03491 radians	0.05 radians
zangle	Material axis about z	0.08727 radians	0.05 radians
E	Elastic modulus	18.38E6 psi	1.838E6 psi
Shear	Shear modulus	18.63E6 psi	1.863E6 psi

Table 51.4 Solution strategy for assessing model uncertainties using NESSUS

Analysis	Model runs	Results	Assessments
Central difference sensitivity study	2 × number of variables	Sensitivity plots	<ul style="list-style-type: none"> • Evaluate trends for correct model behavior • Compare response values to expected values
Build linear response surface and perform MC analysis and global sensitivity analysis	Existing runs	CDF Global sensitivities	Check parameter importance for consistency with physics
Run exact model with LHS	100	CDF Training data for GP model	
Probabilistic analysis and variance sensitivity analysis using GP model	All existing runs	<ul style="list-style-type: none"> • GP model accuracy <ul style="list-style-type: none"> ◦ Cross-validation ◦ Compare to exact model at select points ◦ Compare CDF to LHS results • CDF • Variance-based sensitivities 	<ul style="list-style-type: none"> • GP model accuracy • Identify unimportant variables for elimination for future analyses • Refine input definitions for important variables
Perform probabilistic analysis for different correlation values (correlated variables)	None, use existing GP model	CDF	Define correlations if they are important
Assess uncertainty of the model prediction using refined model and inputs (e.g., AMV+, LHS, EGRA)	~100–1000	CDF	<ul style="list-style-type: none"> • Compare reduced variable results to original results • Assess model uncertainty for required decisions

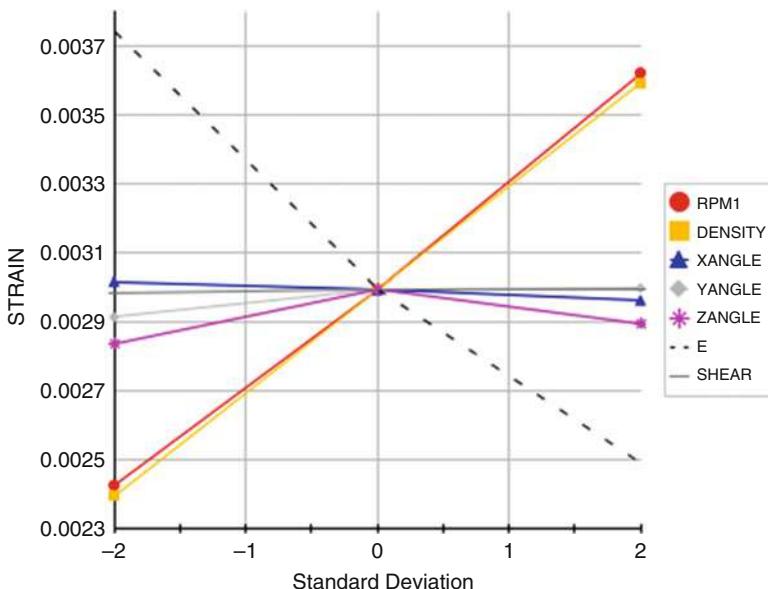


Fig. 51.22 Deterministic parameter variation study to verify correct trends of the turbine blade model

number of model runs, results, and assessments to be made in each step. The first analysis step is used as part of the model verification and to check for correct trends and importance. The deterministic parameter variation capability is used to perform a central difference sensitivity study. Each parameter is perturbed by 2 standard deviations below and above the mean. Figure 51.22 shows the plot from NESSUS for the central difference sensitivity study. Correct trends are observed such as increased strain with increasing rotational speed (RPM1) and material density.

The next step was to start the 100-run LHS analysis to provide a simulation-based CDF using the exact model and create additional training data for more accurate response surface models. While these LHS runs were in process, an initial assessment of prediction uncertainty and important parameters was made using a linear response surface trained with the model evaluations from the deterministic sensitivity study. Figure 51.23 shows the CDF and global sensitivities using the linear response surface. These studies are intended to provide initial information about important parameters and prediction uncertainty.

The results from the LHS study were used in addition to the central difference results to train a GP model to improve the accuracy of the CDF and global sensitivities. An accurate model fit is demonstrated by the cross-validation R-squared value of 0.99997, which is reported by NESSUS along with other goodness-of-fit information. A comparison of the CDF using the linear response surface, LHS with the exact model, and Monte Carlo simulation with the GP model are

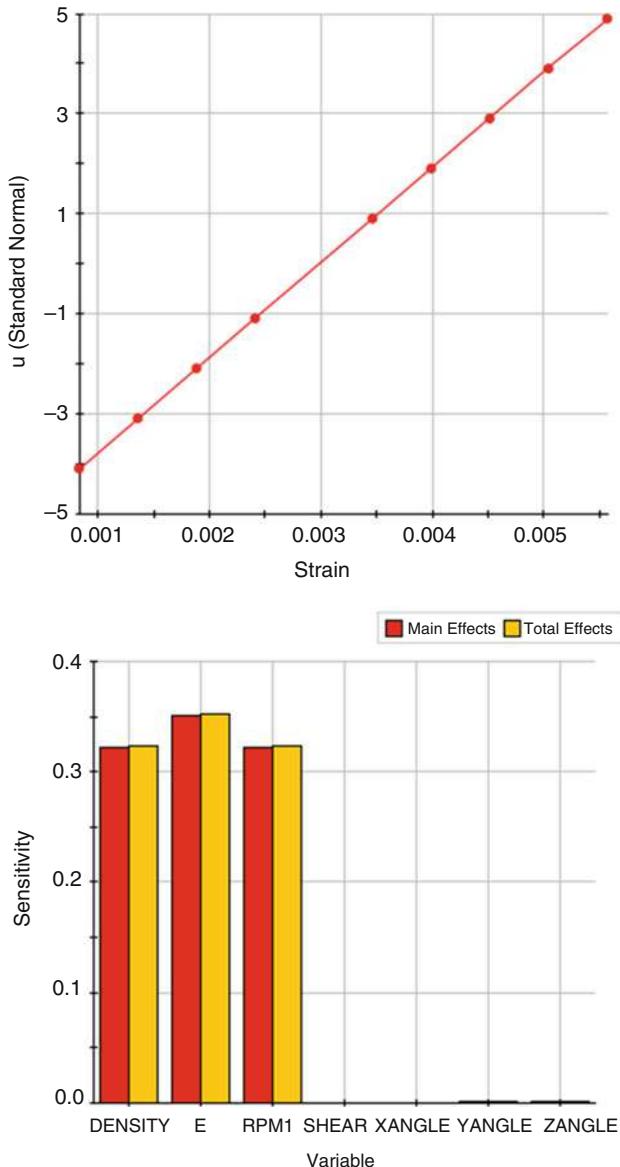


Fig. 51.23 CDF and global sensitivities using a linear response surface used to estimate prediction uncertainty and important parameters

shown in Fig. 51.24 along with global sensitivities using the GP model. The LHS solution was used as a partial verification for the accuracy of the GP model. The global sensitivities identify four variables that contribute very little to the

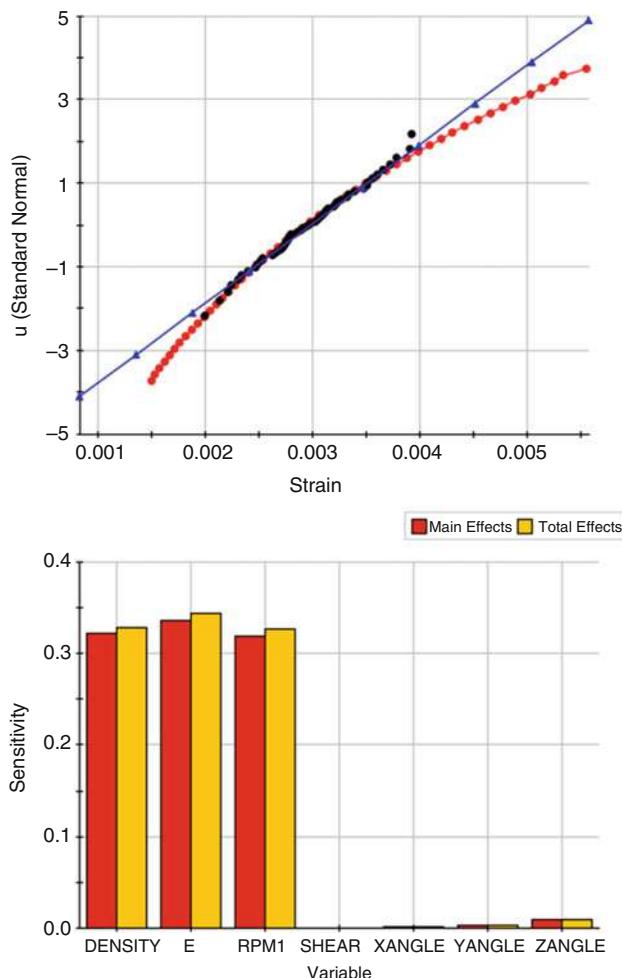


Fig. 51.24 CDF using the linear response surface (blue), LHS with exact model (black), and Monte Carlo simulation with the GP model (red) and the global sensitivities

predicted strain uncertainty: the three material angles and the shear modulus. These are candidate for removal in future uncertainty assessments for strain at this location.

Additional uncertainty studies can be quickly performed using the GP model. For example, sufficient experimental data are often not available to determine distribution types or correlations between different parameters. NESSUS can be used to study the impact of different distributions (such as normal or lognormal) on the predicted uncertainty by simply changing the distribution type. To demonstrate

the process for studying the uncertainty in correlations for this model, two additional NESSUS analyses were performed using Monte Carlo simulation with the GP model for correlation coefficients of -0.8 and 0.8 between the material stiffness and density. Correlations between these parameters influence the model uncertainty predictions as shown in Fig. 51.25. These results provide valuable information for allocating resources to either gather additional data/information to better quantify the correlation or to include this uncertainty when using the model.

Based on the global sensitivities, only important variables are retained to more efficiently predict the full distribution of strain using the exact model to refine the uncertainty assessment. The AMV+ method is used with the finite element model after removing the unimportant variables to predict the CDF. The NESSUS problem statement and CDF for the reduced variable set is compared to the LHS solution using the exact model and full variable set in Fig. 51.26. The AMV+ solution required 32 finite element analysis runs to predict the full range of the CDF.

This example shows some of the capabilities for performing uncertainty quantification in NESSUS. The combination of response surface models and efficient probabilistic methods allows for quick and informative studies to understand and communicate the model uncertainties. More large-scale problems solved using NESSUS can be found in Ref. [10].

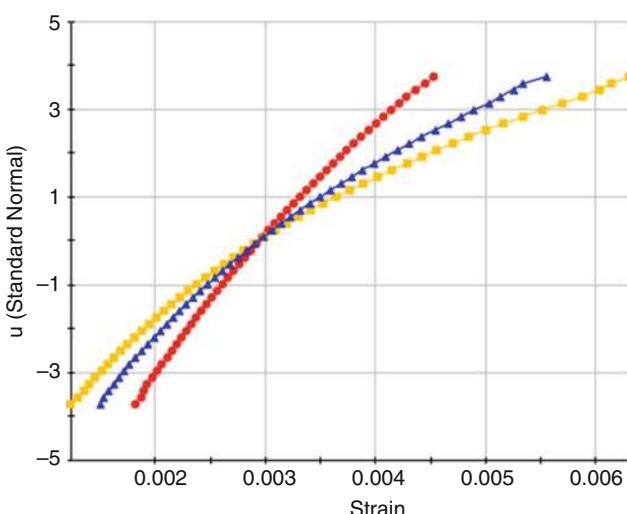


Fig. 51.25 NESSUS was used to study the uncertainty in the correlation between material stiffness and density by computing the CDF for correlation coefficients of -0.8 (red), 0.0 (blue), and 0.8 (yellow)

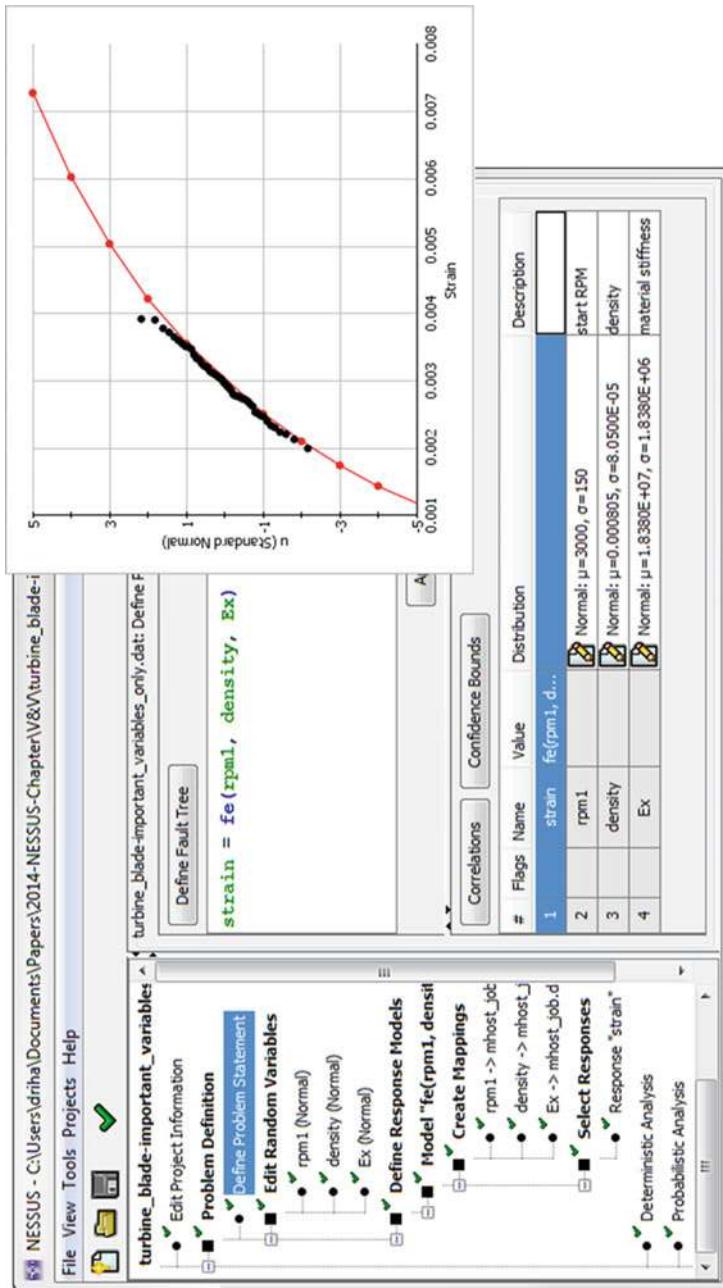


Fig. 51.26 NESSUS problem statement for the model after removing unimportant variables and the CDF for the AMV+ solution (red) and the original model LHS solution (black)

6 Conclusions

The NESSUS probabilistic analysis software is designed to simplify and streamline the process of setting up and executing reliability analysis, uncertainty quantification, and sensitivity analysis studies. The software has been applied to a wide array of problems in a variety of engineering disciplines, but its development has been largely motivated by the need for practical tools that can be used for large-scale engineering applications and complex models with long runtimes. NESSUS was created in the 1980s with the goal of developing new technology for probabilistic design and analysis of space shuttle main engine components. Since then, the capabilities have continued to expand to include advanced response surface and sensitivity analysis methods.

NESSUS includes a variety of flexible and powerful capabilities for interfacing with deterministic performance models. These range from a simple algebraic equation syntax to sophisticated interfaces with external third party or user-defined codes, including a graphical interface for defining variable mappings. With the ability to set up user-defined interfaces based on either file input/output or direct calls into user-created shared libraries, it is possible to configure NESSUS to interface with virtually any numerical model.

NESSUS includes 16 probabilistic analysis methods, the majority of which were developed specifically for working with long-running performance models. These include the advanced mean value plus (AMV+) method as well as several methods based on Gaussian process response surface modeling, such as the efficient global reliability analysis (EGRA) method. NESSUS includes traditional probabilistic sensitivity results for all forward and inverse reliability analysis methods, and the software has recently been expanded to include variance-based global sensitivity analysis.

References

1. Wu, Y.-T.: Computational methods for efficient structural reliability and reliability sensitivity analysis. *AIAA J.* **32**(8), 1717–1723 (1994)
2. Martin, J., Simpson, T.: Use of kriging models to approximate deterministic computer models. *AIAA J.* **43**(4), 853–863 (2005)
3. Iman, R.L., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat. Part B. Simul. Comput.* **11**(3), 311–334 (1982)
4. Wu, Y.-T., Millwater, H.R., Cruse, T.A.: Advanced probabilistic structural analysis methods for implicit performance functions. *AIAA J.* **28**(9), 1663–1669 (1990)
5. Riha, D.S., Thacker, B.H., Fitch, S.H.K.: NESSUS capabilities for ill-behaved performance functions. In: Proceedings of the AIAA/ASME/ASCE/AHS/ASC 45th Structures, Structural Dynamics, and Materials (SDM) Conference, AIAA 2004-1832, Palm Springs, 19–22 Apr 2004
6. Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M.: Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA J.* **46**(10), 2459–2468 (2008)

7. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**, 259–270 (2010)
8. Riha, D.S., Thacker, B.H., Pepin, J.E., Fitch, S.H.K.: Uncertainty modeling to relate component assembly uncertainties to physics-based model parameters. In: Proceedings of the AIAA/ASME/ASCE/AHS/ASC 49th Structures, Structural Dynamics, and Materials Conference, AIAA 2008-2158, Schaumburg, 7–10, April 2008
9. Thacker, B.H., McClung, R.C., Millwater, H.R.: Application of the probabilistic approximate analysis method to a turbopump blade analysis. In: Proceedings of the 31st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Part 2, AIAA-90-1098, Long Beach, pp. 1039–1047, Apr 1990
10. Thacker, B.H., Riha, D.S., Huyse, L.J., Fitch, S. HK.: Probabilistic engineering analysis using the NESSUS software. *Struct. Saf.* **28**, 83–107 (2006)

Eric T. Phipps and Andrew G. Salinger

Abstract

Stokhos (Phipps, Stokhos embedded uncertainty quantification methods. <http://trilinos.org/packages/stokhos/>, 2015) is a package within Trilinos (Heroux et al., ACM Trans Math Softw 31(3), 2005; Michael et al., Sci Program 20(2):83–88, 2012) that enables embedded or intrusive uncertainty quantification capabilities to C++ codes. It provides tools for implementing stochastic Galerkin methods and embedded sample propagation through the use of template-based generic programming (Pawlowski et al., Sci Program 20:197–219, 2012; Roger et al., Sci Program 20:327–345, 2012) which allows deterministic simulation codes to be easily modified for embedded uncertainty quantification. It provides tools for forming and solving the resulting linear and nonlinear equations these methods generate, leveraging the large-scale linear and nonlinear solver capabilities provided by Trilinos. Furthermore, Stokhos is integrated with the emerging many-core architecture capabilities provided by the Kokkos (Edwards et al., Sci Program 20(2):89–114, 2012; Edwards et al., J Parallel Distrib Comput 74(12):3202–3216, 2014) and Tpetra packages (Baker and Heroux, Sci Program 20(2):115–128, 2012; Hoemmen et al., Tpetra: next-generation distributed linear algebra. <http://trilinos.org/packages/tpetra>, 2015) within Trilinos, allowing these embedded uncertainty quantification capabilities to be applied in both shared and distributed memory parallel computational environments. Finally, the Stokhos tools have been incorporated into the Albany simulation code (Pawlowski et al., Sci Program 20:327–345, 2012; Salinger et al., Albany multiphysics simulation code. <https://github.com/gahansen/Albany>, 2015) enabling embedded uncertainty quantification of a wide variety of large-scale PDE-based simulations.

E.T. Phipps • A.G. Salinger (✉)

Sandia National Laboratories, Center for Computing Research, Albuquerque, NM, USA
e-mail: etphipp@sandia.gov; agsalin@sandia.gov

Keywords

Stochastic Galerkin methods • Embedded sampling methods • Polynomial chaos • Sparse grids • C++ templates • Operator overloading • Linear solvers • Preconditioning • Parallel programming • Shared memory parallelism • Distributed memory parallelism • Fine-grained parallelism • Multicore architectures

Contents

1	Introduction	1766
2	Background	1767
3	Stochastic Galerkin Methods with Stokhos	1768
4	Uncertainty Quantification on Emerging Computer Architectures	1771
5	Sample Propagation Methods with Stokhos	1773
6	Examples	1774
6.1	Obtaining, Configuring, and Compiling the Stokhos Package	1774
6.2	Simple Polynomial Chaos Example	1776
6.3	Simple Ensemble Propagation Example	1780
6.4	Nonlinear Stochastic Galerkin Example	1784
6.5	Nonlinear Fluid Flow Problem with Albany	1802
7	Conclusions	1804
	References	1805

1 Introduction

Stokhos [27] is a package for implementing *embedded* uncertainty quantification methods in C++ codes that is part of the Trilinos framework [16, 17]. Here the term embedded refers to methods that require substantial modification of the simulation source code to implement (also called intrusive). Stokhos itself does not provide any uncertainty quantification capabilities; rather it provides a set of tools for building those capabilities in general C++ simulation codes. There are two general classes of methods it enables: stochastic Galerkin methods based on a (generalized) polynomial chaos discretization and embedded sample propagation for any kind of sampling-based method. In both cases, Stokhos provides tools to support propagation of uncertainty information at the lowest levels of a simulation code via a technique called template-based generic programming [25, 26] as well as tools for forming and solving any resulting linear and nonlinear equations. It is geared toward uncertainty propagation in partial differential equations (explicit or implicit), but will work with any type of deterministic simulation code. In the sections that follow, stochastic Galerkin methods and their implementation with Stokhos are described, followed by a simpler discussion of embedded sample propagation. Next, use of these tools and techniques for exposing fine-grained parallelism on emerging multi-core and many-core architectures is described, followed by several examples describing the use of Stokhos for uncertainty propagation in C++ codes. The paper concludes with an example of applying these techniques to an uncertain fluid flow problem implemented through the Albany simulation code [26, 35]. More details on Stokhos and in particular its application in the emerging computer architecture context can be found here [30, 31].

2 Background

To describe the software tools for implementing embedded uncertainty quantification methods with Stokhos and to fix notation, a brief review of stochastic Galerkin methods is provided. Following [31], only steady-state problems are considered for simplicity, as the extension of the software to transient problems is straightforward. Let $(\Omega, \mathcal{B}, \mu)$ be a complete probability space where Ω is the set of outcomes, \mathcal{B} is the σ -algebra of measurable events, and $\mu : \Omega \rightarrow [0, 1]$ is the probability measure. Assume the problem depends on a finite set of M independent random variables $\xi_i : \Omega \rightarrow \Gamma_i \subset \mathbb{R}$, $i = 1, \dots, M$, representing uncertain input data with corresponding density function $\rho_i(y_i)$, $i = 1, \dots, M$. Let $\xi = [\xi_1, \dots, \xi_M]$, $\Gamma = \Gamma_1 \times \dots \times \Gamma_M$ and $\rho(y) = \rho_1(y_1) \dots \rho_M(y_M)$ be the density of ξ for $y = (y_1, \dots, y_M) \in \Gamma$. Finally, assume the problem of interest can be modeled by the discrete nonlinear system

$$f(x; \xi(\omega)) = 0, \quad (52.1)$$

where $x \in \mathbb{R}^n$ is the corresponding discrete unknown solution vector and $f : \mathbb{R}^{n+M} \rightarrow \mathbb{R}^n$. By the Doob-Dynkin Lemma [23], the solution x can be parameterized by the same random vector ξ : $x(\omega) = x(\xi(\omega))$. As described previously in this volume, this mapping can be approximated through the use of global polynomials $\{\Psi^i\}$ orthogonal with respect to the joint density function ρ [11, 12, 39, 40]:

$$\langle \Psi^j \Psi^k \rangle \equiv \int_{\Gamma} \Psi^j(y) \Psi^k(y) \rho(y) dy = \langle (\Psi^j)^2 \rangle \delta_{jk}, \quad j, k = 0, 1, 2, \dots \quad (52.2)$$

These polynomials are constructed in Stokhos in the traditional manner by tensorization of one-dimensional orthogonal polynomials. For each $i = 1, \dots, M$, let $\{\psi_i^j\}$ be a family of polynomials orthonormal with respect to the measure ρ_i :

$$\langle \psi_i^j \psi_i^k \rangle_i \equiv \int_{\Gamma_i} \psi_i^j(y_i) \psi_i^k(y_i) \rho_i(y_i) dy_i = \langle (\psi_i^j)^2 \rangle_i \delta_{jk}, \quad j, k = 0, 1, 2, \dots \quad (52.3)$$

ordered in such a way that the degree of polynomial ψ_i^j is j . For a given multi-index $\alpha = (\alpha_1, \dots, \alpha_M)$, define

$$\Psi^\alpha(y) = \psi_1^{\alpha_1}(y_1) \dots \psi_M^{\alpha_M}(y_M), \quad y = (y_1, \dots, y_M) \in \Gamma. \quad (52.4)$$

Then for a given $N \geq 0$, denote by $\mathcal{L}_{M,N}$ the complete polynomial space of total order at most N ,

$$\mathcal{L}_{M,N} = \text{span}\{\Psi^\alpha : |\alpha| \equiv \alpha_1 + \dots + \alpha_M \leq N\} \subset L_\rho^2(\Gamma), \quad (52.5)$$

and define $P + 1 \equiv \dim \mathcal{L}_{M,N} = \frac{(M+N)!}{M!N!}$. The solution $x(\xi)$ is then approximated in $\mathcal{L}_{M,N}$ as

$$x(\xi) \approx \hat{x}(\xi) \equiv \sum_{i=0}^P x^i \Psi^i(\xi). \quad (52.6)$$

The stochastic Galerkin method approximates the unknown coefficients x^i through orthogonal projection of the residual f into $\mathcal{L}_{M,N}$:

$$f^i = \frac{\langle f \Psi^i \rangle}{\langle (\Psi^i)^2 \rangle} = \frac{1}{\langle (\Psi^i)^2 \rangle} \int_{\Gamma} f(\hat{x}(y); y) \Psi^i(y) \rho(y) dy = 0, \quad i = 0, \dots, P. \quad (52.7)$$

Define

$$X = \begin{bmatrix} x^0 \\ \vdots \\ x^P \end{bmatrix} = \sum_{k=0}^P e^k \otimes x^k, \quad F = \begin{bmatrix} f^0 \\ \vdots \\ f^P \end{bmatrix} = \sum_{k=0}^P e^k \otimes f^k \quad (52.8)$$

where e_k is the k th column of the $(P+1) \times (P+1)$ identity matrix, $k = 0, \dots, P$. This defines a fully-coupled (spatial-stochastic) nonlinear system

$$F(X) = 0 \quad (52.9)$$

of $n(P+1)$ equations in $n(P+1)$ unknowns.

3 Stochastic Galerkin Methods with Stokhos

Stokhos provides a set of software tools for forming and solving the stochastic Galerkin nonlinear system (52.9) using Newton-type nonlinear solver schemes:

$$\frac{\partial F}{\partial X} \Delta X_l = -F(X_l), \quad X_{l+1} = X_l + \Delta X_l, \quad l = 0, 1, \dots \quad (52.10)$$

This involves several challenges, the first of which is evaluating the stochastic Galerkin residual and Jacobian matrix. Note that from the definition of F , the components f^i of F are just the polynomial chaos coefficients of $f(\hat{x}(\xi); \xi)$. Furthermore for $i, j = 0, \dots, P$,

$$\frac{\partial f^i}{\partial x^j} = \frac{1}{\langle (\Psi^i)^2 \rangle} \int_{\Gamma} \frac{\partial f}{\partial x}(\hat{x}(y); y) \Psi^i(y) \Psi^j(y) \rho(y) dy \approx \sum_{k=0}^P A^k \frac{\langle \Psi^i \Psi^j \Psi^k \rangle}{\langle (\Psi^i)^2 \rangle} \quad (52.11)$$

where

$$\begin{aligned} \frac{\partial f}{\partial x}(\hat{x}(\xi); \xi) &\approx \sum_{k=0}^P A^k \Psi^k(\xi), \\ A^k &= \frac{1}{\langle (\Psi^i)^2 \rangle} \int_{\Gamma} \frac{\partial f}{\partial x}(\hat{x}(y); y) \Psi^k(y) \rho(y) dy, \quad k = 0, \dots, P \end{aligned} \quad (52.12)$$

is the truncated polynomial chaos approximation to the Jacobian operator $\partial f / \partial x$. Given C++ computer code to evaluate f and $\partial f / \partial x$ for given values of x and y , Stokhos generates computer code to evaluate $\{f^i\}$ and $\{A^i\}$ using a technique called template-based generic programming [25, 26] based on the ideas of automatic differentiation (see, e.g., [13] and the references contained within). In brief, the code to evaluate f and $\partial f / \partial x$ is decomposed into a sequence of elementary operations (addition, subtraction, multiplication, and division) and simple functions (e.g., sine, cosine, exponential, logarithms). The idea is to compute projections into $\mathcal{L}_{M,N}$ vis-à-vis Eq. 52.7 for each intermediate variable used in the evaluation of f or $\partial f / \partial x$ using simple rules for each of these elementary operations and simple functions.

Let a and b be two intermediate variables in a given calculation and assume, by way of induction, their polynomial chaos projections

$$a(\xi) \approx \hat{a}(\xi) \equiv \sum_{i=0}^P a^i \Psi^i(\xi), \quad b(\xi) \approx \hat{b}(\xi) \equiv \sum_{i=0}^P b^i \Psi^i(\xi)$$

have already been computed. Let $c = \varphi(a, b)$ where φ is some elementary operation/simple function and the polynomial chaos projection of c must be computed. For the elementary arithmetic operations, simple formulas for the coefficients $\{c^i\}$ can be readily obtained as shown in Table 52.1. Note the rule for division requires solving a linear system for the coefficients $\{c^i\}$. The tensor $C_{ijk} \equiv \langle \Psi^i \Psi^j \Psi^k \rangle / \langle (\Psi^i)^2 \rangle$ is precomputed and stored in a sparse format. Rules for transcendental operations and non-smooth functions (e.g., min/max, abs) are more challenging, but a number of approaches have been investigated [7, 21]. The approach recommended in Stokhos is the use of numerical integration, e.g.,

$$\begin{aligned} c^i &= \frac{1}{\langle (\Psi^i)^2 \rangle} \int_{\Gamma} \varphi(\hat{a}(y), \hat{b}(y)) \Psi^i(y) \rho(y) dy \\ &\approx \frac{1}{\langle (\Psi^i)^2 \rangle} \sum_{k=0}^Q w_k \varphi(\hat{a}(y_k), \hat{b}(y_k)) \Psi^i(y_k) \end{aligned} \quad (52.13)$$

Table 52.1 Projection rules for elementary operations

Operation	Rule
$c = a \pm b$	$c^i = a^i \pm b^i$
$c = ab$	$c^i = \sum_{j,k=0}^P a^j b^k \langle \Psi^i \Psi^j \Psi^k \rangle / \langle (\Psi^i)^2 \rangle$
$c = a/b$	$\sum_{j,k=0}^P b^j c^k \langle \Psi^i \Psi^j \Psi^k \rangle / \langle (\Psi^i)^2 \rangle = a^i$

where $\{(w_k, y_k) : k = 0, \dots, Q\}$ is a quadrature rule determined by the measure ρ . Stokhos provides a number of sparse-grid quadrature rules [3, 22, 36], allowing (52.13) to be implemented for many standard simple math functions (Note the use of sparse-grid quadrature in Eq. 52.13 can result in additional numerical error if care is not taken to preserve orthogonality of the polynomial chaos basis functions $\{\Psi^i\}$ using the discrete inner product defined by the quadrature rule. This can be remedied by replacing Eq. 52.13 by a more general rule based on the Smolyak formula [5, 6]).

Stokhos provides a C++ data type designed to store the polynomial chaos coefficients for each variable and overloads of all of the elementary/simple functions to compute projections in the manner described above, referred to as the *polynomial chaos scalar type*. The code to evaluate $\{f^i\}$ and $\{A^i\}$ is then obtained by replacing the fundamental floating-point scalar type in the code to evaluate f and $\partial f / \partial x$ with this new data type, leveraging the standard operator overloading resolution rules for the C++ language. While not required, it is recommended to facilitate this transformation by instead turning the code into general template code, where the scalar type becomes a template parameter. Then the original code can be obtained by instantiating the template code on the original floating-point type and the transformed code by instantiating it on the polynomial chaos scalar type.

With an ability to evaluate the stochastic Galerkin residual F and Jacobian $\partial f / \partial X$ available, the next challenge is solving the resulting linear systems appearing in Eq. 52.10. Due to the very large size of these systems, Stokhos provides interfaces and data structures for solving these systems using iterative solver methods such as CG and GMRES implemented by other packages in the Trilinos framework. In this context, the Jacobian matrix $\partial f / \partial X$ does not need to be formed directly; rather matrix-vector products may be computed efficiently using the Kronecker product structure of the Jacobian:

$$\frac{\partial F}{\partial X} \approx A \equiv \sum_{k=0}^P G^k \otimes A^k \quad (52.14)$$

where each matrix $G^k \in \mathbb{R}^{(P+1) \times (P+1)}$ satisfies $G_{ij}^k = C_{ijk} = \langle \Psi^i \Psi^j \Psi^k \rangle / \langle (\Psi^i)^2 \rangle$, $i, j, k = 0, \dots, P$. An algorithm to efficiently evaluate matrix-vector products using this representation is described in detail here [30].

Critical to the efficiency of these solvers are effective preconditioning strategies, and Stokhos provides implementations of several preconditioners that have been studied in the literature, including mean-based [32], relaxation [33], Kronecker product [38], and Schur complement [37] that couple to a variety of algebraic preconditioners implemented in Trilinos designed for the original system $f(x) = 0$ (such as incomplete factorizations and multi-grid). Finally Stokhos provides implementations of several interfaces in Trilinos allowing nonlinear, transient, and optimization solvers to be applied to the stochastic Galerkin system. Examples of applying these capabilities to several simple problems will be described later in this chapter.

4 Uncertainty Quantification on Emerging Computer Architectures

Over the coming decade, it is expected that computer architectures will evolve considerably in order to achieve increasing levels of computing throughput with only modest increases in overall power consumption. Harnessing this increased computational power is likely to place tremendous demands on the efficiency and scalability of scientific and engineering simulation codes by requiring:

- Regular memory access patterns to contiguous regions of memory in order to avoid long access latencies.
- Arithmetic that maps well to arithmetic on wide vectors (vectorization).
- Good spatial and temporal data locality in order to effectively utilize deep memory hierarchies that can be shared by multiple independent execution contexts.

Unfortunately many simulation codes struggle to meet these demands, in particular those involving complex physics and sparse linear algebra on unstructured meshes. Sparse linear algebra inherently introduces indirect memory addressing which introduces latency effects into the overall calculation and generally results in poor cache utilization making effective use of hardware threads impossible. Also complicated loop structures in complex simulation codes often make translating those loops to arithmetic on wide vectors challenging. Thus it is entirely possible that many simulation codes may achieve lower performance on these next-generation architectures than they do today. Considering that the aggregate performance of sampling-based uncertainty quantification is limited by the performance of the underlying simulation, this reduction in performance will directly translate into increased cost for uncertainty quantification calculations built on top of these simulations.

However embedded uncertainty quantification methods such as the stochastic Galerkin methods described above provide an opportunity to improve performance on these architectures. The operator overloading-based, scalar-level uncertainty propagation method discussed above for evaluating Galerkin residual and Jacobian values naturally replaces scalar floating-point operations with operations on arrays of polynomial chaos coefficients. This will generate more regular memory access patterns if these coefficients are stored in continuous memory locations. It also exposes new dimensions of fine-grained parallelism by parallelizing the operations in Table 52.1 through fine-grained threading and vectorization. Finally, it improves temporal data locality and cache-reuse as only one uncertainty propagation sweep through the simulation code is required instead of numerous sweeps at each sample.

To take advantage of these features for the solution of the stochastic Galerkin linear systems however, some modifications of the approach discussed above are required. The Kronecker structure of the Galerkin matrix (52.14) gives rise to a block system with a two-level sparsity structure: an outer structure as determined

by the sparse C_{ijk} tensor and an inner structure determined by the deterministic Jacobian operator as demonstrated in Fig. 52.1a. In many cases, this structure is not ideal for emerging architectures when the sparsity pattern of the deterministic Jacobian operator is unstructured. However by commuting the terms of the Kronecker product,

$$\tilde{A} = \sum_{k=0}^P A^k \otimes G^k, \quad (52.15)$$

the sparsity structure is inverted as shown in Fig. 52.1b. This amounts to merely a reordering of the stochastic and deterministic degrees-of-freedom, ordering all of the stochastic degrees-of-freedom corresponding to a given deterministic degree-of-freedom consecutively.

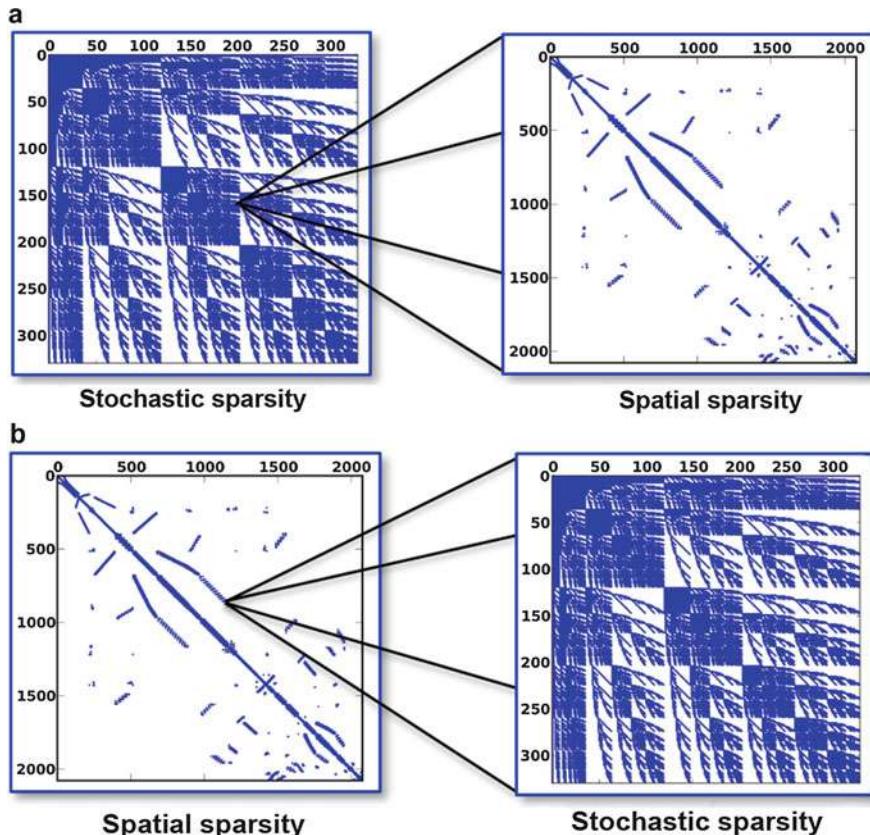


Fig. 52.1 Two-level sparsity structure of stochastic Galerkin operator using traditional layout (a) corresponding to Eq. 52.14 and commuted layout (b) corresponding to Eq. 52.15. For each figure, every nonzero in the *left*, outer sparsity structure is a block of size and sparsity indicated by the *right*, inner sparsity structure. This can result in a denser outer structure with sparse inner structure for the traditional layout (a) and a sparse outer structure with denser inner structure for the commuted layout (b)

This ordering allows all of the properties described above with regard to evaluation of the stochastic Galerkin residual and Jacobian entries to also be applied to sparse linear algebra with the Galerkin matrix. In Stokhos, solution of stochastic Galerkin linear systems in this layout is implemented by instantiating next-generation templated solver and preconditioner libraries in Trilinos such as Tpetra [2, 19], Belos [4], Ifpack2 [18], and MueLu [10, 20] on the polynomial chaos scalar type discussed above. Furthermore, Stokhos provides specializations of the Kokkos many-core parallelism portability library [8, 9] to map fine-grained parallelism across the uncertainty dimension for use in the solver libraries mentioned previously as well as simulation codes that use Kokkos directly in their residual/Jacobian evaluations. For a more thorough description of how Stokhos interacts with these libraries, discussion of custom linear algebra kernels exploiting the commuted Kronecker-product structure (52.15), and performance comparisons of solving representative PDEs with uncertain input data on a variety of emerging multi-core and many-core architectures, see [30, 31].

5 Sample Propagation Methods with Stokhos

In addition to providing tools for implementing embedded stochastic Galerkin methods, Stokhos also provides similar tools for embedded sampling methods. The idea behind this approach is to enable some of the same computer architecture benefits the Galerkin approach provides, such as improved memory access patterns and exposing new dimensions of structured fine-grained parallelism, without the additional solver challenges that arise from the Galerkin approach. In particular, collections of samples called ensembles are propagated together at the scalar level of the simulation code as in the Galerkin approach described above. Stokhos provides an *ensemble scalar type* containing an array (whose length is fixed at compile time) to store the ensemble values for each intermediate variable, and corresponding overloaded operators are provided for all elementary/simple functions that trivially map those operations across the ensemble array. By choosing the length of the ensemble array to be a multiple of the natural vector width of the architecture, it becomes simple for vectorization and fine-grained threading to implement the operations on this ensemble array in parallel. Furthermore, scalar loads and stores become loads/stores of entire ensemble arrays resulting in improved memory access patterns. By implementing the method through the template-based generic programming approach, applying the method to a code already templated on the scalar amounts to just another template instantiation for that code. Just as in the Galerkin case, Stokhos provides the necessary specializations for incorporating the approach into the Kokkos many-core programming library and the template solver libraries in Trilinos.

This approach is amenable to any sampling-based method such as stochastic collocation, nonintrusive polynomial chaos, and Monte Carlo. Samples generated from the method are grouped into ensembles of the chosen size and then propagated together. This provides the best-of-both-worlds hybrid between purely intrusive and nonintrusive methods by propagating some UQ information together at the

scalar level of the code in order to exploit fine-grained parallelism but still allows traditional coarse-grained parallelism between ensembles. The primary challenge of this approach is determining how to group samples to get the most benefit out of the ensemble propagation. Current research is underway to address that question, particularly in the context of adaptive sampling methods applied to systems with non-smooth responses.

6 Examples

In this section, several examples demonstrating the use of Stokhos for simple uncertainty propagation problems are described. First, the steps necessary for obtaining, configuring, and compiling stokhos are described. Then three examples of using Stokhos are covered: computing a polynomial chaos approximation for a simple function using the polynomial chaos scalar type and its overloaded operators, the same example instead using the embedded ensemble propagation approach within the context of nonintrusive spectral projection, and demonstration of the nonlinear solver interface in Stokhos for formulating and solving nonlinear, steady-state stochastic Galerkin problems. These techniques form the building blocks of applying the methods to large-scale systems such as discretized partial differential equations. The section then concludes with a demonstration of these capabilities for solving a fluid flow problem with uncertain input data as computed by the Albany simulation code [26, 35]. Applying Stokhos in multi-core/many-core programming environments is beyond the scope of this chapter. Trilinos provides a complete PDE example called FENL capable of running on modern multi-core and many-core architectures in the TrilinosCouplings package including uncertainty quantification capabilities provided by Stokhos. For performance results in that context, please see here [31].

6.1 Obtaining, Configuring, and Compiling the Stokhos Package

Stokhos is a package within Trilinos and therefore is obtained by downloading Trilinos. Trilinos is currently available for download from <https://trilinos.org/download>, including both periodic source code releases as well as the most recent development sources through the `publicTrilinos` Git repository (`git clone https://software.sandia.gov/trilinos/repositories/publicTrilinos`). In the future Trilinos will be available from GitHub (<https://github.com>). Trilinos is released as open-source software; however the licensing requirements of packages within Trilinos vary. Stokhos is released under the BSD license. Currently only Trilinos sources are provided for download, requiring the user to configure and build Trilinos based on their intended use. Due to the large number of packages within Trilinos and myriad of ways Trilinos can be configured, it is not feasible to provide precompiled binary versions.

Once the Trilinos sources have been obtained, it must be configured before compiling. Trilinos uses CMake (<https://cmake.org>) to manage the configuration

and build process, and documentation is provided by CMake on its use for a variety of computer architectures. For Unix-like environments, this is most often accomplished through a configuration script such as the one shown below.

```
rm -f CMakeCache.txt;
rm -rf CMakeFiles

cmake \
-D CMAKE_INSTALL_PREFIX=$PWD/../install_uq_handbook \
-D CMAKE_BUILD_TYPE=RELEASE \
-D Trilinos_ENABLE_EXPLICIT_INSTANTIATION=ON \
-D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES=ON \
-D BUILD_SHARED_LIBS=ON \
-D CMAKE_CXX_COMPILER=g++ \
-D CMAKE_C_COMPILER=gcc \
-D CMAKE_Fortran_COMPILER=gfortran \
-D Trilinos_ENABLE_Stokhos=ON \
-D Stokhos_ENABLE_TESTS=ON \
-D Stokhos_ENABLE_EXAMPLES=ON \
-D Trilinos_ENABLE_Trikota=OFF \
../Trilinos
```

This configure script enables Stokhos, its tests and examples, and all of the other packages that it may optionally use (except for TriKota, which is explicitly disabled, since it additionally requires the Dakota library [1]). It configures Trilinos for a serial build (no MPI or shared memory parallelism), using the GNU C (gcc), C++ (g++), and Fortran (gfortran) compilers (Note that the compilers used must support the C++11 standard. In the case of GNU compilers, this is version 4.7 or later). It is likely the user will need/want to customize this configure script based on their computer environment, desired set of Trilinos packages, and third-party libraries. Describing the full set of configuration options for Trilinos/Stokhos is well beyond the scope of this chapter. More information for controlling which packages are enabled, enabling parallelism, and enabling various third-party libraries is described at <https://trilinos.org/docs/files/TrilinosBuildReference.html>. The last line of the configure script is the path to the Trilinos sources, which in this case assumes a separate build directory next to the Trilinos source directory (it is not recommended to compile Trilinos within the source directory).

Trilinos is configured by running the configuration script such as the one described above. In a Unix environment, CMake defaults to generating GNU Makefiles to compile, test, and install Trilinos (Other generation options are available for integrating with common IDEs such as Xcode, Eclipse, and VisualStudio). For the default Makefile generation, Trilinos is then compiled by executing

```
make -j
```

The tests for all enabled packages can then be run (to ensure Trilinos configured and compiled correctly) by executing

```
make test
```

Optionally, the headers and libraries can be installed to the location specified by `CMAKE_INSTALL_PREFIX` via

```
make -j install
```

This will also install various CMake configuration files and GNU Makefiles allowing Trilinos to be easily compiled in to external applications. However this will not install any of the examples or tests. Those must be run from within the build directory for each package.

The above steps will build all of the examples described below within `path_to_build_directory/packages/stokhos/examples`. The executable for each example begins with the prefix `Stokhos_uq_handbook`.

6.2 Simple Polynomial Chaos Example

A brief example of applying Stokhos to compute a polynomial chaos approximation of a simple function using the template-based generic programming features of Stokhos is now described. This example is contained within the Stokhos code distribution as `Trilinos/packages/stokhos/example/uq_handbook/pce_example.cpp`.

This example computes a polynomial chaos approximation of the simple function

$$v(\xi) = \frac{1}{\log^2(u(\xi)) + 1} \quad (52.16)$$

where

$$u(\xi) = 1.0 + 0.1\xi_1 + 0.05\xi_2 + 0.01\xi_3 \quad (52.17)$$

and ξ_1, ξ_2, ξ_3 are Gaussian random variables with zero mean and unit variance. Thus $v(\xi)$ is approximated using Hermite polynomials. First the function is defined (written as a C++ template function), to which the polynomial chaos discretization will be applied.

```
#include "Stokhos_Sacado.hpp"

// The function to compute the polynomial chaos expansion of,
// written as a template function
template <class ScalarType>
ScalarType simple_function(const ScalarType& u) {
    ScalarType z = std::log(u);
    return 1.0/(z*z + 1.0);
}
```

Next comes the boilerplate code for the example, enabling short-hand for several classes used by the example.

```

int main(int argc, char **argv)
{
    // Typename of Polynomial Chaos scalar type
    typedef Stokhos::StandardStorage<int,double> storage_type;
    typedef Sacado::PCE::OrthogPoly<double, storage_type> pce_type;

    // Short-hand for several classes used below
    using Teuchos::Array;
    using Teuchos::RCP;
    using Teuchos::rcp;
    using Stokhos::OneDOrthogPolyBasis;
    using Stokhos::HermiteBasis;
    using Stokhos::CompletePolynomialBasis;
    using Stokhos::Quadrature;
    using Stokhos::TensorProductQuadrature;
    using Stokhos::Sparse3Tensor;
    using Stokhos::QuadOrthogPolyExpansion;

    try {

```

This includes a type alias for the polynomial chaosscalar type `pce_type` used for construction of a polynomial chaos approximation of the function contained within `simple_function`. Next a complete polynomial basis of total order 4 in 3 random variables using Hermite orthogonal polynomials is constructed.

```

// Basis of dimension 3, order 4
const int d = 3;
const int p = 4;
Array< RCP<const OneDOrthogPolyBasis<int,double> > > bases(d);
for (int i=0; i<d; i++) {
    bases[i] = rcp(new HermiteBasis<int,double>(p));
}
RCP<const CompletePolynomialBasis<int,double> > basis =
    rcp(new CompletePolynomialBasis<int,double>(bases));

```

Stokhos uses the reference-counted smart pointer class `RCP` extensively to enable correct memory management (The `RCP` class is similar to the smart pointer class `std::shared_ptr` now provided by the C++11 standard). Next a set of quadrature points commensurate with this polynomial basis are built, in this case a tensor product of one-dimensional Gauss-Hermite abscissas (Similarly, sparse-grid rules may be constructed. Stokhos provides an interface to the Dakota library [1] to construct these sparse grids as well as a smaller internal implementation for a more limited set of sparse-grid rules.) The sparse C_{ijk} tensor for this basis and an object implementing the overloaded math operators are also constructed. The quadrature data will be used for the transcendental operations and the sparse tensor for multiplications.

```

// Triple product tensor
RCP<Sparse3Tensor<int,double> > Cijk =
    basis->computeTripleProductTensor();

// Expansion method
RCP<QuadOrthogPolyExpansion<int,double> > expn =
    rcp(new QuadOrthogPolyExpansion<int,double>(basis, Cijk, quad));

```

The polynomial chaos expansion of u is then initialized (By default, each one-dimensional basis in Stokhos is constructed using the standard normalization of that basis. This ensures $\psi_i^1(\xi_i) = \xi_i$. By adding `true` to the second argument of the basis constructor, e.g., `HermiteBasis<int, double>(p, true)`, the basis is normalized to unit norm. However in this case $\psi_i^1(\xi_i) \neq \xi_i$ in general.), followed by the approximation of v using the overloaded operators:

```
// Polynomial expansion of u
pce_type u(expn);
u.term(0,0) = 1.0;           // zeroth order term
u.term(0,1) = 0.1;           // first order term for dimension 0
u.term(1,1) = 0.05;          // first order term for dimension 1
u.term(2,1) = 0.01;          // first order term for dimension 2

// Compute PCE expansion of function
pce_type v = simple_function(u);
```

Finally, the polynomial chaos coefficients, the mean and variance of v , and the value of the polynomial chaos approximation at a point are printed.

```
// Print u and v
std::cout << "\tu_=";
u.print(std::cout);
std::cout << "\tv_=";
v.print(std::cout);

// Compute moments
double mean = v.mean();
double std_dev = v.standard_deviation();

// Evaluate PCE and function at a point = 0.25 in each dimension
Teuchos::Array<double> pt(d);
for (int i=0; i<d; i++)
    pt[i] = 0.25;
double up = u.evaluate(pt);
double vp = simple_function(up);
double vp2 = v.evaluate(pt);

// Print results
std::cout << "\tv_mean_=" << mean << std::endl;
std::cout << "\tv_std_dev_=" << std_dev << std::endl;
std::cout << "\tv(0.25)_("true)_=" << vp << std::endl;
std::cout << "\tv(0.25)_("pce)_=" << vp2 << std::endl;
}

catch (std::exception& e) {
    std::cout << e.what() << std::endl;
}
```

This results in the following terminal output:

```
u = Stokhos::OrthogPolyApprox of size 35 in basis
Complete polynomial basis (Hermite, Hermite, Hermite):
(0, 0, 0) = 1
(1, 0, 0) = 0.1
(0, 1, 0) = 0.05
(0, 0, 1) = 0.01
(2, 0, 0) = 0
```

```
(1, 1, 0) = 0
(1, 0, 1) = 0
(0, 2, 0) = 0
(0, 1, 1) = 0
(0, 0, 2) = 0
(3, 0, 0) = 0
(2, 1, 0) = 0
(2, 0, 1) = 0
(1, 2, 0) = 0
(1, 1, 1) = 0
(1, 0, 2) = 0
(0, 3, 0) = 0
(0, 2, 1) = 0
(0, 1, 2) = 0
(0, 0, 3) = 0
(4, 0, 0) = 0
(3, 1, 0) = 0
(3, 0, 1) = 0
(2, 2, 0) = 0
(2, 1, 1) = 0
(2, 0, 2) = 0
(1, 3, 0) = 0
(1, 2, 1) = 0
(1, 1, 2) = 0
(1, 0, 3) = 0
(0, 4, 0) = 0
(0, 3, 1) = 0
(0, 2, 2) = 0
(0, 1, 3) = 0
(0, 0, 4) = 0
v = Stokhos::OrthogPolyApprox of size 35 in basis
Complete polynomial basis (Hermite, Hermite, Hermite):
(0, 0, 0) = 0.987468
(1, 0, 0) = 0.00350873
(0, 1, 0) = 0.00175446
(0, 0, 1) = 0.000350892
(2, 0, 0) = -0.00987111
(1, 1, 0) = -0.00987232
(1, 0, 1) = -0.00197446
(0, 2, 0) = -0.00246807
(0, 1, 1) = -0.000987232
(0, 0, 2) = -9.87232e-05
(3, 0, 0) = 0.000857605
(2, 1, 0) = 0.0012876
(2, 0, 1) = 0.00025752
(1, 2, 0) = 0.000644051
(1, 1, 1) = 0.000257622
(1, 0, 2) = 2.57621e-05
(0, 3, 0) = 0.000107326
(0, 2, 1) = 6.44014e-05
(0, 1, 2) = 1.28803e-05
(0, 0, 3) = 8.5869e-07
(4, 0, 0) = 2.14286e-05
(3, 1, 0) = 5.42848e-05
(3, 0, 1) = 1.0857e-05
(2, 2, 0) = 3.9738e-05
(2, 1, 1) = 1.58954e-05
(2, 0, 2) = 1.58954e-06
(1, 3, 0) = 1.32679e-05
(1, 2, 1) = 7.94889e-06
(1, 1, 2) = 1.5898e-06
(1, 0, 3) = 1.05987e-07
(0, 4, 0) = 1.57928e-06
(0, 3, 1) = 1.32759e-06
(0, 2, 2) = 3.97685e-07
(0, 1, 3) = 5.30256e-08
(0, 0, 4) = 2.64633e-09
```

```

v mean          = 0.987468
v std. dev.    = 0.0182702
v(0.25) (true) = 0.998464
v(0.25) (pce)  = 0.998569

```

In printing the polynomial chaos expansions of u and v , the tuple before each term indicates the polynomial orders for each random variable corresponding to that term, e.g., $(0, 1, 3)$ represents the zeroth-, first-, and third-order Hermite polynomial terms for ξ_1 , ξ_2 , and ξ_3 respectively.

6.3 Simple Ensemble Propagation Example

Next, a variation on the previous example that employs the embedded ensemble propagation capabilities is described. The source code for this example is contained within `Trilinos/packages/stokhos/example/uq_handbook/ensemble_example.cpp`. As seen below, much of the boilerplate setup is similar to the polynomial chaos example above.

```

#include "Stokhos_Sacado.hpp"
#include "Stokhos_Sacado_Kokkos.hpp"

// The function to compute the polynomial chaos expansion of,
// written as a template function
template <class ScalarType>
ScalarType simple_function(const ScalarType& u) {
    ScalarType z = std::log(u);
    return 1.0/(z*z + 1.0);
}

int main(int argc, char **argv)
{
    // Typename of Polynomial Chaos scalar type
    typedef Stokhos::StandardStorage<int,double> pce_storage_type;
    typedef Sacado::PCE::OrthogPoly<double, pce_storage_type> pce_type;

    // Typename of ensemble scalar type
    const int EnsembleSize = 8;
    typedef Kokkos::DefaultExecutionSpace ExecSpace;
    typedef Stokhos::StaticFixedStorage<int,double,EnsembleSize,ExecSpace>
        ensemble_storage_type;
    typedef Sacado::MP::Vector<ensemble_storage_type> ensemble_type;

    // Short-hand for several classes used below
    using Teuchos::Array;
    using Teuchos::RCP;
    using Teuchos::rcp;
    using Stokhos::OneDOrthogPolyBasis;
    using Stokhos::HermiteBasis;
    using Stokhos::CompletePolynomialBasis;
    using Stokhos::Quadrature;
    using Stokhos::TensorProductQuadrature;
    using Stokhos::Sparse3Tensor;
    using Stokhos::QuadOrthogPolyExpansion;

    try {

        // Basis of dimension 3, order 4
        const int d = 3;
        const int p = 4;

```

```

Array< RCP<const OneDOrthogPolyBasis<int,double> > > bases(d);
for (int i=0; i<d; i++) {
    bases[i] = rcp(new HermiteBasis<int,double>(p));
}
RCP<const CompletePolynomialBasis<int,double> > basis =
    rcp(new CompletePolynomialBasis<int,double>(bases));

// Quadrature method
RCP<const Quadrature<int,double> > quad =
    rcp(new TensorProductQuadrature<int,double>(basis));

// Triple product tensor
RCP<Sparse3Tensor<int,double> > Cijk =
    basis->computeTripleProductTensor();

// Expansion method
RCP<QuadOrthogPolyExpansion<int,double> > expn =
    rcp(new QuadOrthogPolyExpansion<int,double>(basis, Cijk, quad));

// Polynomial expansion of u
pce_type u(expn);
u.term(0,0) = 1.0;           // zeroth order term
u.term(0,1) = 0.1;           // first order term for dimension 0
u.term(1,1) = 0.05;          // first order term for dimension 1
u.term(2,1) = 0.01;          // first order term for dimension 2

```

The only differences are an additional include for the ensemble scalar type and a set of type aliases for defining the ensemble scalar type `ensemble_type` (The execution space template parameter, which determines where and how Kokkos employs shared memory parallelism, is necessary for controlling where the ensemble array data is allocated when a dynamically sized storage scheme is chosen. In this case the array size is fixed at compile time (implying no dynamic memory allocation) and therefore the execution space parameter is not strictly necessary, but is kept for compatibility purposes). The length of the ensemble array is determined by the `EnsembleSize` template parameter, which is chose to be 8 in this case.

Next, instead of computing the polynomial chaos approximation of $v(u)$ directly using the Stokhos overloaded operators for the polynomial chaos scalar type, these coefficients will be computed through numerical integration

$$\begin{aligned}
v(u(\xi)) &\approx \sum_{i=0}^P v^i \Psi^i(\xi), \\
v^i &= \frac{1}{\langle (\Psi^i)^2 \rangle} \int_{\Gamma} v(u(x)) \Psi^i(y) \rho(y) dy \\
&\approx \frac{1}{\langle (\Psi^i)^2 \rangle} \sum_{k=0}^Q w_k v(u(y_k)) \Psi^i(y_k), \quad i = 0, \dots, P,
\end{aligned} \tag{52.18}$$

using ensemble propagation to evaluate v at multiple points simultaneously. To implement this calculation, first the quadrature weights $\{w_k\}$, quadrature points $\{y_k\}$, and values of basis functions at the quadrature points $\{\Psi^i(y_k)\}$ are extracted.

```

// Compute PCE expansion with ensemble propagation
//

// Extract quadrature data
const int pce_size = basis->size();
const int num_quad_points = quad->size();
const Array<double>& quad_weights = quad->getQuadWeights();
const Array< Array<double> >& quad_points= quad->getQuadPoints();
const Array< Array<double> >& quad_values= quad->getBasisAtQuadPoints();

```

Then a loop occurs over the quadrature points in blocks of size EnsembleSize where u is evaluated at each quadrature point within the ensemble.

```

// Loop over quadrature points in blocks of size EnsembleSize
pce_type v(expn);
ensemble_type u_ensemble;
for (int qp_block=0; qp_block<num_quad_points; qp_block+=EnsembleSize) {
    const int qp_sz = qp_block+EnsembleSize <= num_quad_points ?
        EnsembleSize : num_quad_points-qp_block;

    // Evaluate u at each quadrature point
    for (int qp=0; qp<qp_sz; ++qp)
        u_ensemble.fastAccessCoeff(qp) =
            u.evaluate(quad_points[qp_block+qp], quad_values[qp_block+qp]);
    for (int qp=qp_sz; qp<EnsembleSize; ++qp)
        u_ensemble.fastAccessCoeff(qp) =
            u_ensemble.fastAccessCoeff(qp_sz-1);
}

```

If the number of quadrature points is not evenly divisible by EnsembleSize, the remaining values in the last ensemble are duplicated from the last quadrature point. Then v is evaluated for each value of u within its ensemble, using the Stokhos ensemble overloaded operators:

```

// Evaluate function at each quadrature point
ensemble_type v_ensemble = simple_function(u_ensemble);

```

Next the polynomial chaos coefficients of v are computed using Eq. 52.18.

```

// Sum results into PCE integral
for (int pc=0; pc<pce_size; ++pc) {
    const double inv_nrm_sq = 1.0 / basis->norm_squared(pc);
    for (int qp=0; qp<qp_sz; ++qp) {
        const double w = quad_weights[qp_block+qp];
        const double psi = quad_values[qp_block+qp][pc];
        v.fastAccessCoeff(pc) +=
            inv_nrm_sq * w * v_ensemble.fastAccessCoeff(qp) * psi;
    }
}

```

Finally the chaos coefficients are printed out as before, along with statistics on v and the true and polynomial chaos approximations of v at a given point.

```

// Print u and v
std::cout << "\tu=";
u.print(std::cout);
std::cout << "\tv=";
v.print(std::cout);

// Compute moments
double mean = v.mean();
double std_dev = v.standard_deviation();

// Evaluate PCE and function at a point = 0.25 in each dimension
Teuchos::Array<double> pt(d);
for (int i=0; i<d; i++)
    pt[i] = 0.25;
double up = u.evaluate(pt);
double vp = simple_function(up);
double vp2 = v.evaluate(pt);

// Print results
std::cout << "\tv_mean=" << mean << std::endl;
std::cout << "\tv_std.dev=" << std_dev << std::endl;
std::cout << "\tv(0.25)_(true)=" << vp << std::endl;
std::cout << "\tv(0.25)_(pce)=" << vp2 << std::endl;
}
catch (std::exception& e) {
    std::cout << e.what() << std::endl;
}
}

```

This results in screen output similar to the previous example, with slightly different values due to the different method in computing the polynomial chaos coefficients.

```

u = Stokhos::OrthogPolyApprox of size 35 in basis
Complete polynomial basis (Hermite, Hermite, Hermite):
(0, 0, 0) = 1
(1, 0, 0) = 0.1
(0, 1, 0) = 0.05
(0, 0, 1) = 0.01
(2, 0, 0) = 0
(1, 1, 0) = 0
(1, 0, 1) = 0
(0, 2, 0) = 0
(0, 1, 1) = 0
(0, 0, 2) = 0
(3, 0, 0) = 0
(2, 1, 0) = 0
(2, 0, 1) = 0
(1, 2, 0) = 0
(1, 1, 1) = 0
(1, 0, 2) = 0
(0, 3, 0) = 0
(0, 2, 1) = 0
(0, 1, 2) = 0
(0, 0, 3) = 0
(4, 0, 0) = 0
(3, 1, 0) = 0
(3, 0, 1) = 0
(2, 2, 0) = 0
(2, 1, 1) = 0
(2, 0, 2) = 0
(1, 3, 0) = 0
(1, 2, 1) = 0
(1, 1, 2) = 0
(1, 0, 3) = 0

```

```

(0, 4, 0) = 0
(0, 3, 1) = 0
(0, 2, 2) = 0
(0, 1, 3) = 0
(0, 0, 4) = 0
v = Stokhos::OrthogPolyApprox of size 35 in basis
Complete polynomial basis (Hermite, Hermite, Hermite):
(0, 0, 0) = 0.987468
(1, 0, 0) = 0.00350857
(0, 1, 0) = 0.00175432
(0, 0, 1) = 0.000350865
(2, 0, 0) = -0.00987065
(1, 1, 0) = -0.00987137
(1, 0, 1) = -0.00197427
(0, 2, 0) = -0.00246783
(0, 1, 1) = -0.000987133
(0, 0, 2) = -9.87133e-05
(3, 0, 0) = 0.00085764
(2, 1, 0) = 0.00128778
(2, 0, 1) = 0.000257557
(1, 2, 0) = 0.0006439974
(1, 1, 1) = 0.00025759
(1, 0, 2) = 2.5759e-05
(0, 3, 0) = 0.000107318
(0, 2, 1) = 6.43954e-05
(0, 1, 2) = 1.28791e-05
(0, 0, 3) = 8.58606e-07
(4, 0, 0) = 2.09834e-05
(3, 1, 0) = 4.95884e-05
(3, 0, 1) = 9.91768e-06
(2, 2, 0) = 3.64855e-05
(2, 1, 1) = 1.45945e-05
(2, 0, 2) = 1.45945e-06
(1, 3, 0) = 1.21902e-05
(1, 2, 1) = 7.30616e-06
(1, 1, 2) = 1.46126e-06
(1, 0, 3) = 9.74177e-08
(0, 4, 0) = 1.4714e-06
(0, 3, 1) = 1.21915e-06
(0, 2, 2) = 3.65346e-07
(0, 1, 3) = 4.87138e-08
(0, 0, 4) = 2.43247e-09
v mean      = 0.987468
v std. dev. = 0.018269
v(0.25) (true) = 0.998464
v(0.25) (pce)  = 0.998565

```

6.4 Nonlinear Stochastic Galerkin Example

The last example demonstrates the use of Stokhos for solving a nonlinear stochastic Galerkin problem. It computes the solution to the following simple nonlinear system of equations:

$$\begin{aligned} a^2 - x_0 &= 0 \\ x_1^2 - x_0 &= 0 \end{aligned} \tag{52.19}$$

where a is a model parameter. It is easy to see that the solution is $x_0 = a^2$, $x_1 = a$. The problem will be made stochastic by letting a become a random variable:

$$a(\xi) = 2 + \xi \tag{52.20}$$

where ξ is uniform random variable over $(-1, 1)$. Thus x_0 and x_1 will be approximated in terms of Legendre polynomials $\{\Psi^i(\xi)\}$ up to a given order N :

$$\begin{aligned} x_0(\xi) &\approx \hat{x}_0(\xi) \equiv \sum_{i=0}^N x_0^i \Psi^i(\xi) \\ x_1(\xi) &\approx \hat{x}_1(\xi) \equiv \sum_{i=0}^N x_1^i \Psi^i(\xi). \end{aligned} \quad (52.21)$$

Equations for the unknown coefficients $\{x_j^i\}$ are formulated according to the stochastic Galerkin method described above:

$$\left. \begin{aligned} \langle (a^2 - \hat{x}_0) \Psi^i \rangle &= 0 \\ \langle (\hat{x}_1^2 - \hat{x}_0) \Psi^i \rangle &= 0 \end{aligned} \right\}, \quad i = 0, \dots, N. \quad (52.22)$$

It is easy to see the exact solution is

$$\begin{aligned} x_0(\xi) &= a(\xi)^2 = 4 + 4\xi + \xi^2 = 4 \frac{1}{3} \Psi^0(\xi) + 4 \Psi^1(\xi) + \frac{2}{3} \Psi^2(\xi), \\ x_1(\xi) &= a(\xi) = 2 + \xi = 2\Psi^0(\xi) + \Psi^1(\xi). \end{aligned} \quad (52.23)$$

since $\Psi^0(\xi) = 1$, $\Psi^1(\xi) = \xi$, and $\Psi^2(\xi) = \frac{3}{2}\xi^2 - \frac{1}{2}$.

The steps necessary for formulating and solving this problem using Stokhos are described below. As with the two previous examples, the source code for this example may be found within the Stokhos distribution in the directory `Trilinos/packages/stokhos/examples/uq_handbook`. Unlike the previous two examples however, it is split into multiple source files. This example demonstrates:

- Encapsulating a nonlinear, stochastic problem into the EpetraExt ModelEvaluator [15] nonlinear interface.
- Using the Stokhos overloaded operators for computing the stochastic Galerkin residual and Jacobian coefficients in the implementation of this interface.
- The Stokhos interface for formulating the nonlinear stochastic Galerkin problem.
- Solving this nonlinear problem using the NOX [24] nonlinear solver package in Trilinos.
- Directing the NOX nonlinear solver to use various stochastic Galerkin preconditioners provided by Stokhos.

First, a class called `SimpleME` declared below contains an implementation of the nonlinear system (52.19). This class implements the `EpetraExt::ModelEvaluator` interface, which is an interface designed to provide functionality for a variety of different nonlinear solution and analysis methods: explicit and implicit time integration, direct-to-steady-state nonlinear solvers, optimization, stability

analysis, and uncertainty quantification. It leverages the Epetra package [14] in Trilinos for distributed memory (i.e., MPI) parallelism. Fully describing the Epetra object model and use of Epetra classes is beyond the scope of this chapter; however the basic Epetra classes used within this example are:

- **Epetra_Comm**: An abstract object representing a distributed memory communicator. The most common examples are **Epetra_SerialComm** when there is no parallelism and **Epetra_MPIComm** for an MPI-based communicator. It provides methods for determining the size and layout of a parallel machine, as well as basic methods for communicating data between processes.
- **Epetra_Map**: A concrete class representing the distribution of an Epetra object across a parallel machine. Each entry in the object (such as a vector entry or row of a matrix) is given a unique identifier across the parallel distribution (called a GID). **Epetra_Map** then encapsulates which GIDs are located on each processor. While GIDs must be unique, the entry corresponding to a given GID may be located on multiple processors, often called at overlapped map.
- **Epetra_Vector**: A concrete class representing double-precision vector distributed across a parallel machine. It contains an **Epetra_Map** which describes how the vector entries are distributed. The class provides various routines for accessing vector entries and performing vector arithmetic.
- **Epetra_CrsGraph**: A concrete class which represents the sparsity pattern of a general, unstructured sparse matrix, stored in the compressed row storage format. The rows of the graph are distributed in parallel through the specification of the graph's row map.
- **Epetra_CrsMatrix**: A concrete class representing a double-precision sparse matrix, which uses **Epetra_CrsGraph** to store the matrix sparsity pattern. It provides methods for accessing matrix entries as well as common linear algebra operations such as matrix-vector products and sparse triangular solves.

The **EpetraExt::ModelEvaluator** is designed to provide an interface for general (differential) algebraic systems of the form

$$\begin{aligned} 0 &= f(\dot{x}, x, p_1, \dots, p_{N_p}, t), \\ h_j &= g_j(\dot{x}, x, p_1, \dots, p_{N_p}, t), \quad k = 1, \dots, N_g. \end{aligned} \tag{52.24}$$

Here f is the set of residual equations defining the nonlinear system, g_j , $j = 1, \dots, N_g$ is a set of response functions, x is the solution vector, \dot{x} is its time derivative, p_i , $i = 1, \dots, N_p$ are model parameter vectors, and t is time. Each quantity except t above is represented by an **Epetra_Vector** with an arbitrary **Epetra_Map**. The input quantities x , \dot{x} , p_i , and t are called in-args, whereas the output quantities f , g_j , and various derivatives of these quantities such as

$$W = \alpha \frac{\partial f}{\partial \dot{x}} + \beta \frac{\partial f}{\partial x} \tag{52.25}$$

are called out-args. The `EpetraExt::ModelEvaluator` then provides an interface for specifying what in- and out-args are supported by the model, their parallel maps, the initial values of the in-args, and a function to compute the supported out-args from given values of the in-args.

For stochastic Galerkin methods, each in- and out-arg has a corresponding version that represents the polynomial chaos expansion of that quantity (with the name of that in- or out-arg appended with `_sg`). For example, the x in-arg (called `IN_ARG_x`) has a corresponding in-arg (called `IN_ARG_x_sg`) that represents

$$\hat{x}(\xi) = \sum_{i=0}^P x^i \Psi^i(\xi). \quad (52.26)$$

These in- and out-args are encapsulated within the `EpetraVectorOrthogPoly` and `EpetraOperatorOrthogPoly` classes provided by Stokhos. These classes are similar to the polynomial chaos scalar type in Stokhos; however their coefficients are Epetra vectors or matrices instead of scalars.

The declaration of the model evaluator representing (52.19) is shown below and is contained within the file `SimpleME.hpp`. In most cases the function names should be self-explanatory. For this problem there are no response functions ($N_g = 0$) and so functions dealing with response functions resort to their default (empty) implementation. Furthermore, there is only a single parameter vector to store a . The class stores maps for constructing the solution vector x where each vector entry is stored on a unique processor (`x_map`) and where all processors have every entry of x (`x_overlapped_map`), an object for importing entries between these two maps (`importer`), the initial guess for x (`x_init`), a vector to store x in the overlapped distribution (`x_overlapped`), a map to generate the parameter vector p (`p_map`), the initial value of p (`p_init`), the string name of the parameter (`p_names`), and the sparse matrix graph for the matrix $\partial f / \partial x$ (`graph`).

```
#include "Teuchos_RCP.hpp"
#include "Teuchos_Array.hpp"

#include "EpetraExt_ModelEvaluator.h"

#include "Epetra_Map.h"
#include "Epetra_LocalMap.h"
#include "Epetra_Import.h"
#include "Epetra_CrsGraph.h"
#include "Epetra_CrsMatrix.h"

/* Simple model evaluator demonstrating how to use the Stokhos
 * ModelEvaluator to solve a nonlinear stochastic Galerkin problem.
 * It represents the simple function
 *
 *   f(x,a) = | a^2 - x_0 |
 *             | x_1^2 - x_0 |
 *
 * where x=[x_0 x_1]^T and a is a parameter. It has a root at x=[a^2 a]^T.
 * The parameter a may be represented by a given polynomial chaos expansion,
 * and the corresponding expansion for x computed by Stokhos.
 */
```

```
class SimpleME : public EpetraExt::ModelEvaluator {
public:

    ///! Constructor
    SimpleME(const Teuchos::RCP<const Epetra_Comm>& comm);

    /** \name Overridden from EpetraExt::ModelEvaluator . */
    // @{
    // ! Return solution vector map
    Teuchos::RCP<const Epetra_Map> get_x_map() const;

    // ! Return residual vector map
    Teuchos::RCP<const Epetra_Map> get_f_map() const;

    // ! Return parameter vector map
    Teuchos::RCP<const Epetra_Map> get_p_map(int l) const;

    // ! Return array of parameter names
    Teuchos::RCP<const Teuchos::Array<std::string>> get_p_names(int l) const;

    // ! Return initial solution
    Teuchos::RCP<const Epetra_Vector> get_x_init() const;

    // ! Return initial parameters
    Teuchos::RCP<const Epetra_Vector> get_p_init(int l) const;

    // ! Create W = alpha*M + beta*J matrix
    Teuchos::RCP<Epetra_Operator> create_W() const;

    // ! Create InArgs
    InArgs createInArgs() const;

    // ! Create OutArgs
    OutArgs createOutArgs() const;

    // ! Evaluate model on InArgs
    void evalModel(const InArgs& inArgs, const OutArgs& outArgs) const;
    //}

protected:

    // ! Solution vector map
    Teuchos::RCP<Epetra_Map> x_map;

    // ! Overlapped solution vector map
    Teuchos::RCP<Epetra_Map> x_overlapped_map;

    // ! Importer to overlapped distribution
    Teuchos::RCP<Epetra_Import> importer;

    // ! Initial guess
    Teuchos::RCP<Epetra_Vector> x_init;

    // ! Overlapped solution vector
    Teuchos::RCP<Epetra_Vector> x_overlapped;

    // ! Parameter vector map
    Teuchos::RCP<Epetra_Map> p_map;

    // ! Initial parameters
    Teuchos::RCP<Epetra_Vector> p_init;

    // ! Parameter names
    Teuchos::RCP<Teuchos::Array<std::string>> p_names;
```

```
//! Jacobian graph
Teuchos::RCP<Epetra_CrsGraph> graph;
};
```

Next the implementation of the member functions of `SimpleME` is shown below, which is contained within the file `SimpleME.cpp`. The C++ function defining f is encapsulated in the nonmember function `func` that is templated on the scalar type. This allows a single implementation compute the residual vector f , its derivative $\partial f / \partial x$ using Sacado [28, 29] automatic differentiation [13], and the polynomial chaos approximations of these functions using the Stokhos polynomial chaos scalar type and overloaded operators.

```
#include "SimpleME.hpp"
#include "Teuchos_Assert.hpp"
#include "Stokhos_Epetra.hpp"
#include "Stokhos_Sacado.hpp"
#include "Sacado.hpp"

namespace {

template <typename ScalarA, typename ScalarX>
void func(const ScalarA& a, const ScalarX x[2], ScalarX y[2]) {
    y[0] = a*a - x[0];
    y[1] = x[1]*x[1] - x[0];
}

}
```

The constructor initializes all of the class data members described above. Since the solution vector x only has two entries, `x_map` stores two GIDs, starting with 0. The overlapped version of the map is constructed using `Epetra_LocalMap` which creates a map with all entries replicated on each processor. The initial guess for x is initialized to 1.5. The parameter map only has one entry, and the corresponding vector is initialized to 2.0. Finally the sparse matrix graph is constructed as a dense 2×2 matrix, with either of the two rows placed on a processor only if the GID for that row is contained within the map on that processor.

```
SimpleME::SimpleME(const Teuchos::RCP<const Epetra_Comm>& comm)
{
    // Solution vector map
    x_map = Teuchos::rcp(new Epetra_Map(2, 0, *comm));

    // Overlapped solution vector map
    x_overlapped_map = Teuchos::rcp(new Epetra_LocalMap(2, 0, *comm));

    // Importer
    importer = Teuchos::rcp(new Epetra_Import(*x_overlapped_map, *x_map));

    // Initial guess, initialized to 1.5
    x_init = Teuchos::rcp(new Epetra_Vector(*x_map));
    x_init->PutScalar(1.5);

    // Overlapped solution vector
    x_overlapped = Teuchos::rcp(new Epetra_Vector(*x_overlapped_map));
```

```

// Parameter vector map
p_map = Teuchos::rcp(new Epetra_LocalMap(1, 0, *comm));

// Initial parameters
p_init = Teuchos::rcp(new Epetra_Vector(*p_map));
(*p_init)[0] = 2.0;

// Parameter names
p_names = Teuchos::rcp(new Teuchos::Array<std::string>(1));
(*p_names)[0] = "alpha";

// Jacobian graph (dense 2x2 matrix)
graph = Teuchos::rcp(new Epetra_CrsGraph(Copy, *x_map, 2));
int indices[2];
indices[0] = 0; indices[1] = 1;
if (x_map->MyGID(0))
    graph->InsertGlobalIndices(0, 2, indices);
if (x_map->MyGID(1))
    graph->InsertGlobalIndices(1, 2, indices);
graph->FillComplete();
graph->OptimizeStorage();
}
}

```

Next comes the implementation of several boilerplate functions whose implementation is self-explanatory.

```

Teuchos::RCP<const Epetra_Map>
SimpleME::get_x_map() const
{
    return x_map;
}

Teuchos::RCP<const Epetra_Map>
SimpleME::get_f_map() const
{
    return x_map;
}

Teuchos::RCP<const Epetra_Map>
SimpleME::get_p_map(int l) const
{
    TEUCHOS_TEST_FOR_EXCEPTION(l != 0,
        std::logic_error,
        std::endl <<
        "Error! SimpleME::get_p_map(): " <<
        "Invalid parameter index_l=" << l << std::endl);

    return p_map;
}

Teuchos::RCP<const Teuchos::Array<std::string> >
SimpleME::get_p_names(int l) const
{
    TEUCHOS_TEST_FOR_EXCEPTION(l != 0,
        std::logic_error,
        std::endl <<
        "Error! SimpleME::get_p_names(): " <<
        "Invalid parameter index_l=" << l << std::endl);

    return p_names;
}

Teuchos::RCP<const Epetra_Vector>
SimpleME::get_x_init() const
{
}

```

```

    return x_init;
}

Teuchos::RCP<const Epetra_Vector>
SimpleME::get_p_init(int l) const
{
    TEUCHOS_TEST_FOR_EXCEPTION(l != 0,
                               std::logic_error,
                               std::endl <<
                               "Error! SimpleME::get_p_init(): "
                               "Invalid parameter index l = " << l << std::endl);

    return p_init;
}

Teuchos::RCP<Epetra_Operator>
SimpleME::create_W() const
{
    Teuchos::RCP<Epetra_CrsMatrix> A =
        Teuchos::rcp(new Epetra_CrsMatrix(Copy, *graph));
    A->FillComplete();
    A->OptimizeStorage();
    return A;
}

```

Next are the functions that determine what in- and out-args are supported by this model. In this case the model supports x and a single parameter vector as in-args, as well as the polynomial chaos expansions of these quantities. It also supports the Stokhos basis, quadrature, and expansion objects which are used to compute the stochastic Galerkin residual and Jacobian. As out-args it supports f , $W = \partial f / \partial x$, and their polynomial chaos expansions.

```

EpPetraExt::ModelEvaluator::InArgs
SimpleME::createInArgs() const
{
    InArgsSetup inArgs;
    inArgs.setModelEvalDescription("Simple_Model_Evaluator");

    // Deterministic InArgs
    inArgs.setSupports(IN_ARG_x,true);
    inArgs.set_Np(1); // 1 parameter vector

    // Stochastic InArgs
    inArgs.setSupports(IN_ARG_x_sg,true);
    inArgs.setSupports(IN_ARG_p_sg, 0, true); // 1 SG parameter vector
    inArgs.setSupports(IN_ARG_sg_basis,true);
    inArgs.setSupports(IN_ARG_sg_quadrature,true);
    inArgs.setSupports(IN_ARG_sg_expansion,true);

    return inArgs;
}

EpPetraExt::ModelEvaluator::OutArgs
SimpleME::createOutArgs() const
{
    OutArgsSetup outArgs;
    outArgs.setModelEvalDescription("Simple_Model_Evaluator");

    // Deterministic OutArgs
    outArgs.set_Np_Ng(1, 0);
    outArgs.setSupports(OUT_ARG_f,true);
    outArgs.setSupports(OUT_ARG_W,true);

    // Stochastic OutArgs

```

```

    outArgs.setSupports(OUT_ARG_f_sg, true);
    outArgs.setSupports(OUT_ARG_W_sg, true);

    return outArgs;
}

```

Next the evalModel function is shown below that computes the requested out-args from the given set of in-args. There are essentially two large blocks of code within its implementation. The first implements a deterministic calculation where x and p are supplied as in-args and f and/or W are requested as out-args. This is signaled by `inArgs.get_x()` returning a non-null RCP. In this case, the vector is copied to the overlapped distribution so that all entries are on all processors, x_0 and x_1 are extracted, the parameter value a is extracted, and the residual f or Jacobian W are computed based on which out-args are non-null. These quantities are obtained by evaluating `func` with x_0 and x_1 or AD objects representing x_0 and x_1 to compute the necessary partial derivatives.

```

void
SimpleME::evalModel(const InArgs& inArgs, const OutArgs& outArgs) const
{
    //
    // Deterministic calculation
    //

    // Solution vector
    Teuchos::RCP<const Epetra_Vector> x = inArgs.get_x();
    if (x != Teuchos::null) {
        x_overlapped->Import(*x, *importer, Insert);
        double x0 = (*x_overlapped)[0];
        double x1 = (*x_overlapped)[1];

        // Parameters
        Teuchos::RCP<const Epetra_Vector> p = inArgs.get_p(0);
        if (p == Teuchos::null)
            p = p_init;
        double a = (*p)[0];

        // Residual
        // f = | a*a - x0 |
        //     | x1*x1 - x0 |
        // where a = p[0].
        Teuchos::RCP<Epetra_Vector> f = outArgs.get_f();
        if (f != Teuchos::null) {
            double x[2] = { x0, x1 };
            double y[2];
            func(a, x, y);

            if (x_map->MyGID(0)) {
                int row = 0;
                f->ReplaceGlobalValues(1, &y[0], &row);
            }
            if (x_map->MyGID(1)) {
                int row = 1;
                f->ReplaceGlobalValues(1, &y[1], &row);
            }
        }

        // Jacobian
        // J = | -1   0   |
        //      | -1  2*x1 |
        Teuchos::RCP<Epetra_Operator> W = outArgs.get_W();
    }
}

```

```

if (W != Teuchos::null) {
    typedef Sacado::Fad<double,2> fad_type;
    fad_type x[2], y[2];
    x[0] = fad_type(2, 0, x0);
    x[1] = fad_type(2, 1, x1);
    func(a, x, y);
}

Teuchos::RCP<Epetra_CrsMatrix> jac =
    Teuchos::rcp_dynamic_cast<Epetra_CrsMatrix>(W, true);
int indices[2] = { 0, 1 };
if (x_map->MyGID(0)) {
    int row = 0;
    jac->ReplaceGlobalValues(row, 2, y[0].dx(), indices);
}
if (x_map->MyGID(1)) {
    int row = 1;
    jac->ReplaceGlobalValues(row, 2, y[1].dx(), indices);
}
}

```

The second block, shown below, implements the stochastic Galerkin calculation and is evaluated if the `x_sg` in-arg is not null. In this case, the orthogonal polynomial basis and expansion object are also extracted, which are used in the polynomial chaos evaluation of f and W (this code uses type aliases such as `InArgs::sg_const_vector_t` for `EpetraVectorOrthogPoly` for simplicity). The workflow is similar to the deterministic calculation above; however values for x_0 and x_1 are created using the polynomial chaos scalar type instead of double-precision numbers. This requires looping over each vector entry of `x_sg`, importing the entries of that vector, extracting each coefficient, and setting the corresponding polynomial chaos coefficient in `x0` and `x1`. In a similar manner the polynomial chaos expansion of a is extracted from the `p_sg` in-arg. Finally the polynomial chaos expansions of f and W are computed using the Stokhos overloaded operators described above by calling the same template function `func`. For the Jacobian matrix W , this uses the polynomial chaos scalar type nested within the Sacado automatic differentiation scalar type.

```

// Stochastic Galerkin calculation
//
// Stochastic solution vector
InArgs::sg_const_vector_t x_sg = inArgs.get_x_sg();
if (x_sg != Teuchos::null) {
    // Get stochastic expansion data
    Teuchos::RCP<const Stokhos::OrthogPolyBasis<int,double> > basis =
        inArgs.get_sg_basis();
    Teuchos::RCP<Stokhos::OrthogPolyExpansion<int,double> > expn =
        inArgs.get_sg_expansion();
    typedef Stokhos::StandardStorage<int,double> storage_type;
    typedef Sacado::PCE::OrthogPoly<double, storage_type> pce_type;

    pce_type x0(expn), x1(expn);
    for (int i=0; i<basis->size(); i++) {
        x_overlapped->Import((*x_sg)[i], *importer, Insert);
        x0.fastAccessCoeff(i) = (*x_overlapped)[0];
        x1.fastAccessCoeff(i) = (*x_overlapped)[1];
    }
}

```

```

// Stochastic parameters
InArgs::sg_const_vector_t p_sg = inArgs.get_p_sg(0);
pce_type a(expn);
if (p_sg != Teuchos::null) {
    for (int i=0; i<basis->size(); i++) {
        a.fastAccessCoeff(i) = (*p_sg)[i][0];
    }
}

// Stochastic residual
// f[i] = |<a*x - x0, psi_i>/<psi_i^2>| +
//         |<x1*x1 - x0, psi_i>/<psi_i^2>|
OutArgs::sg_vector_t f_sg = outArgs.get_f_sg();
if (f_sg != Teuchos::null) {
    pce_type x[2] = { x0, x1 };
    pce_type y[2];
    func(a, x, y);

    if (x_map->MyGID(0)) {
        int row = 0;
        for (int i=0; i<basis->size(); i++) {
            double c = y[0].coeff(i);
            (*f_sg)[i].ReplaceGlobalValues(1, &c, &row);
        }
    }
    if (x_map->MyGID(1)) {
        int row = 1;
        for (int i=0; i<basis->size(); i++) {
            double c = y[1].coeff(i);
            (*f_sg)[i].ReplaceGlobalValues(1, &c, &row);
        }
    }
}

// Stochastic Jacobian
// J[0] = | -1      0      |,   J[i] = |  0      0      |,   i > 0
//          | -1  2*x0[0] |
OutArgs::sg_operator_t W_sg = outArgs.get_W_sg();
if (W_sg != Teuchos::null) {
    typedef Sacado::Fad::SFad<pce_type,2> fad_type;
    fad_type x[2], y[2];
    x[0] = fad_type(2, 0, x0);
    x[1] = fad_type(2, 1, x1);
    func(a, x, y);

    for (int i=0; i<basis->size(); i++) {
        Teuchos::RCP<Epetra_CrsMatrix> jac =
            Teuchos::rcp_dynamic_cast<Epetra_CrsMatrix>(W_sg->getCoeffPtr(i),
                                                          true);
        int indices[2] = { 0, 1 };
        if (x_map->MyGID(0)) {
            int row = 0;
            double values[2] = { y[0].dx(0).coeff(i), y[0].dx(1).coeff(i) };
            jac->ReplaceGlobalValues(row, 2, values, indices);
        }
        if (x_map->MyGID(1)) {
            int row = 1;
            double values[2] = { y[1].dx(0).coeff(i), y[1].dx(1).coeff(i) };
            jac->ReplaceGlobalValues(row, 2, values, indices);
        }
    }
}
}

```

One can see that the use of the Stokhos polynomial chaos scalar type in conjunction with C++ templates for implementing the equations defining the nonlinear model makes evaluation of a stochastic Galerkin residual and Jacobian nearly as simple as evaluating deterministic residuals and Jacobians. Furthermore, most of the code shown above is independent of the actual problem being solved and only depends on the number of degrees-of-freedom, the parallel layout, and the sparsity pattern of the matrix. Thus in a large simulation code, this code can often be written once and then applied to a wide range of different physical problems with different implementations of the templated, physics-specific code (which are in turn independent of Stokhos).

Finally the driver code that uses the `SimpleME` model evaluator above to actually solve the stochastic Galerkin system is shown below and is contained within the file `nonlinear_sg_example.cpp`. It includes a function to create and initialize the NOX nonlinear solver (`create_nox_solver()`) whose implementation is suppressed for brevity.

```
#include <iostream>

// NOX
#include "NOX.H"
#include "NOX_Epetra.H"

// Epetra communicator
#ifndef HAVE_MPI
#include "Epetra_MpiComm.h"
#else
#include "Epetra_SerialComm.h"
#endif

// Stokhos Stochastic Galerkin
#include "Stokhos_Epetra.hpp"

// Utilities
#include "Teuchos_GlobalMPISession.hpp"
#include "Teuchos_StandardCatchMacros.hpp"

// Our model
#include "SimpleME.hpp"

// Function to setup the NOX nonlinear solver from a given model evaluator
Teuchos::RCP<NOX::Solver::Generic>
create_nox_solver(int MyPID,
                  const Teuchos::RCP<EpetraExt::ModelEvaluator>& model);
```

First the Epetra communicator is initialized, depending on whether MPI was enabled (note that the configure script above does not enable MPI).

```
int main(int argc, char *argv[]) {
    using Teuchos::rcp;
    using Teuchos::RCP;
    using Teuchos::Array;
    using Teuchos::ParameterList;
    using Stokhos::OneDOrthogPolyBasis;
    using Stokhos::LegendreBasis;
    using Stokhos::CompletePolynomialBasis;
```

```

using Stokhos::OrthogPolyExpansion;
using Stokhos::AlgebraicOrthogPolyExpansion;
using Stokhos::ParallelData;
using Stokhos::SGModelEvaluator;
using Stokhos::EpetraVectorOrthogPoly;

// Initialize MPI
Teuchos::GlobalMPISession mpiSession(&argc, &argv);

int MyPID;
bool success = true;
try {

    // Create a communicator for Epetra objects
    RCP<const Epetra_Comm> globalComm;
#ifndef HAVE_MPI
    globalComm = rcp(new Epetra_MpiComm(MPI_COMM_WORLD));
#else
    globalComm = rcp(new Epetra_SerialComm);
#endif
    MyPID = globalComm->MyPID();
}

```

Then the Stokhos objects describing the polynomial chaos discretization are generated, as described before. This code creates a basis for a single uniform random variable using Legendre polynomials up to order $N = 5$. Included here as well is code to enable Stokhos to distribute polynomial chaos coefficients across the parallel machine. By changing `num_spatial_procs` to 1, the polynomial chaos coefficients for x would be distributed in parallel (the default -1 used here means to use all processors for distributing the entries of x in parallel).

```

// Create Stochastic Galerkin basis and expansion
const int p = 5;
Array< RCP<const OneDOrthogPolyBasis<int,double> > > bases(1);
bases[0] = rcp(new LegendreBasis<int,double>(p));
RCP<const CompletePolynomialBasis<int,double>> basis =
    rcp(new CompletePolynomialBasis<int,double>(bases));
RCP<Stokhos::Sparse3Tensor<int,double>> Cijk =
    basis->computeTripleProductTensor();
RCP<OrthogPolyExpansion<int,double>> expansion =
    rcp(new AlgebraicOrthogPolyExpansion<int,double>(basis, Cijk));

// Create stochastic parallel distribution
int num_spatial_procs = -1;
ParameterList parallelParams;
parallelParams.set("Number_of_Spatial_Processors", num_spatial_procs);
RCP<ParallelData> sg_parallel_data =
    rcp(new ParallelData(basis, Cijk, globalComm, parallelParams));
RCP<const Epetra_Comm> app_comm = sg_parallel_data->getSpatialComm();

```

Next, the model evaluator encapsulating the model is created. Parameters for describing how the stochastic Galerkin matrix should be preconditioned are also specified. In this case a mean-based preconditioner using an incomplete LU factorization of the mean is chosen. Other options for both the stochastic and mean preconditioners are available (most stochastic preconditioners require specifying the mean preconditioner to control how the inverse of the mean blocks is approximated). For example, by replacing “mean-based” with “approximate Gauss-Seidel” allows a Gauss-Seidel method applied to the stochastic blocks to be chosen.

```
// Create application model evaluator
RCP<EpetraExt::ModelEvaluator> model = rcp(new SimpleME(app_comm));

// Setup stochastic Galerkin algorithmic parameters
RCP<ParameterList> sgParams = rcp(new ParameterList);
ParameterList& sgPrecParams = sgParams->sublist("SG_Preconditioner");
sgPrecParams.set("Preconditioner_Method", "Mean-based");
//sgPrecParams.set("Preconditioner Method", "Approximate Gauss-Seidel");
sgPrecParams.set("Mean_Preconditioner_Type", "Ifpack");
```

Next an adapter called `SGModelEvaluator` is created that wraps the `SimpleME` model evaluator and translates the stochastic Galerkin system to a deterministic system suitable for traditional nonlinear solution and analysis algorithms. For example, it translates a polynomial chaos representation of x to and from a block-vector representation:

$$\hat{x} \equiv \sum_{i=0}^P x^i \Psi^i(\xi) \leftrightarrow X \equiv \begin{bmatrix} x^0 \\ \vdots \\ x^P \end{bmatrix} \quad (52.27)$$

Similarly, given the polynomial chaos approximation $\partial f / \partial x \approx \sum_{i=0}^P A^i \Psi^i(\xi)$, this adapter creates an operator for $\partial F / \partial X$ using Eq. 52.14. This allows stochastic models implementing the polynomial chaos in- and out-args to be applied to standard time integrators, nonlinear solvers, and optimizers. In this case, this model evaluator will be given to the NOX nonlinear solver to solve for the unknown polynomial chaos coefficients of x .

```
// Create stochastic Galerkin model evaluator
RCP<SGModelEvaluator> sg_model =
    rcp(new SGModelEvaluator(model, basis, Teuchos::null, expansion,
                           sg_parallel_data, sgParams));
```

Next the initial guess for `x_sg` is constructed by taking the initial guess from `SimpleME` and setting the mean term with the higher-order coefficients set to zero. Similarly the polynomial chaos expansion of the parameter is set to the mean term provided by `SimpleME`, the first-order term set to 1, and all higher-order terms set to 0.

```
// Stochastic Galerkin initial guess
// Set the mean to the deterministic initial guess, higher-order terms
// to zero
RCP<EpetraVectorOrthogPoly> x_init_sg = sg_model->create_x_sg();
x_init_sg->init(0.0);
(*x_init_sg)[0] = *(model->get_x_init());
sg_model->set_x_sg_init(*x_init_sg);

// Stochastic Galerkin parameters
// Linear expansion with the mean given by the deterministic initial
// parameter values, linear terms equal to 1, and higher order terms
// equal to zero.
RCP<EpetraVectorOrthogPoly> p_init_sg = sg_model->create_p_sg(0);
p_init_sg->init(0.0);
(*p_init_sg)[0] = *(model->get_p_init());
```

```

for (int i=0; i<model->get_p_map(0)->NumMyElements(); i++)
    (*p_init_sg)[i+1][i] = 1.0;
sg_model->set_p_sg_init(0, *p_init_sg);
std::cout << "Stochastic_Galerkin_parameter_expansion_=" << std::endl
    << *p_init_sg << std::endl;

```

Finally, the NOX nonlinear solver is created and solved. The details of creating this solver are not shown here. Once the nonlinear solver terminates, the final solution is extracted, converted to a polynomial chaos expansion, and then printed to the screen.

```

// Build nonlinear solver (implemented above)
RCP<NOX::Solver<Generic> solver = create_nox_solver(MyPID, sg_model);

// Solve the system
NOX::StatusTest::StatusType status = solver->solve();

// Get final solution
const Epetra_Vector& finalSolution = get_final_solution(*solver);

// Convert block Epetra_Vector to orthogonal polynomials
RCP<Stokhos::EpetraVectorOrthogPoly> x_sg =
    sg_model->create_x_sg(View, &finalSolution);

if (MyPID == 0)
    std::cout << "Final_Solution_=" << std::endl;
    std::cout << *x_sg << std::endl;

if (status != NOX::StatusTest::Converged)
    success = false;
}
TEUCHOS_STANDARD_CATCH_STATEMENTS(true, std::cerr, success);

if (success && MyPID == 0)
    std::cout << "Example_Passed!" << std::endl;

if (!success)
    return 1;
return 0;
}

```

The screen output of the nonlinear solution process is shown below, along with the computed final solution.

```

Teuchos::GlobalMPISession::GlobalMPISession(): started serial run
Stochastic Galerkin parameter expansion =
Stokhos::VectorOrthogPoly of global size 6, local size 6 in basis
Complete polynomial basis (Legendre):
Term 0 (0):
    MyPID          GID          Value
    0              0              2
Term 1 (1):
    MyPID          GID          Value
    0              0              1
Term 2 (2):
    MyPID          GID          Value
    0              0              0
Term 3 (3):
    MyPID          GID          Value
    0              0              0

```

```

Term 4 (4):
    MyPID      GID      Value
    0          0          0
Term 5 (5):
    MyPID      GID      Value
    0          0          0
*****
-- Status Test Results --
***** OR Combination ->
***** F-Norm = 1.444e+00 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
***** Number of Iterations = 0 < 10
*****
-- Nonlinear Solver Step 0 --
||F|| = 5.003e+00 step = 0.000e+00 dx = 0.000e+00
*****

Creating a new preconditioner

Time required to create preconditioner : 0.000854 (sec.)

*****
***** Problem: Stokhos Matrix Free Operator
***** Preconditioned GMRES solution
***** Stokhos Mean-Based Preconditioner:
***** IFPACK ILU (fill=0, relax=0.000000, athr=0.000000)
***** No scaling
*****
iter:    0      residual = 1.000000e+00
iter:    1      residual = 1.655127e-16

Solution time: 0.000000 (sec.)
total iterations: 1
*****
-- Status Test Results --
***** OR Combination ->
***** F-Norm = 8.135e-01 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
***** Number of Iterations = 1 < 10
*****
-- Nonlinear Solver Step 1 --
||F|| = 2.818e+00 step = 1.000e+00 dx = 5.175e+00
*****



Destroying preconditioner

Creating a new preconditioner

Time required to create preconditioner : 0.000298 (sec.)

*****
***** Problem: Stokhos Matrix Free Operator
***** Preconditioned GMRES solution
***** Stokhos Mean-Based Preconditioner:
***** IFPACK ILU (fill=0, relax=0.000000, athr=0.000000)
***** No scaling
*****

```

```
iter:      0          residual = 1.000000e+00
iter:      5          residual = 4.581349e-05

Solution time: 0.000000 (sec.)
total iterations: 5
*****
-- Status Test Results --
***** OR Combination ->
***** F-Norm = 6.800e-02 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
***** Number of Iterations = 2 < 10
*****
-- Nonlinear Solver Step 2 --
||F|| = 2.356e-01 step = 1.000e+00 dx = 4.069e-01
*****
Destroying preconditioner
Creating a new preconditioner
Time required to create preconditioner : 8.8e-05 (sec.)

***** Problem: Stokhos Matrix Free Operator
***** Preconditioned GMRES solution
***** Stokhos Mean-Based Preconditioner:
***** IFPACK ILU (fill=0, relax=0.000000, athr=0.000000)
***** No scaling
*****
iter:      0          residual = 1.000000e+00
iter:      4          residual = 3.755168e-05

Solution time: 0.000000 (sec.)
total iterations: 4
*****
-- Status Test Results --
***** OR Combination ->
***** F-Norm = 7.242e-04 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
***** Number of Iterations = 3 < 10
*****
-- Nonlinear Solver Step 3 --
||F|| = 2.509e-03 step = 1.000e+00 dx = 3.977e-02
*****
Destroying preconditioner
Creating a new preconditioner
Time required to create preconditioner : 9e-05 (sec.)

***** Problem: Stokhos Matrix Free Operator
***** Preconditioned GMRES solution
***** Stokhos Mean-Based Preconditioner:
***** IFPACK ILU (fill=0, relax=0.000000, athr=0.000000)
***** No scaling
*****
```

```

        iter:      0          residual = 1.000000e+00
        iter:      4          residual = 3.615531e-05

        Solution time: 0.000000 (sec.)
        total iterations: 4
*****
-- Status Test Results --
***** OR Combination ->
***** F-Norm = 9.685e-08 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
***** Number of Iterations = 4 < 10
*****

*****
-- Nonlinear Solver Step 4 --
||F|| = 3.355e-07 step = 1.000e+00 dx = 4.265e-04
*****


Destroying preconditioner

Creating a new preconditioner

Time required to create preconditioner : 8.9e-05 (sec.)

*****
***** Problem: Stokhos Matrix Free Operator
***** Preconditioned GMRES solution
***** Stokhos Mean-Based Preconditioner:
***** IFPACK ILU (fill=0, relax=0.000000, athr=0.000000)
***** No scaling
*****


        iter:      0          residual = 1.000000e+00
        iter:      5          residual = 4.585404e-05

        Solution time: 0.000000 (sec.)
        total iterations: 5
*****


-- Nonlinear Solver Step 5 --
||F|| = 1.538e-11 step = 1.000e+00 dx = 6.082e-08 (Converged!)
*****


***** Final Status Test Results --
Converged....OR Combination ->
Converged....F-Norm = 4.441e-12 < 1.000e-10
      (Length-Scaled Two-Norm, Absolute Tolerance)
??.Number of Iterations = -1 < 10
*****


Final Solution =
Stokhos::VectorOrthogPoly of global size 6, local size 6 in basis
Complete polynomial basis (Legendre):
Term 0 (0):
    MyPID          GID          Value
      0              0          4.33333
      0              1              2
Term 1 (1):
    MyPID          GID          Value
      0              0              4
      0              1              1
Term 2 (2):
    MyPID          GID          Value

```

```

      0          0          0.666667
      0          1        1.68629e-12
Term 3 (3):
  MyPID      GID      Value
  0          0          0
  0          1        6.25093e-13
Term 4 (4):
  MyPID      GID      Value
  0          0          0
  0          1        -1.61449e-12
Term 5 (5):
  MyPID      GID      Value
  0          0          0
  0          1        1.13799e-12

```

Destroying preconditioner
Example Passed!

In this case, five Newton iterations are required to drive the nonlinear residual norm of the stochastic Galerkin system down below a tolerance of 10^{-10} . The linear system for each Newton iteration is solved by GMRES using the mean-based preconditioner provided by Stokhos, where the inverse of the mean is approximated by an incomplete LU factorization. The linear solver tolerance was set to 10^{-6} . By examining the computed solution, one can see the correct solution is obtained as derived above (up to the nonlinear solver tolerance).

6.5 Nonlinear Fluid Flow Problem with Albany

The previous examples demonstrate the utility of the template-based generic programming approach provided by Stokhos in making complicated, intrusive uncertainty propagation methods simpler to implement in simulation codes. The application must incorporate a small amount of code to set up the various Stokhos objects, provide Stokhos-specific code to evaluate stochastic Galerkin linear algebra objects such as residual vectors and Jacobian matrices, and then “merely” template the rest of the simulation code on the scalar type. While templating the code on the scalar type can involve substantial effort, this only needs to occur once. It has been observed that once the code has been templated, maintaining and extending the code (e.g., adding new physics or features) become simple for code developers that aren’t experts in uncertainty quantification methods. This approach has been used quite effectively in the Albany simulation code [26, 35], which implements PDE simulations of many types of complex physics. Furthermore, the code necessary to setup Stokhos is largely independent of the details of the simulation code and therefore can often be written once for a large class of simulation codes. This setup code has been incorporated into the Trilinos package Piro [34] making it possible to add Stokhos capabilities with only a few additional lines of code.

As a demonstration of applying these techniques in more realistic scientific computing scenarios, results from applying Stokhos to compute the solution to a fluid flow problem with uncertain input data are provided. A two-dimensional domain $D = [-5, 25] \times [-4, 4]$ is considered where fluid flows in from the left side

of the domain with a prescribed velocity and flows around a small cylinder within the domain and out the right side. No-slip boundary conditions are applied at the top and bottom walls of the domain as well as the cylinder. In the interior of the domain, the fluid is modeled by the incompressible Navier-Stokes equations:

$$\begin{aligned} -\nu \Delta u + u \cdot \nabla u + \nabla p &= 0, \\ \nabla \cdot \rho u &= 0, \end{aligned} \quad (52.28)$$

where u is the two-dimensional fluid velocity, p is pressure, ν is the kinematic viscosity, and ρ is the fluid density. In this case ν is considered uncertain and modeled by a truncated Karhunen-Loëve expansion

$$\nu(x, \xi) = \nu_0 + \sigma \sum_{i=0}^M \nu_i(x) \xi_i, \quad (52.29)$$

where each ξ_i is a uniform random variable over $[-1, 1]$, ν_0 is the mean field, σ is its standard deviation, the ν_i are eigenfunctions of the exponential covariance function

$$C(x, x') = \sigma^2 e^{-\frac{\|x-x'\|_1}{L}}, \quad x, x' \in D \quad (52.30)$$

and L is the correlation length. For this problem, $\rho = 1$, $\nu_0 = 1$, $\sigma = 0.1$, $L = 1$, $M = 2$, and the inlet velocity is fixed at 40.

The spatial domain D is discretized into roughly 13,300 quadrilateral cells and discretized via linear finite elements with Albany. Albany incorporates all of the embedded uncertainty quantification tools described above. It implements the `EpetraExt::ModelEvaluator` described above to formulate the nonlinear stochastic Galerkin problem and can use the Stokhos polynomial chaos scalar type for evaluating the finite element residual and Jacobian entries for each mesh cell. These coefficients are then assembled into Epetra residual and Jacobian objects in a manner similar to the one shown above. The Piro package handles the initialization of the relevant Stokhos objects from an input XML file as well as the NOX nonlinear solver. In this case, a total polynomial order of $N = 3$ was chosen for building the polynomial chaos approximation of u and p . The resulting nonlinear equations were solved by NOX using Newton's method, where the solution to each linear system was computed by GMRES. The Stokhos approximate Gauss-Seidel stochastic preconditioner was used to precondition each linear system using an incomplete LU factorization of the mean blocks. The linear solver tolerance was set to 10^{-3} and six Newton iterations were required to achieve relative solution updates smaller than 10^{-6} . The nonlinear stochastic Galerkin system was then solved in parallel using MPI on 32 processors. Once the stochastic Galerkin solution was computed, the mean and standard deviation of the fluid velocity were computed directly from its polynomial chaos expansion and are shown in Fig. 52.2.

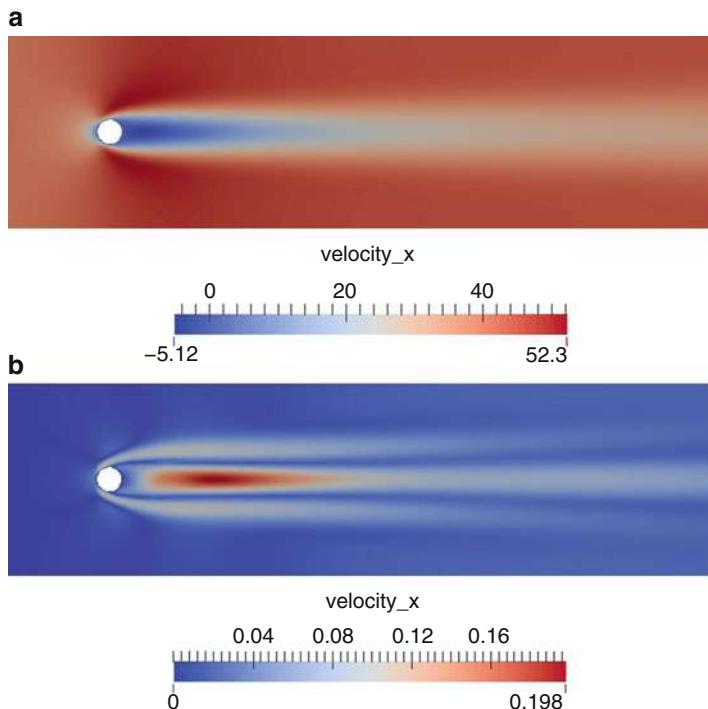


Fig. 52.2 Mean (a) and standard deviation (b) of the horizontal component of the fluid velocity for fluid flow around a cylinder with uncertain viscosity field

7 Conclusions

As described above, Stokhos provides a complete set of software tools for implementing embedded uncertainty quantification methods such as stochastic Galerkin and embedded sampling methods in general C++ simulation codes. It provides polynomial chaos and ensemble scalar types that allow uncertainty information to be propagated at the lowest levels of the simulation code using the operator overloading capabilities of the C++ language. These scalar types, in conjunction with judicious use of C++ templates, allow simulation codes to be easily modified to incorporate these embedded uncertainty propagation capabilities. Furthermore, these scalar types have been integrated with the Kokkos and Tpetra packages enabling their use in both shared and distributed memory parallel environments and enable application of fine-grained hardware parallelism across uncertainty dimensions. Finally, Stokhos provides capabilities for formulating the large-scale linear and nonlinear systems these embedded uncertainty quantification methods generate through a general nonlinear system interface that has been integrated with numerous nonlinear simulation and analysis packages within Trilinos. These packages, in conjunction with custom linear solver and preconditioning algorithms provided by Stokhos, allow large-scale embedded uncertainty quantification systems to be solved

in a scalable and efficient manner. All of these techniques have been incorporated into the Albany simulation code, enabling embedded uncertainty quantification of a diverse set of physical simulations.

Acknowledgements This work was supported by the Advanced Simulation and Computing (ASC) and Laboratory Directed Research and Development (LDRD) programs at Sandia National Laboratories, as well as based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR). Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. Adams, B.M., Dalbey, K.R., Eldred, M.S., Gay, D.M., Swiler, L.P., Bohnhoff, W.J., Eddy, J.P., Haskell, K., Hough, P.D.: DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, Uncertainty Quantification, and Sensitivity Analysis. Sandia National Laboratories, technical report sand2010-2183 edition, May 2010
2. Baker, C.G., Heroux, M.A.: Tpetra, and the use of generic programming in scientific computing. *Sci. Program.* **20**(2), 115–128 (2012)
3. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.* **12**(4), 273–288 (2000)
4. Bavier, E., Hoemmen, M., Rajamanickam, S., Thornquist, H.: Amesos2 and Belos: direct and iterative solvers for large sparse linear systems. *Sci. Program.* **20**(3), 241–255 (2012)
5. Conrad, P.R., Marzouk, Y.M.: Adaptive Smolyak pseudospectral approximations. *SIAM J. Sci. Comput.* **35**(6), A2643–A2670 (2013)
6. Constantine, P.G., Eldred, M.S., Phipps, E.T.: Sparse pseudospectral approximation method. *Comput. Methods Appl. Mech. Eng.* **229–232**(C), 1–12 (2012)
7. Debusschere, B.J., Najm, H.N., Pebay, P.P., Knio, O.M., Ghanem, R.G., Le Maître, O.P.: Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.* **26**(2), 698–719 (2004)
8. Edwards, H.C., Sunderland, D., Porter, V., Amsler, C., Mish, S.: Manycore performance-portability: Kokkos multidimensional array library. *Sci. Program.* **20**(2), 89–114 (2012)
9. Edwards, H.C., Trott, C.R., Sunderland, D.: Kokkos: enabling manycore performance portability through polymorphic memory access patterns. *J. Parallel Distrib. Comput.* **74**(12), 3202–3216 (2014)
10. Gaidamour, J., Hu, J., Siefert, C., Tuminaro, R.: Design considerations for a flexible multigrid preconditioning library. *Sci. Program.* **20**(3), 223–239 (2012)
11. Ghanem, R., Spanos, P.D.: Polynomial chaos in stochastic finite elements. *J. Appl. Mech.* **57**, 197–202 (1990)
12. Ghanem, R.G., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
13. Griewank, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Number 19 in *Frontiers in Applied Mathematics*. SIAM, Philadelphia (2000)
14. Heroux, M.A.: Epeta parallel linear algebra data structures. <http://trilinos.org/packages/epetra/> (2015)
15. Heroux, M.A.: EpetaExt extended epetra utilities. <http://trilinos.org/packages/epetraext/> (2015)
16. Heroux, M.A., Willenbring, J.M.: A new overview of the Trilinos project. *Sci. Program.* **20**(2), 83–88 (2012)
17. Heroux, M.A., Bartlett, R.A., Howle, V.E., Hoekstra, R.J., Hu, J.J., Kolda, T.G., Lehoucq, R.B., Long, K.R., Pawlowski, R.P., Phipps, E.T., Salinger, A.G., Thornquist, H.K., Tuminaro, R.S.,

- Willenbring, J.M., Williams, A.B., Stanley, K.S.: An overview of the Trilinos package. *ACM Trans. Math. Softw.* **31**(3) (2005). <http://trilinos.org/>
18. Hoemmen, M.F., Hu, J.J., Siefert, C.S.: Ifpack2: incomplete factorizations, relaxations, and domain decomposition library. <http://trilinos.org/packages/ifpack2> (2015)
19. Hoemmen, M.F., Thornquist, H.K., Heroux, M.A., Parks, M.: Tpetra: next-generation distributed linear algebra. <http://trilinos.org/packages/tpetra> (2015)
20. Hu, J., Prokopenko, A., Siefert, C., Tuminaro, R.: MueLu multigrid framework. <http://trilinos.org/packages/muelu> (2015)
21. Le Maître, O.P., Knio, O.M.: Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics. *Scientific Computation*. Springer, New York (2010)
22. Novak, E., Ritter, K.: High dimensional integration of smooth functions over cubes. *Numerische Mathematik* **75**, 79–97 (1996)
23. Øksendal, B.: Stochastic Differential Equations. Springer, Berlin (1998)
24. Pawlowski, R.P., Kolda, T.G.: NOX object-oriented nonlinear solver package. <http://trilinos.org/packages/nox> (2015)
25. Pawlowski, R.P., Phipps, E.T., Salinger, A.G.: Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: template-based generic programming. *Sci. Program.* **20**, 197–219 (2012)
26. Pawlowski, R.P., Phipps, E.T., Salinger, A.G., Owen, S.J., Siefert, C.M., Staten, M.L.: Automating embedded analysis capabilities and managing software complexity in multiphysics simulation Part II: application to partial differential equations. *Sci. Program.* **20**, 327–345 (2012)
27. Phipps, E.T.: Stokhos embedded uncertainty quantification methods. <http://trilinos.org/packages/stokhos> (2015)
28. Phipps, E.T., Gay, D.M.: Sacado automatic differentiation package. <http://trilinos.sandia.gov/packages/sacado> (2015)
29. Phipps, E., Pawlowski, R.: Efficient expression templates for operator overloading-based automatic differentiation. In: Forth, S., Hovland, P., Phipps, E., Utke, J., Walther, A. (eds.) Recent Advances in Algorithmic Differentiation. Volume 87 of Lecture Notes in Computational Science and Engineering, pp. 309–319. Springer, Berlin (2012)
30. Phipps, E., Edwards, H.C., Hu, J., Ostien, J.T.: Exploring emerging manycore architectures for uncertainty quantification through embedded stochastic Galerkin methods. *Int. J. Comput. Math.* **91**(4), 707–729 (2014)
31. Phipps, E.T., Edwards, H.C., Hu, J.: Exploring heterogeneous multicore architectures for advanced embedded uncertainty quantification. Technical report SAND2014-17875, Sandia National Laboratories, Sept 2014
32. Powell, C.E., Elman, H.C.: Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA J. Numer. Anal.* **29**(2), 350–375 (2009)
33. Rosseel, E., Vandewalle, S.: Iterative solvers for the stochastic finite element method. *SIAM J. Sci. Comput.* **32**(1), 372–397 (2010)
34. Salinger, A.G.: Piro embedded nonlinear analysis capabilities package. <http://trilinos.org/packages/piro> (2015)
35. Salinger, A., et al.: Albany multiphysics simulation code. <https://github.com/gahansen/Albany> (2015)
36. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR* **4**, 240–243 (1963)
37. Sousedík, B., Ghanem, R.G., Phipps, E.T.: Hierarchical schur complement preconditioner for the stochastic galerkin finite element methods. *Numer. Linear Algebra Appl.* **21**(1), 136–151 (2014)
38. Ullmann, E.: A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. *SIAM J. Sci. Comput.* **32**(2), 923–946 (2010)
39. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**, 897–936 (1938)
40. Xiu, D.B., Karniadakis, G.E.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)

Bert Debusschere, Khachik Sargsyan, Cosmin Safta, and Kenny Chowdhary

Abstract

The UQ Toolkit (UQTk) is a collection of tools for uncertainty quantification, ranging from intrusive and nonintrusive forward propagation of uncertainty to inverse problems and sensitivity analysis. This chapter first outlines the UQTk design philosophy, followed by an overview of the available methods and the way they are implemented in UQTk. The second part of this chapter is a detailed example that illustrates a UQ workflow from surrogate construction, and calibration, to forward propagation and attribution.

Keywords

UQTk • UQ toolkit • Polynomial chaos • Quadrature • Bayesian inference • Markov chain Monte Carlo • Global sensitivity analysis • Surrogate modeling • Uncertainty propagation

Contents

1	Introduction	1808
2	Methods	1808
3	Implementation and Installation	1810
4	Example Workflow	1811
4.1	Polynomial Chaos Surrogate Construction	1813
4.2	Global Sensitivity Analysis	1817

B. Debusschere (✉)

Mechanical Engineering, Sandia National Laboratories, Livermore, CA, USA

Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA

e-mail: bjdebus@sandia.gov

K. Sargsyan

Reacting Flow Research Department, Sandia National Laboratories, Livermore, CA, USA

e-mail: ksargsy@sandia.gov

C. Safta • K. Chowdhary

Quantitative Modeling and Analysis, Sandia National Laboratories, Livermore, CA, USA

e-mail: csafta@sandia.gov; kchowdh@sandia.gov

4.3	Model Parameter Inference	1818
4.4	Polynomial Chaos Construction via Rosenblatt Transformation	1820
4.5	Uncertainty Propagation and Attribution	1822
4.6	Command-Line Reproduction	1824
5	Conclusion	1826
	Cross-References	1826
	References	1826

1 Introduction

As the name implies, the UQ Toolkit (UQTk) is a collection of tools for uncertainty quantification (UQ) in computational models. Both the terms *tools* and *UQ* are to be interpreted quite broadly here. The term *tools* refers to C++ classes and libraries, stand-alone applications, or *apps*, as well as Python scripts. UQ refers to all operations involved in assessing and improving the confidence in a numerical model prediction such as parameter inference, sensitivity analysis, surrogate construction, or forward propagation of uncertainty.

In terms of design philosophy, UQTk is meant to be a set of UQ tools that is straightforward to download, install, and use. The target users are people who are learning about UQ methods, or who are developing new UQ algorithms, or who have a need for UQ tools that are tailored to their applications. Rather than implementing a comprehensive set of functionalities in one single software library, UQTk offers a set of tools, each implementing specific operations that can be strung together to build a complete UQ workflow. The current UQTk version is 3.0, which is available on the web at <http://www.sandia.gov/UQToolkit> [7] under the Lesser General Public License (LGPL).

The next section gives an overview of the various UQ algorithms and methods that are implemented in UQTk, followed by a section that covers some of the implementation and installation details. An example of comprehensive UQ workflow with UQTk is detailed at the end.

2 Methods

The UQTk library implements several methodologies using a modular approach facilitated by the C++ programming language. The current software release, UQTk v3.0, provides capabilities for numerical quadrature, polynomial chaos representations, Markov chain Monte Carlo, Bayesian compressive sensing, and random field representations via Karhunen-Loeve expansions. These capabilities are implemented via C++ classes or functions and are briefly described below.

The numerical quadrature C++ class, *Quad*, constructs multidimensional full tensor and sparse quadrature rules based on several 1D quadrature rules. The weights for 1D Newton-Cotes (NC) rules [13] are computed via the solution of Vandermonde systems, while the weights and grid points for Clenshaw-Curtis (CC) [4] rules are computed analytically. Both closed and open NC and CC rules are provided by the library. In addition to NC and CC rules, 1D quadrature rules are also

Table 53.1 Polynomial bases implemented in UQTk and the corresponding random variables

Polynomial basis	Random variable
Legendre	Uniform
Hermite	Gaussian
Jacobi	Beta
Laguerre	Gamma

computed based on a series of orthogonal polynomials in the Askey scheme [1, 25]: e.g., Legendre, Hermite, Jacobi, and Laguerre. The quadrature points and weights for this set of rules are computed based on the Golub-Welsch algorithm [11]. The Quad class can also create custom quadrature rules for orthogonal polynomials given via two-term recurrence coefficients provided by the user. Multidimensional rules are created using either full tensor products, generally feasible for low dimensionality. Smolyak constructions [9, 23] are employed for sparse quadrature rules.

Methods pertaining to polynomial chaos constructions [10] are implemented via two C++ classes, *PCSet* and *PCBasis*. The latter provides implementations for polynomial bases orthogonal with respect to various continuous random variables. These are shown in Table 53.1.

This class is connected with the *Quad* class for the purpose of retrieving quadrature points and weights corresponding to a specific polynomial basis. The *PCBasis* class provides these functionalities to the *PCSet* class which implements operations for both intrusive (ISP) and nonintrusive (NISP) spectral projection [6]. Briefly, the latter class provides several methods for unitary operations, such as logarithm, and binary operations, e.g., addition or division, with PC expansions (PCE). Arithmetic operations involving a larger number of PCEs can be cast in a sequence of unitary and binary operations. In order to reduce the aliasing errors, *PCSet* also provides a method for products of three PCEs. Methods for Galerkin projection include both quadrature and Monte Carlo integration approaches. The *PCSet* class provides several other functionalities like computing main and joint effect sensitivity indices or evaluating PCEs at custom germ values.

For inverse problems, Bayesian inference methods are provided in the *MCMC* class. Several Markov chain Monte Carlo (MCMC) algorithms are implemented in this class, such as Gibbs sampling [8], the adaptive MCMC methodology [12], and the Metropolis-adjusted Langevin algorithm (MALA) [15] to sample probability densities that are otherwise difficult to estimate analytically. The *MCMC* class provides several methods to set algorithm parameters, including parameter bounds and initial proposal densities. Similarly, several methods are implemented for the retrieval of information from the sample chain. If necessary, this class connects to an optimization library to compute and start the chain samples at the maximum a posteriori estimate.

The Bayesian compressive sensing (BCS) approach of Babacan [2] is implemented via a C++ function named *bcs*. This function uses Laplace priors and a greedy algorithm to detect sparsity in high-dimensional settings. Python scripts, available via PyUQTk, wrap around this function to provide an iterative BCS formulation [19].

The *kle* class implements functionalities for representing random fields via Karhunen-Loëve (KL) expansions [10]. This class constructs orthogonal basis functions via the solution of an eigensystem constructed from the random field covariance matrix provided by the user. Further, the *kle* class allows the user to specify custom grid weights, thus enabling KL expansions for random fields of arbitrary meshes and dimensionality. Currently, this class uses the LAPACK library to compute eigenvalues/eigenvectors for the KL expansion. The class also provides methods for projecting the random field samples on the eigenmodes and retrieving samples of random variables corresponding to each mode.

3 Implementation and Installation

UQTk is primarily a collection of C++ classes and functions for quadrature, polynomial approximation, Markov chain Monte Carlo sampling, Bayesian compressive sensing, and Karhunen-Loeve construction. UQTk relies on a few additional FORTRAN and C libraries for linear algebra (LAPACK and BLAS), quadrature, optimization (LBFGS), ordinary different equation solvers (cvode), and other general purpose mathematical and statistics routines (SLATEC). These third-party libraries are packaged as part of UQTk and do not need to be downloaded separately. UQTk v3.0 also comes with the option of installing a Python interface to the C++ classes and functions using SWIG (Simplified Wrapper Interface Generator).

The software can be downloaded from <http://www.sandia.gov/UQToolkit> [7] as a tar file. UQTk uses CMake for cross-platform building and installation. In order to compile the default C++ libraries, the user will need C, C++, and FORTRAN compilers. To install the Python interface to UQTk, PyUQTk, the user will also need Python and SWIG. For more details about compiler versions and architectures, e.g., Mac OSX and Linux compatibility, etc., the reader is referred to the INSTALL file located in the UQTk source code directory or to the UQTk manual, which is available online at <http://www.sandia.gov/UQToolkit>.

After UQTk is built and installed, the user can run a series of tests using the CTest command to make sure the libraries were successfully compiled. Assuming the installation was successful, the installation directory should contain *include* and *lib* directories, which contain all the C++ header files and libraries, respectively. The user can use these header files and libraries to integrate UQTk into their existing C++ codes by linking their existing C++ code to these libraries. UQTk is designed to be modular in that the user has the option of linking to a desired subset of the libraries. For example, if the user wants to utilize the quadrature library, then he or she only needs the *libuqtarray.a* and *libuqtquad.a* libraries along with their respective header files. For simplicity, the *libuqt.a* library contains all UQTk classes and routines and *libuqtdep.a* contains all dependent libraries, e.g., LAPACK, BLAS, SLATEC, etc., except for cvode, which is included in the libraries *libdep_cvode.a* and *libdep_nvvec.a*. Examples of how to integrate UQTk into existing C++ code can be found under the *cpp/tests* directory. It is important to note that most of these libraries rely on a UQTk-specific array class defined in the *ArrayID.h* and

Array2D.h header files. These array classes are simply a wrapper for the standard C++ vector class.

So far, this text covered the integration of UQTk with existing C++ code. However, UQTk also provides a collection of separate executables or applications to be used outside the scope of the user code. These executables can be found in the *bin* folder in the installation directory. For example, the executable *model_inf* can be used to infer model parameters from user-provided data. More examples of how to use these applications can be found in the *examples* directory or in the *Workflows* section of this chapter.

The most recent addition to UQTk is PyUQTk, which is a collection of pure Python and SWIG interface files to the UQTk C++ classes routines. The pure Python routines mostly consist of functions for plotting and post-processing of data. In order to also compile and create the SWIG interface, the user needs to set the *PyUQTk* flag to ON in the CMake setup stage. After successful compilation, this will generate Python libraries that wrap UQTK's C++ classes and methods and allow easy translation from Numpy style arrays to the arrays used in UQTk. To see examples of how to use the SWIG interface, see the test directory under the *PyUQTk* folder.

4 Example Workflow

In this section, an example workflow is performed demonstrating many of the forward and inverse UQ capabilities of UQTk. Consider a model $g(\lambda)$ that is used to compute some quantity of interest. In this model, λ is a set of parameters that have some uncertainty associated with them. *For example*, $g(\lambda)$ may be a model to compute the heat losses of a house, while λ is the set of conductive and convective heat transfer coefficients that are needed to compute the heat fluxes through the walls and windows of the house. The main goal of the UQ workflow described here is to determine the uncertainty in the predictions made with $g(\lambda)$ due to the uncertainty in the model inputs λ . In essence, this is a typical forward UQ problem. However, the uncertainty in λ first needs to be characterized before it can be propagated through $g(\lambda)$. This is done through calibration, i.e., an inverse problem. Often, the calibration relies on a different model $f(\lambda)$ instead of $g(\lambda)$. In the example of the computation of the heat losses of a house, $f(\lambda)$ may represent an experiment for the measurement of heat transfer coefficients by applying known heat fluxes to well-controlled samples of the materials that make up the walls and windows of the house.

A schematic of the resulting overall workflow is depicted in more detail in Fig. 53.1. In the calibration step, the parameters λ are determined given data $\mathcal{D} = \{y_i\}$ and the model $f(\lambda)$ associated with the data. Such parameter estimation requires *training* evaluations of the model $f(\lambda)$ at selected values of λ . This is typically computationally intensive when $d \gg 1$, i.e., the input parameter vector λ is high dimensional, as well as when $f(\cdot)$ itself is a complex model that takes significant computational resources to run for a fixed set of parameters λ . In such

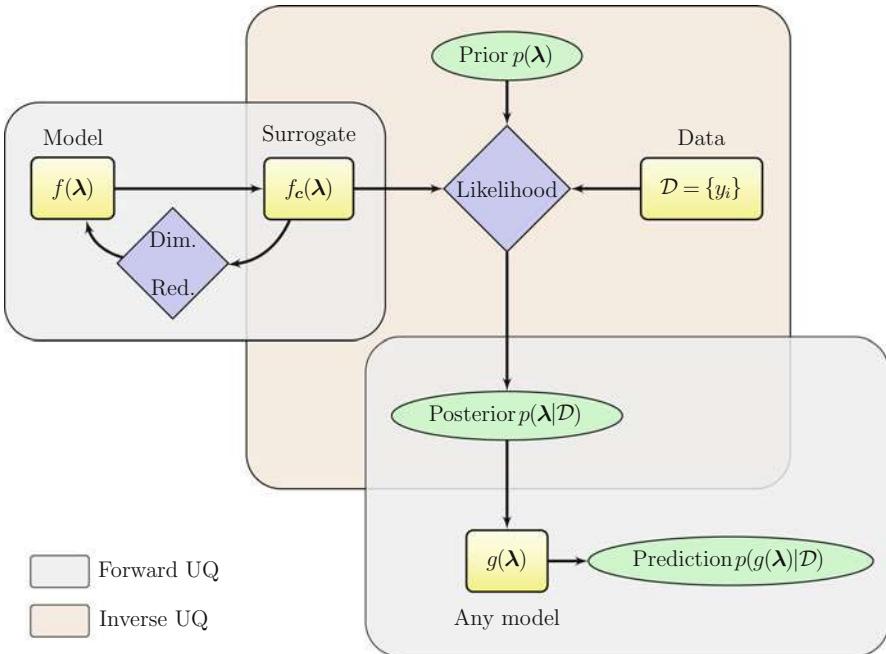


Fig. 53.1 Forward and inverse UQ schematic. Inverse UQ corresponds to parameter estimation, also called calibration. The first forward UQ task consists of sensitivity analysis, dimensionality reduction, and surrogate construction, while the second forward UQ task is for forward prediction with already calibrated parameters

situations, it is common to precompute a computationally cheap surrogate for the model $f(\lambda)$ to replace it in computationally intensive studies. While UQTk offers a few different types of surrogates (e.g., radial basis function expansions or Gaussian process regression), here the use of polynomial chaos (PC) surrogates is highlighted. Besides being cheap to evaluate, a PC surrogate offers readily available sensitivity indices that can help identify the parameters that actually impact the model output, which can in turn be used to derive a reduced-dimensionality surrogate. Such surrogate construction and subsequent sensitivity analysis can be viewed as a *forward UQ* task since it is equivalent to propagating uniformly distributed, *i.i.d* input parameters λ through the model $f(\lambda)$. If there is a clear indication that the problem dimensionality can be reduced, one should repeat this forward UQ step to construct a lower-dimensional surrogate that approximates the model over the range of variability of the most important parameters only, while keeping the non-important ones at their nominal values. Such a lower-dimensional surrogate will in principle be more accurate as more training points per dimension can be afforded in the low-dimensional space, and it will render the subsequent calibration to be much more well-conditioned.

The outcome of the calibration or *inverse UQ* step is the characterization of the uncertainties in the relevant parameters λ in the form of the posterior distribution for these parameters. This posterior distribution can then in turn be pushed through any predictive model of interest $g(\lambda)$. As mentioned before, this is also a forward UQ step, but with a much more informed input parameter distribution for λ compared to the uniform *i.i.d.* one that was used for the surrogate construction. Note that one could also push the parameter posterior through the calibrated model $f(\lambda)$ itself in order to validate or check the quality of calibration. However, in general, any model of interest can be employed for forward UQ even if there is no observed data associated with it.

In the sections below, the implementation with UQTK for the elements in the workflow of Fig. 53.1 is illustrated for arbitrarily chosen functions $f(\lambda)$ and $g(\lambda)$.

4.1 Polynomial Chaos Surrogate Construction

Consider a function $f(\lambda)$, where $\lambda = (\lambda_1, \dots, \lambda_d)$ is a d -dimensional input parameter vector defined on a hypercube $\Omega = \prod_{i=1}^d [a_i, b_i]$. Computationally intensive studies that require many model evaluations often necessitate the usage of *surrogates*, i.e., fast-to-evaluate approximations $f_c(\lambda) \approx f(\lambda)$ over the domain of interest Ω , parameterized by a vector c . This workflow will demonstrate surrogates based on polynomial chaos machinery. To this end, it is worth observing that surrogate construction is a special case of PC-based forward propagation of uncertainty. One first casts the input vector λ as uniformly distributed, *i.i.d.* random variables or a linear PC expansion with respect to standard uniform, *i.i.d.* random vector ξ with component-wise linear scaling:

$$\lambda_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} \xi_i, \quad \text{for } i = 1, \dots, d. \quad (53.1)$$

The forward uncertainty propagation task then becomes constructing a PC for the output:

$$f(\lambda(\xi)) \approx \sum_{\alpha \in \mathcal{S}} c_\alpha \Psi_\alpha(\xi), \quad (53.2)$$

where $\lambda(\xi)$ is the relationship encoded in the linear PC (53.1). From the surrogate construction viewpoint, one seeks an approximation $f(\lambda) \approx f_c(\lambda) = \sum_{\alpha \in \mathcal{S}} c_\alpha \Psi_\alpha(\xi(\lambda))$ over the range of the inputs λ . The basis $\Psi_\alpha(\xi) = \psi_{\alpha_1}(\xi_1) \cdots \psi_{\alpha_d}(\xi_d)$ is a multivariate Legendre polynomial, corresponding to a *multiindex* $\alpha = (\alpha_1, \dots, \alpha_d)$, and each individual α_i denotes the order of the univariate Legendre polynomial $\psi_{\alpha_i}(\xi_i)$. The basis/multiindex set \mathcal{S} can be truncated according to predefined rules. Here the most commonly used total-order truncation is employed, i.e., $\mathcal{S} = \left\{ \alpha : \sum_{i=1}^d \alpha_i \leq p \right\}$ for some *order* p . With such truncation, the cardinality of the basis is $|\mathcal{S}| = K = (p+d)!/(p!d!)$.

The forward UQ task or, in this case, surrogate construction now boils down to computing the PC coefficient vector $\mathbf{c} = c_\alpha$ given a set of *training* simulations of the model at parameter settings $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}(\boldsymbol{\xi}^{(i)})$ corresponding to values $(\boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(N)})$ of the underlying standard random variable $\boldsymbol{\xi}$, i.e., a set of input-output pairs $\left\{(\boldsymbol{\lambda}^{(i)}, f^{(i)})\right\}_{i=1}^N$, where $f^{(i)} = f(\boldsymbol{\lambda}^{(i)})$.

Typically, the number of PC coefficients is smaller than the number of data points, i.e., $K < N$, and therefore it is generally impossible to exactly interpolate $f_c(\boldsymbol{\lambda}^{(i)}) = f^{(i)}$. Regression-based approaches search for a minimum of some distance measure between the vectors of model simulation $\mathbf{f} = (f^{(1)}, \dots, f^{(N)})$ and surrogate evaluation at the training points $\mathbf{f}_c = (f_c(\boldsymbol{\lambda}^{(1)}), \dots, f_c(\boldsymbol{\lambda}^{(N)}))$, i.e., $\mathbf{c} = \arg \min \rho(\mathbf{f}, \mathbf{f}_c)$. For example, the most commonly used least-squares approach corresponds to $\rho(\mathbf{f}, \mathbf{f}_c) = \|\mathbf{f} - \mathbf{f}_c\|_2$.

For the purposes of this workflow, another approach, *orthogonal projection*, is employed, which makes use of the orthogonality of the basis polynomials $\Psi_\alpha(\boldsymbol{\xi})$, and minimizes the function-norm $R(f, f_c) = \int_{\Omega} [f(\boldsymbol{\lambda}) - f_c(\boldsymbol{\lambda})]^2 \pi_\lambda(\boldsymbol{\lambda}) d\boldsymbol{\lambda}$ with respect to the probability density function (PDF) of $\boldsymbol{\lambda}$, denoted by $\pi_\lambda(\boldsymbol{\lambda})$. Namely, the coefficient c_α is found by:

$$c_\alpha = \frac{1}{\|\Psi_\alpha(\boldsymbol{\xi})\|^2} \langle f(\boldsymbol{\lambda}(\boldsymbol{\xi})) \Psi_\alpha(\boldsymbol{\xi}) \rangle, \quad (53.3)$$

where $\|\Psi_\alpha(\boldsymbol{\xi})\| = \sqrt{\langle \Psi_\alpha(\boldsymbol{\xi})^2 \rangle}$ are precomputed norms, and the expectation is defined as $\langle f(\boldsymbol{\lambda}(\boldsymbol{\xi})) \Psi_\alpha(\boldsymbol{\xi}) \rangle = \frac{1}{2^d} \int_{[-1,1]^d} f(\boldsymbol{\lambda}(\boldsymbol{\xi})) \Psi_\alpha(\boldsymbol{\xi}) d\boldsymbol{\xi}$. For smooth functions $f(\boldsymbol{\lambda})$ that are amenable to reasonably accurate polynomial representation (53.2), Gaussian quadrature integration is the most appropriate approach. Namely:

$$\frac{1}{2^d} \int_{[-1,1]^d} f(\boldsymbol{\lambda}(\boldsymbol{\xi})) \Psi_\alpha(\boldsymbol{\xi}) d\boldsymbol{\xi} \approx \sum_{i=1}^N f\left(\boldsymbol{\lambda}\left(\boldsymbol{\xi}^{(i)}\right)\right) \Psi_\alpha\left(\boldsymbol{\xi}^{(i)}\right) w_i, \quad (53.4)$$

requiring the training points $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}(\boldsymbol{\xi}^{(i)})$ to correspond to Gaussian quadrature points with associated weights w_i that integrate polynomials up to a certain order exactly. This is one drawback of the projection approach compared to regression techniques – the training set is predefined at Gaussian quadrature locations.

UQTk primarily focuses on Galerkin projection approaches for forward UQ, with most of the methods offered involving nonintrusive quadrature-based approaches using either full or sparse tensor products. However, for cases where the samples are very sparse in a high-dimensional space, UQTk also offers a Bayesian compressed sensing method.

In this specific workflow, a Gauss quadrature approach is used. The UQTk app `generate_quad` generates quadrature points $\boldsymbol{\xi}^{(i)}$, while another app, called `pce_eval`, will map the points to the “physical” model inputs $\boldsymbol{\lambda}$ according to the

input PC coefficients defined in a given file. Note that for surrogate construction, the input PC expansion (53.1) is linear, and, writing the input PC in the full form:

$$\lambda_i = l_{0i} + \sum_{k=1}^d l_{ki} \xi_i, \quad (53.5)$$

the input PC matrix $L_{(d+1) \times d}$ to be given to the app `pce_eval` is “nearly” diagonal. That is, the first row of L is defined as $l_{0i} = \frac{a_i+b_i}{2}$, while the rest of the matrix is diagonal, $l_{ki} = \delta_{ki} \frac{b_i-a_i}{2}$ using the Kronecker- δ . The matrix L in a text format should be stored in a file that is given as a command-line argument to `pce_eval` (Note that the linear map (53.1) can be easily accomplished by other means; here `pce_eval` is selected simply to illustrate general PC evaluation capabilities.).

The accuracy of the surrogate is estimated by comparing the model and PC surrogate outcomes at a set of M randomly chosen input samples, called *validation* samples. This can be accomplished prior to projection, by random sampling of the input PC *germ* ξ using the UQTk application `pce_rv`, followed by input PC evaluation using another app, called `pce_eval`. This concludes steps F1-F4 listed in Table 53.2 at the end of this section. With both training and validation samples in place, one can proceed to model evaluation (step F5) and to the projection itself (step F6), which is implemented via the UQTk app `pce_resp`, essentially carrying out the integration according to (53.4). Note that the model should be evaluated *offline* – the fully nonintrusive approach is taken – as a black-box, generally, independent of UQTk.

Figure 53.2 demonstrates the surrogate construction results for two test functions:

$$\begin{aligned} \text{Gaussian: } f^G(\boldsymbol{\lambda}) &= \exp\left(-\sum_{i=1}^d a_i^2 \lambda_i^2\right) \\ \text{Exponential: } f^E(\boldsymbol{\lambda}) &= \exp\left(\sum_{i=1}^d a_i \lambda_i\right). \end{aligned} \quad (53.6)$$

The dimensionality is set to $d = 5$, and the dimensional coefficients are set to $\boldsymbol{a} = (0.4, 0.3, 0.2, 0.1, 0.05)$. The surrogate order was set to $p = 3$, while the quadrature rule was chosen to be a product grid with 4 points per dimension, i.e., a total of $N = 4^5 = 1024$ training points. Finally, $M = 111$ random validation samples are selected in order to assess the accuracy of the surrogates. In these tests, the relative error was below 0.1%, suggesting the surrogates are very accurate for the current purposes. Creating good-quality high-dimensional surrogates is generally challenging, however. One reason is the high number of samples needed to inform the surrogate construction. Also, it is not uncommon that some of the input parameters do not significantly impact the outputs of interest, which makes the subsequent parameter estimation or inverse UQ studies challenging. Therefore, surrogates should generally be investigated for potential dimensionality reduction. To illustrate this process, a global sensitivity analysis is applied to the surrogate derived here.

Table 53.2 Forward UQ steps, Forward (DIM, PCTYPE, INPC). Note that the arguments DIM, PCTYPE, INPC are kept general to capture various forward UQ tasks, such as high- and low-dimensional surrogate construction, as well as uncertainty propagation given a general input PC expansion. Besides the command-line inputs, the relevant “hardwired” input/output file names are highlighted, with the necessary file saving/copying commands to help navigate between the command-line apps

<p>(F1) Quadrature generation</p> <ul style="list-style-type: none"> - generate_quad -d <DIM> -g <PCTYPE> -x full -p 4 - Output: quadrature points qdpts.dat quadrature weights wghts.dat <p>(F2) Generate model training inputs</p> <ul style="list-style-type: none"> - cp qdpts.dat xdata.dat - Input: quadrature points $\xi \in [-1, 1]^d$ xdata.dat - pce_eval -x PC -s <PCTYPE> -o 1 -f <INPC> - Output: function inputs $\lambda \in [\prod_{i=1}^d [a_i, b_i]]^d$ ydata.dat - cp ydata.dat ptrain.dat <p>(F3) Random sample generation</p> <ul style="list-style-type: none"> - pce_rv -w PCvar -d <DIM> -n 111 -p <DIM> -x <PCTYPE> - Output: random samples rvar.dat <p>(F4) Generate model inputs for validation</p> <ul style="list-style-type: none"> - cp rvar.dat xdata.dat - Input: random samples $\xi \in [-1, 1]^d$ xdata.dat - pce_eval -x PC -s <PCTYPE> -o 1 -f <INPC> - Output: function inputs for validation $\lambda \in [a_i, b_i]^d$ ydata.dat - cp ydata.dat pval.dat <p>(F5) Model evaluation (typically performed outside UQTk, as a black-box)</p> <ul style="list-style-type: none"> - Inputs: ptrain.dat and pval.dat - Outputs: ytrain.dat and yval.dat, correspondingly. <p>(F6) Projection</p> <ul style="list-style-type: none"> - cp ytrain.dat ydata.dat - Input: qdpts.dat wghts.dat ydata.dat - pce_resp -d <DIM> -x <PCTYPE> -o 3 -e - Output: multiindex mindex.dat PC coefficients PCcoeff_quad.dat <p>(F7) Sensitivity analysis/Uncertainty attribution</p> <ul style="list-style-type: none"> - pce_sens -m mindex.dat -f PCcoeff_quad.dat -x <PCTYPE> - Output: Main, total and joint sensitivities mainsens.dat, totsens.dat, jointsens.dat, respectively. <p>(F8) Generate output PC samples</p> <ul style="list-style-type: none"> - pce_rv -w PCmi -n 1000000 -p <DIM> -f PCcoeff_quad.dat \ -m mindex.dat -x <PCTYPE> - Output: PC samples rvar.dat <p>(F9) PDF computation</p> <ul style="list-style-type: none"> - pdf_cl -i rvar.dat -g 100 - Output: PDF evaluations dens.dat 	<p>Forward UQ</p>
---	-------------------

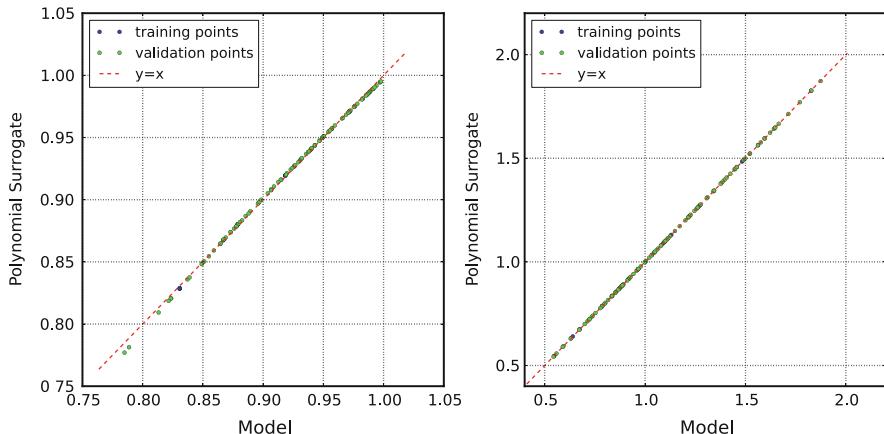


Fig. 53.2 Comparison between the model and its surrogate for two test functions, (left) Gaussian and (right) exponential

4.2 Global Sensitivity Analysis

Global sensitivity analysis (GSA) relies on variance-based decomposition and computes Sobol sensitivity indices [3, 5, 24]. For generic functions, one can employ a sampling-based approach [14, 17]. However, accurate estimation of these indices often requires prohibitively many model evaluations; hence, surrogate construction can provide a computationally feasible route. Here the PC surrogate approach renders itself particularly useful as one can compute sensitivity indices *exactly* from the polynomial form (53.2) without having to sample the surrogate model itself. To this end, the UQTK app `pce_sens` is employed to demonstrate the main sensitivity index computation for the test models outlined in the previous subsection (see step F7 in Table 53.2). The formula for the main sensitivity index for the i -th dimension reads:

$$S_i = \frac{\sum_{\alpha \in S_i} \|\Psi_\alpha(\xi)\|^2}{\sum_{\alpha \in S} \|\Psi_\alpha(\xi)\|^2}, \quad (53.7)$$

where $S_i = \left\{ \left(0, 0, \dots, \underbrace{0}_{i-1}, \underbrace{k}_{i}, \underbrace{0}_{i+1}, \dots, 0, 0 \right) \right\}_{k=1}^p$ is the set of multi-indices that contain only dimension i , while the denominator is nothing more than the variance of the surrogate model assuming uniform inputs. The formula (53.7) corresponds to the fraction of the total variance that can be attributed to the i -th parameter only.

Figure 53.3 demonstrates the main sensitivity indices for the two models considered in this workflow. Clearly, as the dimensional importance coefficients

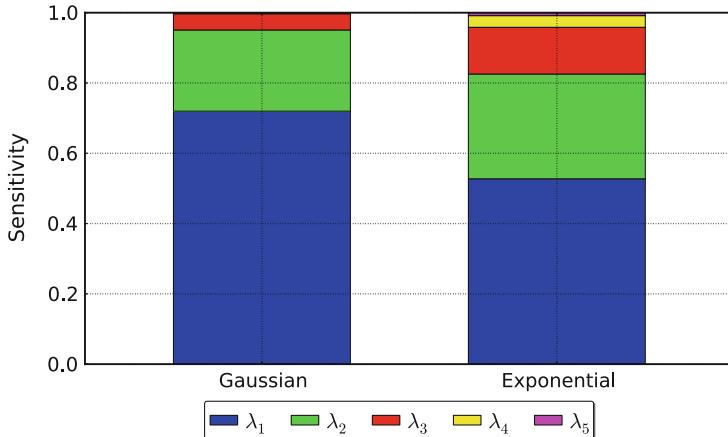


Fig. 53.3 Main sensitivity indices, i.e., variance fractions, for two test functions

of the test functions (53.6) are selected as $\boldsymbol{a} = (0.4, 0.3, 0.2, 0.1, 0.05)$, most of the variance can be attributed to the leading dimensions. This type of analysis allows for dimensionality reduction. For example, more than 80% of the variance can be attributed to the first two dimensions; therefore, one may want to focus on these two dimensions for more accurate surrogate construction as well as for the subsequent parameter inference to be well defined. This will be demonstrated within the workflow as follows.

As the initial, higher-dimensional study suggests, one can reduce the dimensionality of the problem from $d = 5$ to $d = 2$ without considerable accuracy reduction. Therefore, the rest of the inputs will be fixed at their nominal values $\lambda_3 = \lambda_4 = \lambda_5 = 0$, and the models are reformulated as *two*-dimensional, i.e., $d = 2$ and $\boldsymbol{a} = (0.4, 0.3)$. The surrogate construction should ideally be repeated for $d = 2$ or, in the absence of extra computational resources for running the model, one can analytically reduce the already constructed *five*-dimensional surrogate and cast it as a function of 2 parameters only. Now, assuming the *two*-dimensional surrogate is in place, the workflow proceeds to the inverse UQ task, i.e., parameter estimation given observational data.

4.3 Model Parameter Inference

Consider observational data that corresponds to the output of the test models Gaussian $f^G(\boldsymbol{\lambda})$ and Exponential $f^E(\boldsymbol{\lambda})$. Namely, let us assume y_j^G and y_j^E , for $j = 1, \dots, R$ are R experimental observations of the “physical” quantities that are outputs of the models $f^G(\boldsymbol{\lambda})$ and $f^E(\boldsymbol{\lambda})$, correspondingly. Assuming there is Gaussian noise in the data, one can write a *data noise model*:

$$y_j^{G(E)} = f^{G(E)}(\boldsymbol{\lambda}) + \sigma \underbrace{\epsilon_j^{G(E)}}_{N(0,1)}, \quad (53.8)$$

Table 53.3 Inverse UQ steps, `Inverse()`

<p>(I1) Generate data (typically performed outside UQTk)</p> <ul style="list-style-type: none"> - Output: <code>xfile.dat</code> <code>yfile.dat</code> <p>(I2) Parameter inference</p> <ul style="list-style-type: none"> - Input: <code>pdomain.dat</code> <code>mindex.dat</code> <code>pccf.dat</code> - <code>model_inf -x xfile.dat -y yfile.dat -f pc -l classical \ -m100000 -e 0.01 -i normal</code> - Output: <code>means.dat</code> <code>vars.dat</code> <code>chain.dat</code> <p>(I3) Build PC expansion for calibrated input</p> <ul style="list-style-type: none"> - <code>pce_quad -o 1 -f chn.dat -x HG</code> - Output: <code>PCcoeff.dat</code> - <code>cp PCcoeff.dat pce_calib.dat</code> 	Inverse UQ
--	-------------------

Similar to model evaluation, this synthetic data generation is in principle carried out independently from UQTk (Step I1 in Table 53.3). Given observational data, one then employs Bayesian techniques [22] to infer the posterior distribution for λ , given synthetically generated data $\mathbf{y} = (\mathbf{y}^G, \mathbf{y}^E)$, denoted by \mathcal{D} . For this, one relies on the UQTk application `model_inf` (step I2) that applies MCMC sampling to generate samples from the posterior distribution of λ defined via Bayes formula

$$\overbrace{p(\lambda|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{p(\mathcal{D}|\lambda)}^{\text{Likelihood}} \overbrace{p(\lambda)}^{\text{Prior}}. \quad (53.9)$$

For the sake of illustration, normal, independent identically distributed priors with vanishing mean, and standard deviation $s = 0.3$ for the components of λ will be employed $p(\lambda) \propto \exp\left(-\frac{\lambda_1^2 + \lambda_2^2}{2s^2}\right)$, while the likelihood is dictated directly from the assumed noise model (53.8):

$$\mathcal{L}_{\mathcal{D}}(\lambda) = p(\mathcal{D}|\lambda) \propto \prod_{j=1}^R \exp\left(-\frac{(y_j^G - f^G(\lambda))^2}{2\sigma^2}\right) \exp\left(-\frac{(y_j^E - f^E(\lambda))^2}{2\sigma^2}\right). \quad (53.10)$$

Note that the choice of Gaussian likelihood is implemented in the app `model_inf` (for the full set of options implemented in the app, the reader is referred to the UQTk manual), but in general one should view the app as a use case illustration of the *MCMC* library. For non-Gaussian likelihoods, a custom posterior function evaluation can be written in C++ and linked to the MCMC library, similar to the `model_inf` app. Samples of the posterior obtained via MCMC are shown in Fig. 53.4. The data is generated as $y^G \sim N(1.0, 0.05)$ and $y^E \sim N(0.7, 0.05)$, using $R = 13$ replicas. This step is typically assumed to be performed by the user, outside UQTk. After the posterior samples are obtained, one can analyze them

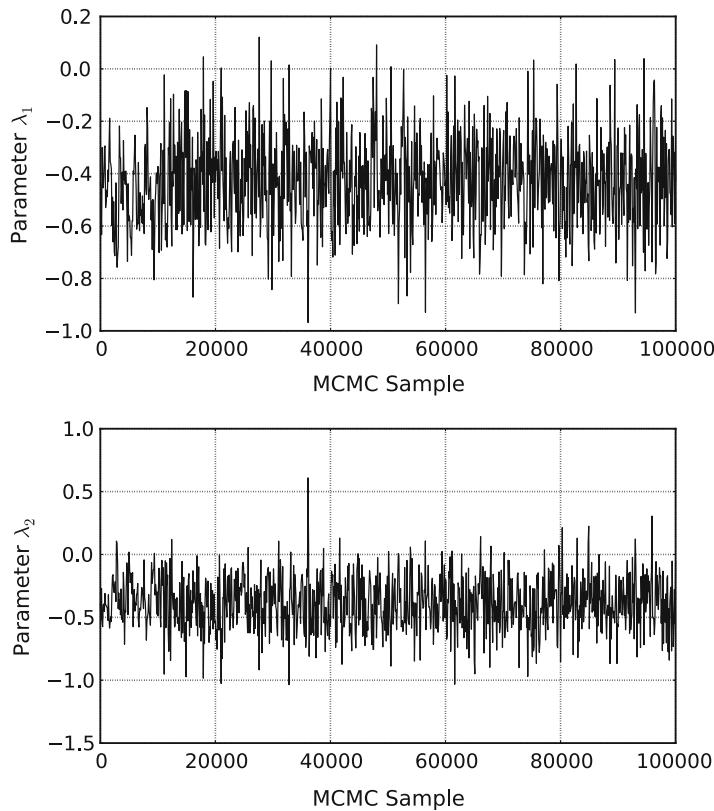


Fig. 53.4 MCMC posterior samples of inferred parameters λ_1 and λ_2

further. For example, Fig. 53.5 illustrates marginal and joint posterior distributions computed via kernel density estimation (UQTk app `pdf_c1` can be employed for this) using the MCMC posterior samples.

Having used the data to calibrate the model parameters, the next task is, armed with the posterior samples of (λ_1, λ_2) , to propagate the uncertainty associated with these parameters through an arbitrary model $g(\lambda)$ to obtain predictions of $g(\lambda)$ with quantified uncertainty. However, since at this point this parametric uncertainty is captured by a set of samples from the posterior distribution, it first needs to be represented in a format that is more suitable to the tools used here for forward propagation.

4.4 Polynomial Chaos Construction via Rosenblatt Transformation

In this example, the nonintrusive spectral projection (NISP) with polynomial chaos is employed in order to propagate posterior uncertainties of the model parameters

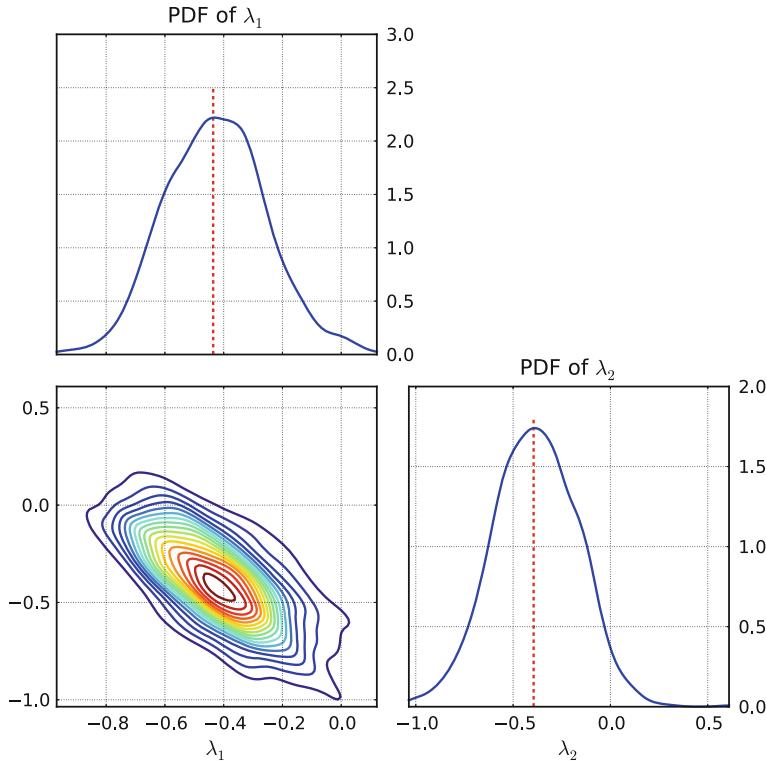


Fig. 53.5 Illustration of the marginal posterior distributions $p(\lambda_1|\mathcal{D})$ and $p(\lambda_2|\mathcal{D})$, as well as the full posterior $p(\lambda_1, \lambda_2|\mathcal{D})$ computed via kernel density estimation of the MCMC posterior samples. The dashed red line indicates maximum a posteriori (MAP) value

through models of interest. This requires a construction of polynomial chaos expansions for the input parameters given the samples obtained by MCMC in the previous stage. The application `pce_quad` performs exactly this task (step I3 in Table 53.3), i.e., given samples $\boldsymbol{\lambda}^{(j)}$ for $j = 1, \dots, M$ of a random vector $\boldsymbol{\lambda}$, build a set of joint PC expansions:

$$\lambda_i \approx \sum_{\alpha \in \mathcal{S}} l_{i\alpha} \Psi_\alpha(\xi), \text{ for } i = 1, \dots, d, \quad (53.11)$$

for a chosen truncation $\mathcal{S} = \{\alpha : \sum_{i=1}^d \alpha_i \leq s\}$ of a total order s . For this task, Gauss-Hermite PC is employed; therefore, $\Psi_\alpha(\cdot)$ are Hermite polynomials and ξ are i.i.d. standard normal random variables. The PC coefficients $l_{i\alpha}$ are determined via orthogonal projection

$$l_{i\alpha} = \frac{1}{||\Psi_\alpha(\xi)||^2} \langle \lambda_i \Psi_\alpha(\xi) \rangle, \quad (53.12)$$

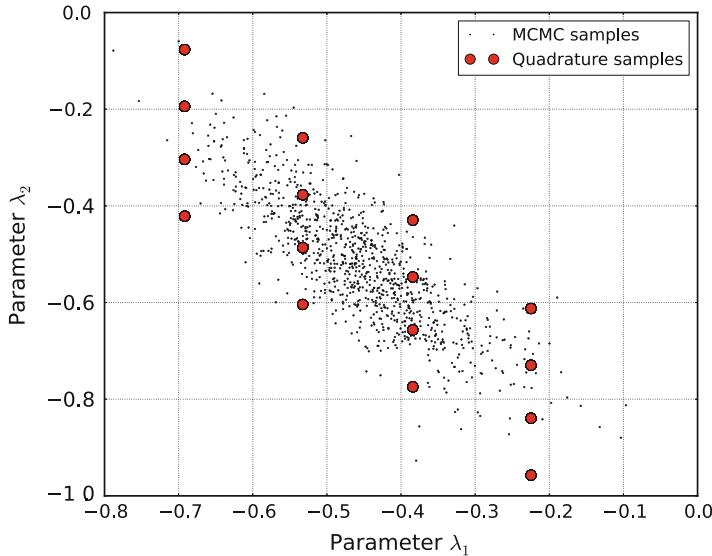


Fig. 53.6 Posterior samples of parameters λ_1, λ_2 , as well as the Gauss quadrature points mapped from (ξ_1, ξ_2) to (λ_1, λ_2)

where the integrals $\langle \lambda_i \Psi_\alpha(\xi) \rangle = \int_{\xi} \lambda_i(\xi) \Psi_\alpha(\xi) \pi_\xi(\xi) d\xi \approx \sum_{q=1}^Q \lambda_i(\xi_q) \Psi_\alpha(\xi_q) w_q$ require the mapping $\xi \rightarrow \lambda$ accomplished by the inverse Rosenblatt transformation [16] of the Gauss quadrature points ξ . For details of the Rosenblatt transformation and the overall procedure, see [18] or the UQTk manual [7]. Figure 53.6 illustrates MCMC samples of the two parameters (λ_1, λ_2) together with the 4×4 quadrature point set $\lambda_{1,2}(\xi_q)$ for $q = 1, \dots, 16$.

4.5 Uncertainty Propagation and Attribution

Having built a PC expansion for the inputs λ , one can proceed to construct the output PC expansion for any model $g(\lambda)$, including the ones used in the calibration. This is once again the same forward UQ propagation task used in surrogate construction, albeit with potentially higher order Gauss-Hermite input PC expansions (53.11) instead of the linear Legendre-Uniform PC (53.1) that was used in the surrogate construction. Both training (quadrature) and validation samples should be obtained via a combination of the UQTk apps `generate_quad`, `pce_rv` and `pce_eval` (Steps F1-F4 in Table 53.2), this time with a more involved input PC coefficient matrix that incorporates the values $l_{i\alpha}$. The PC expansion for the model $g(\lambda)$ can be written as:

$$g(\lambda) \approx \sum_{\alpha \in \mathcal{S}} g_\alpha \Psi_\alpha(\xi(\lambda)), \quad (53.13)$$

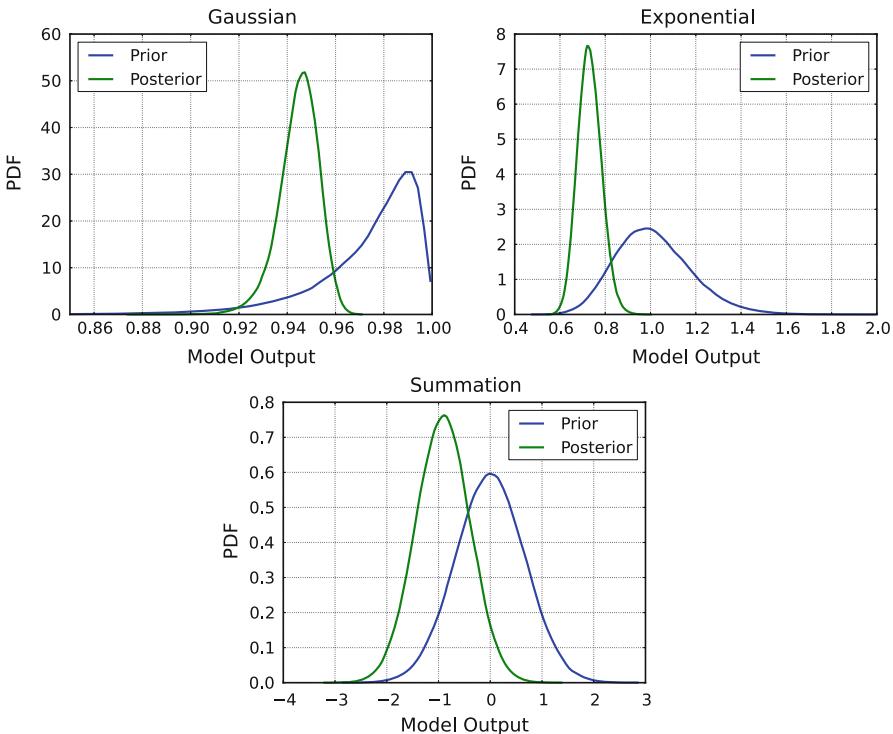


Fig. 53.7 Model output PDFs constructed by sampling the resulting PC expansions

with the orthogonal projection formula (similar to the surrogate construction, this can be accomplished using the application `pce_resp`):

$$g_\alpha = \frac{1}{||\Psi_\alpha(\xi)||^2} \langle g(\lambda(\xi)) \Psi_\alpha(\xi) \rangle, \quad (53.14)$$

and the implied relationship $\xi \leftrightarrow \lambda$ encoded in input PC (53.11). As forward propagation examples, consider a simple summation model $f^S(\lambda) = \sum_{i=1}^d \lambda_i$, as well as the two test models (53.6) used for calibration, $f^G(\lambda)$ and $f^E(\lambda)$. For the sake of the full workflow illustration, these models are considered as functions of the full set of $d = 5$ inputs. While the first two calibrated parameters, λ_1 and λ_2 , are described by their joint posterior distribution, the last three less important parameters will still obey their prior distributions, i.e., independent, $N(0, 0.3)$ normal random variables. Figure 53.7 illustrates the estimated PDFs using 100,000 samples (obtained via `pce_rv`, or step F8) of the 3-rd order output PC expansions for all three models, before and after the calibration, i.e., employing the full 5-dimensional normal i.i.d. prior or the posterior that has a joint structure in the first two parameters.

In all three cases, the effect of calibration is the narrowing of the output distribution, which reflects the fact that the information obtained from the data reduces the output uncertainty. The PDF computation is performed using UQTk app `pdf_c1` (step F9) which implements kernel density estimation (KDE) [20,21] with improved fast Gauss transform [26] for computational efficiency gains – the latter is at least an order of magnitude faster than Python’s native KDE estimators for high-dimensional data sets. Note that steps F8 and F9 of forward propagation, i.e., PDF estimation, are not necessary steps during surrogate construction as the latter is developed in order to replace the model and assess parameter sensitivities rather than for output PDF estimation.

Having computed the output PC expansion (53.13), one can evaluate the sensitivities of the model outputs with respect to components of ξ , as before, using the UQTk apps `gen_mi` and `pce_sens`. However, while ξ_3, ξ_4, ξ_5 can be fully attributed to the inputs $\lambda_3, \lambda_4, \lambda_5$, respectively, the first two inputs are now correlated. While the sampling methods would in principle help estimate the sensitivities with respect to λ_1 and λ_2 , here, they are considered together. As such, the total variance contribution from the pair (ξ_1, ξ_2) is identified with that of the pair (λ_1, λ_2) . Figure 53.8 demonstrates the main sensitivities before and after the calibration. For example, for the summation model, before the calibration, all five inputs have equal contribution to the total variance, while, after the calibration of the first two parameters, their respective contribution has reduced considerably, while the three other input parameters remain equal contributors. For the Gaussian and exponential models also, the contribution of the pair (λ_1, λ_2) , after calibration, is smaller than the sum of their contributions before calibration, due to their reduced variance.

4.6 Command-Line Reproduction

To reproduce the exact command-line executions for the example workflow described here, all of the steps are summarized in Tables 53.2 and 53.3 in terms of Forward and Inverse UQ tasks, respectively. The steps in these tables have been referred to in the previous subsections. The exact order of the workflow execution is presented below, where `Forward(DIM, PCTYPE, INPC)` and `Inverse()` are *not* specific UQTk functions – they denote groups of command-line UQTk applications including appropriate file management, model evaluation, and data generation.

- `Forward(DIM=5, PCTYPE='LU', INPC='pcf_diag.dat')`
[High-d Surrogate]
 - Generate input parameter values (Steps F1-F4)
 - For each model, evaluate it (Step F5), project to obtain PC coefficients (Step F6), and postprocess (Steps F7-F9)

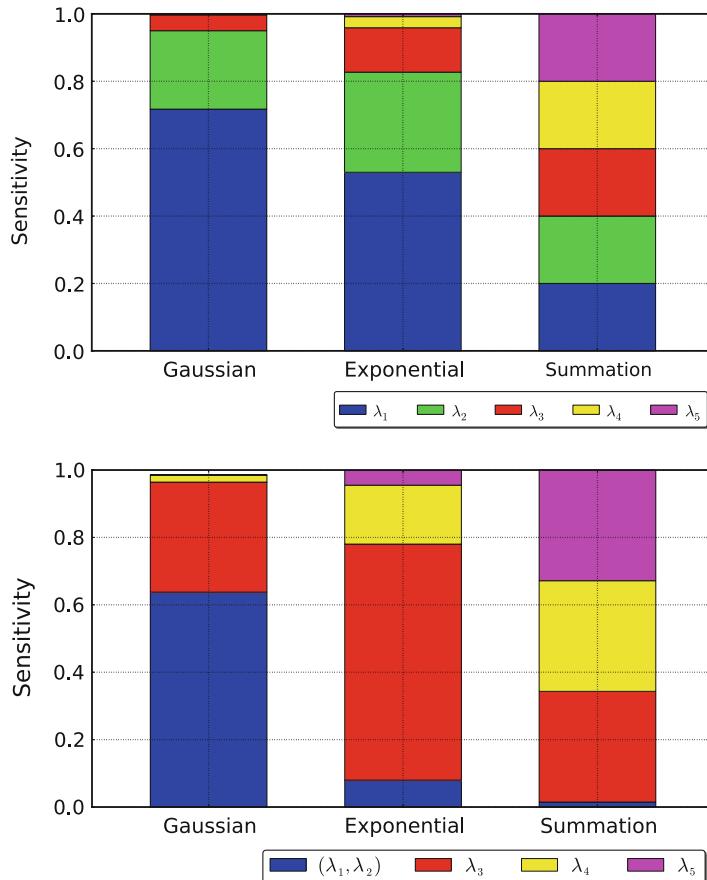


Fig. 53.8 Illustration of the variance decomposition for three models before and after the calibration of the first two inputs

- `Forward(DIM=2, PCTYPE='LU', INPC='pcf_diag.dat')` [Low-d Surrogate]
- `Inverse()` [Model calibration]
 - Obtain/generate data (Step I1)
 - Perform Bayesian inference via MCMC (Step I2)
 - Construct PC for the MCMC samples (Step I3)
- `Forward(DIM=2, PCTYPE='HG', INPC='pcf_calib.dat')` [Prediction]

The two PC input files correspond to the near-diagonal coefficients in Eq. (53.5) with $d = 5$ or $d = 2$ (`pce_diag.dat`) and to the calibrated PC expansion in Eq. (53.11) that is the output of the step I3 (`pce_calib.dat`).

As an illustration of the forward workflow, together with some of the plotting routines that have been used to produce figures in this chapter, UQTk contains a set of python scripts in *examples/uqpc* that essentially carry out the steps contained in `Forward(...)`.

5 Conclusion

The UQ Toolkit offers a collection of tools for a wide range of UQ operations, ranging from forward UQ to surrogate construction, sensitivity analysis, and inverse problems. UQTk is geared toward tutorials, UQ algorithm development, and custom UQ workflow design. The key methods are implemented as C++ libraries but are also accessible through stand-alone apps or through a Python interface. As illustrated in this chapter, the tools can be easily combined to form UQ workflows of arbitrary complexity.

Acknowledgements This material is based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under the Applied Math Research (AMR), and Scientific Discovery through Advanced Computing (SciDAC) programs. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Cross-References

- ▶ [The Bayesian Approach to Inverse Problems](#)
- ▶ [Polynomial Chaos: Modeling, Estimation, and Approximation](#)
- ▶ [Surrogate Models for Uncertainty Propagation and Sensitivity Analysis](#)

References

1. Askey, R., Wilson, J.: Some basic hypergeometric polynomials that generalize Jacobi polynomials. *Mem. Am. Math. Soc.* **319**, 1–55 (1985)
2. Babacan, S., Molina, R., Katsaggelos, A.: Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **19**(1), 53–63 (2010)
3. Blatman, G., Sudret, B., et al.: Efficient global sensitivity analysis of computer simulation models using an adaptive least angle regression scheme. In: 41èmes Journées de Statistique, SFdS, Bordeaux (2009)
4. Clenshaw, C.W., Curtis, A.R.: A method for numerical integration on an automatic computer. *Numerische Mathematik* **2**, 197–205 (1996)
5. Crestaux, T., Le Maître, O., Martinez, J.: Polynomial chaos expansion for sensitivity analysis. *Reliab. Eng. Syst. Saf.* **94**(7), 1161–1172 (2009)
6. Debusschere, B., Najm, H., Pébay, P., Knio, O., Ghanem, R., Le Maître, O.: Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.* **26**(2), 698–719 (2004)

7. Debusschere, B., Sargsyan, K., Safta, C., Chowdhary, K.: UQ toolkit. <http://www.sandia.gov/UQToolkit> (2015). Accessed 15 Jan 2015
8. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 721–741 (1984). doi:10.1109/TPAMI.1984.4767596
9. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms* **18**, 209–232 (1998). doi:10.1023/A:1019129717644, also as SFB 256 preprint 553, Univ. Bonn, 1998
10. Ghanem, R., Spanos, P.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (1991)
11. Golub, G.H., Welsch, J.H.: Calculation of Gauss quadrature rules. *Math. Comput.* **23**, 221–230 (1969). doi:10.1090/S0025-5718-69-99647-1
12. Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. *Bernoulli* **7**, 223–242 (2001). doi:10.2307/3318737
13. Hildebrand, F.: *Introduction to Numerical Analysis*. Dover Publications, Inc. New York (1987)
14. Kucherenko, S., Tarantola, S., Annoni, P.: Estimation of global sensitivity indices for models with dependent variables. *Comput. Phys. Commun.* **183**, 937–946 (2012). doi:10.1016/j.cpc.2011.12.020
15. Roberts, G.O., Tweedie, R.L.: Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2**(4), 341–363 (1996). <http://projecteuclid.org/euclid.bj/1178291835>
16. Rosenblatt, M.: Remarks on a multivariate transformation. *Ann. Math. Stat.* **23**(3), 470–472 (1952). doi:10.1214/aoms/1177729394
17. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**(2), 259–270 (2010)
18. Sargsyan, K., Debusschere, B., Najm, H., Le Maître, O.: Spectral representation and reduced order modeling of the dynamics of stochastic reaction networks via adaptive data partitioning. *SIAM J. Sci. Comput.* **31**(6), 4395–4421 (2010)
19. Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., Thornton, P.: Dimensionality reduction for complex models via Bayesian compressive sensing. *Int. J. Uncertain. Quantif.* **4**(1), 63–93 (2014). doi:10.1615/Int.J.UncertaintyQuantification.2013006821
20. Scott, D.: *Multivariate Density Estimation. Theory, Practice and Visualization*. Wiley, New York (1992)
21. Silverman, B.: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London (1986)
22. Sivia, D.S., Skilling, J.: *Data Analysis: A Bayesian Tutorial*, 2nd edn. Oxford University Press, Oxford/New York (2006)
23. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Sov. Math. Dokl.* **4**, 240–243 (1963)
24. Sobol, I.M.: Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.* **1**, 407–414 (1993)
25. Xiu, D., Karniadakis, G.: The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
26. Yang, C., Duraiswami, R., Gumerov, N.A., Davis, L.: Improved fast Gauss transform and efficient Kernel density estimation. In: *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 664–671. IEEE Computer Society, Washington, DC (2003)

The Parallel C++ Statistical Library for Bayesian Inference: QUESO

54

Damon McDougall, Nicholas Malaya, and Robert D. Moser

Abstract

The Parallel C++ Statistical Library for the Quantification of Uncertainty for Estimation, Simulation, and Optimization (QUESO) is a collection of statistical algorithms and programming constructs supporting research into the quantification of uncertainty of models and their predictions. QUESO is primarily focused on solving statistical inverse problems using Bayes' theorem, which expresses a distribution of possible values for a set of uncertain parameters (the posterior distribution) in terms of the existing knowledge of the system (the prior) and noisy observations of a physical process, represented by a likelihood distribution. The posterior distribution is not often known analytically and so requires computational methods. It is typical to compute probabilities and moments from the posterior distribution, but this is often a high-dimensional object, and standard Riemann-type methods for quadrature become prohibitively expensive. The approach QUESO takes in this regard is to rely on Markov chain Monte Carlo (MCMC) methods which are well suited to evaluating quantities such as probabilities and moments of high-dimensional probability distributions. QUESO's

D. McDougall (✉) • N. Malaya

Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

e-mail: damon@ices.utexas.edu; nick@ices.utexas.edu

R.D. Moser

Department of Mechanical Engineering, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA

Predictive Engineering and Computational Science, Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, TX, USA
e-mail: rmoser@ices.utexas.edu

intended use is as tool to assist and facilitate coupling uncertainty quantification to a specific application called a *forward problem*. While many libraries presently exist that solve Bayesian inference problems, QUESO is a specialized piece of software primarily designed to solve such problems by utilizing parallel environments demanded by large-scale forward problems. QUESO is written in C++, uses MPI, and utilizes libraries already available to the scientific community. Thus, the target audience of this library are researchers who have solid background in Bayesian methods, are comfortable with UNIX concepts and the command line, and have knowledge of a programming language, preferably C/C++.

Keywords

Bayesian inference • Inverse problems • Monte Carlo methods • Computational Markov chains • Mathematical software • Parallel computation • Parallel algorithms

Contents

1	Introduction	1831
2	Motivation	1831
3	Alternatives to QUESO	1832
4	Formulation	1833
	4.1 The Forward Problem	1834
	4.2 The Inverse Problem	1834
	4.3 Prediction	1835
5	Examples	1835
	5.1 A Template Example	1835
	5.2 Defining the Likelihood Distribution	1839
	5.3 Ball Drop Example	1840
	5.4 Statistical Inverse Problem	1840
	5.5 Statistical Forward Problem	1842
	5.6 Example Code	1843
	5.7 Running the Gravity Example with Several Processors	1846
	5.8 Data Post-processing and Visualization	1847
	5.9 Infinite-Dimensional Inverse Problems	1848
6	Extensibility	1852
	6.1 Custom Priors	1852
7	The QUESO Design and Implementation	1854
	7.1 Software Engineering	1854
	7.2 QUESO Internals	1855
8	Algorithms	1857
	8.1 DRAM	1857
	8.2 Multilevel	1857
	8.3 Preconditioned Crank-Nicolson	1858
9	Input File	1858
10	Conclusions	1861
	10.1 QUESO-Provided Likelihoods	1861
	10.2 Emulators	1863
	10.3 API Considerations	1863
	10.4 Exascale	1864
	References	1865

1 Introduction

The Parallel C++ Statistical Library for the Quantification of Uncertainty for Estimation, Simulation, and Optimization (QUESO), is a collection of statistical algorithms and programming constructs supporting research into the uncertainty quantification (UQ) of models and their predictions. It has been designed with three objectives: (a) to be sufficiently abstract in order to handle a large spectrum of large-scale computationally intensive models; (b) to be extensible, allowing easy creation of custom-defined objects; and (c) leverage parallel computing through use of high-performance vector and matrix libraries. Such objectives demand a combination of an object-oriented design with robust software engineering practices. QUESO is written in C++, uses MPI, and utilizes libraries already available to the scientific community.

The purpose of this book chapter is not to teach uncertainty quantification methods, but rather to introduce the QUESO library so it can be used as a tool to assist and facilitate coupling UQ to a specific application (forward problem). Thus, the target audience of this chapter is researchers who have solid background in Bayesian methods, are comfortable with UNIX concepts and the command line, and have knowledge of a programming language, preferably C/C++.

The rest of the document is organized as follows. Section 2 has a brief discussion of statistical inverse problems, and in doing so, provides the impetus behind the QUESO library. Section 4 then discusses the types of problems the library is designed to solve, as well as introducing the notation used for the rest of this document. Several illustrative examples, including the new infinite-dimensional capability, are provided in Sect. 5 along with code snippets demonstrating typical software call-patterns. Section 6 discusses how the library design can easily be extended for bespoke user-defined random variables, probability distribution functions, and realizers. Section 7 discusses the design and internals of the library, as well as providing a software snapshot of the current library status. Finally, we conclude by discussing several areas in which to focus future QUESO development efforts.

All of the examples in this document are present in the QUESO source tree of the latest release, 0.53.0. Users should consult the website, libqueso.com, for the latest news and source code.

This chapter builds on the 2012 paper that introduced the QUESO library [1] and the current QUESO user's manual [2] by including a myriad of changes that have since been incorporated into the library.

2 Motivation

Statistical inverse problems using a Bayesian formulation model all quantities as random variables, where probability distributions of the quantities capture the uncertainty in their values. The solution to the inverse problem is then the probability distribution of the quantity of interest when all information available has

been incorporated in the model. This (posterior) distribution describes the degree of confidence about the quantity after the measurement has been performed [3].

Thus, the solution to the statistical inverse problem is given by Bayes' formula, which expresses the posterior distribution in terms of the prior distribution and the data represented through the likelihood function.

For all but toy problems, the likelihood function has an open form and its evaluation is highly computationally intensive. Worse, simulation-based posterior inference often requires a large number of these evaluations of the forward model. Therefore, fast and efficient sampling techniques are desirable for posterior inference.

It is often not straightforward to obtain explicit posterior point estimates of the solution, since it usually involves the evaluation of a high-dimensional integral with respect to a possibly non-smooth posterior distribution. In such cases, an alternative integration technique is the Markov chain Monte Carlo method where posterior moments may be estimated using the samples from a series of (correlated) random draws from the posterior distribution.

QUESO is designed in an abstract way so that it can be used by any computational model, as long as a likelihood function (in the case of statistical inverse problems) and a quantity of interest (QoI) function (in the case of statistical forward problems) are provided by the user application.

With this framework in mind, QUESO provides tools for both sampling algorithms for statistical inverse problems, following Bayes' formula, and statistical forward problems. It provides Markov chain Monte Carlo algorithms using the Metropolis-Hastings acceptance ratio [4, 5]: these are the multilevel Monte Carlo [6] method and DRAM [7]. QUESO is also capable of handling several chains in parallel computational environments.

3 Alternatives to QUESO

QUESO is certainly not the only quality statistical software library. There are many different libraries that can be used to solve Bayesian inference problems. QUESO is a specialized piece of software, primarily designed to solve such problems utilizing the, often required, parallel environment demanded by large-scale forward problems. This focus is simultaneously the QUESO's greatest strength and weakness, depending on user's target problem. For instance, QUESO would be less effective to use for serial problems than several alternative libraries, as there is significant turnaround time from learning how to build QUESO and link a custom forward code to it. In instances where parallelization is not necessary and the forward problem is relatively cheap to execute, there are good alternatives to QUESO. We now provide a simple survey of several other major libraries that we consider useful for problems of Bayesian inference, along with a brief discussion some unique strengths and weaknesses.

As discussed above, for inference problems that do not require parallelization, serial libraries can be leveraged with less development. An excellent example of this is PyMC [8]. A modern software package, PyMC is a python-based software library for Bayesian estimation and MCMC. Its strengths lie in its flexibility and excellent post-processing, especially when coupled with matplotlib [9]. emcee [10] is another python-based package, with a particular emphasis on Bayesian parameter estimation. Both of these libraries are useful for rapid software prototyping using serial inference problems.

On the other end of the spectrum, there are complete statistical software languages. These are often more mature software projects which are capable of much more general statistical computations than QUESO. However, these languages are often weaker for specialized problems, because they are not as well optimized for solving Bayesian inference problems, particularly at scale. The ultimate example of this is R [11]. R is a free software programming language and software environment for statistical computing and graphics. R is arguably the most general and complete source of open-source statistical packages in the world. It is not limited to Bayesian techniques and has packages across a wide range of topics in statistics. However, it is not easy to couple R with other codebases (for the forward problem, for instance). Furthermore, while some packages supporting parallelism are now being developed, the language is still primarily focused on serial computations. Another alternative is Stan [12]. Stan is a probabilistic programming language written in C++ implementing full Bayesian statistical inference.

Another major library is WinBUGS [13]. WinBUGS is statistical software for Bayesian analysis using MCMC methods. WinBUGS is of particular historical importance, as it was one of the earliest openly available MCMC libraries, with development starting the late 1980s. It is also unique in that it is developed for the Windows platform, instead of Linux. It is also primarily based on the Gibbs sampler algorithm.

Finally, the DAKOTA [14] toolkit is a very general library developed at Sandia National Laboratories, containing a vast array of algorithms with applications to uncertainty quantification, optimization, emulation, experimental design, prediction, and sensitivity analysis. DAKOTA is written in C++ and supported on Linux, OS X, and Windows and represents 20 years of advanced algorithms research. Furthermore, given DAKOTA's advanced certainty propagation algorithms, the QUESO and DAKOTA development teams are working together to establish a seamless integration of QUESO's algorithms into DAKOTA to give users a matured and coupled forward and inverse UQ software solution.

4 Formulation

Here we give a rigorous description of the types of problems that QUESO solves. This will crystallize both the terminology and notation in an attempt to make everything in this chapter self-contained.

4.1 The Forward Problem

Here we set out the auspices under which we will operate. We make two high-level assumptions: (1) we have access to a set of observations of some physical phenomenon; and (2) we have a mathematical model that attempts to model the observed physical phenomenon. Ensuring that the mathematical model is *valid* is an exercise left to the reader. We will denote the observed data by y and the mathematical model by \mathcal{G} . The model will certainly depend on various parameters, and we call the process of mapping these parameters to model output the *forward problem*. In many physical engineering applications, the forward problem is expensive and may involve the solution of a set of partial differential equations.

4.2 The Inverse Problem

In the subsection above, we described the forward problem. It may be the case that the mathematical model in the forward problem may depend on some parameters that are unknown and that we wish to estimate. We will refer to these unknown parameters as θ . The process of estimating θ given observations goes by many names, but is generally referred to as the *inverse problem*. There are several frameworks for solving inverse problems. We will focus only on the *Bayesian framework*, which we rigorously describe now.

As noted above, we are given a set of observations y . This dataset is corrupted by errors made during the experiment. These errors could be human errors, equipment errors, or errors in the setup of the experimental scenario. In complete generality, it is difficult to say with certainty what statistical distribution these errors follow. In a lot of experimental cases, however, a Gaussian distribution with some, perhaps unknown, variance is quite a reasonable characterization.

The unknown parameters themselves might have some inherent constraining property. For example, if the unknown parameter were a concentration of a contaminant underground then it is not possible for this unknown parameter to be negative. The constraint varies depending on the physical domain, but it is rarely the case one knows *nothing* about the unknown parameters. This information can be translated to constraints on a prior distribution.

To regroup, we have a statistical distribution governing the behavior of the experimental errors given the unknown parameters, $\mathbb{P}(y|\theta)$. We also have some prior distribution on the unknown parameters $\mathbb{P}(\theta)$. The Bayesian solution to the inverse problem of finding θ is the distribution of θ given y , $\mathbb{P}(\theta|y)$. By Bayes' rule, this can be written as follows:

$$\mathbb{P}(\theta|y) \propto \mathbb{P}(y|\theta)\mathbb{P}(\theta).$$

The left-hand side is referred to as the posterior distribution. The right-hand side is the product of the likelihood distribution and the prior distribution. QUESO solves

the Bayesian inverse problem by providing samples that are distributed according to the posterior distribution using Markov chain Monte Carlo. This chapter does not provide the details of how MCMC works. The authors refer the reader to the expansive body of available literature on the topic cited throughout this work.

4.3 Prediction

The prediction step in the Bayesian framework is that of estimating some quantity $\mathcal{Q}(\theta)$ dependent on the unknown parameters. This is usually referred to as a *statistical forward problem*. QUESO is equipped to solve statistical forward problems, but throughout this chapter we will focus mainly on the statistical inverse problem.

5 Examples

5.1 A Template Example

Here we walk through a template example. This template should be general enough to serve as a good starting point for most Bayesian inverse problems. Before we step through the example, here it is in its entirety:

```
#include <queso/GslVector.h>
#include <queso/GslMatrix.h>
#include <queso/UniformVectorRV.h>
#include <queso/StatisticalInverseProblem.h>
#include <queso/ScalarFunction.h>
#include <queso/VectorSet.h>

template<class V = QUESO::GslVector, class M = QUESO::GslMatrix>
class Likelihood : public QUESO::BaseScalarFunction<V, M>
{
public:

    Likelihood(const char * prefix, const QUESO::VectorSet<V, M> & domain)
        : QUESO::BaseScalarFunction<V, M>(prefix, domain)
    {
        // Setup here
    }

    virtual ~Likelihood()
    {
        // Deconstruct here
    }

    virtual double lnValue(const V & domainVector, const V * domainDirection,
                          V * gradVector, M * hessianMatrix, V * hessianEffect) const
    {
        // 1) Run the forward code at the point domainVector
        //     domainVector[0] is the first element of the parameter vector
        //     domainVector[1] is the second element of the parameter vector
        //     and so on
    }
}
```

```
//  
// 2) Compare to data, y  
//     Usually we compute something like  
//     || MyModel(domainVector) - y ||^2 / (sigma * sigma)  
//  
// 3) Return below  
  
double misfit = 0.0;  
  
return -0.5 * misfit;  
}  
  
virtual double actualValue(const V & domainVector, const V * domainDirection,  
    V * gradVector, M * hessianMatrix, V * hessianEffect) const  
{  
    return std::exp(this->lnValue(domainVector, domainDirection, gradVector,  
        hessianMatrix, hessianEffect));  
}  
  
private:  
    // Maybe store the observed data, y, here.  
};  
  
int main(int argc, char ** argv) {  
    MPI_Init(&argc, &argv);  
  
    // Step 0 of 5: Set up environment  
    QUESO::FullEnvironment env(MPI_COMM_WORLD, argv[1], "", NULL);  
  
    // Step 1 of 5: Instantiate the parameter space  
    QUESO::VectorSpace<> paramSpace(env,  
        "param_", 1, NULL);  
  
    double min_val = 0.0;  
    double max_val = 1.0;  
  
    // Step 2 of 5: Set up the prior  
    QUESO::GslVector paramMins(paramSpace.zeroVector());  
    paramMins.cwSet(min_val);  
    QUESO::GslVector paramMaxs(paramSpace.zeroVector());  
    paramMaxs.cwSet(max_val);  
  
    QUESO::BoxSubset<> paramDomain("param_", paramSpace, paramMins, paramMaxs);  
  
    // Uniform prior here. Could be a different prior.  
    QUESO::UniformVectorRV<> priorRv("prior_", paramDomain);  
  
    // Step 3 of 5: Set up the likelihood using the class above  
    Likelihood<> lhood("llhd_", paramDomain);  
  
    // Step 4 of 5: Instantiate the inverse problem  
    QUESO::GenericVectorRV<> postRv("post_", paramSpace);  
  
    QUESO::StatisticalInverseProblem<> ip("", NULL, priorRv, lhood, postRv);  
  
    // Step 5 of 5: Solve the inverse problem  
    QUESO::GslVector paramInitials(paramSpace.zeroVector());  
  
    // Initial condition of the chain  
    paramInitials[0] = 0.0;
```

```

paramInitials[1] = 0.0;

QUESO::GslMatrix proposalCovMatrix(paramSpace.zeroVector());

for (unsigned int i = 0; i < 2; i++) {
    // Might need to tweak this
    proposalCovMatrix(i, i) = 0.1;
}

ip.solveWithBayesMetropolisHastings(NULL, paramInitials, &proposalCovMatrix);

MPI_Finalize();

return 0;
}

```

Notice that this template example is fairly short, weighing in at roughly 100 lines of boilerplate C++ code. Incorporating a specific physical model into the likelihood will certainly increase the size of the statistical application. In the meantime, we will walk through the boilerplate setup that will be common to many use cases.

We will start with the main function. This is where most of the setup takes place. Firstly, since QUESO uses MPI, we must call the MPI_Init function before using any of the classes in QUESO. The next line,

```
QUESO::FullEnvironment env(MPI_COMM_WORLD, argv[1], "", NULL);
```

sets up the QUESO environment. The constructor parameters are, in order, an MPI communicator and could be a custom sub-communicator; the filename of a QUESO input file; a prefix, if a different from the default is desired, for input file options specific to the QUESO environment; and an optional EnvOptionsValues object so that the user can set environment options programmatically. The next thing we do is define the dimension of the state space by created a object representing a vector space:

```
QUESO::VectorSpace<> paramSpace(env, "param_", 1, NULL);
```

In this particular example, the dimension of the state space is 1. The constructor parameters here are the QUESO environment; a prefix, if a different from the default is desired, for input file options specific to this parameter space object; and a vector of strings to name components of the vectors belonging to this vector space. Now we are in a position to set up the domain of the statistical inverse problem. QUESO only supports box domains, but the bounds for the box may be arbitrary. We store the bounds for the domain in GslVector objects like so:

```
QUESO::GslVector paramMins(paramSpace.zeroVector());
paramMins.cwSet(min_val);
QUESO::GslVector paramMaxs(paramSpace.zeroVector());
paramMaxs.cwSet(max_val);
```

Here min_val and max_val will be specific to the user's problem. The box domain uses these bounds and is constructed as follows:

```
QUESO::BoxSubset<> paramDomain("param_",
    paramMins,
    paramMaxs);
```

We have finished setting up the domain of the statistical inverse problem. Recall the ingredients we need for a well-posed statistical inverse problem; a prior distribution and a likelihood distribution. QUESO supports many statistical distributions that can all be used as a prior, and the user may choose to implement their own prior distribution if (see Sect. 6) such customization is needed. The following line creates an object representing a uniform random variable:

```
QUESO::UniformVectorRV<> priorRv("prior_",
    paramDomain);
```

This object contains all the necessary information to fully define a uniformly distributed random variable, namely, its probability density function and mechanisms by which one can make draws with this density. The second ingredient needed for a statistical inverse problem is the definition of a likelihood distribution, and this is done now:

```
Likelihood<> lhood("lhood_", paramDomain);
```

This line may look different to the one for your specific application, as it is intended to interact with a specific physical model. The `Likelihood` class is a custom-defined class. We will come back to the full `Likelihood` class in Sects. 5.3 and 5.2 explain how it is implemented. For now, we will continue with the setup of the inverse problem and all the necessary code needed to initialize the sampling. We construct a placeholder object that represents a posterior random variable:

```
QUESO::GenericVectorRV<> postRv("post_", paramSpace);
```

QUESO will operate on this object during the sampling. After QUESO has finished its sampling, this object is then available to you for post-processing. Next, we pass the prior, likelihood, and posterior over to the `StatisticalInverseProblem` class like so:

```
QUESO::StatisticalInverseProblem<> ip("", NULL,
    priorRv, lhood, postRv);
```

We are now ready to finalize the setup of the inverse problem. We do this by giving QUESO an initial condition for the sampler:

```
QUESO::GslVector paramInitials(
    paramSpace.zeroVector());
paramInitials[0] = 0.0;
paramInitials[1] = 0.0;
```

We also give QUESO an initial covariance matrix:

```
QUESO::GslMatrix proposalCovMatrix(
    paramSpace.zeroVector());
```

```
for (unsigned int i = 0; i < 1; i++) {
    proposalCovMatrix(i, i) = 0.1;
}
```

The closer this matrix is to the covariance between parameters under the posterior measure, the better the Markov chain will perform. Providing a bad initial covariance does not change the posterior distribution in the limit of infinite samples. Finally, we begin sampling with the following call:

```
ip.solveWithBayesMetropolisHastings(NULL,
    paramInitials, &proposalCovMatrix);
```

5.2 Defining the Likelihood Distribution

As can be observed in the example illustrated above, the user must pass a likelihood to QUESO. QUESO expects, as a likelihood, anything that subclasses the `BaseScalarFunction` abstract base class. This base class has two pure virtual functions that must be implemented in any subclass. These functions are `lnValue()` and `actualValue()`. The function `lnValue` takes a number of parameters, the most important of which is `const V & domainVector`. When the user implements this function, it should return the natural logarithm of the likelihood distribution evaluated at the point `domainVector`. A concrete example of this can be seen in the next subsection. The function `actualValue` should return exactly the likelihood distribution evaluated at the point `domainVector`. For most practical applications, this function will usually just return `std::exp` of `lnValue`, but the user has the freedom to implement a more optimized computation if one is needed.

A typical Gaussian likelihood distribution will look something like this:

```
template<class V, class M>
double
Likelihood<V = QUESO::GslVector,
            M = QUESO::GslMatrix>::lnValue(
    const V & domainVector,
    const V * domainDirection, V * gradVector,
    M * hessianMatrix, V * hessianEffect) const
{
    double misfit = 0.0;
    unsigned int vec_len = domainVector.sizeLocal()

    for (unsigned int i = 0; i < vec_len; i++) {
        misfit += domainVector[i] - observation[i];
    }

    return -0.5 * misfit;
}
```

To avoid numerical problems computing the acceptance probability in an MCMC algorithm, QUESO will call `lnValue` instead of `actualValue` to do the accept-reject step in log space.

5.3 Ball Drop Example

This section presents an example of how to use QUESO as an application that solves a statistical inverse problem (SIP) and a statistical forward problem (SFP), where the solution of the former serves as input to the later. This example will use the canonical “ball drop” problem, a standard problem in uncertainty quantification. The objective of the SIP is to infer the acceleration due to gravity on an object in free fall near the surface of the Earth. During the SFP, the distance traveled by a projectile launched at a given angle and altitude is calculated using the calibrated magnitude of the acceleration of gravity gathered during the SIP. As expressed in Sect. 4, both the inference and forward problem will be performed using a Bayesian methodology, and so, the resulting quantities of interest (QoIs) will be expressed as probability distributions.

5.4 Statistical Inverse Problem

A deterministic mathematical model for the vertical motion of an object in free fall near the surface of the Earth is given by

$$h(t) = -\frac{1}{2}gt^2 + v_0t + h_0. \quad (54.1)$$

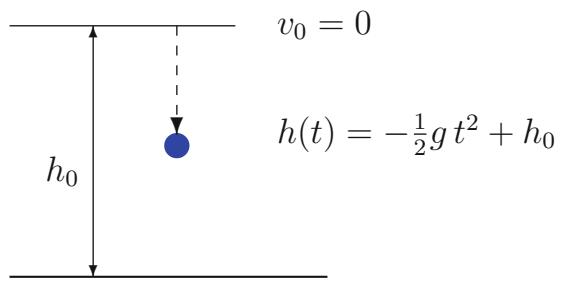
where, v_0 [m/s] is the initial velocity, h_0 [m] is the initial altitude, $h(t)$ [m] is the altitude with respect to time, t [s] is the elapsed time, and g [m/s^2] is the magnitude of the acceleration due to gravity (the parameter which cannot be directly measured and will be statistically inferred).

This model is an expression of a high-fidelity model, Newton’s second law of motion. However, the model is imperfect, as it does not account resistive force of air resistance, for example.

5.4.1 Experimental Data

The experimental data will be generated from an identical object falling from several different heights, each with zero initial velocity (see Fig. 54.1). We collect data, y , of the time taken for the ball to impact the ground starting from various different initial heights. Each experimental observation error is treated as Gaussian with some known mean and variance standard deviation, σ . The error is a result of measurement uncertainties, such as estimates of the actual height the object was dropped from, the human error introduced by operating a stopwatch for time measurement, and any other possible sources of error. The actual observation values can be found in the accompanying source code that will follow shortly.

Fig. 54.1 An object falls from altitude h_0 with zero initial velocity ($v_0 = 0$)



5.4.2 The Prior, Likelihood, and Posterior

In Bayesian inference, the prior probability signifies the modeler's expectation of the result of an experiment before any data is provided. In this problem, a prior must be provided for the parameter g . Near the surface of the Earth, an object in free fall in a vacuum will accelerate at approximately 9.8 m/s^2 , independent of its mass. For this gravitational inference problem, we will place a uniform prior on our unknown variable θ , over the interval [8,11]:

$$\mathbb{P}(\theta) = \mathcal{U}(8, 11). \quad (54.2)$$

We select a Gaussian likelihood function that assigns greater probabilities to parameter values that result in model predictions close to the data:

$$\mathbb{P}(\mathbf{y}|\theta) \propto \exp\left(-\frac{1}{2} (\mathcal{G}(\theta) - \mathbf{y})^T \mathbf{C}^{-1} (\mathcal{G}(\theta) - \mathbf{y})\right), \quad (54.3)$$

where \mathbf{C} is a given covariance matrix, \mathbf{y} is the experimental data, and $\mathcal{G}(\theta)$ is the model output.

Using the deterministic model for the acceleration of gravity (Eq. 54.1) with no initial velocity, the observations \mathbf{y} , and equation (54.3), we have

$$\theta \stackrel{\text{def.}}{=} g, \quad \mathcal{G}(\theta) = \begin{bmatrix} \sqrt{\frac{2h_1}{g}} \\ \sqrt{\frac{2h_2}{g}} \\ \vdots \\ \sqrt{\frac{2h_{n_d}}{g}} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n_d} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_{n_d}^2 \end{bmatrix}, \quad (54.4)$$

where $n_d = 14$ is the number of observations. We now invoke Bayes' formula in order to obtain the posterior PDF $\mathbb{P}(\theta|\mathbf{y})$:

$$\mathbb{P}(\theta|\mathbf{y}) \propto \mathbb{P}(\mathbf{y}|\theta)\mathbb{P}(\theta). \quad (54.5)$$

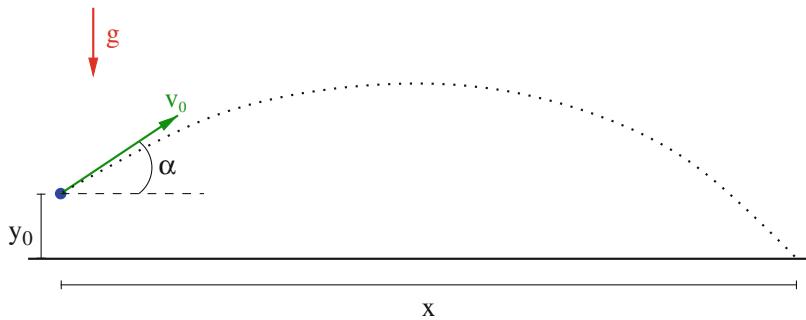


Fig. 54.2 Object traveling with projectile motion

5.5 Statistical Forward Problem

Projectile motion refers to the motion of an object projected into the air at an angle, e.g., a soccer ball being kicked, a baseball being thrown, or an athlete long jumping. In the absence of a propulsion system and neglecting air resistance, the only force acting on the object is proportional to a constant gravitational acceleration g . A deterministic two-dimensional mathematical model for the vertical motion of an object projected from near the surface of the Earth is given by

$$v_x = v_{0x}, \quad (54.6)$$

$$v_y = v_{0y} - gt, \quad (54.7)$$

$$x = v_{0x}t, \quad (54.8)$$

$$h = h_0 + v_{0y}t - \frac{1}{2}gt^2, \quad (54.9)$$

where h_0 is the initial height, $x = x(t)$ is the distance traveled by the object, $\mathbf{v}_0 = (v_{0x}, v_{0y})$ is the initial velocity, $v_{0x} = v_0 \cos(\alpha)$, $v_{0y} = v_0 \sin(\alpha)$, and $v_0 = \|\mathbf{v}_0\|^2$. Figure 54.2 displays the projectile motion of an object in these conditions.

In this example, we assume that h_0 , α , and v_0 are all known, with $h_0 = 0$, $\alpha = \pi/4$, $v_0 = 5$, and g is the result of the SIP described in Sect. 5.4.

Since the result of the SIP is a PDF on g , the output of the mathematical model (54.6) will be a random variable, and our forward problem result will also be statistical in nature.

5.5.1 The Input Random Variable, QoI, and Output Random Variable

The input for the statistical forward problem is the random variable g , the acceleration of gravity. This is the solution (posterior PDF) of the inverse problem described in Sect. 5.4. The QoI for this example is the distance x traveled by an object in projectile motion.

Combining the expressions in Eq. (54.6) and rearranging them, the QoI function for x is

$$x = \frac{v_0 \cos \alpha}{g} \left(v_0 \sin \alpha + \sqrt{(v_0 \sin \alpha)^2 + 2g y_0} \right). \quad (54.10)$$

Here x is the distance traveled and our quantity of interest (QoI).

5.6 Example Code

The source code for the SIP and the SFP is composed of several files. Three of them are common for both problems, `gravity_main.C`, `gravity_compute.h`, and `gravity_compute.C`; they combine both problems and use the solution of the SIP (the posterior PDF for the gravity) as an input for the SFP. We present only the statistical inverse problem here. The forward problem is very similar to the inverse problem, and the user is encouraged to visit the source tree (<https://libqueso.com>) for the full treatment.

The files common to the inverse (and forward) problem are in Listings 1 and 2. Two files specifically handle the SIP: `gravity_likelihood.h` and `gravity_likelihood.C`. These are displayed in Listings 3 and 4.

Listing 1 File `gravity_main.C`.

```
#include <gravity_compute.h>

int main(int argc, char* argv[])
{
    // Initialize QUESO environment
    MPI_Init(&argc,&argv);
    QUESO::FullEnvironment* env =
        new QUESO::FullEnvironment(MPI_COMM_WORLD, argv[1], "", NULL);

    // Call application
    computeGravityAndTraveledDistance(*env);

    // Finalize QUESO environment
    delete env;
    MPI_Finalize();

    return 0;
}
```

Listing 2 File `gravity_compute.C`. The first part of the code (lines 4–44) handles the statistical forward problem, whereas the second part of the code (lines 53–76) handles the statistical forward problem.

```
1 void computeGravityAndTraveledDistance(const QUESO::FullEnvironment& env) {
2     // Statistical inverse problem (SIP): find posterior PDF for 'g'
3
4     // SIP Step 1 of 6: Instantiate the parameter space
5     QUESO::VectorSpace<QUESO::GslVector, QUESO::GslMatrix> paramSpace(env,
6         "param_", 1, NULL);
7 }
```

```

8 // SIP Step 2 of 6: Instantiate the parameter domain
9 QUESO::GslVector paramMinValues(paramSpace.zeroVector());
10 QUESO::GslVector paramMaxValues(paramSpace.zeroVector());
11 paramMinValues[0] = 8.;
12 paramMaxValues[0] = 11.;
13
14 QUESO::BoxSubset<QUESO::GslVector, QUESO::GslMatrix> paramDomain("param_",
15 paramSpace, paramMinValues, paramMaxValues);
16
17 // SIP Step 3 of 6: Instantiate the likelihood object to be used by QUESO.
18 Likelihood<QUESO::GslVector, QUESO::GslMatrix>lhood("like_", paramDomain);
19
20 // SIP Step 4 of 6: Define the prior RV
21 QUESO::UniformVectorRV<QUESO::GslVector, QUESO::GslMatrix> priorRv
22 ("prior_", paramDomain);
23
24 // SIP Step 5 of 6: Instantiate the inverse problem
25 QUESO::GenericVectorRV<QUESO::GslVector, QUESO::GslMatrix>
26 postRv("post_", // Extra prefix before the default "rv_" prefix
27 paramSpace);
28
29 QUESO::StatisticalInverseProblem<QUESO::GslVector, QUESO::GslMatrix>
30 ip("", // No extra prefix before the default "ip_" prefix
31 NULL,
32 priorRv,
33 lhood,
34 postRv);
35
36 // SIP Step 6 of 6: Solve the inverse problem, that is,
37 // set the 'pdf' and the 'realizer' of the posterior RV
38 QUESO::GslVector paramInitials(paramSpace.zeroVector());
39 priorRv.realizer().realization(paramInitials);
40
41 QUESO::GslMatrix proposalCovMatrix(paramSpace.zeroVector());
42 proposalCovMatrix(0,0) = std::pow(std::abs(paramInitials[0]) / 20.0, 2.0);
43
44 ip.solveWithBayesMetropolisHastings(NULL, paramInitials,
45 &proposalCovMatrix);
46
47 // Statistical forward problem (SFP): find the max distance
48 // traveled by an object in projectile motion; input pdf for 'g'
49 // is the solution of the SIP above.
50
51 // SFP Step 1 of 6: Instantiate the parameter *and* qoi spaces.
52 // SFP input RV = FIP posterior RV, so SFP parameter space
53 // has been already defined.
54 QUESO::VectorSpace<QUESO::GslVector, QUESO::GslMatrix> qoiSpace(env,
55 "qoi_", 1, NULL);
56
57 // SFP Step 2 of 6: Instantiate the parameter domain
58 // NOTE: Not necessary because input RV of the SFP = output RV of SIP.
59 // Thus, the parameter domain has been already defined.
60
61 // SFP Step 3 of 6: Instantiate the qoi object to be used by QUESO.
62 Qoi<QUESO::GslVector, QUESO::GslMatrix, QUESO::GslVector, QUESO::GslMatrix>
63 qoi("qoi_", paramDomain, qoiSpace);
64
65 // SFP Step 4 of 6: Define the input RV
66 // NOTE: Not necessary because input RV of SFP=output RV of SIP (postRv).

```

```

67 // SFP Step 5 of 6: Instantiate the forward problem
68 QUESO::GenericVectorRV<QUESO::GslVector, QUESO::GslMatrix> qoiRv("qoi_",
69   qoiSpace);
70
71 QUESO::StatisticalForwardProblem<QUESO::GslVector, QUESO::GslMatrix,
72   QUESO::GslVector, QUESO::GslMatrix> fp("", NULL, postRv, qoi, qoiRv);
73
74 // SFP Step 6 of 6: Solve the forward problem
75 fp.solveWithMonteCarlo(NULL);
76 }

```

Listing 3 File gravity_likelihood.h.

```

template<class V, class M>
class Likelihood : public QUESO::BaseScalarFunction<V, M>
{
public:
    Likelihood(const char * prefix, const QUESO::VectorSet<V, M> & domain);
    virtual ~Likelihood();
    virtual double InValue(const V & domainVector, const V * domainDirection,
        V * gradVector, M * hessianMatrix, V * hessianEffect) const;
    virtual double actualValue(const V & domainVector, const V * domainDirection,
        V * gradVector, M * hessianMatrix, V * hessianEffect) const;

private:
    std::vector<double> m_heights; // heights
    std::vector<double> m_times; // times
    std::vector<double> m_stdDevs; // uncertainties in time measurements
};

```

Listing 4 File gravity_likelihood.C.

```

#include <gravity_likelihood.h>

template<class V, class M>
Likelihood<V, M>::Likelihood(const char * prefix,
    const QUESO::VectorSet<V, M> & domain)
: QUESO::BaseScalarFunction<V, M>(prefix, domain),
  m_heights(0),
  m_times(0),
  m_stdDevs(0)
{
    // Observational data
    double const heights[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110,
        120, 130, 140};

    double const times [] = {1.41, 2.14, 2.49, 2.87, 3.22, 3.49, 3.81, 4.07,
        4.32, 4.47, 4.75, 4.99, 5.16, 5.26};

    double const stdDevs[] = {0.020, 0.120, 0.020, 0.010, 0.030, 0.010, 0.030,
        0.030, 0.030, 0.050, 0.010, 0.040, 0.010, 0.09};

    std::size_t const n = sizeof(heights) / sizeof(*heights);
    m_heights.assign(heights, heights + n);
    m_times.assign(times, times + n);
    m_stdDevs.assign(stdDevs, stdDevs + n);
}

template<class V, class M>

```

```

Likelihood<V, M>::~Likelihood ()
{
    // Deconstruct here
}

template<class V, class M>
double
Likelihood<V, M>::lnValue(const V & domainVector, const V * domainDirection,
                           V * gradVector, M * hessianMatrix, V * hessianEffect) const
{
    double g = domainVector[0];

    double misfitValue = 0.0;
    for (unsigned int i = 0; i < m_heights.size(); ++i) {
        double modelTime = std::sqrt(2.0 * m_heights[i] / g);
        double ratio = (modelTime - m_times[i]) / m_stdDevs[i];
        misfitValue += ratio * ratio;
    }

    return -0.5 * misfitValue;
}

template<class V, class M>
double
Likelihood<V, M>::actualValue(const V & domainVector,
                               const V * domainDirection, V * gradVector, M * hessianMatrix,
                               V * hessianEffect) const
{
    return std::exp(this->lnValue(domainVector, domainDirection, gradVector,
                                    hessianMatrix, hessianEffect));
}

template class Likelihood<QUESO::GslVector, QUESO::GslMatrix>;

```

5.7 Running the Gravity Example with Several Processors

QUESO requires MPI, so any compilation of the user's statistical application will look like this:

```
mpicxx -I/path/to/boost/include -I/path/to/gsl/include \
       -I/path/to/queso/include -L/path/to/queso/lib \
       YOURAPP.C -o YOURAPP -lqueso
```

This will produce a file in the current directory called YOURAPP. To run this application with QUESO in parallel, you can use the standard `mpirun` command:

```
mpirun -np N ./YOURAPP
```

Here `N` is the number of processes you would like to give to QUESO. They will be divided equally among the number of chains requested (see `env_numSubEnvironments` below). If the number of requested chains does not divide the number of processes, an error is thrown.

Even though the application described in Sect. 5.6 is a serial code, it is possible to run it using more than one processor, i.e., produce multiple chains. Supposing

the user's workstation has $N_p = 8$ processors, then, the user may choose to have $N_s = 1, \dots, 8$ subenvironments. This complies with the requirement that the total number of processors in the environment (eight) must be a multiple of the specified number of subenvironments (one). Each subenvironment has only one processor because the forward code is serial.

Thus, to build and run the application code with $N_p = 8$, and $N_s = 8$ subenvironments, the user must set the variable `env_numSubEnvironments = 8` in the input file and enter the following command lines:

```
mpirun -np 8 ./gravity_gsl gravity_inv_fwd.inp
```

The steps above will create a total number of eight raw chains, of size defined by the variable `ip_mh_rawChain_size`. QUESO internally combines these eight chains into a single chain of size $8 \times \text{ip_mh_rawChain_size}$ and saves it in a file named according to the variable `ip_mh_rawChain_dataOutputFileName`. QUESO also provides the user with the option of writing each chain—handled by its corresponding processor—in a separate file, which is accomplished by setting the variable `ip_mh_rawChain_dataOutputAllowedSet = 0 1 ... Ns-1`.

Note: Although the discussion in the previous paragraph refers to the raw chain of a SIP, the analogous is true for the filtered chains (SIP), and for the samples employed in the SFP (`ip_mh_filteredChain_size`, `fp_mc_qseq_size` and `fp_mc_qseq_size`, respectively). See the QUESO user's manual for further details.

5.8 Data Post-processing and Visualization

5.8.1 Statistical Inverse Problem

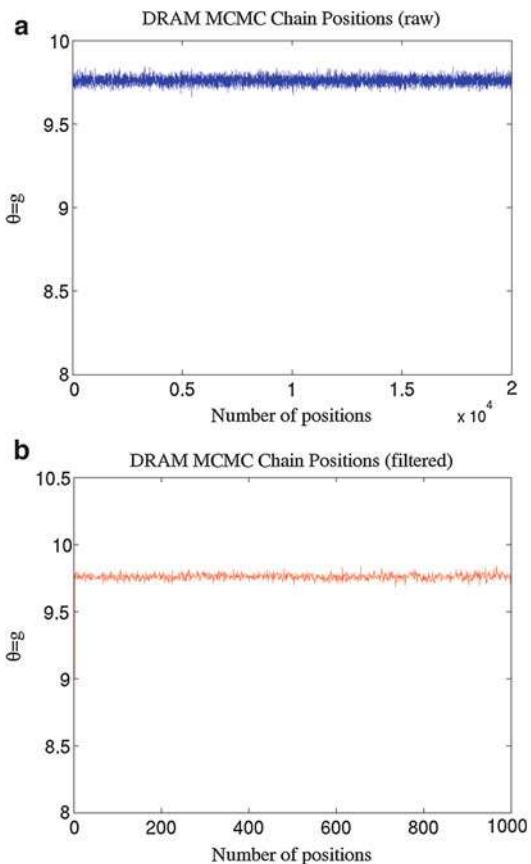
QUESO supports both python and Matlab for post-processing. This section illustrates several forms of visualizing QUESO output and discusses the results computed by QUESO with the code of Sect. 5.6. For Matlab-ready commands for post-processing the data generated by QUESO, refer to the QUESO user's manual.

It is quite simple to plot, using Matlab, the chain of positions used in the DRAM algorithm implemented within QUESO. Figure 54.3a, b show what raw and filtered chain output look like, respectively.

Predefined Matlab and numpy/matplotlib functions exist for converting the raw or filtered chains into histograms. The resulting output can be seen in Fig. 54.4a, b, respectively.

There are also standard built-in functions in Matlab and SciPy to compute kernel density estimates. Resulting output for the raw and filtered chains can be seen in Fig. 54.5a, b, respectively.

Fig. 54.3 MCMC raw chain with 20,000 positions and a filtered chain with lag of 20 positions (a) Raw chain.
(b) Filtered chain

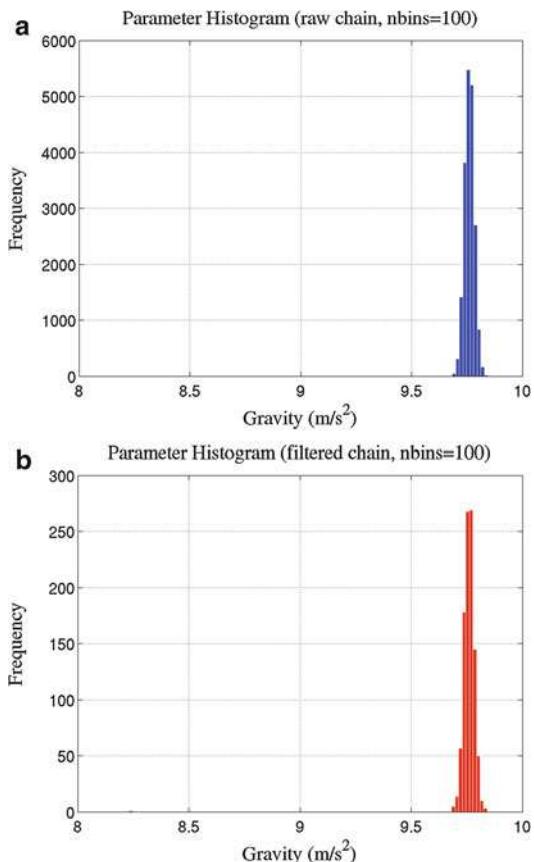


5.9 Infinite-Dimensional Inverse Problems

QUESO has functional but limited support for solving infinite-dimensional inverse problems. Infinite-dimensional inverse problems are problems for which the posterior distribution is formally defined on a function space. After implementation, this distribution will lie on a discrete space, but the MCMC algorithm used is robust to mesh refinement of the underlying function space.

There is still substantial work to be done to bring the formulation of these class of inverse problems in QUESO in line with that of the finite-dimensional counterpart described above, but what currently exists in QUESO is usable. The reason for the departure in design pattern to that of the finite-dimensional code is that for infinite-dimensional problems, QUESO must be agnostic to any underlying vector type representing the random functions that are sampled. To achieve this, a finite element back end is needed to represent functions. There are many choices of finite element libraries that are freely available to download and use, and the design of the infinite-dimensional part of QUESO is such that addition of new back ends should

Fig. 54.4 Histograms of parameter $\theta = g$. (a) Raw chain. (b) Filtered chain



be attainable without too much effort. libMesh is the default and only choice currently available in QUESO. libMesh is open source and freely available to download and use. Visit the libMesh website for further details: <http://libmesh.github.io>

We proceed with showing a concrete example of how to formulate an infinite-dimensional inverse problem in QUESO.

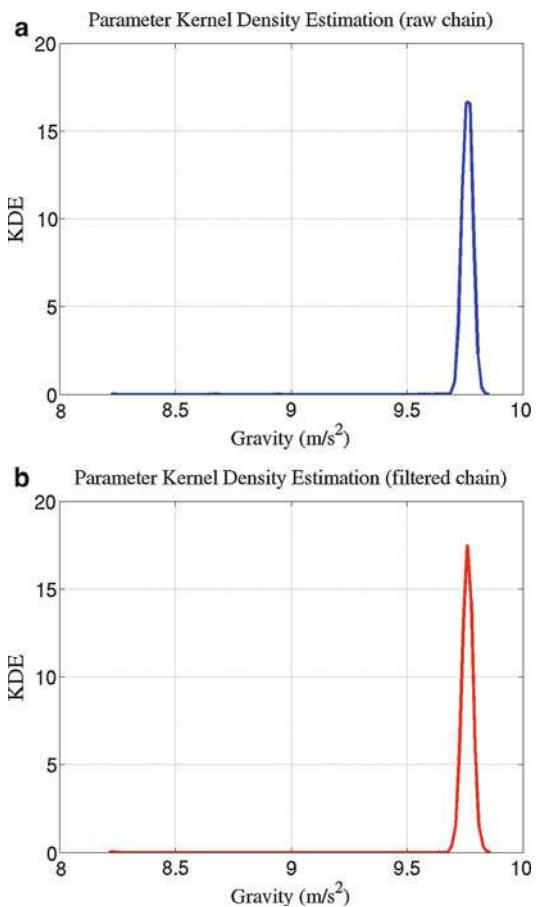
First, we assume the user has access to a `libMesh::Mesh` object on which their forward problem is defined. In what follows, we shall call this object `mesh`.

5.9.1 Defining the Prior

Currently, the only measure you can define is a Gaussian measure. This is because Gaussian measures are well-defined objects on function space and their properties are well understood.

To define a Gaussian measure on function space, one needs a mean function and a covariance operator. QUESO has a helper object to help the user build functions

Fig. 54.5 Kernel density estimation (a) Raw chain.
(b) Filtered chain



and operators called `FunctionOperatorBuilder`. This object has properties that are set by the user that define the type and order of the finite elements used by `libMesh` to represent functions:

```
// Use a helper object to define some of the properties
// of our samples
QUESO::FunctionOperatorBuilder fobuilder;
fobuilder.order = "FIRST";
fobuilder.family = "LAGRANGE";
fobuilder.num_req_eigenpairs = num_pairs;
```

This object will be passed to the constructors of functions and operators and will instruct `libMesh`, in this case, to use first-order Lagrange finite elements. The `num_req_eigenpairs` variable dictates how many eigenpairs to solve for in an eigenvalue problem needed for the construction of random functions. The more eigenpairs used in the construction of Gaussian random functions, the more high-

frequency information is present in the function. The downside to asking for a large number of eigenpairs is that the solution of the eigenvalue problem will take longer. Solving the eigenvalue problem, however, is a one-time cost. The details of the construction of Gaussian random fields can be found in [15–17]. To define a function, one can do the following:

```
QUESO::LibMeshFunction mean(fobuilder, mesh);
```

This function is initialized to be exactly zero everywhere. For more fine-grained control over point values, one can access the internal `libMesh EquationSystems` object using the `get_equation_systems()` method.

Specifying a Gaussian measure on a function space is often more convenient to do in terms of the precision operator rather than the covariance operator. Currently, the only precision operators available in QUESO are powers of the Laplacian operator. However, the design of the class hierarchy for precision operators in QUESO should be such that implementation of other operators is easily achievable. To create a Laplacian operator in QUESO one can do the following:

```
QUESO::LibMeshNegativeLaplacianOperator
    precision(fobuilder, mesh);
```

The Gaussian measure can then be defined by the mean and precision above (where the precision can be taken to a power) as such:

```
QUESO::InfiniteDimensionalGaussian
    mu(env, mean, precision, alpha, beta);
```

Here `beta` is the coefficient of the precision operator, and `alpha` is the power to raise the precision operator to.

5.9.2 Defining the Likelihood

Defining the likelihood is very similar to the ball drop example. We have to subclass `InfiniteDimensional LikelihoodBase` and implement the `evaluate(FunctionBase & flow)` method. This method should return the logarithm of the likelihood distribution evaluated at the point `flow`.

One's specific likelihood implementation will vary from problem to problem, but an example, which is actually independent of `flow`, is shown here for completeness:

```
double
Likelihood::evaluate(QUESO::FunctionBase & flow)
{
    const double obs_stddev = this->obs_stddev();
    const double obs = gsl_ran_gaussian(this->r, obs_stddev);
    return obs * obs / (2.0 * obs_stddev * obs_stddev);
}
```

The reader is reminded that a full working implementation of this example is available in the source tree. See <http://libqueso.com>.

5.9.3 Sampling the Posterior

The following code will use the prior and the likelihood defined above to set up the inverse problem and start sampling:

```
QUESO::InfiniteDimensionalMCMCSamplerOptions opts(env, "");

// Set the number of iterations to do
opts.m_num_iters = 1000;

// Set the frequency with which we save samples
opts.m_save_freq = 10;

// Set the RWMH step size
opts.m_rwmh_step = 0.1;

// Construct the sampler, and set the name of the output file (will only
// write HDF5 files)
QUESO::InfiniteDimensionalMCMCSampler s(env, mu, llhd, &opts);

for (unsigned int i = 0; i < opts.m_num_iters; i++) {
    s.step();
    if (i % 100 == 0) {
        std::cout << "sampler iteration: " << i << std::endl;
        std::cout << "avg acc prob is: " << s.avg_acc_prob() << std::endl;
        std::cout << "l2 norm is: " << s.llhd_val() << std::endl;
    }
}
```

The infinite-dimensional inverse problem work is still considered experimental but should produce meaningful results for a large class of simple problems. Work is ongoing to bring the user interface in line with that of the finite-dimensional inverse problem API.

6 Extensibility

QUESO is written in C++. The choice of the language inspired design decisions that the user can take advantage of. One such benefit of having a well-defined inverse problem setup and workflow is that the user is offered the freedom to extend many of the abstract base classes in QUESO. A good example of this we have seen already is the specification of the likelihood distribution by subclassing `BaseScalarFunction`.

In this section we will take this a step further and show how to extend some of the other classes in QUESO to define a custom prior measure. All of the classes we deal with here have their relationships with other objects discussed in Sect. 7.2.

6.1 Custom Priors

We will look at one of the existing measures in QUESO to get a feel for a how a measure QUESO is built. Take, for example, the Gamma distribution.

In QUESO, the user will interact with a Gamma measure by instantiating a `GammaVectorRV` class. This object has two main members that QUESO is

interested in, an object representing a probability distribution function and an object called a “realizer” through which random variates are drawn. These classes are called `GammaJointPdf` and `GammaVectorRealizer`, respectively.

The user does not, usually, need to interact with the probability distribution function or the realizer; these are objects that QUESO will utilize during the execution of the Markov chain Monte Carlo procedure.

6.1.1 PDF Objects

Probability distribution functions are represented by C++ objects. If the user wishes to create a custom prior measure, for example, then they will also have to implement a probability distribution class. The probability distribution class must derive from the `BaseJointPdf`. The `BaseJoinPdf` class subclasses from `BaseScalarFunction`, as we have seen before, and therefore any probability distribution class must implement the `lnValue` and `actualValue` methods. These methods have exactly the same purpose as when the user defines their likelihood. That is, `lnValue` returns the log of the probability distribution function evaluated at `domainVector`, and `actualValue` returns the actual value of the distribution evaluated at `domainVector`.

`BaseJointPdf` has an extra method called `computeLogOfNomalizationFactor` and so this must also be implemented. This method computes the logarithm of the normalizing constant of the probability distribution. If it is known analytically, the user can implement it here. For many distributions, this is not known analytically. In these circumstances one can use the `numSamples` argument to approximate this quantity using samples from the distribution instead. A basic algorithm for computing the log of the normalizing constant from samples is implemented in the `commonComputeLogOfNormalizationFactor` method of `BaseJointPdf`. Indeed the computation of the log of the normalization constant for the Gamma distribution is handed off to this method:

```
template<class V, class M>
double
GammaJointPdf<V, M>::computeLogOfNormalizationFactor(
    unsigned int numSamples,
    bool updateFactorInternally) const
{
    value =
        BaseJointPdf<V,M>::commonComputeLogOfNormalizationFactor(
            numSamples, updateFactorInternally);
    return value;
}
```

Notice that when we defined a custom likelihood object, we only subclassed `BaseScalarFunction` and not `BaseJointPdf`. This is because for most applications, the likelihood is not a probability distribution since it does not integrate to 1. Furthermore, it avoids needing to implement the `computeLogOfNormalizationFactor` method. This is because the normalizing constant is usually not known analytically, and computing it by samples is often intractable for large engineering problems. Note, however, that

the normalizing constant for the likelihood is not needed since MCMC methods do not require knowledge of any normalizing constant in order to draw random samples. This is crystallized in the following section.

6.1.2 Realizer Objects

Realizer objects are objects that QUESO interacts with to draw random samples from the appropriate distribution. A realizer object must subclass `BaseVectorRealizer` and must therefore implement the `realization(V & nextValues) const` method. This method fills the `nextValues` vector with a random draw from the associated distribution. The size of the vector `nextValues` is equal to the dimension of the state space on which the measure is defined.

In the case of the Gamma distribution, QUESO falls back to GSL to draw samples that are Gamma distributed.

A warning to the user: it is possible to define a measure on a space that is improper. In this case drawing realizations from the associated realizer object produces meaningless results.

6.1.3 Random Variable Objects

Random variable objects, named `*VectorRV` in QUESO, are encapsulating objects that hold references to the associated probability distribution function object and the associated realizer object. A random variable object must subclass `BaseVectorRV` which implements the getter methods `realizer()` and `pdf()` that return references to the realizer object and PDF object, respectively.

The user never has to deal with constructing the PDF object or the realizer object explicitly. Construction of these objects is handled by the random variable object's constructor.

7 The QUESO Design and Implementation

7.1 Software Engineering

High-quality software is essential for developing, analyzing, and scaling up new UQ algorithmic ideas involving complex simulation codes running on HPC platforms. QUESO helps researchers to bootstrap statistical inverse problems for large-scale models widely seen in the physics and engineering domains in parallel compute environments. With ongoing effort to enhance the API in terms of extensibility (see Sect. 10.3), in the future it will be possible to quickly prototype new algorithms in a sophisticated computation environment, rather than first coding and testing them with a scripting language and only then recoding in a C++/MPI environment. QUESO also allows researchers to more naturally translate the mathematical language present in algorithms to a concrete program in the library and to concentrate their efforts on algorithmic, load balancing, and parallel scalability issues.

We utilize various community tools to manage the QUESO development cycle. Source code traceability is provided via Git, and the GNU Autotools suite is used to

provide a portable, flexible build system, with the standard GNU package pattern: `configure`; `make`; `make check`; `make install` steps. We also utilize GitHub for project management, which provides a web-based mechanism to manage releases, milestone developments, issues, bugs, and source code changes. In case the build system or application development processes change, please consult the website (<http://libqueso.com>) for a detailed and up-to-date guide on how to build and install QUESO.

As of the latest QUESO release, 0.53.0, the library is comprised of approximately 73,000 source lines of code, with the vast majority of this instantiated across approximately 200 C/C++ source files and headers. At a minimum, QUESO compilation requires MPI and linkage against two external libraries: boost and GSL. QUESO also has several optional dependencies that enable additional functionality: Teuchos, GRVY, HDF5. The optional infinite-dimensional capabilities of QUESO in particular require libMesh and HDF5.

We employ an active regression testing, with approximately thirty regression tests, and can test latest GitHub builds using Travis-CI in order to have a continuous integration analysis of source code commits.

Contributing QUESO has been made easy with the recent explosion in popularity of GitHub. We employ the feature branch model by Driessen (<http://nvie.com/posts/a-successful-git-branching-model>), and further instructions for contribution to QUESO can be found by mirroring some of the other contributions we have merged (<https://github.com/libqueso/queso/issues>).

7.2 QUESO Internals

In this subsection, we show and discuss several of the inheritance diagrams behind the principle objects in the QUESO library. This is in order to:

- Document the QUESO internal structure
- Provide context for leveraging the existing QUESO objects in extending the library (as in Sect. 6).

This subsection addresses some of the C++ objects for the finite-dimensional Bayesian inverse problem. Objects associated with the infinite-dimensional problem exist and are available on the online documentation, but are not discussed here since development work to get the finite- and infinite-dimensinoal APIs consistent with each other is ongoing.

`BaseScalarFunction` is a templated base class for handling generic scalar functions. This provides a high-level interface and member functions for the QUESO generic class, `BaseJointPDF`, which is discussed below.

`BaseJointPdf` is a templated (base) class for handling joint PDFs. For example, Fig. 54.6 shows the inheritance of the Gamma joint PDF class, which is a derived class from the `BaseScalarFunction` class. QUESO presently has several provided joint PDFs for a wide variety of statistical distributions, including: `InvLogitGaussianJointPdf`, `ConcatenatedJointPdf`,

Fig. 54.6 Class Reference for the Gamma Joint PDF

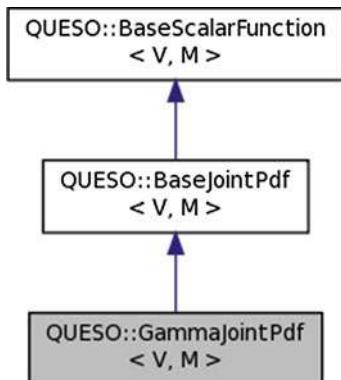
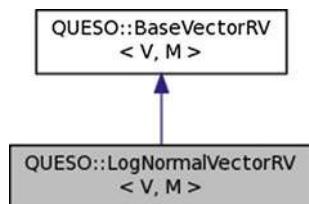


Fig. 54.7 Class reference for the LogNormalVectorRV



`GaussianJointPdf`, `BaseJointPdf`, `BayesianJointPdf`, `LogNormalJointPdf`, `PoweredJointPdf`, `BetaJointPdf`, `GammaJointPdf`, `InverseGammaJointPdf`, `WignerJointPdf`, `GenericJointPdf`, `UniformJointPdf`, `JeffreysJointPdf`, `GenericScalarFunction`, and `ConstantScalarFunction`. However, implementing a new distribution is intended to be straightforward and is detailed in Sect. 6.

Another useful internal QUESO object, `BaseVectorRV`, is a templated base class for handling vector random variables. For example, Fig. 54.7 shows the inheritance diagram of the `LogNormalVectorRV` class, which is a class that contains member functions and associated utilities to provide a random vector of draws from a `LogNormal` distribution.

Presently included in QUESO are the following: `GaussianVectorRV`, `GenericVectorRV`, `BetaVectorRV`, `GammaVectorRV`, `InverseGammaVectorRV`, `InvLogitGaussianVectorRV`, `JeffreysVectorRV`, `LogNormalVectorRV`, `UniformVectorRV`, and `WignerVectorRV`. In other words, nearly all canonical distributions from classical statistics are already available in the library. However, as stated above, QUESO is designed with extensibility in mind, and the user can implement any `*VectorRV` by deriving from the `BaseVectorRV` class. In principle, this permits a series of draws from any distribution.

Another important base class contained within QUESO is the realizer object, `BaseVectorRealizer`. A realizer is an object that, simply put, contains a `realization()` operation that returns a sample of a random variable. `BaseVectorRealizer` is therefore an abstract base class that provides the necessary interface for sampling from random variables. As before, the realizer

object contains most of the common statistical distributions. It also contains a sequence realizer class for storing samples of a MH algorithm.

8 Algorithms

8.1 DRAM

A simple Metropolis-Hastings sampling algorithm [4] can be improved by adding both “Delayed Rejection” [18–21] and “Adaptive Metropolis”. Taken together, these form the “DRAM” algorithm, which is available in QUESO. In particular, the QUESO implements the DRAM algorithm of Haario, Laine, Mira, and Saksman [7].

A “vanilla” Metropolis-Hastings sampler involves a proposal at each step, and accepts or rejects this proposal based on the ratio between proposal and prior likelihoods. Typically, the proposal is drawn from some fixed distribution, such as a Gaussian distribution, with fixed covariance and a mean centered at the value of the current state of the chain. However, this has several deficiencies. Should the proposal variance be set too high, many proposals will be rejected. This is undesirable, as it increases the auto-correlation of the chain. Furthermore, should the target distribution deviate greatly from the proposal distribution, the proposal will not match the local shape of the distribution, resulting in poor sampling.

Delayed rejection attempts to circumvents these issues. Before rejecting a sample, a series of back-up proposals each with successively smaller jumps in state space are pushed through the Metropolis-Hastings acceptance probability rejection. They are tested in order of decreasing jump size, and if one of them is accepted, the sampler continues. If they are all rejected, the sampler rejects the sample and starts again.

Conversely, when the proposal variance is too small to efficiently sample the target distribution, the sampler will randomly walk through regions of higher likelihood in the posterior distribution, without efficiently sampling the tails. This results in too high an acceptance rate.

In order to mitigate this, Adaptive Metropolis sampling continuously adapts the proposal covariance. This is accomplished by using the covariance of the history of the Markov chain as the proposal covariance matrix of the Gaussian proposal distribution instead of the arbitrary proposal covariance imposed at the start. Adapting the proposal to match the posterior covariance structure results in a better chain performance than a static proposal covariance.

8.2 Multilevel

Multilevel Monte Carlo [6] is an algorithm available in QUESO that attempts to sample probability distributions with multiple modes. Sampling multi-modal distributions is a heavily researched topic. The way multilevel Monte Carlo attempts to solve the problem of metastability in Markov chains is by “heating up” the posterior distribution to flatten out some of the modes, allowing a Markov chain to

sample the flattened distribution and then “cooling down” the posterior distribution before doing a final sampling run. The idea is identical to that of simulated tempering or simulated annealing, except that the multilevel algorithm allows for convenient and efficient computation of the posterior normalizing constant. This constant is usually intractable to compute but is essential for Bayesian model selection purposes.

8.3 Preconditioned Crank-Nicolson

The preconditioned Crank-Nicolson proposal [15] is used by QUESO for solving infinite-dimensional Bayesian inverse problems (Sect. 5.9). This particular form of proposal is typical for sampling on formally infinite-dimensional spaces since the Metropolis-Hastings acceptance probability remains unchanged when the state undergoes mesh refinement, a popular technique in large-scale engineering models involving the solution of partial differential equations by finite element methods.

9 Input File

Here we provide some of the default input file options QUESO recognizes. For detailed descriptions of the behavior of each option and how they interact with other options, consult the online QUESO documentation. For example, for the description of each DRAM option, consult the documentation for the `MhOptionsValues` object. For the description of each `FullEnvironment` option, see the documentation for the `EnvOptionsValues` object. The documentation for these is available at <http://libqueso.com> (Tables 54.1, 54.2, 54.3, 54.4, and 54.5).

Table 54.1 Input file options for a QUESO environment

Option name	Default	Description
<code>env_help</code>		Produces help message for environment class
<code>env_numSubEnvironments</code>	1	Number of subenvironments
<code>env_subDisplayFileName</code>	"."	Output filename for sub-screen writing
<code>env_subDisplayAllowAll</code>	0	Allows all subenvironments to write to output file
<code>env_subDisplayAllowedSet</code>	" "	Subenvironments that will write to output file
<code>env_displayVerbosity</code>	0	Sets verbosity
<code>env_syncVerbosity</code>	0	Sets synchronized verbosity
<code>env_seed</code>	0	Set seed

Table 54.2 Input file options for a QUESO statistical inverse problem

Option name	Default	Description
<code>ip_help</code>		Produces help message for statistical inverse problem
<code>ip_comPUTEsolution</code>	1	Computes solution process
<code>ip_dataOutputFileName</code>	"."	Name of data output file
<code>ip_dataOutputAllowedSet</code>	" "	Subenvironments that will write to data output file

Table 54.3 Input file options for a QUESO DRAM solver

Option name	Default value
<code>mh_dataOutputFileName</code>	“.”
<code>mh_dataOutputAllowAll</code>	0
<code>mh_initialPositionDataInputFileName</code>	“.”
<code>mh_initialPositionDataInputFileType</code>	“m”
<code>mh_initialProposalCovMatrixDataInputFileName</code>	“.”
<code>mh_initialProposalCovMatrixDataInputFileType</code>	“m”
<code>mh_rawChainDataInputFileName</code>	“.”
<code>mh_rawChainDataInputFileType</code>	“m”
<code>mh_rawChainSize</code>	100
<code>mh_rawChainGenerateExtra</code>	0
<code>mh_rawChainDisplayPeriod</code>	500
<code>mh_rawChainMeasureRunTimes</code>	1
<code>mh_rawChainDataOutputPeriod</code>	0
<code>mh_rawChainDataOutputFileName</code>	“.”
<code>mh_rawChainDataOutputFileType</code>	“m”
<code>mh_rawChainDataOutputAllowAll</code>	0
<code>mh_filteredChainGenerate</code>	0
<code>mh_filteredChainDiscardedPortion</code>	0.
<code>mh_filteredChainLag</code>	1
<code>mh_filteredChainDataOutputFileName</code>	“.”
<code>mh_filteredChainDataOutputFileType</code>	“m”
<code>mh_filteredChainDataOutputAllowAll</code>	0
<code>mh_displayCandidates</code>	0
<code>mh_putOutOfBoundsInChain</code>	1
<code>mh_tkUseLocalHessian</code>	0
<code>mh_tkUseNewtonComponent</code>	1
<code>mh_drMaxNumExtraStages</code>	0
<code>mh_drDuringAmNonAdaptiveInt</code>	1
<code>mh_amKeepInitialMatrix</code>	0
<code>mh_amInitialNonAdaptInterval</code>	0
<code>mh_amAdaptInterval</code>	0
<code>mh_amAdaptedMatricesDataOutputPeriod</code>	0
<code>mh_amAdaptedMatricesDataOutputFileName</code>	“.”
<code>mh_amAdaptedMatricesDataOutputFileType</code>	“m”
<code>mh_amAdaptedMatricesDataOutputAllowAll</code>	0
<code>mh_amEta</code>	1.
<code>mh_amEpsilon</code>	1×10^{-5}
<code>mh_enableBrooksGelmanConvMonitor</code>	0
<code>mh_BrooksGelmanLag</code>	100

Table 54.4 Input file options for a QUESO multilevel solver

Option name	Default value
ml_restartOutput_levelPeriod	0
ml_restartOutput_baseNameForFiles	“.”
ml_restartOutput_fileType	“m”
ml_restartInput_baseNameForFiles	“.”
ml_restartInput_fileType	“m”
ml_stopAtEnd	0
ml_dataOutputFileName	“.”
ml_dataOutputAllowAll	0
ml_loadBalanceAlgorithmId	2
ml_loadBalanceThreshold	1.0
ml_minEffectiveSizeRatio	0.85
ml_maxEffectiveSizeRatio	0.91
ml_scaleCovMatrix	1
ml_minRejectionRate	0.50
ml_maxRejectionRate	0.75
ml_covRejectionRate	0.25
ml_minAcceptableEta	0.
ml_totallyMute	1
ml_initialPositionDataInputFileName	“.”
ml_initialPositionDataInputFileType	“m”
ml_initialProposalCovMatrixDataInputFileName	“.”
ml_initialProposalCovMatrixDataInputFileType	“m”
ml_rawChainDataInputFileName	“.”
ml_rawChainDataInputFileType	“m”
ml_rawChainSize	100
ml_rawChainGenerateExtra	0
ml_rawChainDisplayPeriod	500
ml_rawChainMeasureRunTimes	1
ml_rawChainDataOutputPeriod	0
ml_rawChainDataOutputFileName	“.”
ml_rawChainDataOutputFileType	“m”
ml_rawChainDataOutputAllowAll	0
ml_filteredChainGenerate	0
ml_filteredChainDiscardedPortion	0.
ml_filteredChainLag	1
ml_filteredChainDataOutputFileName	“.”
ml_filteredChainDataOutputFileType	“m”
ml_filteredChainDataOutputAllowAll	0
ml_displayCandidates	0
ml_putOutOfBoundsInChain	1
ml_tkUseLocalHessian	0
ml_tkUseNewtonComponent	1

(continued)

Table 54.4 (continued)

ml_drMaxNumExtraStages	0
ml_drScalesForExtraStages	0
ml_drDuringAmNonAdaptiveInt	1
ml_amKeepInitialMatrix	0
ml_amInitialNonAdaptInterval	0
ml_amAdaptInterval	0
ml_amAdaptedMatricesDataOutputPeriod	0
ml_amAdaptedMatricesDataOutputFileName	“.”
ml_amAdaptedMatricesDataOutputFileType	“m”
ml_amAdaptedMatricesDataOutputAllowAll	0
ml_amEta	1.
ml_amEpsilon	1.e-5

Table 54.5 Input file options for a QUESO pCN solver

Option name	Default value
infcmc_dataOutputDirName	“chain”
infcmc_dataOutputFileName	“out.h5”
infcmc_num_iters	1000
infcmc_save_freq	1
infcmc_rwmh_step	1e-2

10 Conclusions

We conclude this chapter with a discussion of several of the areas the QUESO development team is investing time into implementing, extending, and improving along with some of the newest features recently made available in v0.53.0. Previously, we have covered only the basics of how to interact with QUESO and to provide a resource that is accessible and can be used to bootstrap a user’s statistical inverse problem quickly and efficiently. With this in mind, there are still many areas in which QUESO can improve to become more user friendly, consistent, and extensible. In what follows, we discuss some major areas of development that would likely encourage widespread adoption of QUESO in the computational applied mathematics and engineering community.

10.1 QUESO-Provided Likelihoods

In many large-scale physics and engineering-based experimental settings, it is often the case that observations of a physical quantity are performed several times. These observations are then averaged to homogenize the effect of experimental observation error. In the case of independent experimental errors, this average will be normally distributed. Therefore, a reasonable choice for a likelihood in many applications would be a Gaussian.

At present, the user must derive from `BaseScalarFunction` and implement `lnValue` explicitly. This is a tedious task if all that is needed is the standard Gaussian error in the Euclidean 2-norm:

$$\mathbb{P}(y|\theta) = Z \exp\left(\frac{1}{2} (\mathcal{G}(\theta) - y)^\top \Sigma^{-1} (\mathcal{G}(\theta) - y)\right), \quad (54.11)$$

where Z is a normalizing constant.

A recently released and much leaner approach is to provide an abstract base class of `BaseScalarFunction` called `BaseGaussianLikelihood` with a pure virtual method called `evaluateModel` that asks for the output of the map \mathcal{G} at the point `domainVector`. Equipped with an implementation of `lnValue` that computes the log of (54.11), the user would only need to provide Σ and y , which can be passed in from the constructor. An example follows:

Here the user would pass an instance of `Likelihood` to `StatisticalInverseProblem`, as per usual.

Extensions of this idea are also available, where one wishes to treat Σ as a hyper-parameter to be sampled along with θ in so-called “hierarchical Bayesian” methods. The design described above is easily applied to this situation.

Ongoing work is being invested in developing other pre-made likelihood objects representing other likelihood forms that are commonly used.

10.2 Emulators

The two main forms of emulation used in the statistical modeling community are Gaussian processes and generalized polynomial chaos. These are both important methods in statistical inference as they can considerably reduce the computational cost of computing the posterior.

Gaussian process emulators, similar to the ready-made Gaussian likelihoods discussed in the previous section, are also a form of baked likelihood, but where the user is not required to implement a method returning the output of \mathcal{G} . For Gaussian process emulators, the user would only need to instantiate an emulator with a specific dataset and observational error covariance matrix. The rest of the statistical application the user writes is identical to any other statistical application and the output (samples) is processed as per usual.

Generalized polynomial chaos methods require different algorithms for solution, since no Markov chain Monte Carlo is done. This type of emulator is not currently on the QUESO development road map for the near future, but contributions in the area are more than welcome.

As of QUESO v0.53.0, the only supported emulator is a linear interpolation of model output values. Interested users should consult the documentation and, in particular, the example called `4d_interp.C`.

10.3 API Considerations

As mentioned in the infinite-dimensional example, the infinite-dimensional and finite-dimensional APIs are not aligned. Although the user interacts with only one of these APIs at any given time, an aligned API structure exposes the opportunity for algorithms designed on function space, which tend to be more robust algorithms, to be used in the finite-dimensional setting. Moreover, an aligned API eases the maintenance, documentation, and testing burden.

Currently, there are only two (finite-dimensional) algorithms the user can use, DRAM and multilevel. At present, there is no organized structure that Markov chains (`MetropolisHastingsSG` objects) inherit from, meaning that there is a significant hurdle involved in bootstrapping one’s own MCMC algorithm for the purposes of testing and research. Just as above, a consistent class hierarchy for MCMC algorithms would ease the burden for software maintenance.

A rather cumbersome design choice made early on in the development of QUESO was the hot-swappability of vector and matrix implementations for all of QUESO's classes. The net result of this is that any QUESO class that involves an operation with a vector or a matrix is templated around the type of that vector or matrix. This was done to ensure that optimized code could be generated that dealt with the specifics of each vector and matrix library. Assuming that, in high-performance uncertainty quantification, likelihood evaluations are the dominating cost of Markov chain Monte Carlo sampling, one need not encumber the QUESO API with such templates. Furthermore, a hierarchical class structure for vector and matrix types exists in QUESO and therefore necessitates the run-time overhead of virtual table lookups. Efforts are currently ongoing to enrich the vector and matrix class hierarchy in QUESO sufficiently such that the particulars of vector and matrix implementations still remain opaque but significantly shorten unnecessarily long class names with a negligibly small impact on run-time performance. This enrichment would also allow QUESO to pick a high-tuned vector/matrix implementation at configure time for high-performance problems in exascale compute environments. For example, QUESO's build system could default to using PETSc vectors optimized for multi-core architectures, while the user need not deal explicitly with MPI calls. All parallel logic would be handled under the hood. This offers a pleasing software experience while maintaining performance.

Python has become a very popular environment to do post-processing and visualization in multi-core HPC systems. A desirable feature to have in QUESO would be the automatic generation of python bindings. This would offer the possibility to do uncertainty quantification in statistical inverse problems as a quick-turnaround experiment for cheap forward models in an interpreted language environment. This implementation will likely leverage the Simplified Wrapper and Interface Generator (SWIG) which is not limited to Python and can provide interfaces to many modern programming languages, such as Perl, Python, Ruby, and Tcl.

10.4 Exascale

Uncertainty quantification has pushed the limits of current computational power by requiring many evaluations of large-scale engineering systems described by partial differential equations. Utilizing more information about the system can significantly increase the performance of MCMC algorithms [22–24]. In particular QUESO does not currently implement MCMC algorithms that use gradient or Hessian information to construct proposal distributions. However, the design of the API for the pure virtual methods in `BaseScalarFunction` allows this information to be passed to QUESO easily, in the form of a pointer `V * gradVector`. For more details on the parameters passed to the `lnValue` function, the reader is directed to the QUESO documentation which be found online here: <http://libqueso.com>.

References

1. Prudencio, E.E., Schulz, K.W.: Euro-Par 2011: Parallel Processing Workshops, pp. 398–407. Springer (2012). http://dx.doi.org/10.1007/978-3-642-29737-3_44
2. Estacio-Hiroms, K.C., Prudencio, E.E.: Quantification of Uncertainty for Estimation, Simulation, and Optimization (QUESO), User's Manual (2008). Unpublished, <http://www.libqueso.com/>.
3. Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems, vol. 160. Springer, New York (2005)
4. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: *J. Chem. Phys.* **21**(6), 1087 (1953). doi:10.1063/1.1699114. <http://link.aip.org/link/JCPA6/v21/i6/p1087/s1&Agg=doi>
5. Hastings, W.K.: *Biometrika* **57**(1), 97 (1970). doi:10.1093/biomet/57.1.97. <http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/57.1.97>
6. Cheung, S.H., Prudencio, E.E.: *Int. J. Uncertain. Quantif.* **2**(3), p. 215–237 (2012)
7. Haario, H., Laine, M., Mira, A., Saksman, E.: *Stat. Comput.* **16**(4), 339 (2006). doi:10.1007/s11222-006-9438-0. <http://link.springer.com/10.1007/s11222-006-9438-0>
8. Patil, A., Huard, D., Fonnesbeck, C.J.: *J. Stat. Softw.* **35**(4), p. 1–81. (2010)
9. Hunter, J.D.: *Comput. Sci. Eng.* **9**(3), 90 (2007)
10. Foreman-Mackey, D., Hogg, D.W., Lang, D., Goodman, J.: *Publ. Astron. Soc. Pac.* **125**(925), 306 (2013)
11. Core Team, R.: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2013). ISBN 3-900051-07-0, <http://www.R-project.org/>.
12. Stan Development Team: Stan: a c++ library for probability and sampling, version 2.5.0 (2014). <http://mc-stan.org/>
13. Lunn, D.J., Thomas, A., Best, N., Spiegelhalter, D.: *Stat. Comput.* **10**(4), 325 (2000)
14. Adams, B.M., Hart, W.E., Eldred, M.S., Dunlavy, D.M., Hough, P.D., Giunta, A.A., Griffin, J.D., Martinez-Canales, M.L., Watson, J.P., Kolda, T.G.: DAKOTA, a Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 4.0 Uers's Manual (2006)
15. Cotter, S.L., Roberts, G.O., Stuart, A.M., White, D.: <http://arxiv.org/abs/1202.0709> (2012)
16. Bogachev, V.I.: Gaussian Measures. American Mathematical Society, Providence (1998)
17. Lifshits, M.A.: Gaussian Random Functions. Springer Netherlands (1995)
18. Mira, A.: *LIX*(3–4), 231 (2001)
19. Mira, A.: *Stat. Sci.* **16**(4), 340 (2002). doi:10.1214/ss/1015346319. <http://projecteuclid.org/euclid.ss/1015346319>
20. Tierney, L., Mira, A.: *Stat. Med.* **18**, 2507 (1999)
21. Green, P.J., Mira, A.: *Biometrika* **88**, 1035 (2001)
22. Girolami, M., Calderhead, B.: *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **73**(2), 123 (2011). doi:10.1111/j.1467-9868.2010.00765.x. <http://doi.wiley.com/10.1111/j.1467-9868.2010.00765.x>
23. Martin, J., Wilcox, L., Burstedde, C., Ghattas, O.: *SIAM J. Sci. Comput.* **34**(3), 1460 (2012)
24. Bui-thanh, T., Ghattas, O., Higdon, D.: *SIAM J. Sci. Comput.* **34**(6), 2837 (2012)

Gaussian Process-Based Sensitivity Analysis and Bayesian Model Calibration with GPMSA

55

James Gattiker, Kary Myers, Brian J. Williams, Dave Higdon, Marcos Carzolio, and Andrew Hoegh

Abstract

The Gaussian Process Models for Simulation Analysis (GPMSA) package is a set of functions written in the Matlab programming language aimed at *emulating* a computer model of a system being studied, *calibrating* this computer model to observations of the system, and giving *predictions* of the expected system response. Collectively, these capabilities comprise uncertainty quantification (UQ) in model-supported inference.

This chapter will first discuss some background and motivation for the GPMSA code, then demonstrate the code's function interfaces in the context of a series of illustrative example problems.

Keywords

Bayesian analysis • Design and analysis of computer experiments • Gaussian process • Markov chain Monte Carlo • Statistical analysis of computer models

Contents

1	Introduction	1868
2	Summary of the GPMSA Statistical Model	1871
3	Example 1: Ball Drop with Varying Radii	1873
3.1	How We Use the Gaussian Process Model	1874

J. Gattiker (✉) • K. Myers • B.J. Williams

Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM, USA
e-mail: gatt@lanl.gov; kary@lanl.gov; brianw@lanl.gov

D. Higdon

Social Decision Analytics Laboratory, Virginia Bioinformatics Institute Virginia Tech,
Arlington, VA, USA
e-mail: dhigdon@vbi.vt.edu

M. Carzolio • A. Hoegh

Department of Statistics, Virginia Tech, Blacksburg, VA, USA
e-mail: cmarcos8@vt.edu; ahoegh@vt.edu

3.2	Preparing the Data for Use by GPMSA	1876
3.3	Model Initialization and MCMC	1882
3.4	Examining the Estimated Parameters' Posterior Distribution	1885
3.5	Assessing Emulator Adequacy	1887
3.6	System Predictions	1891
3.7	The <code>pout</code> Object	1892
4	Example 2: Ball Drop with Different Radii and Densities	1892
4.1	How We Use the Gaussian Process Model	1893
4.2	Preparing the Data for Use by GPMSA	1894
4.3	Model Initialization and MCMC	1898
4.4	Examining the Estimated Parameter's Posterior Distribution	1899
4.5	System Predictions	1900
5	Example 3: Ball Drop Exploiting Kronecker-Separable Design	1901
6	Example 4: Specifying Priors on C and g	1904
6.1	Prior Specification for g and C	1904
7	Example 5: Inferring the Type of Ball Dropped	1905
7.1	Specifying a Categorical Ball-Type Parameter in GPMSA	1906
8	Conclusion	1907
	References	1907

1 Introduction

We begin with an overview of how a scientific computer model is used in inference and how UQ is performed. There is a system S whose behavior we would like to understand and predict. To study that system, observations of its state O_s are made, where these observations also have some error. To understand the system, a model for its behavior is specified that estimates the system response as a function of some conditions that influence the system model. For an example, consider studying gravity. We might notice that the position x of a dropped weight seems to be a quadratic function of time t and so postulate the system model $x = \alpha t^2$. To determine α , where α in this case corresponds to the constant of gravity, observations of the system would be taken, measuring x at time t . Observations x have Gaussian uncertainty.

Given these observations, the values of α that are consistent can be determined. Performing this evaluation in a Bayesian framework, a likelihood function defines the probability density of the observations given a value for α (appropriate for the system model). Prior probabilities express information known about the value of α before the experiment. Now, the posterior probability distribution for α can be evaluated in principle, and in simple cases, this can be performed analytically. Given the posterior for α , predictions for x at unobserved t can be made, with uncertainty. This is the operation of UQ in *calibration*, or finding posterior probability distributions of unknown settings of system model parameters.

When the system model is complicated, as with models that require computer evaluation, this procedure cannot be performed analytically. Instead, in the Bayesian setting, Monte Carlo methods are used to recover samples from the posterior distribution of α . These samples of α can be similarly used for UQ in *prediction*,

propagating them through the model to get samples of the distribution of x at specified, perhaps unobserved, values of t .

UQ becomes more complex in the case where α represents several variables instead of single scalar. When α is low dimensional, say ≤ 5 , it is possible to sample the joint posterior distribution at regular intervals, but this becomes untenable in higher dimensions. The solution in this case is to use Markov chain Monte Carlo (MCMC), a technique for generating samples of the posterior distribution. Using MCMC introduces some additional complications, which are described in a rich literature on the topic. Two typical issues are tuning the MCMC sample step size in a Metropolis-Hastings framework so that the process is efficient and in the possible need for variable transformations to avoid issues that arise with highly correlated parameters.

Scaling this up, we come to the case where the system model is a complicated computer simulation that takes hours or days to evaluate. This is not at all unusual in many areas of scientific study. Now the Monte Carlo procedures cannot be performed as needed, as hundreds or perhaps thousands of evaluations of the model (i.e., runs of the computer simulation) are required to study α . In this case, a statistical model, specifically a Gaussian process (GP) model, is used to *emulate* the system model. The strategy is to use a number of instances of the system model's input-output relationship to construct an approximation to that relationship that is fast to evaluate and accurate over the domain of interest and delivers defensible prediction uncertainty. This presumes that the system model's total behavior over the domain of interest can be well characterized by a reasonable (in evaluation time and dataset size) number of samples. Once the emulator has been constructed and its accuracy validated, it is used to provide surrogate fast system model evaluations. The main positive features of a Gaussian process model as used in GPMSA are an appropriate expressiveness of two aspects of uncertainty associated with an emulator – the uncertainty in interpolation and the uncertainty in prediction – and the ability of GP models to function in high-dimensional spaces.

It was recognized some time ago that a key feature in using complex system models in inference is an appreciation that the estimated system model is not the same as the true system model and must be expected to present a biased prediction of the system. This is often called “structural error” in the model and implies the incorporation of a *discrepancy* representing the difference in response between the system and the model. In the case of GPMSA, this discrepancy model is an additive GP. In the Bayesian framework, priors say that the discrepancy should be small if it can be, but that it can supply bias when the system model is consistently biased compared to system observations.

As a final complication (optional to GPMSA), complex system models typically have a multivariate nature, as do the system observations. In the example of studying gravity, observations may be the full curves $x(t)$. There is a mechanism to represent the multivariate system response as a linear combination of a smaller number of orthogonal basis functions, in practice typically the first few principal components constructed on examples obtained from the system model. This has several implications. A linear basis such as principal components is a consilient

dimension reduction method. It can be seen as a form of covariance definition between the raw outputs, appropriately limiting the potential expressiveness of the model output to an appropriate number of degrees of freedom and thus also limiting the complexity of the emulation problem. The dimension reduction also simplifies the computations associated with the GP model, by defining orthogonal responses that can be taken to be independent. In the discrepancy GP, it is even more appropriate to reduce the flexibility of the function with a linear basis, constraining the discrepancy response in a manner that is different from the representation of the system model response. This is typically accomplished in practice with an approximate smooth kernel basis approach.

Addressing all of these complications from the basic analytical UQ framework introduces additional parameters beyond the system model parameters that describe the GP emulator. Their value is posed in the same Bayesian framework as the system model, so their uncertainty is included in the UQ analysis and in subsequent predictions and model evaluation. GPMSA presumes that the datasets are scaled and normalized, which allowed the assignment of default priors for these additional parameters. However, it is also important to realize the function and relevance of these parameters in validating the emulator, as they are often diagnostic of emulator performance. This requires an understanding of the underlying statistical framework.

Ultimately, the joint posterior distribution of all the parameters, system model, and GP is sampled using MCMC. The core functions of GPMSA are the setting up of the emulator and model calibration as a Bayesian inference problem, the MCMC sampling of these parameters, and making predictions of system response, with uncertainty. These predictions can then be used for exploring the expected response with conditional visualizations, sensitivity analysis of system model response to parameters (a GPMSA function), and evaluation of uncertainty.

An analytical description of GPMSA is given in the next section, in notation more related to the GPMSA code compared to the detailed description in Higdon et al. [3].

GPMSA has been implemented in the Matlab environment for two main reasons. First, setting up a GPMSA analysis typically involves a significant amount of exploratory data analysis (EDA), customization according to particulars of the problem, and custom statistical and graphical analysis of the results; Matlab provides the necessary full-featured EDA environment. Second, the core GPMSA computations are based in matrix operations and linear algebra, and Matlab provides an efficient computational infrastructure supporting these types of computations.

In further discussion, *observations* will be used for observations of the system under study and *simulator* or simulation for the system model. The observations were collected in *experiments*, which included taking *measurements* of some aspects of the experimental setup. The relevant settings for the simulator are referred to as *simulator inputs* or *parameters*, and results of simulation, corresponding to system observations, are *simulator output*, or *response*.

2 Summary of the GPMSA Statistical Model

In this section, the GPMSA model, described in Higdon et al. [3], is defined using notation that more closely matches the variable names used in GPMSA code.

There are n physical (observational) experiments. From the i th physical experiment at p inputs $\mathbf{x}_i^{\text{obs}} = (x_{i1}^{\text{obs}}, \dots, x_{ip}^{\text{obs}})$, the observation $\mathbf{y}_i^{\text{obs}}(\mathbf{x}_i^{\text{obs}})$ (an $n_{y_i^{\text{obs}}} \times 1$ vector) is modeled by

$$\mathbf{y}_i^{\text{obs}}(\mathbf{x}_i^{\text{obs}}) = \eta(\mathbf{x}_i^{\text{obs}}, \boldsymbol{\theta}) + \delta(\mathbf{x}_i^{\text{obs}}) + \mathbf{e}_i^{\text{obs}}, \quad (55.1)$$

where the observation error vector $\mathbf{e}_i^{\text{obs}}$ is modeled by

$$\mathbf{e}_i^{\text{obs}} \sim MVN(\mathbf{0}_{n_{y_i^{\text{obs}}}}, \frac{1}{\lambda_{y_i^{\text{obs}}}^{\text{obs}}} \Sigma_i^{\text{obs}}). \quad (55.2)$$

$\eta(\cdot)$ is an emulator from a simulation code, $\boldsymbol{\theta}$ corresponds to inputs of the parameter, and $\delta(\cdot)$ is a discrepancy from reality.

There are m simulation experiments. From the i th simulation experiment at $p+q$ inputs $\mathbf{x}_i^{\text{sim}} = (x_{i1}^{\text{sim}}, \dots, x_{ip}^{\text{sim}})$ and $\mathbf{t}_i^{\text{sim}} = (t_{i1}^{\text{sim}}, \dots, t_{iq}^{\text{sim}})$, the observation $\mathbf{y}_i^{\text{sim}}(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}})$ (an $n_{y_i^{\text{sim}}} \times 1$ vector) is modeled by

$$\mathbf{y}_i^{\text{sim}}(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) = \eta(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) + \mathbf{e}_i^{\text{sim}}, \quad (55.3)$$

where the error vector $\mathbf{e}_i^{\text{sim}}$ is modeled by $MVN(\mathbf{0}_{n_{y_i^{\text{sim}}}}, \frac{1}{\lambda_{y_i^{\text{sim}}}^{\text{sim}}} \mathbf{I}_m)$ and \mathbf{I}_m is the $m \times m$ identity matrix.

We reexpress $\eta(\mathbf{x}_i, \mathbf{t}_i)$ and $\delta(\mathbf{x}_i)$ by linear combinations of basis functions and approximate them using a subset of the complete set of basis functions. Consequently,

$$\eta(\mathbf{x}_i^{\text{obs}}, \boldsymbol{\theta}) \approx \sum_{j=1}^{p_u} \mathbf{K}_j^{\text{obs}} u_j(\mathbf{x}_i^{\text{obs}}, \boldsymbol{\theta}) \quad (55.4)$$

for p_u basis functions $\mathbf{K}_j^{\text{obs}}$. So the matrix $\mathbf{K}^{\text{obs}} = (\mathbf{K}_1^{\text{obs}} \cdots \mathbf{K}_{p_u}^{\text{obs}})$. Similarly,

$$\delta(\mathbf{x}_i^{\text{obs}}) \approx \sum_{j=1}^{p_v} \mathbf{D}_j^{\text{obs}} v_j(\mathbf{x}_i^{\text{obs}}) \quad (55.5)$$

for p_v basis functions $\mathbf{D}_j^{\text{obs}}$. So the matrix $\mathbf{D}^{\text{obs}} = (\mathbf{D}_1^{\text{obs}} \cdots \mathbf{D}_{p_v}^{\text{obs}})$. We use the right-hand sides of (55.4) and (55.5) in (55.1) so that $\mathbf{e}_i^{\text{obs}}$ also includes their differences from reality. Also,

$$\eta(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) \approx \sum_{j=1}^{p_u} \mathbf{K}_j^{\text{sim}} w_j(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) \quad (55.6)$$

for p_u basis functions $\mathbf{K}_j^{\text{sim}}$, where $w_j(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) = u_j(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}}) + \epsilon_j^{\text{sim,nug}}$. Note that $\frac{1}{\lambda_{\epsilon_j^{\text{sim,nug}}}^{\text{ws}}}$ is the variance of an i.i.d. Normally distributed nugget $\epsilon_j^{\text{sim,nug}}$ with mean zero intended to account for small numerical fluctuations in the simulator. Note that only in fitting is the nugget used. In predicting, the nugget is dropped. So the matrix $\mathbf{K}^{\text{sim}} = (\mathbf{K}_1^{\text{sim}} \cdots \mathbf{K}_{p_u}^{\text{sim}})$. We use the right-hand side of (55.6) in (55.3) so that $\mathbf{e}_i^{\text{sim}}$ also includes its difference from reality.

The $u_j(\mathbf{x}, \mathbf{t})$, $j = 1, \dots, p_u$ are modeled as a GP with mean $\mathbf{0}_n$ and variance covariance matrix $\frac{1}{\lambda_{u_j}^{\text{uz}}} R_j^u$, where

$$R_j^u((\mathbf{x}_i, \mathbf{t}_i), ((\mathbf{x}_l, \mathbf{t}_l)) = \prod_{k=1}^p (\rho_{jk}^u)^{4|x_{ik} - x_{lk}|^2} \prod_{k=1}^q (\rho_{(j+p)k}^u)^{4|t_{ik} - t_{lk}|^2}, \quad (55.7)$$

whose more familiar form is

$$R_j^u((\mathbf{x}_i, \mathbf{t}_i), (\mathbf{x}_l, \mathbf{t}_l)) = \prod_{k=1}^p \exp(-\beta_{jk}^u |x_{ik} - x_{lk}|^2) \prod_{k=1}^q \exp(-\beta_{(j+p)k}^u |x_{ik} - x_{lk}|^2), \quad (55.8)$$

so that $\beta_{jk}^u = -4 \log(\rho_{jk}^u)$ or $\rho_{jk}^u = \exp(-\frac{\beta_{jk}^u}{4})$.

Similarly, the $v_j(\mathbf{x}_i^{\text{obs}})$, $j = 1, \dots, n$ are modeled as a GP with mean $\mathbf{0}_n$ and variance covariance matrix $\frac{1}{\lambda_{v_j^{\text{obs}}}^{\text{vz}}} R_j^v$, where

$$R_j^v(\mathbf{x}_i^{\text{obs}}, \mathbf{x}_l^{\text{obs}}) = \prod_{k=1}^p (\rho_{jk}^v)^{4|x_{ik} - x_{lk}|^2}, \quad (55.9)$$

whose more familiar form is

$$R_j^v(\mathbf{x}_i^{\text{obs}}, \mathbf{x}_l^{\text{obs}}) = \prod_{k=1}^q \exp(-\beta_{jk}^v |x_{ik} - x_{lk}|^2), \quad (55.10)$$

so that $\beta_{jk}^v = -4 \log(\rho_{jk}^v)$ or $\rho_{jk}^v = \exp(-\frac{\beta_{jk}^v}{4})$.

Note that the $\mathbf{x}, \mathbf{t}, \boldsymbol{\theta}$ are normalized to $[0,1]$ and the $\mathbf{y}_i^{\text{obs}}(\mathbf{x}_i^{\text{obs}})$ and $\mathbf{y}_i^{\text{sim}}(\mathbf{x}_i^{\text{sim}}, \mathbf{t}_i^{\text{sim}})$ are normalized to sample mean $\mathbf{0}$ and covariance matrices equal to identity matrices. Consequently, the Σ_i^{obs} for $\mathbf{y}_i^{\text{obs}}(\mathbf{x}_i^{\text{obs}})$ in (55.2) has to be normalized in the same way that the $\mathbf{y}_i^{\text{obs}}(\mathbf{x}_i^{\text{obs}})$ are.

A series of examples will now be presented, showing how to set up, run, and visualize the results using GPMSA. The first example will include tutorial detail; the subsequent examples illustrate alternatives and additional options in less detail. Specifically, this tutorial shows how to:

- set up a problem using GPMSA,
- calibrate model parameters using physical observations,
- make predictions for the physical system at new input settings
- assess the accuracy of the Gaussian process emulator used in the modeling.

3 Example 1: Ball Drop with Varying Radii

For the first example, the trial system will be the trajectory of balls of various radii R dropped from a tower, from heights of 5, 10, 15, and 20 m. Each dropped ball experiment produces vector output in the form of a height-time curve, i.e., a curve showing the time as function of the current height $t = t(h)$ of the ball at a set of measured heights h .

Here, these experiments will not actually be performed, but rather the system observations will be generated from

$$\frac{d^2h}{dt^2} = g - \frac{C}{R} \left(\frac{dh}{dt} \right)^2. \quad (55.11)$$

where g is the acceleration due to gravity, C is the coefficient of drag, and, as introduced above, R is the radius of the ball. For the purposes of the illustration of GPMSA, these will henceforth be referred to as system observations, although knowing their generating model will allow insight into the calibration process and results.

The system model for this example is given by

$$\frac{d^2h}{dt^2} = g - \frac{C}{R} \frac{dh}{dt} \quad (55.12)$$

that will be used as a specified radius R and a coefficient of drag C , producing a height-time curve giving the computed times t at some set of tower heights h (which are not necessarily the same heights used in the field experiments). The model is deterministic, so the same inputs (R, C) will always produce the same height-time curve.

Note that Eqs. 55.11 and 55.12 are differential equations that can be solved for height h as a function of time t and ball radius R . Observations will be the inverse of this: the data have time as the output recorded as a function of height and radius. An optimizer performs this inversion in MATLAB resulting in model output congruent with observations.

Insight into the physical process comes from Eq. 55.12 and the system observations collected at the known initial tower heights, and in analysis the true physical process, presumed unknown, described by Eq. 55.11. While Eq. 55.11 has a squared velocity term (dh/dt), Eq. 55.12 includes velocity as a linear term. Thus the system model is systematically biased from the system observations.

Acceleration due to gravity g is assumed known, but the coefficient of drag C is not and is the target of inference. That is, the goal is to find the distribution of parameters C that correspond to the simulator output being similar to the system observations. In other words, the model will be calibrated to the observations, by finding the posterior distribution on C that corresponds to the GPMSA statistical model fit to all available data.

The first step in preparing the emulator is to collect system observations and examples of the computational *simulator* that represents the system model over the feasible domain. System observations are ideally collected according to a careful design, but often they are opportunistically collected according to experimental constraints. As has been developed as a standard practice in the study of complex computer codes, a (space-filling) Latin hypercube (LH) design is used to efficiently collect these samples over the domain of the model parameters of interest, in this case (R, C) .

Here is a summary of the datasets, where n denotes the size of the experimental data set, m denotes the number of runs of the simulator, and n_η denotes the number of heights used by the simulator:

- Data are from $n = 3$ field experiments, one each for balls of radius $R \in \{0.1, 0.2, 0.4\}$ m. Each experiment produces a curve of drop times made of three or four height-time pairs. For the two smallest balls, the experimental heights are $h_e \in \{5, 10, 15, 20\}$ m. The largest has a drop time measured only from experimental heights $h_e \in \{5, 10, 15\}$ m (perhaps representing experimental or cost constraints in data collection)
- A scaled Latin hypercube design was used to select the $m = 25$ (R, C) pairs, shown in Fig. 55.1, at which to run the computer model. For each (R, C) pair in the design, the simulator produces a curve of $n_\eta = 16$ height-time pairs, where the simulation heights h_s are evenly spaced in $[1.5, 24]$ m.

3.1 How We Use the Gaussian Process Model

The Gaussian process model requires specification of inputs of two kinds:

1. $\mathbf{x} = (x_1, x_2, \dots, x_p)$ denotes inputs that are under the control of (or are observable by) the experimenter in both the field experiments and the simulator runs. In the example, there is $p = 1$ input of this type: $\mathbf{x} = x = R$, the radius of the ball being dropped.
2. $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_q)$ denotes inputs to the simulator that are needed to estimate using the experimental data. These $\boldsymbol{\theta}$ could correspond to real physical quantities or could be parameters of the simulator code. In the example, there is $q = 1$ input of this type: $\boldsymbol{\theta} = \theta = C$, the coefficient of drag.

Note that h will be used as an index of the multivariate response rather than a model parameter.

run	R	C
1	0.0996	0.2105
2	0.3995	0.1795
3	0.2956	0.1167
4	0.4033	0.1457
5	0.4478	0.0610
6	0.0971	0.2376
7	0.1222	0.0982
8	0.3155	0.1072
9	0.1676	0.0742
10	0.1480	0.0979
11	0.2331	0.1639
12	0.2251	0.2275
13	0.0795	0.1977
14	0.3272	0.2237
15	0.2460	0.1914
16	0.2881	0.2466
17	0.3502	0.0702
18	0.3613	0.1345
19	0.3833	0.1564
20	0.2095	0.1762
21	0.1308	0.1498
22	0.1898	0.1236
23	0.4264	0.2087
24	0.0579	0.0544
25	0.2676	0.0885

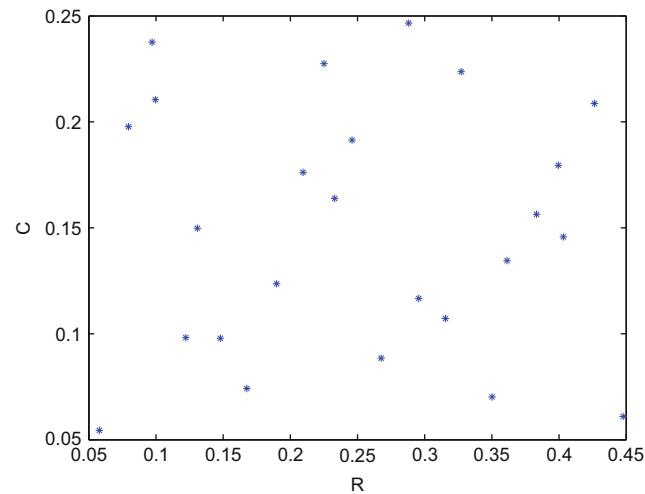


Fig. 55.1 *Left:* Scaled Latin hypercube design with $m = 25$ rows of (R, C) pairs. *Right:* A plot of the space-filling design

There are also the two types of output from system observations and the system model simulation:

1. $\mathbf{y}^{\text{obs}}(\mathbf{x})$, the system observations. For each experiment, \mathbf{y}^{obs} can be a scalar, or, as in the example, multivariate. For the tower experiments, \mathbf{y}^{obs} is a three or four element vector of times, one for each corresponding drop height.

Not all experiments produce output of the same size; in the tower experiment, there are four recorded times for the two smaller balls but only three for the largest ball.

2. $\mathbf{y}^{\text{sim}}(\mathbf{x}, \boldsymbol{\theta})$, the output of the simulation runs. \mathbf{y}^{sim} can be scalar or multivariate, corresponding to the observations. For the tower example, the simulator uses an evenly spaced grid of $n_\eta = 16$ heights and computes a time for each height on the grid, so $\mathbf{y}^{\text{sim}} = \mathbf{t}_{\text{sim}}$.

Unlike the observed data, the simulator output will always have the same size, i.e., the same number of computed times. The grid of 16 equally spaced heights between 1.5 and 24 will be the same from run to run.

3.2 Preparing the Data for Use by GPMSA

The GPMSA code is used to calibrate the statistical model parameters and to make predictions from the model. Before involving GPMSA, the user needs to read in the data, transform the inputs and outputs, compute basis functions for transforming the standardized outputs and the discrepancy term, and package the data into MATLAB structures that can be passed to GPMSA. These aspects are performed by the user because they fundamentally involve choices in the analysis. This will now be illustrated for the tower example.

3.2.1 Reading the Data

For this example, there are three data files for the observations and three for the simulator runs: one with the measured R_{obs} for the observations and corresponding inputs R_{sim} and C_{sim} for the simulator runs, one with the observations and corresponding simulator outputs (t_{obs} and t_{sim}), and one with the heights (\mathbf{h}_{obs} and \mathbf{h}_{sim}).

```
% read in the field (observed) data
>> Robs = textread('field.radius'); % radii R

Robs =
0.1000    0.2000    0.4000

>> hobs = textread('field.height'); % heights h
% hobs has all the tower heights;
% we only use the lowest 3 for the largest ball
hobs =
5      10      15      20

>> tobs = textread([dirstr 'field.dat']); % times t
% tobs has one column per experiment, one row per tower height.
% The NaN in experiment 3 indicates we didn't drop the largest
% ball from the highest tower.
tobs =
1.2180    1.1129    1.0611
2.0126    1.7225    1.5740
2.7942    2.2898    2.0186
3.5747    2.8462      NaN

>> tstd = textread([dirstr 'field.sd']); % sd of measured
times
```

```
tstd =
0.1000    0.1000    0.1000
0.1000    0.1000    0.1000
0.1000    0.1000    0.1000
0.1000    0.1000      NaN

% read in the design and the simulator output
>> [Rsim Csim] = textread('sim.design'); % design (R and C)
>> tsim = textread('sim.dat'); % times t
>> hsim = textread('sim.height'); % heights h

>> n = size(tobs, 2); % number of experiments
>> m = size(tsim, 2); % number of simulation runs
```

3.2.2 Transforming x and θ

The GPMSA code requires that the inputs x and θ lie in the interval $[0, 1]^{p+q}$ and responses that are $N(0,1)$. These transformations allow the selection of default prior distributions and MCMC proposal distributions. First the inputs to the simulator (R_{sim} and C_{sim}) are transformed, so they lie in $[0, 1]$; then the minimum and range of R_{sim} are used to transform the corresponding experiment's measurements (R_{obs}).

```
% transform the simulator inputs so each dimension lies in
[0, 1]
>> Rsimmin = min(Rsim);
>> Rsrang = range(Rsim);
>> Rsim01 = (Rsim - Rsimmin) / Rsrang; % transformed R

>> Csimmin = min(Csim);
>> Csrang = range(Csim);
>> Csim01 = (Csim - Csimmin) / Csrang; % transformed C

% transform the field experiment input the same way
>> Robs01 = (Robs - Rsimmin) / Rsrang; % transformed R
```

3.2.3 Transforming y^{sim} and y^{obs}

The GPMSA code requires that the outputs y have mean zero and variance one. As above, the output from the simulator (\mathbf{t}_{sim}) is transformed and then those values are used to transform the observations (\mathbf{t}_{obs}). Here the simulator output should have mean zero at each height h and an overall variance of one.

```
% standardize the simulator output
>> tsimmean = repmat(mean(tsim, 2), [1 m]); % the mean
simulator run
>> tsimStd = tsim - tsimmean; % make mean at each height zero
>> tsimsd = std(tsimStd(:)); % standard deviation of ALL
elements of tsimStd
>> tsimStd = tsimStd / tsimsd; % make overall variance one
```

Transformed observations should use the distribution of the simulator runs (`tsimmean` above) at each experimental height, but the output grid of the simulator doesn't match the experimental observation grid; i.e., the value of the mean simulator run at all the experimental heights is unknown. Instead, `tsimmean` will be interpolated in order to estimate the value of an assumed underlying mean function at each experimental height. This interpolated mean and the overall standard deviation of all elements of the simulator runs will be used `tsimsd`, to transform the observations.

Each multivariate observation may have its elements on a different grid (in this example, different number of heights at which the ball was dropped), so the observations cannot be in a matrix. Instead, the observation data are collected into a Matlab *struct* array called `yobs`.

```
>> for ii = 1:n
    % number of heights with measurements for experiment ii
    numhts = sum(~isnan(tobs(:, ii)));

    % do the interpolation and get the interpolated values at
    % the experimental heights
    yobs(ii).tobsmean = ...
        interp1(hsim, tsimmean(:, 1), hobs(1:numhts),
        'linear', 'extrap');

    % do the standardization
    yobs(ii).tobsStd = (tobs(1:numhts, ii) - yobs(ii)
        .tobsmean') / tsimsd;

    % for convenience, record some extra information in yobs
    yobs(ii).hobs = hobs(1:numhts); % the heights where
        measurements were taken
    yobs(ii).tobs = tobs(1:numhts, ii); % the untransformed
        output

    % now record the observation covariance matrix for the
    % measured times
    yobs(ii).Sigy = diag(tstd(1:numhts, ii).^2);
    % now the observation covariance for the standardized
    % observations
    yobs(ii).SigyStd = yobs(ii).Sigy ./ (tsimsd.^2);
end
```

`yobs(ii).Sigy` holds the covariance matrix for the observations of experiment `ii`. If not specified, it is given a default constant value. `Sigy` is scaled by a parameter calibrated in the GPMSA model that by default is free to vary. The prior specification for this measurement precision scale factor can be changed to ensure that measurement error stays close to the specified value.

3.2.4 Computing the K Matrix for Transforming \mathbf{y}^{sim} and \mathbf{y}^{obs}

As discussed in the model overview, multivariate observations and responses are modeled with a linear basis. The use of *principal components*, or scaled

eigenvectors, computed by the singular value decomposition of the simulator output, is demonstrated. While principal components are commonly used for this, any orthogonal linear transformation will work. For a compact representation, $p_u < m$ basis functions that capture most of the variation in the simulation runs are retained. The choice of how many principal components to use is a trade-off, considering the variability represented by the projection and emulator performance. Note that p_u , the number of basis elements, should not be confused with p , the dimension of the input x .

```
>> pu = 2; % number of basis components to keep
>> [U, S, V] = svd(tsimStd, 0); % compute the SVD
>> Ksim = U(:, 1:pu) * S(1:pu, 1:pu) ./ sqrt(m); % construct
    the basis components
```

This K_{sim} matrix has $n_\eta = 16$ rows (one for each height in the grid used by the simulator) and $p_u = 2$ columns. A corresponding matrix K_{obs} for each experiment in the field data is computed by interpolating the K_{sim} components onto the observation data locations.

```
>> for ii = 1:n
    yobs(ii).Kobs = zeros(length(yobs(ii).tobsStd), pu); %
    allocate space

    % compute each basis component
    for jj = 1:pu
        % do the interpolation and get the interpolated values
        % at the experimental heights
        yobs(ii).Kobs(:, jj) = ...
            interp1(hsim, Ksim(:, jj), yobs(ii).hobs,
            'linear', 'extrap');
    end
end
```

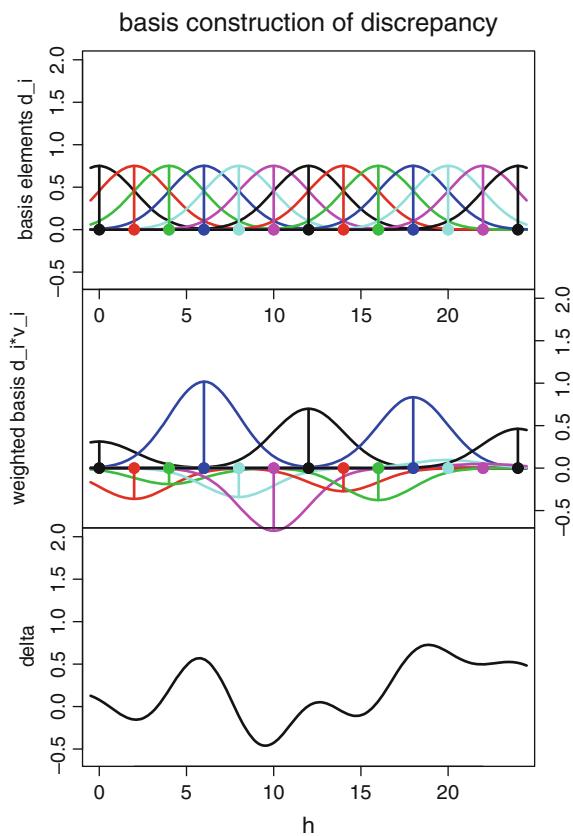
3.2.5 Specifying the D Matrix for Modeling the Discrepancy Term

The discrepancy term $\delta(x)$ models a systematic bias between the simulator (at the best setting for the calibration parameter θ) and is represented as a GP over the x space. In the example, for a given ball radius x , $\delta(x)$ is a function over the possible drop heights $1.5 \leq h \leq 24$ m. The discrepancy is approximated by a linear basis with similar motivation to the emulator response. The discrepancy basis is selected to allow a prior estimated required flexibility. Its form is ultimately problem dependent, but in many problems a reasonable form is the smooth constraint of a normal kernel basis.

Over $h \in [1.5, 24]$, $\delta(x)$ is represented as a linear combination of basis functions

$$\delta(x) = \sum_{i=1}^{p_v} \mathbf{d}_i v_i(x)$$

Fig. 55.2 Basis construction of $\delta(x)$ for the ball dropping example. Here a model for $\delta(x)$ – the discrepancy between the calibrated simulator and experimental observations at x – is modeled by a linear combination of normal kernels. *Top:* 13 normal kernels with $sd = 2$ are placed at heights $h = 0, 2, \dots, 24$. Each of the 13 columns in D corresponds to one of these basis kernels. *Middle:* each basis kernel is multiplied by a random normal variate $v_i(x)$ which is estimated in GPMSA, with dependence over the x -space, using the simulation output and experimental data. *Bottom:* the discrepancy is set to the sum of these weighted kernels, producing a prior realization for $\delta(x)$. In vector form, this is given by $Dv(x)$, where $v(x)$ is the 13 element vector of weights corresponding to input condition x



where \mathbf{d}_i are vectors over the output support h . For this example, the $d_i(\cdot)$'s are taken to be normal kernels with an sd of 2. The kernels are centered at a grid of 13 heights equally spaced between 0 and 24. This model is depicted in Fig. 55.2.

For each experiment, we construct the matrix D_{obs} whose rows correspond to the observed drop heights in the experiment, and whose columns correspond to the $p_v = 13$ basis elements. For analysis purposes, the matrix D_{sim} which has rows corresponding to the h -space is also constructed:

```
% -- D basis --
>> Dgrid = 0:2:max(hsim); % locations on which the kernels
    are centered
>> Dwidth = 2; % width of each kernel
>> pv = length(Dgrid); % number of kernels

% Compute the kernel function map, for each kernel
% Designate space for the Dsim matrix,
% one row per simulated height, one column per kernel
% (consider making the grid of heights much denser for
    plotting)
```

```

>> Dsim = zeros(length(hsim), pv);

% designate space for the Dobs matrix for each experiment,
% one row per experimental height, one column per kernel
>> for ii = 1:n
    yobs(ii).Dobs = zeros(length(yobs(ii).tobsStd), pv);
end

% create each kernel
>> for jj = 1:pv
    % first create kernel jj for each experiment
    for ii = 1:n
        % normpdf computes the value of a Gaussian with mean
        % Dgrid(jj) and variance Dwidth at each element of hobs
        yobs(ii).Dobs(:, jj) = normpdf(yobs(ii).hobs,
            Dgrid(jj), Dwidth);
    end
    % now create kernel jj for the simulations
    Dsim(:, jj) = normpdf(hsim, Dgrid(jj), Dwidth);
end

% normalize the basis elements of D so that the marginal
% variance of delta is about 1
>> Dmax = max(max(Dsim * Dsim'));
>> Dsim = Dsim / sqrt(Dmax);
>> for ii = 1:n
    yobs(ii).Dobs = yobs(ii).Dobs / sqrt(Dmax);
end

```

The D matrices are normalized so that the prior marginal variance for $\delta(x)$ is approximately one when $\lambda_v = 1$.

For normal basis kernels, a rule of thumb is to make the spacing one standard deviation between adjacent kernels to ensure that no sparsity effects appear, while limiting the number of parameters [2].

3.2.6 Package all the Pieces

Having now completed the specification and transformation of required data, it can be collected into a single Matlab structure to be given to GPMSA for model setup. This structure, here called `data`, will contain a field for the simulated data (`simData`) and another for the field data (`obsData`). For both fields, we'll include information that's required by the model as well as extra information (stored in a subfield called `orig`) that will later make it easier for us to return the output to the native scale and to perform analysis and plotting.

```

% required fields
>> simData.x = [Rsim01 Csim01]; % our design (standardized)
>> simData.yStd = tsimStd; % output, standardized
>> simData.Ksim = Ksim;

% extra fields: original data and transform stuff
>> simData.orig.y = tsim;
>> simData.orig.ymean = tsimmmean;

```

```
>> simData.orig.ysd = tsimsd;
>> simData.orig.Dsim = Dsim;
>> simData.orig.t = hsim;
>> simData.orig.xorig = [Rsim Csim]; % original scale for
simulated R, C
```

For the observed data, each experiment is packaged separately since each could have a different length.

```
% loop over experiments
>> for ii = 1:n
    % required fields
    obsData(ii).x = Robs01(ii);
    obsData(ii).yStd = yobs(ii).tobsStd;
    obsData(ii).Kobs = yobs(ii).Kobs;
    obsData(ii).Dobs = yobs(ii).Dobs;
    obsData(ii).Sigy = yobs(ii).Sigy./(tsimsd.^2);

    % extra fields: original data
    obsData(ii).orig.y = yobs(ii).tobs;
    obsData(ii).orig.ymean = yobs(ii).tobsmean;
    obsData(ii).orig.t = yobs(ii).hobs;
end
```

Now we'll put `simData` and `obsData` in a structure called `data` that can be passed to GPMSA.

```
>> data.simData = simData;
>> data.obsData = obsData;
```

3.3 Model Initialization and MCMC

Now that the user setup of data has been completed, the model can be initialized and the posterior distributions of parameters sampled via Markov chain Monte Carlo (MCMC). The code in this section is in the MATLAB file `runmcmc.m`.

`readdata.m` implements the code previously detailed.

```
>> towerdat = readdata()

towerdat =

    simData: [1x1 struct]
    obsData: [1x3 struct]
```

`setupModel()` performs the initial setup of the model, taking the `obsData` and `simData` fields from `towerdat` and returning a structure `pout` (for ‘parameter output’).

```
>> pout = setupModel(towerdat.obsData, towerdat.simData)
SetupModel: Determined data sizes as follows:
```

```

SetupModel: n= 3 (number of observed data)
SetupModel: m= 25 (number of simulated data)
SetupModel: p= 1 (number of parameters known for
observations)
SetupModel: q= 1 (number of additional simulation inputs
(to calibrate))
SetupModel: pu= 2 (response dimension (transformed))
SetupModel: pv= 13 (discrepancy dimension (transformed))

pout =

```

data:	[1x1 struct]
model:	[1x1 struct]
priors:	[1x1 struct]
mcmc:	[1x1 struct]
obsData:	[1x3 struct]
simData:	[1x1 struct]
pvals:	[]

Fields of `pout` include the simulated and observed data transformed by the K and D matrices (`data`), initial values for the parameters of the posterior induced by the model in GPMSA (`model`), priors on the model parameters (`priors`), MCMC controls (e.g., step sizes) in (`mcmc`), and the `obsData` and `simData` structures that were given in the call to `setupModel()`. It also includes a placeholder for the `pvals` field which will hold the MCMC draws. MCMC will be used to get draws from the parameters' posterior distribution with the GPMSA function `gpmmcmc()`.

1. Before performing MCMC, and as an optional step, GPMSA includes a utility `stepsize()` to optimize the MCMC proposal widths, or “step sizes.” Default settings may provide reasonable, although not optimal, performance. The step size estimation procedure is taken from Graves [1].

Computation of the step size starts by collecting MCMC proposal acceptance statistics at a number of possible values (`levels`), using estimates constructed from a number of MCMC draws.

```

>> nsamp = 100; % number of draws to sample

>> nlev = 13; % number of candidate levels used for step
size estimation

>> pout=stepsize(pout,nsamp,nlev)
Setting up structures for stepsize statistics collect ...
Collecting stepsize acceptance stats ...
Drawing 100 samples (nBurn) over 13 levels (nLev)
Started timed counter, vals 1 -> 1300
963..20.29sec
Computing optimal step sizes ...
Step size assignment complete.

```

```
pout =
    data: [1x1 struct]
    model: [1x1 struct]
    priors: [1x1 struct]
    mcmc: [1x1 struct]
    obsData: [1x3 struct]
    simData: [1x1 struct]
    pvals: [1x1300 struct]
```

The `pvals` object holds the result of the MCMC. Here it records the 1300 draws from the posterior distribution for each parameter produced by the MCMC updates carried out so far. In addition to the parameter values at each of the 1300 MCMC steps, the corresponding values for the log likelihood, the log prior, and the log posterior are also recorded. The draws used for step size estimation are not valid samples of the posterior distribution and should not be used in prediction or analysis; they could be discarded at this point, setting `pout.pvals` to an empty matrix.

Here are the updated MCMC settings:

```
>> pout.mcmc

ans =
    thetaWidth: 0.2668
    rhoUWidth: [0.5341 0.4523 2.6462 1.7655]
    rhoVWidth: 0.4656
    lamVzWidth: 433.1763
    lamUzWidth: [0.8726 1.9799]
    lamWsWidth: [1.6396e+03 4.0254e+03]
    lamWosWidth: 2.0908e+04
    lamOsWidth: 3.1539e+04
    pvars: {1x11 cell}
    svars: {'theta' 'betaV' 'betaU' 'lamVz' 'lamUz'
            'lamWs' 'lamWos' 'lamOs'}
    svarSize: [1 1 4 1 2 2 1 1]
    wvars: {1x8 cell}
```

Note that like any MCMC procedure, the values presented here are subject to random variability and will not replicate exactly.

2. Now MCMC draws (realizations) can be generated efficiently from the parameters' joint posterior distribution. These will be added to the `pvals` field of `pout`.

```
>> nmcmc = 10000; % number of draws we want
>> pout = gpmmcmc(pout, nmcmc,'step',1)
Started timed counter, vals 1 -> 10000
    787..1577..2363..3158..3923..4675..5409..6152..6877..
    7614.. 1.7 min, 0.5 min remain
    8344..9077..9817..2min:12.57sec
```

```
>> pout =
    data: [1x1 struct]
    model: [1x1 struct]
    priors: [1x1 struct]
    mcmc: [1x1 struct]
    obsData: [1x3 struct]
    simData: [1x1 struct]
    pvals: [1x11300 struct]
```

There are now 10,000 additional values recorded for each parameter in the `pout` object. These were produced by the 10,000 MCMC iterations carried out by the last call to `gpmmcmc()`.

At this point, the system model parameters and the GPMSA parameters have been calibrated by sampling the joint posterior distribution.

3.4 Examining the Estimated Parameters' Posterior Distribution

Since the model inputs and data are transformed, the posterior sampling carried out in GPMSA corresponds to this transformed scale. Some model diagnostics (e.g., the GP's $u(x, t)$, $v(x)$, precision parameters λ) are more interpretable on their transformed scale, while others (e.g., predictions, model parameters) are more usefully viewed on their original, or native, scale. Hence some of the plotting routines covered here use the transformed scale for showing output, while others use the native scale.

3.4.1 Traces of the MCMC Draws

The GPMSA code function `showPvals()` will produce traces of the MCMC draws for the parameters in the model as shown in Fig. 55.3.

```
>> showPvals(pout.pvals);
Processing pval struct from index 1 to 11300
    theta:   mean           s.d.
      1:       0.3513        0.04313
    betaV:   mean           s.d.
      1:       0.767          0.9483
    betaU:   mean           s.d.
      1:       8.628          2.38
      2:       0.6311         0.3162
      3:       4.365          2.52
      4:       3.466          2.473
    lamVz:   mean           s.d.
      1:       72.42          63.21
    lamUz:   mean           s.d.
      1:       0.4518          0.1354
      2:       0.8458          0.3108
    lamWs:   mean           s.d.
```

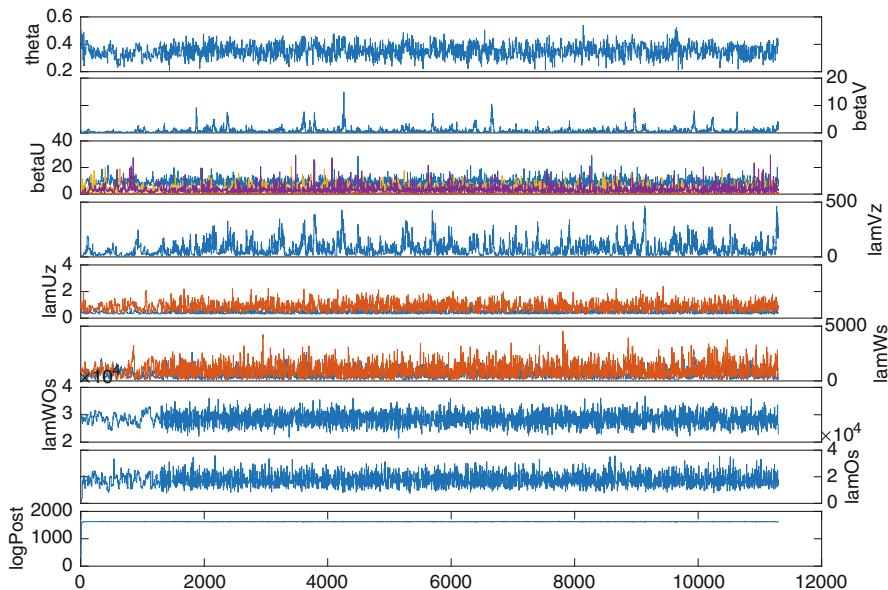


Fig. 55.3 Traces of the MCMC draws of the parameters in `pout.pvals` as generated by the `showPvals()` function

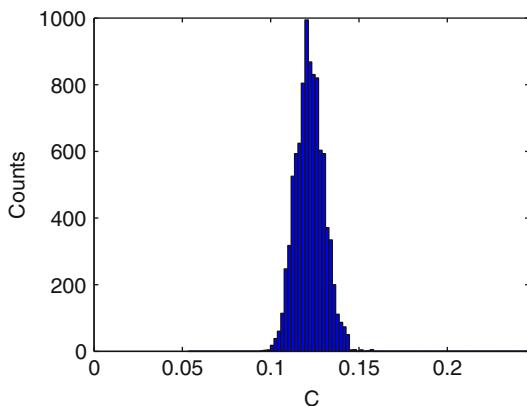
1:	546.2	326.6
2:	963.7	572.3
lamWo:	mean	s.d.
1:	2.846e+04	2095
lamOs:	mean	s.d.
1:	1.779e+04	4275
logPost:	mean	s.d.
1:	1624	28.9

Note that we're calling `showPvals()` with *all* the draws in `pout.pvals`, not just the ones specified by the index `ip` defined above. This includes the draws used for burn in and step size estimation. The resulting plot is shown in Fig. 55.3. The figure shows qualitatively that the MCMC chains of all the parameters after 2000 draws, skipping the step size data collection phase and allowing for a transient, appear mixed and stationary. Note that a rerunning of this MCMC chain will give slightly different values.

3.4.2 Posterior Distribution of C

Figure 55.4 shows the histogram of the MCMC draws from the posterior distribution of the object of calibration C . Because all 10,000 posterior draws may be excessive for plotting or further computation, a subset of 500 equally spaced draws are taken.

Fig. 55.4 Histogram of draws from the posterior distribution of C , on the native scale



```
>> from = 2000; % start getting realizations at this index
>> to = length(pout.pvals); % continue to the last realization
>> thismany = 500; % grab this many evenly spaced realizations
>> ip = round(linspace(from, to, thismany)); % indices of the
pvals to use
```

With the exception of the `showPvals()` function, the plotting functions in this section are not part of the core GPMSA code package, but the example `.m` files used are available as associated examples to GPMSA. The posterior distribution of the target model parameters is θ in the output from `showPvals`.

```
>> thetaphisthist(pout, 2000:11300);
```

Note that the realizations of the standardized parameter θ were scaled and shifted to give realizations of C on its native scale.

3.5 Assessing Emulator Adequacy

It is important to examine diagnostics to understand the quality of the emulator in reproducing the simulator output at new, untried settings. The GPMSA code produces several diagnostic plots, a few of which are displayed here. The following plots are useful in assessing the quality of the emulator fit.

3.5.1 Principal Components

Figure 55.5 shows the $p_u = 2$ principal components used in this example. Note that the vertical scale for the second principal component is much smaller than that of the first; this confirms that most of the variation in the data is being captured by the first principal component. The left graphic on this plot was made as follows:

```
>> plot(pout.simData.orig.h,pout.simData.Ksim);
```

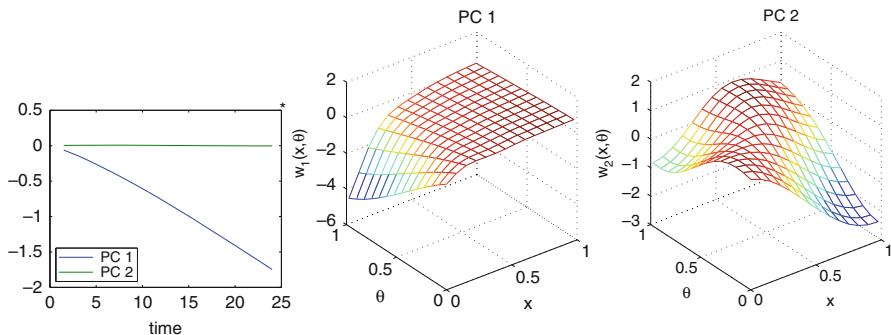


Fig. 55.5 *Left:* The $p_u = 2$ principal components used in modeling this example. *Middle and right:* posterior mean of the Gaussian processes of the weight functions for the two principal components. The weights $w(x, \theta)$ are used to make the predictions

The right two graphs on Fig. 55.5 show the posterior mean of the Gaussian process of the weight functions for the two principal component for the domain of (x, θ) . The weight $w(x, \theta)$ at each (x, θ) pair is used to make the predictions from the model. The code `PCresponsesurf.m` calls the function `gPred()` to make predictions at each grid point and generates the plot as follows:

```
>> PCresponsesurf(pout, ip);
```

3.5.2 Parameters in the Gaussian Process Fit

Using the MCMC draws of the spatial dependence parameters β , $\rho = \exp\{-\beta/4\}$ can be calculated to show the values on a bounded scale. The value of ρ gives us information about the dependence of the simulation output on each input parameter x and θ . Figure 55.6 shows box plots of the posterior draws for the ρ for each x and θ and for each principal component. The figure was generated using a subset of the realizations:

```
>> rhoboxplots(pout, ip);
```

When ρ is near 1 for a particular x or θ principal component, it suggests that particular component of the simulator output is linear in that dimension (the case of exactly 1 is degenerate; theoretically it would be constant). As ρ goes smaller, nonlinear activity is associated with that input. The outputs will vary smoothly with the inputs, with smaller values of ρ indicating less smoothness.

As ρ becomes smaller, the modeled response is increasingly flexible. Eventually, this indicates that the emulator is overfitting the data, interpolating each point instead of fitting a trend. This suggests predictions from the model are suspect. Thus if any of the box plots in Fig. 55.6 shows values that are all close to zero, the model is suspect. Cross-validation diagnostics should be considered before accepting predictions from the model, especially in that case. Correlation parameter(s) for the discrepancy process can be similarly assessed.

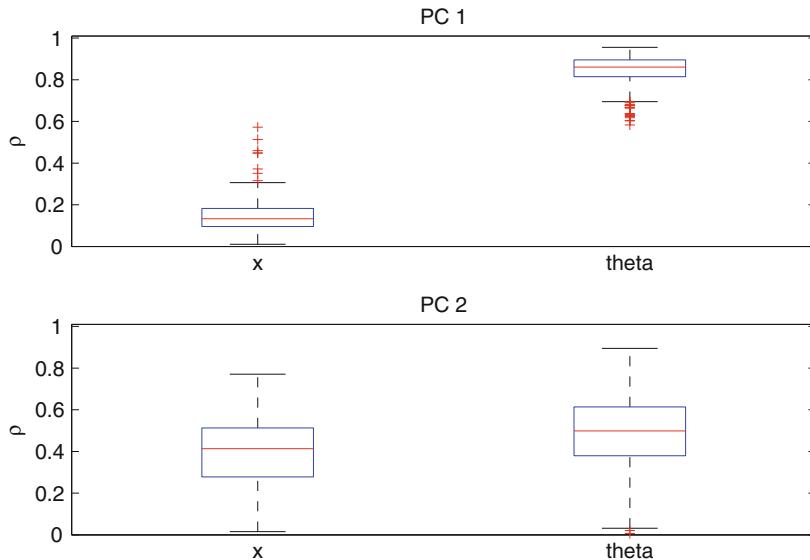


Fig. 55.6 Box plots of $\rho = \exp\{-\beta/4\}$ for the draws of β associated with x and θ for each principal component

3.5.3 Cross-Validation of Simulation Response

The most direct test of the emulator's ability to reproduce the results of the simulator is to compare the emulator predictions with actual simulations for a relevant holdout set. If available, such tests will be very informative. If a holdout test set is not available, one option is to take a cross-validation approach [4].

For each run i of the simulator, a model excluding run i is constructed and then run i predicted as holdout. This allows us to look for trends in the quality of predictions as a function of the inputs R_{sim} and C_{sim} . Figure 55.7 shows the results sorted by R_{sim} . This plot, among others, was produced by the function call `holdoutpreds(pout, ip)`. One could also plot the residuals for a closer look. Here the cross-validated holdout predictions are less accurate when R is small. As expected, larger errors are associated with design points at the edge of the design (see Fig. 55.1) where there is lower neighborhood constraint. Holding out data from the design in a cross-validation fashion will be an overestimate of the expected error of predictions using all constraints.

3.5.4 Conditional Response Sensitivity

An alternative way to understand how the simulator responds to changes in the inputs is to plot the output while varying each input over its range from high to low, at fixed settings of the other parameters. Figure 55.8 shows how the drop time as a function of height varies as x and θ are varied from their low to high values. Here the other input is held at its midpoint to produce this plot. This plot

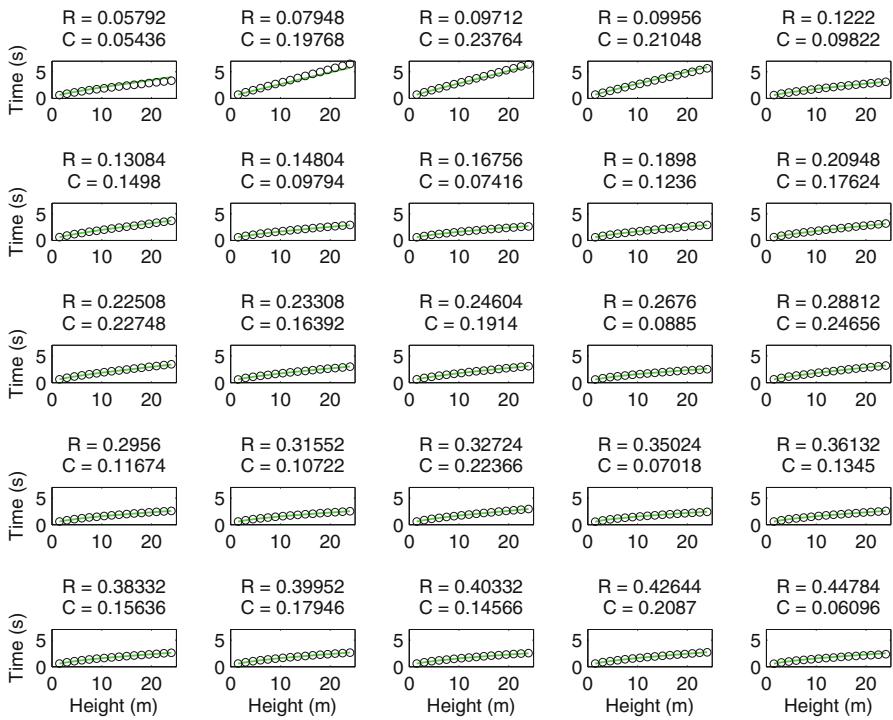


Fig. 55.7 Holdout predictions, sorted by the value of R_{sim} . The black circles are actual simulations and the green lines are emulations

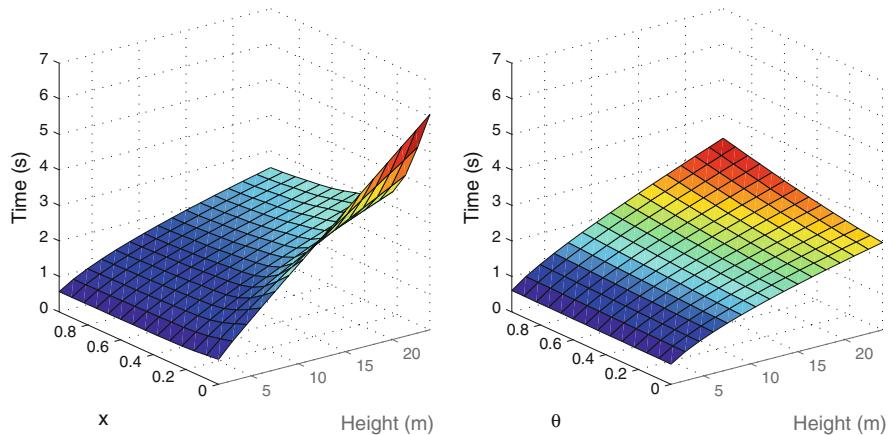


Fig. 55.8 Surface plots showing sensitivity to x (left) and to θ (right)

also highlights the sensitivity of the simulator response to very low values of the standardized ball radius x . Figure 55.8 is generated by the following function call `sensitivities(pout, ip)`.

3.6 System Predictions

Once we're satisfied that the emulator is adequately reproducing the simulator output, GPMSA can produce predictions for the actual system at new experimental conditions \mathbf{x} . Below, the MCMC samples are used to produce predictions with uncertainty.

The system prediction uses the posterior uncertainty for θ , along with the additional GPMSA model parameters, to produce uncertainties for a new experimental outcome \mathbf{y}^* at experimental conditions \mathbf{x}^*

$$\mathbf{y}^* = \eta(\mathbf{x}^*, \theta) + \delta(\mathbf{x}^*) + \mathbf{e}^*.$$

Uncertainties in this prediction come partly from variation in the posterior samples collected. Posterior draws of \mathbf{y}^* use the samples in `pout`. This is carried out in the function `etasdeltas.m`. This produces realizations of the calibrated simulator $\eta(\mathbf{x}^*, \theta)$, the discrepancy $\delta(\mathbf{x}^*)$, which compose the predicted drop time \mathbf{y}^* as a function of height. Making many predictions, for three different balls denoted whose

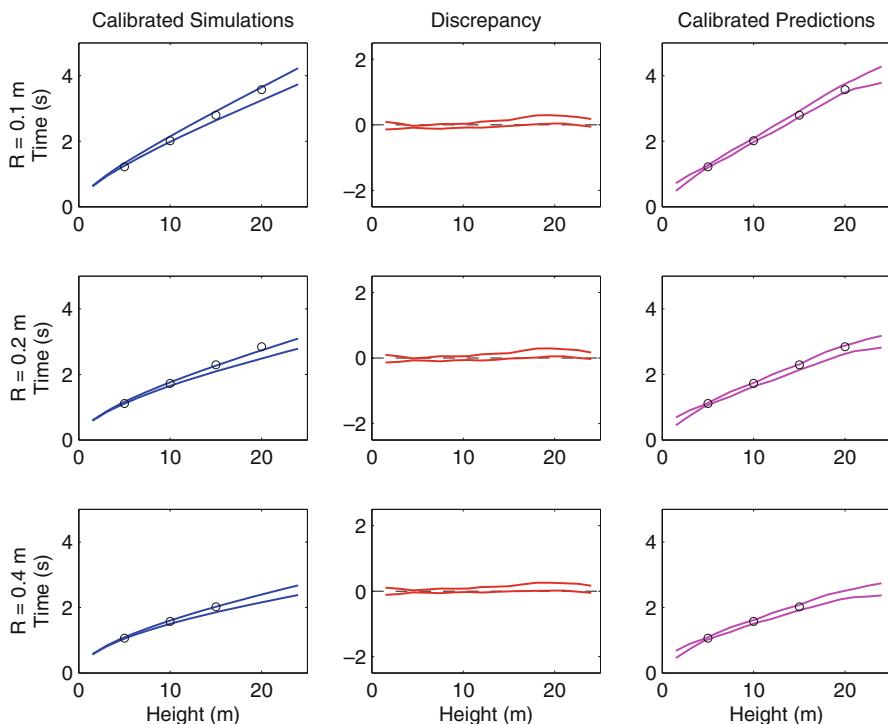


Fig. 55.9 Circles show the field data, and colored lines indicate the 5th and 95th percentiles. Each row is a different ball size. *Left column:* Calibrated simulations. *Center:* Discrepancy term (dashed line shows where zero discrepancy would be). *Right:* Calibrated predictions = calibrated simulations + the discrepancy term

radii are denoted by \mathbf{x}^* , allows the estimate of mean and quantile statistics, as shown in Fig. 55.9. The left column shows the calibrated simulations; the center column shows the discrepancy between the experimental data (circles) and the calibrated simulations; and the right column shows the calibrated predictions made after adding the discrepancy term to the calibrated simulation.

3.7 The pout Object

The ball dropping example has produced `pout` which holds a variety of data, which may in some cases be useful for diagnostics, to follow through the function of calibrating the GPMSA statistical model. The preprocessing function `readdata()` constructs `obsData` and `simData`. `obsData` holds information regarding the physical observation data, while `simData` holds information regarding the simulation output, including the basis representations for the multivariate simulation output and the discrepancy basis.

The function `setupModel()` creates `data`, `model`, `priors`, and `mcmc`. It also creates an empty object `pvals`, which will later hold the MCMC output produced by `gpmcmc()`. Hence the posterior samples for the various parameters will be kept in the `pvals` object. The `data` object holds transformations of the simulation and observed data that are required for likelihood evaluations used in the MCMC algorithm. There should be no need to modify this data, although if unexpected behavior occurs, it may be useful to validate the problem setup by verifying the expected transformations are in place. The `model` object holds all of the additional objects required to evaluate the likelihood and prior, as well as saved partial computations to speed computation. The current value of the MCMC chain is stored here. The `prior` object holds the prior specification for each of the model parameters. This includes upper and lower bounds for each parameter, the name of the log-prior evaluation function, and the parameters. Nonstandard or user-defined priors can be implemented by changing the prior functions and parameters. Finally, the `MCMC` object holds information required to carry out the MCMC sampling, including step sizes used in the Metropolis-Hastings updates for each parameter, indicators for sampling, and the parameters to be logged in `pvals`. These values are modified when the step size estimation is carried out in `gpmcmc()`.

Further descriptions of these fields are provided in the reference manual.

4 Example 2: Ball Drop with Different Radii and Densities

The second example follows a similar framework to Example 1; however, now consider different types of balls that have different densities. In this scenario, three different types of balls, bowling ball, basketball, and a baseball, are again dropped from a tower. Each type of ball has a unique radius R and density ρ . The heights that the balls are dropped range from 10 to 60 m; however, the basketball and baseball are only dropped from 20, 40, and 60 m. The bowling ball is dropped from 10, 20, 30, 40, 50, and 60 m.

The experiments are not actually performed, but the system observations are generated as noisy realizations from

$$\frac{d^2h}{dt^2} = g - \frac{C}{2} \frac{3\rho_{\text{air}}}{4R_{\text{ball}}\rho_{\text{ball}}} \left(\frac{dh}{dt} \right)^2, \quad (55.13)$$

where ρ_{air} is the density of air, R_{ball} is the radius of the ball, ρ_{ball} is the density of the ball, and g and C are the coefficient of gravity and the coefficient of drag as defined in Eq. 55.11. The system model for this example is given by Eq. 55.13. In this example, the acceleration due to gravity, g , and the coefficient of drag, C , are both considered unknown and the goal is to jointly estimate these parameters.

Again the emulator is prepared to collect system observations via a computational simulator representing the system model. In this framework, 20 simulations are conducted for each ball type, which corresponds to a radius-density pair. The radius-density pairs are selected using a space filling Latin hypercube design.

Here is a summary of the data used in this example:

- There are n_i field experiments for each ball. The baseball, $R = 0.0380 \text{ m}$, $\rho_{\text{ball}} = 626 \text{ kg/m}^3$, is dropped from three heights $\{20, 40, 60\} \text{ m}$. The basketball, $R = 0.1200 \text{ m}$, $\rho_{\text{ball}} = 84 \text{ kg/m}^3$, is dropped from three heights $\{20, 40, 60\} \text{ m}$. The bowling ball, $R = 0.1100$ and $\rho_{\text{ball}} = 1304 \text{ kg/m}^3$, is dropped from six heights $\{10, 20, 30, 40, 50, 60\} \text{ m}$.
- A space filling Latin hypercube design selects the $m = 20$ (C, g) pairs, shown in Fig. 55.10, at which to run the computer model. For each (C, g) pair in the design, the simulator produces a curve of $n = 100$ height-time pairs, where the simulation heights are evenly spaced in $[0, 99] \text{ m}$. Each of these 20 parameter settings are run for each of the three balls described above and a softball with $R = 0.0485$ and $\rho_{\text{ball}} = 380.9$.

4.1 How We Use the Gaussian Process Model

1. $\mathbf{x} = (x_1, x_2, \dots, x_p)$ denotes inputs that are under the control of (or are observable by) the experimenter in both the field experiments and the simulator runs. In this example, there are $p = 2$ inputs of this type: $\mathbf{x} = \{R, \rho\}$, the radius and density of the ball being dropped.
2. $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_q)$ denotes inputs to the simulator to be estimated using the experimental data. In this example, there are $q = 2$ inputs of this type: $\boldsymbol{\theta} = \{C, g\}$, the coefficients of drag and gravity.

There are also the two types of output from system observations and the system model simulation:

run	C	g
1	1.3368	10.9474
2	0.4842	9.4737
3	0.7684	12.0000
4	0.6737	8.2105
5	0.9579	10.3158
6	1.1474	8.4211
7	0.3895	11.3684
8	1.9053	11.1579
9	2.0000	9.8947
10	1.6211	9.2632
11	0.5789	10.7368
12	0.2000	10.1053
13	0.2947	8.8421
14	1.8105	8.6316
15	1.0526	11.5790
16	1.2421	9.6842
17	1.5263	11.7895
18	1.4316	8.0000
19	1.7158	10.5263
20	0.8632	9.0526

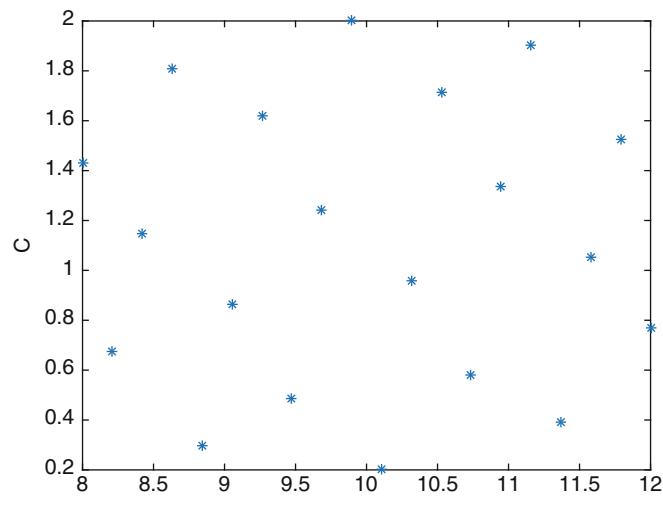


Fig. 55.10 Left: Scaled Latin hypercube design with $m = 20$ rows of (C, g) pairs. Right: A plot of the design

1. $\mathbf{y}^{\text{obs}}(\mathbf{x})$, the system observations. For the synthetic tower experiments, \mathbf{y}^{obs} is a 3- or 6-element vector of times, one time for each corresponding drop height for a given ball.

Not all experiments produce output of the same size; in the tower experiment, there are six recorded times for the bowling ball, but only three for the baseball and basketball.

2. $\mathbf{y}^{\text{sim}}(\mathbf{x}, \boldsymbol{\theta})$, the output of the simulation runs. For the tower example, the simulator uses an evenly spaced grid of $n_\eta = 100$ heights (up to 100 m) and computes a time for each height on the grid, so $\mathbf{y}^{\text{sim}} = \mathbf{t}_{\text{sim}}$.

Unlike the observed data, the simulator output will always have the same size, i.e., the same number of computed times. The grid of 100 equally spaced heights between 0 and 100 m will be the same from run to run.

4.2 Preparing the Data for Use by GPMSA

Before using the GPMSA code, the data are read and transformed. Much of the material here overlaps with Example 1 and will not be explained here in as much detail.

4.2.1 Reading the Data

This example uses one dataset for the tower experiments and three datasets for the simulated data. Note the dataset corresponding the tower experiments includes softball drops, but this information is not used in this experiment.

```
>> % read in the field data
>> fielddat = textread([dirstr 'fieldDat15x6gparam.txt']);
% R_ball rho_air rho_ball height time sd(time); 1st 3 are
    basketball drops,
% 2nd 3 are baseball drops, 3rd 3 are bowling ball drops.
>> fielddat

fielddat =
    .1200    1.1840    84.0000    20    2.2903    0.10
    .1200    1.1840    84.0000    40    3.2610    0.10
    .1200    1.1840    84.0000    60    4.3482    0.10
    .0380    1.1840   626.0000    20    2.0149    0.10
    .0380    1.1840   626.0000    40    2.7790    0.10
    .0380    1.1840   626.0000    60    4.0438    0.10
    .1100    1.1840  1304.0000    10    1.4650    0.10
    .1100    1.1840  1304.0000    20    1.9566    0.10
    .1100    1.1840  1304.0000    30    2.6336    0.10
    .1100    1.1840  1304.0000    40    2.7205    0.10
    .1100    1.1840  1304.0000    50    3.2327    0.10
    .1100    1.1840  1304.0000    60    3.5393    0.10

>> % read in the simulated data and the design
>> tsim = textread([dirstr 'sims101x80Cg.txt']); % times
>> hsim = textread([dirstr 'simHeights101x1']); % heights
>> % design (x=R, rho_ball and theta=C g)
>> designNative = textread([dirstr 'desNative80x4Cg.txt']);
>> designNative(1:5,:)

ans =
    0.1200    84.0000    1.3368    10.9474
    0.1200    84.0000    0.4842     9.4737
    0.1200    84.0000    0.7684    12.0000
    0.1200    84.0000    0.6737     8.2105
    0.1200    84.0000    0.9579    10.3158

>> m = size(tsim, 2); % number of simulation runs
>> n = 3;
>> iball = [1 1 1 2 2 2 3 3 3 3 3]; % index of balls for
    field data
>> nmeas = [3 3 6]; % number of measurements per ball
>> cumnmeas = [0 cumsum(nmeas)];
```

4.2.2 Transforming $x, \theta, y^{\text{sim}}$; and y^{obs}

Again the inputs \mathbf{x} and $\boldsymbol{\theta}$ as well as y^{sim} and y^{obs} need to be standardized according to the same procedure as Experiment 1.

```
>> simData.orig.designNative = designNative;
>> simData.orig.colmax = max(designNative);
```

```

>> simData.orig.colmin = min(designNative);

>> % standardize the inputs to the simulator (x and theta) to
    lie in [0, 1]
>> dmin = simData.orig.colmin;
>> dmax = simData.orig.colmax;
>> drange = dmax-dmin;

>> % standardize the simulator output to have mean zero at
    each height and an
>> % overall variance of one
>> tsimmean = repmat(mean(tsim,2), [1 m]);
>> tsimStd = tsim - tsimmean; % makes mean at each height zero
>> tsimsd = std(tsimStd(:));
>> tsimStd = tsimStd / tsimsd; % makes overall variance one
    (but not at each height)

% standardize the field data.
>> for ii = 1:n
    numhts = nmeas(ii); % how many heights have measurements
    for experiment ii
        yobs(ii).y = fielddat((1+cumnmeas(ii)): (cumnmeas(ii+1)),5);
        yobs(ii).h = fielddat((1+cumnmeas(ii)): (cumnmeas(ii+1)),4);
        yobs(ii).xnative = fielddat(cumnmeas(ii+1), [1 3]);
        yobs(ii).x = (yobs(ii).xnative - dmin(1:2))./drange(1:2);
        yobs(ii).ymean = interp1(hsim, tsimmean(:,1), yobs(ii).h,
            'linear', 'extrap');
        yobs(ii).Sigm = diag(fielddat((1+cumnmeas(ii)):
            (cumnmeas(ii+1)),6).^2);
        yobs(ii).yStd = (yobs(ii).y - yobs(ii).ymean)/tsimsd;
    end

```

4.2.3 Computing the K Basis for Transforming y^{sim} and y^{obs}

Again the multivariate observations and responses are modeled with a linear basis using the K basis. For a compact representation, $p_u < m$ basis functions capture most of the variation in the simulation runs. The choice of how many principal components to use is an issue of experimentation with the variability captured and emulator performance. In this example, three basis functions are used.

```

>> pu = 3; % number of basis components to keep
>> [U, S, V] = svd(tsimStd, 0);
>> Ksim = U(:, 1:pu) * S(1:pu, 1:pu) ./ sqrt(m);
% the pu curves capture variation across simulation runs

```

This K_{sim} matrix of basis elements has $n_\eta = 101$ rows (one for each height in the grid used by the simulator) and $p_u = 3$ columns. A corresponding basis matrix K_{obs} for each experiment in the field data is computed by interpolating the K_{sim} components onto the observation data locations.

```

% now interpolate between height grids to produce a
    corresponding Kobs
>> for ii = 1:n

```

```

yobs(ii).Kobs = zeros(length(yobs(ii).yStd), pu);
for jj = 1:pu % compute for each basis component
    yobs(ii).Kobs(:, jj) = interp1(hsim, Ksim(:, jj),
        yobs(ii).h, 'linear', 'extrap');
end
end

```

4.2.4 Specifying the D Basis for Modeling the Discrepancy Term

The discrepancy term $\delta(x)$ models a systematic bias between the simulator and the experimental observations. A common procedure for fitting the discrepancy is using a normal kernel basis as in Example 1. Here a simple linear discrepancy is used.

```

>> % let's just use a simple linear discrepancy delta(h) = a*h;
>> Dsim = hsim;
>> pv = size(Dsim,2);
>> Dmax = max(max(Dsim * Dsim'));
>> Dsim = Dsim / sqrt(Dmax);
>> for ii = 1:n
    nyobsii = length(yobs(ii).yStd);
    hobsii = yobs(ii).h;
    yobs(ii).Dobs = zeros([nyobsii pv]);
    for jj = 1:pv
        yobs(ii).Dobs(:,jj) = interp1(hsim,Dsim(:,jj),hobsii,
            'linear');
    end
end

```

4.2.5 Package All the Pieces

Having now completed the specification and transformation of required data, it can be collected into a single Matlab structure to be given to GPMSA for model setup. This structure, here called `data`, will contain a field for the simulated data (`simData`) and another for the field data (`obsData`). For both fields, we'll include information that's required by the model as well as extra information (stored in a subfield called `orig`) that will later make it easier for us to return the output to the original scale and to do plots.

```

>> simData.x = design; % our design, standardized
>> simData.yStd = tsimStd; % output, standardized
>> simData.Ksim = Ksim;

% extra fields: original data and transform stuff
>> simData.orig.y = tsim;
>> simData.orig.ymean = tsimmmean;
>> simData.orig.ysd = tsimsd;
>> simData.orig.Dsim = Dsim;
>> simData.orig.h = hsim;
>> simData.orig.xNative = designNative; % original scale
    for simulated R, C

```

For the observed data, each experiment is created separately since each could have a different length.

```
% -- obsData --
>> for ii = 1:n
    % required fields
    obsData(ii).x = yobs(ii).x;
    obsData(ii).yStd = yobs(ii).yStd;
    obsData(ii).Kobs = yobs(ii).Kobs;
    obsData(ii).Dobs = yobs(ii).Dobs;
    obsData(ii).Sigy = yobs(ii).Sigy./ (tsimsd.^2);

    % extra fields
    obsData(ii).orig.y = yobs(ii).y;
    obsData(ii).orig.ymean = yobs(ii).ymean;
    obsData(ii).orig.h = yobs(ii).h;
    obsData(ii).orig.xNative = yobs(ii).xnative;
end
```

4.3 Model Initialization and MCMC

Now that the user setup of data has been completed, we can initialize the model, use the data to compute the posterior distribution of the parameters, and then sample from this distribution via Markov chain Monte Carlo (MCMC). The code in this section is in the MATLAB file `runmcmc.m`.

First we'll call `readdata.m`, which implements the code previously detailed, in order to get the data structure created there; we'll store it in a variable called `towerdat`.

```
>> % read data
>> towerdat = towerreg(1);
```

The initial setup of the model is performed using the GPMSA code function `setupModel()`. The function `setupModel()` takes the `obsData` and `simData` fields from `towerdat`, makes all the structures needed to do MCMC, and returns a structure which we'll call `pout` for “**parameter output**”.

```
>> params = setupModel(towerdat.obsData, towerdat.simData);
SetupModel: Determined data sizes as follows:
SetupModel: n= 3 (number of observed data)
SetupModel: m= 80 (number of simulated data)
SetupModel: p= 2 (number of parameters known for observations)
SetupModel: q= 2 (number of additional simulation inputs
(to calibrate))
SetupModel: pu= 3 (response dimension (transformed))
SetupModel: pv= 1 (discrepancy dimension (transformed))

>> params
params =
    data: [1x1 struct]
    model: [1x1 struct]
    priors: [1x1 struct]
    mcmc: [1x1 struct]
```

```

obsData: [1x4 struct]
simData: [1x1 struct]
optParms: []
pvals: []

```

Fields of `params` include the simulated and observed data transformed by the K and D matrices (`data`), initial values for the parameters of the posterior induced by the model in GPMSA (`model`), priors on the model parameters (`priors`), details (like step sizes) of the MCMC routine for getting draws from the posterior distribution of the parameters (`mcmc`), and the `obsData` and `simData` structures given in the call to `setupModel()`. It also includes a placeholder for the `pvals` field which will hold the MCMC draws.

Next the prior distribution parameters are specified.

```

% require the model stays close to the obs sd of .1 seconds
params.priors.lamOs.params = [10 10];
% allow the white noise component of the W's to get small
params.priors.lamWs.params = repmat([1 .00001],
    [params.model.pu 1]);
% initialize with a small discrepancy error
params.model.lamVz=10000;
% start with the observation precision mulitplier at 1
params.model.lamOs=1.0;
% allow the precision for the discrepancy to get big
params.priors.lamVz.priors = [1 .00001];
params.mcmc.lamVzwidth = 200.0;

```

1. Again as in Example 1, and as an optional step, GPMSA includes a utility `stepsize()` to optimize the MCMC proposal widths, or “step sizes.” Default settings may provide reasonable, although not optimal, performance. Computation of the step size starts by collecting MCMC proposal acceptance statistics at a number of possible values (levels), using estimates constructed from a number of MCMC draws. The code is the same as Example 1 and will not be replicated here.
2. After running the `stepsize()` function, MCMC samples are generated efficiently from the parameters’ joint posterior distribution. Again the code here is the same as Example 1.

4.4 Examining the Estimated Parameter’s Posterior Distribution

All of the diagnostic plots presented in Example 1 can be performed, but here the focus is on a different set from those in the first example.

4.4.1 Joint Posterior Distribution of C and g

In this example, the aim is to learn the joint distribution of the coefficient of drag, C , and the coefficient of gravity, g . The function call `t1Plots(pout, ip, 23)` creates a joint posterior distribution of C and g which can be seen in Fig. 55.11.

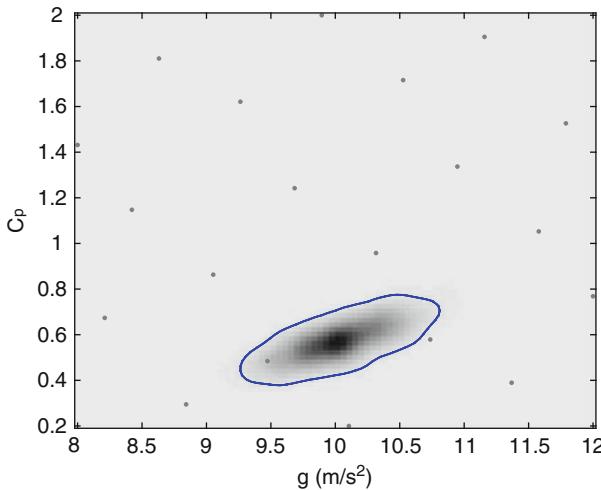


Fig. 55.11 The joint posterior distribution of coefficient of drag, C , and coefficient of gravity, g . The points represent C, g pairs selected in the space filling Latin hypercube design, and the dark contours form the joint posterior of C and g

4.4.2 Parameters Controlling the Gaussian Process Fit

Using the MCMC draws of the spatial dependence parameters $\beta, \rho = \exp\{-\beta/4\}$ is computed. The value of ρ gives information about the dependence of the simulation output on each input parameter \mathbf{x} and $\boldsymbol{\theta}$. Figure 55.12 shows box plots of the posterior draws for the ρ s for each \mathbf{x} and $\boldsymbol{\theta}$ and for each principal component. As above, the figure was generated using a subset of the realizations:

```
>> t1Plots(pout, ip, 1);
```

When ρ is exactly 1 for a particular x or θ principal component, it means that particular component of the simulator output is constant along that dimension. That is, the simulation is not sensitive to that input, and knowing the value of the input gives no information about the value of the output. As ρ goes smaller than 1, this indicates activity associated with that input. The outputs will vary smoothly with the inputs, with smaller values of ρ indicating less smoothness.

4.5 System Predictions

Finally in this experiment, the model is used to predict drop times for each ball type across the heights specified in the emulator. This figure is generated using the following code:

```
>> t1Plots(pout, ip, 3)
```

Note that the estimated discrepancy is nearly zero, so that the prediction uncertainty is accounted for by the model parameters, as well as in uncertainty regarding the

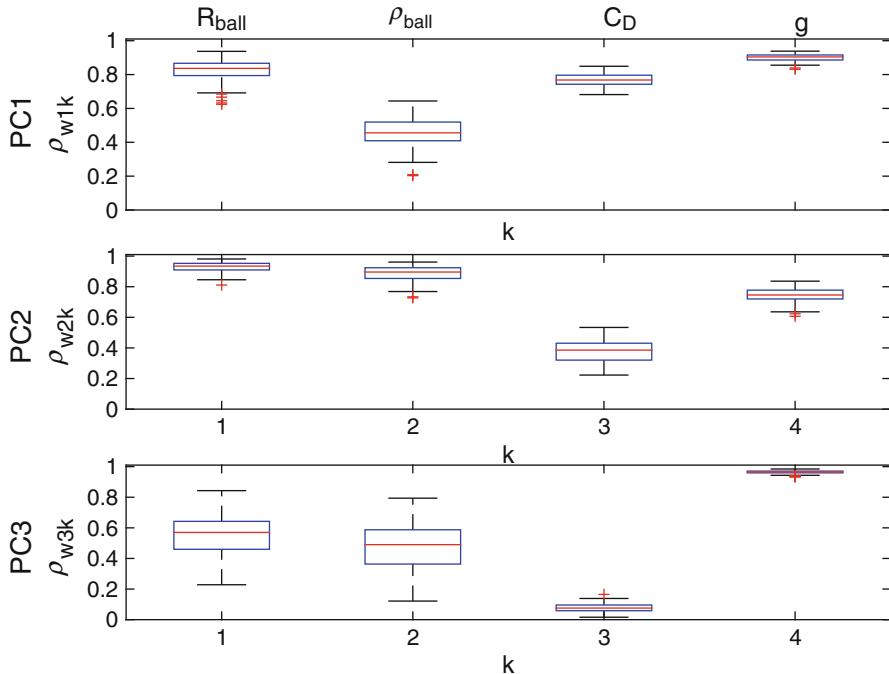


Fig. 55.12 Box plots of $\rho = \exp\{-\beta/4\}$ for the draws of β associated with \mathbf{x} and $\boldsymbol{\theta}$ for each principal component

GP-based response surface. Also, the fact that the design for the simulation runs uses the exact values of R_{ball} and ρ_{ball} for the softball means that the GP interpolation is only over the model parameters $\boldsymbol{\theta} = (C, \theta)$. Hence there will be very little uncertainty due to interpolation, even though the ρ s for the radius and density dimensions can be rather far from 1.

5 Example 3: Ball Drop Exploiting Kronecker-Separable Design

The third example follows the same experimental conditions as Example 2, but takes advantage of the structure present in the input parameters for the emulator. Recall the emulator inputs used a space filling Latin hyper cube to select the twenty pairs for the coefficient of drag and the coefficient of gravity. This dataset was imported directly as a 80×4 matrix with the twenty (C, g) pairs applied to the radius and density of each of the four balls used in the emulator: baseball, basketball, softball, and bowling ball. In this example, two separate datasets are imported, one for the twenty (C, g) and the other with the radius and density for each ball.

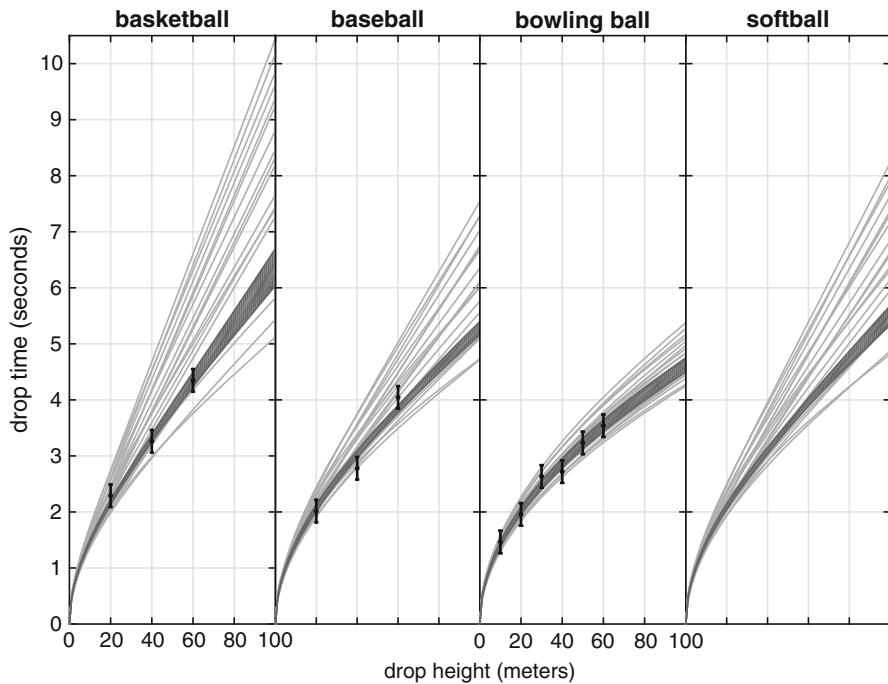


Fig. 55.13 The gray lines represent the emulator runs. The black points represent the experimental observations with uncertainty. The dark bands give a 90% prediction interval produced by propagating the posterior parameter realizations shown in Fig. 55.11 through the emulator

```

>> des1 = textread([dirstr 'desNative4x2Rrho.txt']); % design
    (x=R, rho_ball)
>> des1

des1 =
    0.1200      84
    0.0380     626
    0.1100    1304
    0.0485    380.9
>> des2 = textread([dirstr 'desNative20x2Cg.txt']);
    % design (theta=C,g)
>> des2

des2 =
    1.3368    10.9474
    0.4842     9.4737
    0.7684    12.0000
    0.6737     8.2105
    0.9579    10.3158
    1.1474     8.4211
    0.3895    11.3684

```

```

1.9053    11.1579
2.0000    9.8947
1.6211    9.2632
0.5789    10.7368
0.2000    10.1053
0.2947    8.8421
1.8105    8.6316
1.0526    11.5790
1.2421    9.6842
1.5263    11.7895
1.4316    8.0000
1.7158    10.5263
0.8632    9.0526
>> simData.orig.colmax = [max(des1) max(des2)];
>> simData.orig.colmin = [min(des1) min(des2)];
>> designNative = {des1,des2};

```

The above code is packaged into the `toweregKron.m` Matlab function. By specifying the separable structure, the Matlab function `setupModel.m` recognizes the Kronecker structure.

```

>> % initial set-up
>> params = setupModel(towerdat.obsData, towerdat.simData);
SetupModel: Determined data sizes as follows:
SetupModel: n= 3 (number of observed data)
SetupModel: m= 80 (number of simulated data)
SetupModel: p= 2 (number of parameters known for observations)
SetupModel: q= 2 (number of additional simulation inputs
                 (to calibrate))
SetupModel: pu= 3 (response dimension (transformed))
SetupModel: pv= 1 (discrepancy dimension (transformed))
SetupModel: Kronecker separable design specified

```

After setting up the model, the MCMC algorithm can be initialized using the Matlab function call `gpmcmc.m` just as in Example 2. By specifying the Kronecker-separable design, more efficient matrix algebra techniques can be used to solve the quadratic forms of the matrices, speeding up the log-likelihood evaluations required within the MCMC computations. Note that the `gPredict.m` function does not make use of this structure when producing posterior realizations from the emulator. In the likelihood computation, the computation is usually limited by the number of either observations or simulations, where the computation is $O(n^3)$, quickly dominated by the larger. With a Kronecker-separable design, the computation is $O(n^3)$ dominated by the largest n in the component designs. A design of 10^4 may be intractable to evaluate, but a design of two Kronecker sub-designs each of size 100 will be relatively quick to sample.

6 Example 4: Specifying Priors on C and g

This section extends Example 2, further illustrating how to specify a user-specified prior for the model parameters $\theta = (C, g)$. The “observed” data continues to be noisy realizations from the following ODE:

$$\frac{d^2h}{dt^2} = g - \frac{C}{2} \frac{3\rho_{\text{air}}}{4R_{\text{ball}}\rho_{\text{ball}}} \left(\frac{dh}{dt} \right)^2. \quad (55.14)$$

The experimental design continues to be the same Latin hypercube, which could be decomposed by exploiting its Kronecker structure.

6.1 Prior Specification for g and C

One way to modify priors is to modify their parameters in the `priors` structure, if the prior family does not change. In this example, changing the function that evaluates the log prior to a user-defined function is demonstrated. The prior is specified by defining a function called `logThetaPrior`, which takes as input the parameter $\theta = (C, g)$ and returns the log of the prior density up to an additive constant. For this example, we would like to use the fact that we have very good knowledge about g prior to carrying out this analysis. This is specified by a normal prior distribution for g centered at the gravitational acceleration in Vancouver (9.8134 m/s^2) and with a moderate standard deviation (0.02 m/s^2). A flat, uninformative prior is chosen for the coefficient of drag C . Since $\theta = (C, g)$ is transformed (scaled and shifted according to the min and max of the design range) to reside on the unit interval $[0, 1]^2$, the prior mean and standard deviation are similarly transformed for g to be 0.4533 and 0.005.

GPMSA takes as an argument the name of a function corresponding to this prior with the following line:

```
params.priors.theta.fname='logThetaPrior';
```

This log-prior function is defined in a file called `logThetaPrior.m` with the following code:

```
function lp = logThetaPrior(x,~)
% uniform for C, N(9.8134,.02^2) for g in Vancouver
% on the standardized scale 0 is 8.0, 1 is 12.0
meang = (9.8134 - 8.0)/(12.0 - 8.0);
sdg = .02/(12.0 - 8.0);
lp = - .5/(sdg^2)*(x(2)-meang)^2;
end
```

It requires that the parameter vector be passed to the function. If this function is not provided, GPMSA defaults to specifying independent normal priors $N(0.5, 10)$ for each component of θ , which is very flat over the domain of study in $[0,1]$.

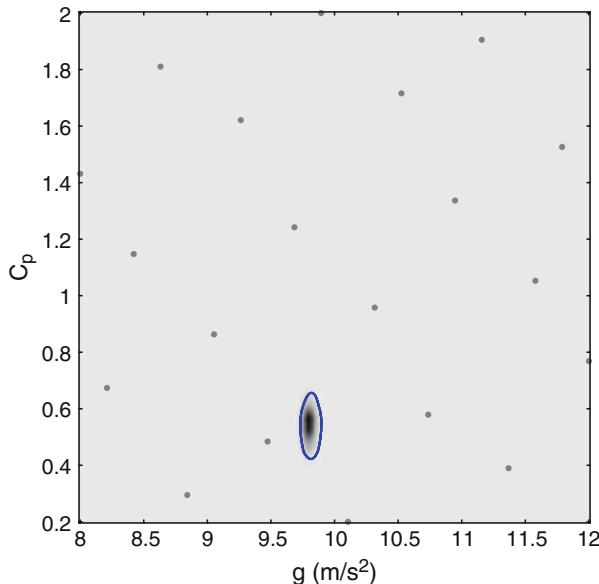


Fig. 55.14 Posterior distribution from Example 4 for the parameter vector θ . Notice that the contours are much narrower after supplying information through the customized prior

By introducing an informative prior, improvements can in principle be made in estimation of the parameters of interest $\theta = (C, g)$. Figure 55.14 shows how doing so results in a tighter posterior distribution, not only for the parameter given a strong prior but also its covariate.

7 Example 5: Inferring the Type of Ball Dropped

In this example, the potential of GPMSA to infer the type of ball dropped (basketball, baseball, bowling ball, or softball) from a new set of drop time data is explored. The general approach is to pose the question as a calibration problem; the unknown example has a category that is to be inferred, with uncertainty, by the GPMSA model. This follows on from the analysis in Example 2, producing a posterior distribution for $\theta = (C, g)$. For this new analysis, the posterior from Example 2 is taken as the prior, specifying a custom prior for (C, g) . This prior is well approximated by the following distribution:

$$\begin{pmatrix} C \\ g \end{pmatrix} \sim N \left(\begin{pmatrix} 0.125 \\ 0.5214 \end{pmatrix}, \begin{pmatrix} 0.0020504 & 0.00283 \\ 0.00283 & 0.007363 \end{pmatrix} \right) \quad (55.15)$$

Given this information, there is a new set of three drop times at heights 20, 40, and 60 m (see Table 55.1).

Table 55.1 Field data for Example 5 with unknown ball type

Height (m)	Time (s)	σ
20	2.1502001	0.1
40	3.1681314	0.1
60	3.9311893	0.1

The aim is to infer which ball was used in these experiments. Thus a categorical latent parameter (ball type) is estimated from these data. Technically, the strong prior for (C, g) expressed above is also updated. Thus, there now is a three-dimensional parameter $\theta = (C, g, \text{balltype})$, where the third component of θ is categorical, taking on values 1, 2, 3, or 4.

7.1 Specifying a Categorical Ball-Type Parameter in GPMSA

The indicator for ball category ranges from 1 to 4, corresponding to basketball, baseball, bowling ball, and softball, respectively. In GPMSA, categorical parameters are specified using the `optParms.catInd` (optional parameters category indicators) argument when calling the function `setupModel`.

In this example, the argument is set to `optParms.catInd = (0, 0, 0, 4)`, a four-dimensional vector with zeros in the first three entries corresponding to the first three entries of (x, θ) which are not categorical. The last entry of `optParms` is 4, indicating that there are four possible categories in the final unknown parameter of (x, θ) . It is worth noting that the first dimension of `optParms.catInd` is a “dummy” variable relating to x , which is set to a constant 0.5 for all instances in the observed and simulated data. GPMSA requires that the length of `optParms.catInd` equals $p + q$, the dimension of (x, θ) . In this case, this is 4, since the dimension of the dummy variable x is 1, while that of $\theta = (C, g, \text{balltype})$ is 3.

For categorical parameters, each iteration of the MCMC algorithm proposes a new category by uniformly randomly sampling from the set of possible categories $\{1, 2, 3, 4\}$, but excluding the current state. That is, if the current state of the sampler is k , the proposal is uniformly generated from $\{1, 2, \dots, k-1, k+1, \dots, K\}$, where in this case $K = 4$. Then the proposal is accepted or rejected according to the usual Metropolis decision rule.

Figure 55.15 shows the posterior distribution of ball type, clearly identifying the fourth category – softball – as the most probable candidate. The estimated posterior probability of the observed data being generated by a softball is 0.67, demonstrating that GPMSA can successfully identify the true underlying ball type. Note that a prediction can also be produced, just as before, but now accounting for the uncertainty regarding the ball type with the call `t1Plots(pout, ip, 32)` as given in the file `runmcmcBallEst.m`.

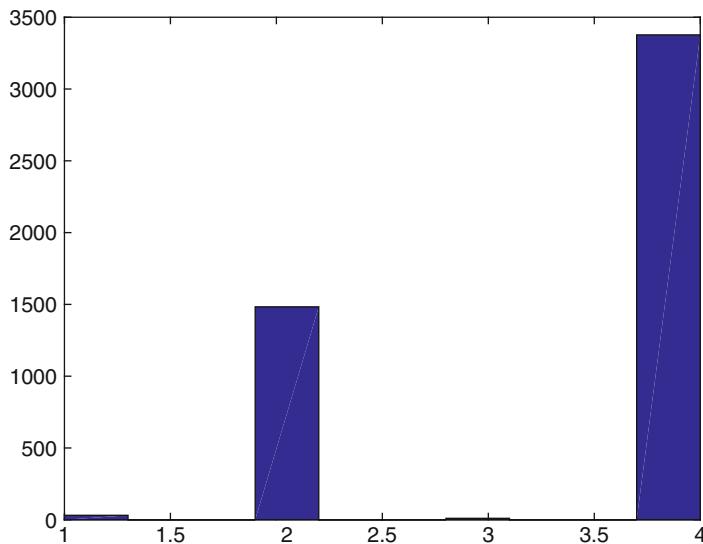


Fig. 55.15 Posterior draws of ball-type category

8 Conclusion

This chapter has discussed the nature of multivariate Bayesian model calibration using a Gaussian process emulator and how this has been implemented in the GPMSA code. This methodology and software have been applied successfully to a number of different examples, where complex scientific phenomena are studied using complex computer models.

References

1. Graves, T.L.: Automatic step size selection in random walk metropolis algorithms. arXiv preprint, arXiv:11035986 (2011)
2. Higdon, D.: Space and space-time modeling using process convolutions. In: Anderson, C., Barnett, V., Chatwin, P.C., El-Shaarawi, A.H. (eds.) Quantitative Methods for Current Environmental Issues, pp. 37–56. Springer, London (2002)
3. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
4. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**, 409–423 (1989)

COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management

56

Edoardo Patelli

Abstract

Computer-aided modeling and simulation is now widely recognized as the third “leg” of scientific method, alongside theory and experimentation. Many phenomena can be studied only by using computational processes such as complex simulations or analysis of experimental data. In addition, in many engineering fields computational approaches and virtual prototypes are used to support and drive the design of new components, structures, and systems. One of the greatest challenges of virtual prototyping is to improve the fidelity of the computational analysis. This can only be achieved by explicitly including variability and uncertainties from different sources. Variability is inherent in many natural systems and therefore cannot be reduced. Uncertainty is also always present since it is not possible to perfectly model or predict future events for which no real-world data is available.

Although stochastic methods offer a much more realistic approach for analysis and design, their utilization in practical applications remains quite limited. One of the reasons is that the developments of software for stochastic analysis have received considerably less attention than their deterministic counterparts. Another common limitation is that the computational cost of stochastic analysis is often by orders of magnitude higher than the deterministic analysis. Hence, robust, efficient, and scalable computational tools are necessary, i.e., by making use of the computational power of a cluster and grid computing.

This chapter presents the COSSAN project: a developed multidisciplinary general-purpose software suite for uncertainty quantification and risk analysis. The computational tools satisfy the industry requirements regarding usability, numerical efficiency, flexibility, and scalability. The software can be used to solve a wide range of engineering and scientific problems. The availability of such software is particularly important for the analysis and design of resilient

E. Patelli (✉)

Institute for Risk and Uncertainty, University of Liverpool, Liverpool, UK

e-mail: edoardo.patelli@liverpool.ac.uk

structures and systems. In fact, despite the different levels of uncertainty, decision makers still need to take clear choices based on the available information. They need to trust the methodology adopted to propagate the uncertainties through multidisciplinary analysis, in order to quantify the risk with the current level of information and to avoid wrong decisions due to artificial restrictions introduced by the modeling.

Keywords

Aleatory and epistemic uncertainty • Computational methods • High-performance computing • Imprecise probability • Matlab • Monte Carlo simulation • Open source • Rare events • Reliability-based optimization • Risk analysis • Robust optimization • Sensitivity analysis • Uncertainty quantification

Contents

1	Introduction	1910
1.1	Background and Motivations	1911
1.2	Importance of Stochastic Analysis	1913
1.3	Needs of an Innovative Software	1914
2	The COSSAN Project	1916
2.1	Overview	1916
2.2	COSSAN-X	1920
2.3	Technical Features	1925
2.4	OPENCOSSAN: An Open-Source Matlab Toolbox	1936
2.5	Engineering Cloud	1941
3	Case Studies	1943
3.1	Application to the Robust Design of a Twin-Jet Aircraft Control System	1943
3.2	Large-Scale Finite Element Model of a Six-Story Building	1952
3.3	Robust Design of a Steel Roof Truss	1957
3.4	Robust Maintenance Scheduling Under Aleatory and Epistemic Uncertainty	1964
4	Conclusions	1970
	References	1972

1 Introduction

Knowledge about the future behavior of engineered systems is the basis for reaching economical and safety relevant decisions in our society and appears in different fields (e.g., automotive and aerospace industry, financial, environmental science, mechanical and energy sector). Together with observed responses, it provides the basis to broaden the understanding between action and reaction. In order to predict accurately the behavior of such systems and/or structures, mathematical models must be constructed and then evaluated. In an increasingly competitive market, engineers are asked to design products faster with rapid prototyping that can be achieved only through computational models and numerical simulations. In fact, nowadays, in many engineering fields computational approaches and virtual prototypes are used to characterize, predict, and simulate complex systems (see

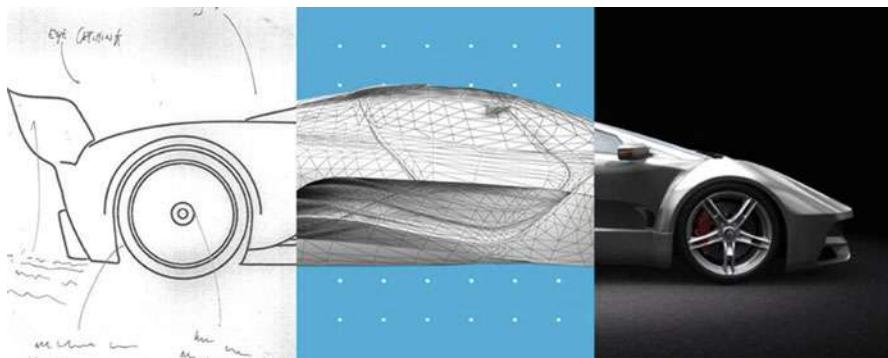


Fig. 56.1 Example of a development strategy by means of virtual prototypes (Courtesy of the Virtual Engineering Centre)

Fig. 56.1). These advancements have allowed engineering practitioners to reduce the number of expensive and destructive tests necessary to qualify new products. In fact, the performance of these products can be tested in a simulated environment and the necessary changes introduced before producing a physical item, reducing the overall development cost and time. Generally any product, component, or system is designed and optimized, in other words engineered, to fulfill requirements and codes, to improve its performance, and to reduce production and maintenance costs.

Since the industrial designs are subject to strict safety, reliability, environmental, and service requirements, the quantification of uncertainties and risks is a necessity and a challenge, needing for innovative software that allows the inclusion of nondeterministic analysis as a practice standard routing in the virtual prototyping.

1.1 Background and Motivations

The continuous advancements in modeling tools allow for very accurate reproduction of the behavior of components and systems at multiple scales. In addition, the exponential growth of computational power allows the analysis for a level of detail and precision that could have not been reached previously. However, even with very advanced models and accurate analysis, a comparison of response predictions with measured data can show an incomplete agreement. The reasons for these discrepancies are the uncertainties in model parameters and in the model itself (see Fig. 56.2). Parameter uncertainty denotes input data in the computational model which are not precisely known and are expected to deviate from the assumed deterministic values. Model uncertainty denotes the fidelity of the mathematical models, which usually involves some abstractions, simplifications, or assumptions to represent with sufficient accuracy the actual mechanical/physical responses. Nowadays, it is widely recognized that essential progress in model prediction can only be accomplished when different sources of uncertainty are explicitly included in the analysis [13, 44, 80].

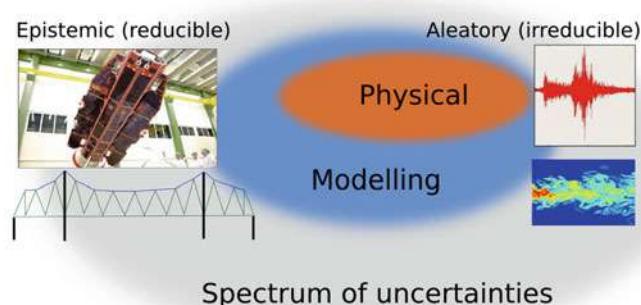


Fig. 56.2 Spectrum of the uncertainties: aleatory (irreducible) uncertainty and epistemic (reducible) uncertainties

Uncertainties are generally classified in two categories: *aleatory* and *epistemic*. Aleatory uncertainty represents the variability of true random or uncontrollable processes (e.g., earthquakes, wind loads, climate change, etc). This kind of uncertainty is irreducible because it is inherent of the system or process (e.g., it is not possible to predict the occurrence and intensity of an earthquake). Epistemic uncertainty represents limited data, limited knowledge, as well as model imprecision due to assumption and simplification. Epistemic uncertainty can be reduced, at least theoretically by collecting new data or using more detailed models. In practice, such new data is either very scarce or impossible to collect and detailed models are difficult to calibrate and validate. These two unavoidable sources of uncertainties (Fig. 56.2) must be appropriately accounted for to guarantee that the components or systems will continue to perform satisfactory despite fluctuations, i.e., the design has to be “robust” (see, e.g., [79]). Ignoring the effects of the uncertainties and/or not including them at the design stage might lead to a poor or unsatisfactory design. For instance, a product may perform well in the laboratory tests but perform unsatisfactory under realistic conditions.

In addition, recent reports have clearly shown that the risk assumed by the decision maker is often wrongly estimated due to inadequate assessment of uncertainties (see, e.g., [50]). Modeling and simulation standards require estimates of uncertainty (and descriptions of any processes used to obtain these estimates) in order to increase confidence and consistency in safety predictions and encourage the development of improved methods for quantifying and managing uncertainty. Hence, uncertainty management is necessary to provide support to the decision makers through a series of different and interconnected analyses. For instance, estimating the importance of collecting additional information allows to characterize and reduce uncertainty; by performing sensitivity analysis, it is possible to identify the parameters that contribute the most to the variability of the output; uncertainty propagation allows to study the effects of uncertainty on the performance of the

system and to identify extreme-case scenarios. Finally, optimizing the design explicitly taking into account the effect of uncertainties allows to design a robust system.

1.2 Importance of Stochastic Analysis

It is quite well accepted that deterministic analyses provide insufficient information to capture the variability of the quantity of interest, while stochastic analysis has been proved to provide a more realistic description taking explicitly into account the effect of uncertainties (see, e.g., [84]).

The merits of considering uncertainties are manifold: it allows for assessing the reliability and variability of the responses, and most importantly, it provides more realistic predictions and information to improve the design. For instance, sensitivity analyses reveal the quantities which are mainly responsible for the variability of the quantity of interest. In case the uncertainty is due to the lack of knowledge (epistemic type) and therefore reducible, the fidelity of the prediction can be improved by gathering additional data for those quantities which cause the most uncertainty in the response(s). On the other hand, irreducible (aleatory) uncertainties lead to irreducible uncertainties in the response and the design must be robust such that adverse events do not jeopardize safe operation.

Similarly, uncertainty quantification and propagation are important aspects when trying to optimize a system or a component. Optimal solutions obtained in a deterministic setting might not perform as expected and can even be dangerous in cases where ignored uncertainties influence the performance considerably. On the other hand, robust design procedures take into account all relevant uncertainties and provide robust and sound solutions, e.g., the failure probability is constrained to be less than an acceptable value [33, 36, 64, 86]. Furthermore, decisions within life cycle management for important infrastructures and investments must be made based on incomplete and generally insufficient data, where a probabilistic Bayesian approach, imprecise probability, and fuzzy methods could provide valuable information [6, 10, 11, 28].

The most widely accepted method to deal rationally with uncertainties is the stochastic approach that includes Bayes and Laplace's subjective interpretation of probability as a state of information [26, 42]. In the stochastic approaches, uncertainties are represented mathematically by random quantities and by suitable probability distributions. However, many of the uncertain phenomena are non-repeatable events. In this case, uncertainty is not embodied by intrinsic or aleatory randomness, but by the lack of knowledge or epistemic uncertainty about the phenomenon. The insight in the underlying physics can also be limited or vague. In other words, the knowledge about the phenomena is not in a complete form that allows the construction of the probability distribution or relevant quantities in the context of classical probability theory. Information may statistically include expert assessments, opinions, or team consensus. Bayesian statistics allows for a rational and systematic treatment of this kind of uncertainty, interpreting probability as a degree of belief, which is by nature subjective. Alternative approaches are

valid options, e.g., fuzzy logic [48], imprecise probability, and possibility theory, which are not so far developed as the theory of probability. To avoid the inclusion of subjective and often unjustified hypothesis, the imprecision and vagueness of the data can be treated by using concepts of imprecise probabilities. Imprecise probability combines probabilistic and set theoretical components in a unified construct (see, e.g., [1]). It allows a rational treatment of the information of possibly different forms without ignoring significant information and without introducing unwarranted assumptions. In the analysis, imprecise probabilities combine, without mixing, randomness and imprecision. Randomness and imprecision are considered simultaneously but viewed separately at any time during the analysis and in the results. The probabilistic analysis is carried out conditional on the elements from the sets, which leads eventually to sets of probabilistic results. This can support economical and safety relevant decisions in many different fields that must be done based on incomplete and generally insufficient data. For instance, interval analysis is a useful tool to explore a variety of set-valued descriptions (nested set of sets), for example, in design problems. These options can be combined with one another to suit the problem. In any case, no assumption is made regarding a distribution of probability over a set. Instead, each and every element from a set is considered as plausible with no weighting with respect to one another.

1.3 Needs of an Innovative Software

The quantification of uncertainties and risks is a key requirement and challenge across various disciplines in order to operate systems of diverse nature safely under the evolutionary dynamics of inputs and boundary conditions. These systems include engineering, infrastructural, and financial systems among others. Industry is fully aware of the truly significant potential of nondeterministic analysis and advanced simulation-based tools for many application fields with large-scale benefits. In fact, in industrial designs that are subject to strict safety, reliability, environmental, and service requirements, the quantification of uncertainties and risks is a necessity and a challenge. Although powerful mathematical basis for comprehensive uncertainty and risk quantification does exist, in practice, expertise in uncertainty quantification is generally not present in industrial design offices. A key issue for this delay in transfer of knowledge and computational technologies into industry on a large scale is the lack of proper software. Consequently, industrial design methods are still predominantly deterministic.

Software for stochastic analysis have received considerably less attention than their deterministic counterparts, and in addition, the computational cost of stochastic analysis is often by orders of magnitude higher than the deterministic analysis. This is because instead of performing the deterministic analysis once by running a detailed model (e.g., FE or CFD model), multiple runs of the deterministic model are required as shown in Fig. 56.3. In general, deterministic analysis provides a map between a single point in the input space (i.e., the model parameters) and a point in the output space (i.e., component or system performance). Stochastic analysis

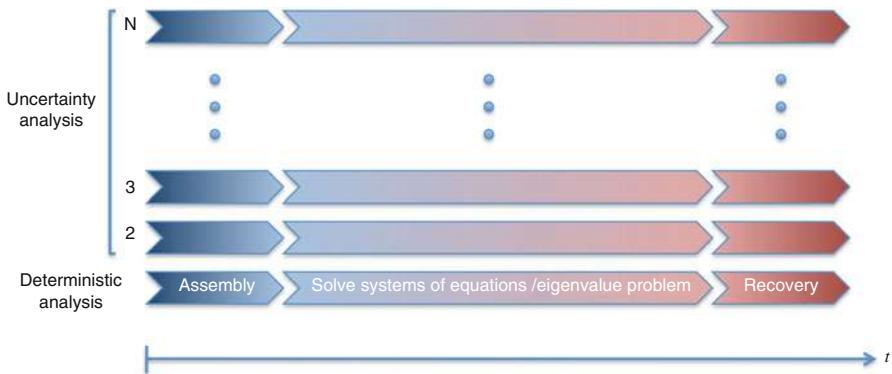
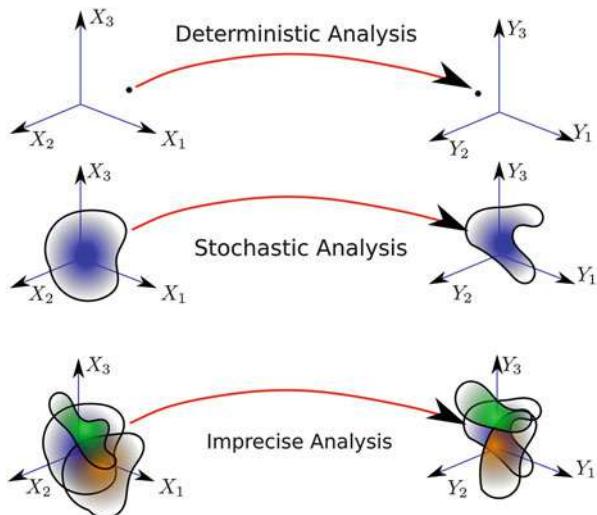


Fig. 56.3 Computational cost of the stochastic analysis versus deterministic analysis. Deterministic analysis requires assembling the model, solving a set of system equations, and then computing the quantity of interest. Uncertainty analysis requires N deterministic analysis

Fig. 56.4 Schematic representation of the different analyses: deterministic analysis provides a map between a point in the input variable space and a point into the output variable space; stochastic analysis maps an area in the input space with an area in the output space and imprecise analysis maps sets in the input space with sets in the output space



extends this map to a region in the input space and a corresponding region in the output space by repeating the deterministic analysis many times as represented in Fig. 56.4.

Recent emerging techniques to deal rationally with uncertainties, such as generalized probabilistic methods, including Bayesian approach fuzzy logic and possibility theory, introduce another layer of computational complexity [14]. These generalized probabilistic models require the evaluation of sets of possible probabilistic models (see Fig. 56.4) with even higher computational costs which might lead to impractical computational costs especially for detailed models. A comparison and details of these techniques can be found in Ref. [73].

In order to include stochastic analysis as standard procedure in engineering practice, it is of paramount importance the availability of innovative software that allows considering explicitly the effects of uncertainties. This software needs also to implement efficient simulation and parallelization strategies allowing a significant reduction of the computational costs of the nondeterministic analyses. Finally, such software should allow the analyst to perform stochastic analyses using the same software and tools used for the *deterministic* design in order to reduce the learning curve. The lack of easy-to-use nondeterministic analysis software, for solving large-scale problems, has motivated the COSSAN project. The current version of the COSSAN software meets these requirements as it will be shown in the next sections.

2 The COSSAN Project

2.1 Overview

The COSSAN project aims at developing a new generation of a general-purpose software for nondeterministic analysis that can be used by industry, academics, and researchers and for teaching purpose as well. The software incorporates the knowledge, understanding, and intellectual property from more than 30 years of research in the field of computational stochastic analysis. The COSSAN software is based on the original development by the group of Prof. Schuëller at the Institute for Engineering Mechanics, University of Innsbruck, Austria [85]. Originally, it was designed to perform stochastic structural analysis only [81] as the software name was indicating (COmputation Stochastic Structural ANalysis).

Starting from 2006, the next-generation software referred to as COSSAN-X is under continuous development, and it is intended for a wider range of applications in different fields, which includes optimization analysis, life cycle management, reliability and risk analysis, sensitivity, optimization, and robust design [65]. The current version of the software is hosted at the Institute for Risk and Uncertainty at the University of Liverpool, UK, and led by Edoardo Patelli. In addition, since 2012, an open-source version of COSSAN-X, called OPENCOSSAN, is available under the GNU Lesser General Public License [30]. This means that the program can be used for free, redistributed, and modified under the terms of the GNU Lesser General Public License. The OPENCOSSAN aims to promote learning and understanding of nondeterministic analysis through the distribution of an intuitive, flexible, powerful, and open computational toolbox in Matlab environment [67]. Recently, COSSAN-X has been integrated into the Engineering Cloud developed and led by the Virtual Engineering Centre (www.virtualengineeringcentre.com). Engineering Cloud is offering *Stochastic Analysis on Demand*, enabling small and medium enterprises to access high-performance computing resources and software capabilities and to cut capital investment requirements for hardware and software purchase.

2.1.1 General-Purpose Software

A framework is a software which provides generic functionalities and can be changed and expanded through user-defined routines and additional codes. In this optic, COSSAN is a *general-purpose computational framework*. Generally highly specialized software are developed to solve very specific problems. These tools can be very efficient and compact, but their applicability remains very limited and the solution for a different problem will require the redevelopment of the entire computational framework. The term *general purpose* means that a reasonably wide range of engineering and scientific problems can be treated by a single software. Such software packages are much more flexible than specialized software, which are developed to solve a specific type of problem within a particular discipline. The complexity of the general-purpose software packages in terms of number of lines of code, structure, and time required for developing and testing represents the major drawback of this kind of software when compared with dedicated software. However, they are developed in a single effort and then they can be used to solve a broad variety of problems. In addition, general-purpose software are usually much simpler to use and thus they can be adopted by less skilled users, resulting in a drastic reduction of analysts' training time required to familiarize with the software.

2.1.2 Historical Developments

Historically, the first developments toward a stand-alone software led to the package ISPUD, an acronym for “Importance Sampling Procedure Using Design Points” (see Fig. 56.5). ISPUD was a multipurpose program package for performing structural reliability analysis. The failure probability was calculated by integrating the joint probability density function using numerical procedures, such as Monte Carlo simulation, and in particular importance sampling around design points [15, 83].

At the beginning of the year 1990, the development of the stand-alone toolbox for structural analysis started, providing a data management system and a command interpreter [17]. The first release of COSSAN is referred to as the COSSAN-A in the development line (see Fig. 56.5). COSSAN-A was an open system, designed to be easily adjustable and expandable to include new computational tasks. Each problem solution was broken down into a set of specific commands, each performing a uniquely defined computational task, denoted as a module. The need to operate

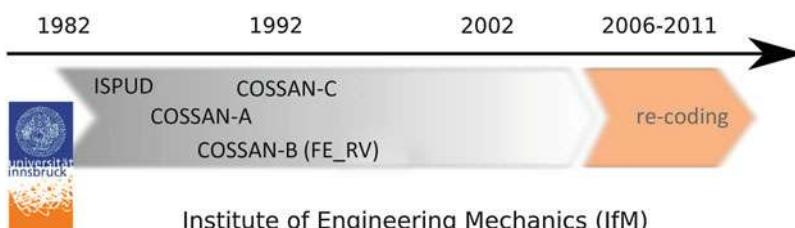


Fig. 56.5 Historical development of the COSSAN software led by Prof. Schuëller at Institute for Engineering Mechanics, University of Innsbruck, Austria

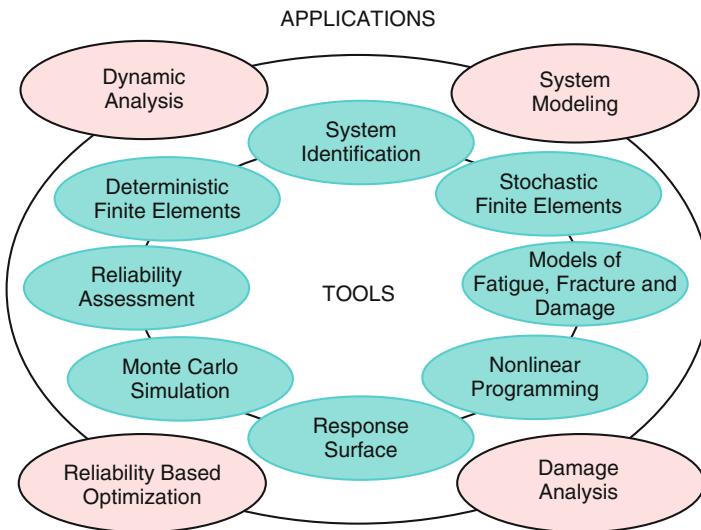


Fig. 56.6 Range of analysis capabilities featured by COSSAN-A: a stand-alone software for computational structural analysis developed in the early 1990s

directly with modules aimed to give the user explicit control over the sequence of specific tasks to be performed. Such extensive control provided substantial advantages to developers who want to expand available capabilities to solve their specific problems. The structure of COSSAN-A is as follows. On the very top of the stand-alone toolbox, there is an event-driven loop of the graphical user interface (GUI). This administration package provides all interactive capabilities of COSSAN-A. The next layer of the toolbox is the command interpreter, which translates and executes a sequence of COSSAN-A commands. The sequence of commands read by the COSSAN-A input file can be controlled by conditional and unconditional jumps, allowing for loops and repeated calls of command sequences. The stand-alone toolbox was composed by 32 module groups and 218 modules. These modules include a library of the most common finite elements, which are needed to perform Monte Carlo simulation. Figure 56.6 shows an overview of the analysis tasks and applications which can be treated within the stand-alone toolbox.

Starting in the mid-1990s, a novel approach to software development in stochastic structural analysis was undertaken: in order to capitalize on existing, widespread general-purpose FE solvers, communication tools started to be developed. These tools were collected in the so-called COSSAN-B development line and merged in the currently operational code FE_RV [18]. The modus operandi of FE_RV is depicted schematically in Fig. 56.7. The central portion (in blue) relates to the user interface; it mainly represents the user-defined specification of the probabilistic model. On the left side, the set of routines implementing the actual probabilistic methods are indicated in green. These routines may be coded in the preferred programming environment, such as PERL, Matlab, C++, etc. As shown in the figure, these engine

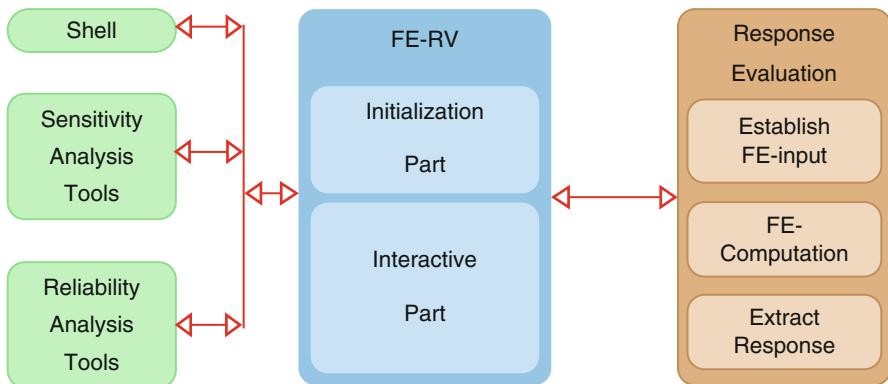


Fig. 56.7 Modus operandi of COSSAN-B, a software for communication with third-party software



Fig. 56.8 Current development of the COSSAN software at the Institute for Risk and Uncertainty at the University of Liverpool, UK

routines encompass sensitivity analysis and, most importantly, reliability analysis methods such as Monte Carlo simulation and advanced simulation methods. The numerical data obtained by these procedures are then transferred to third-party FE codes, represented by the rightmost box (in brown).

2.1.3 Development of the Next-Generation Software

The redevelopment of the COSSAN software started in 2006, in order to merge all previous versions in a more sustainable single software called COSSAN-X (see Fig. 56.5). The associated development efforts aimed at capitalizing on the highly developed, third-party codes for the computational analysis, while using advanced communication tools to interact with the commercial third-party programs and to create a general-purpose software able to solve a number of different problems. The developments aim to create a multidisciplinary software that satisfies industry requirements regarding numerical efficiency and analysis of detailed models.

Since 2011 the development of the COSSAN software is hosted at the Institute for Risk and Uncertainty at the University of Liverpool, UK (see Fig. 56.8). The current

development of COSSAN at the University of Liverpool is led by Edoardo Patelli and supported by Matteo Broggi. The current version of the software implements innovative developments in the computational algorithms, emerging concepts in stochastic mechanics and robust design to cope with model uncertainties, errors in modeling and measurements, and noise in signals [57, 72, 95]. A key element of the software is a comprehensive risk management and uncertainty quantification based on different representations of the uncertainties based on probabilistic approaches, interval and fuzzy methods, imprecise probabilities, and any combination thereof [14].

2.2 COSSAN-X

COSSAN-X represents the latest generation of the COSSAN software. It has been designed to meet the requirements of industry and academics, providing an easy access to the state-of-the-art methodologies and algorithms for stochastic analysis. In fact, it comes with a powerful and easy-to-use interface, allowing a straightforward interaction with third-party (deterministic) software (e.g., finite element solvers), and high-performance computing and data management.

The structure of the software is composed by three main blocks: user interfaces, core components, and the interaction with external code (i.e., third-party software). Each of these main blocks can be formed by a number of additional subcomponents. A scheme of the general-purpose software is shown in Fig. 56.9.

2.2.1 User Interface

COSSAN-X provides a very powerful, interactive, and user-friendly interface (Fig. 56.10). Developed in Eclipse RCP, the user interface provides the state-of-the-art wizards and graphical tools, which construct a comfortable platform to perform the various analyses and applications offered by the program, without excessive training. Available in all operating systems (i.e., Windows, Linux, and Mac OSX), COSSAN-X is designed to provide guidance to its users at every step of the analysis and to assist the inexperienced users in the selection of the most appropriate tools required for the analysis of the problem at hand (Fig. 56.11). In this regard, the user is provided with the necessary warning/error messages, as well as help icons, which enables an easy access to the associated user manual pages (Fig. 56.12).

2.2.2 Interaction with Third-Party Software

The ability to interact with (deterministic) third-party software is a critical point, since the analyst aims to solve the stochastic problem using the same models that they are already familiar with. COSSAN-X interacts with external solvers using a nonintrusive approach derived from the FE_RV code [18]. As shown in Fig. 56.13, the numerical data produced by COSSAN-X are transferred to the external solver by manipulating an ASCII input files. Then, the solver is executed and the output files are generated. Finally, the quantities of interest are from the solver output files (*extracting* data) and passed back to COSSAN-X.

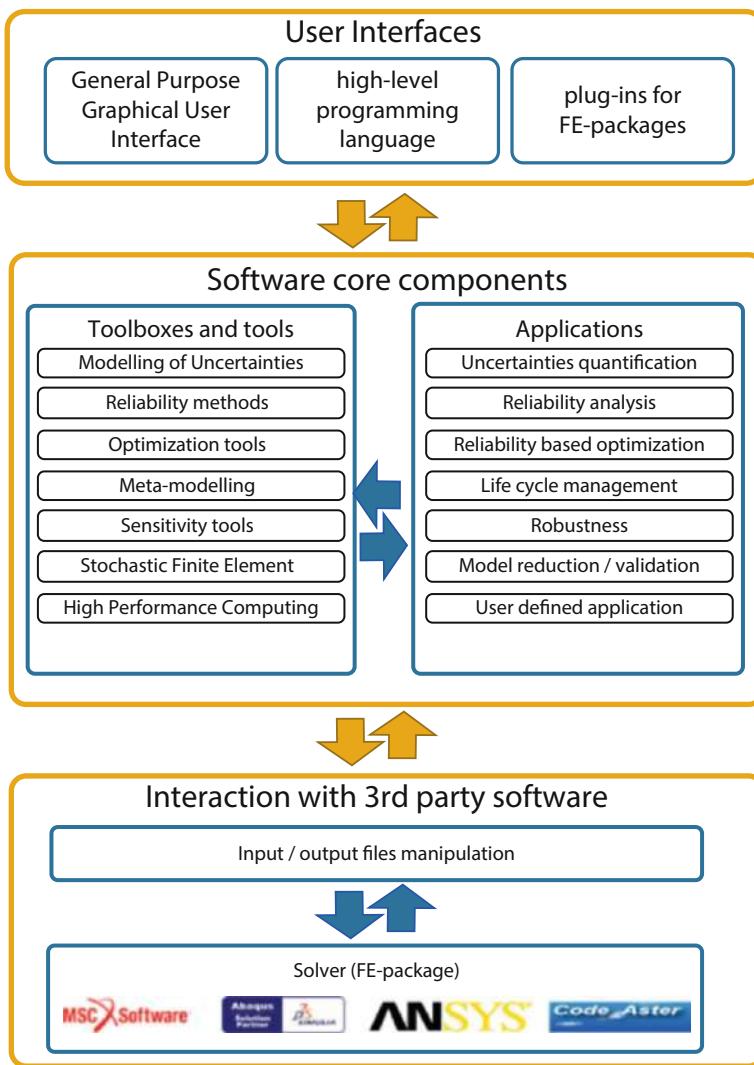


Fig. 56.9 Schematic representation of general-purpose software for computational stochastic analysis

For each realization of random variables, parameters and design variables, etc., the solver input files are modified (i.e., *injecting* data). This approach is very convenient as it allows interaction with any external software without the necessity to write dedicated interfaces. Quantities defined in COSSAN-X are linked with input parameters of the solver by means of XML tags included in the ASCII input file. XML stands for EXtensible Markup Language and it is a software- and hardware-

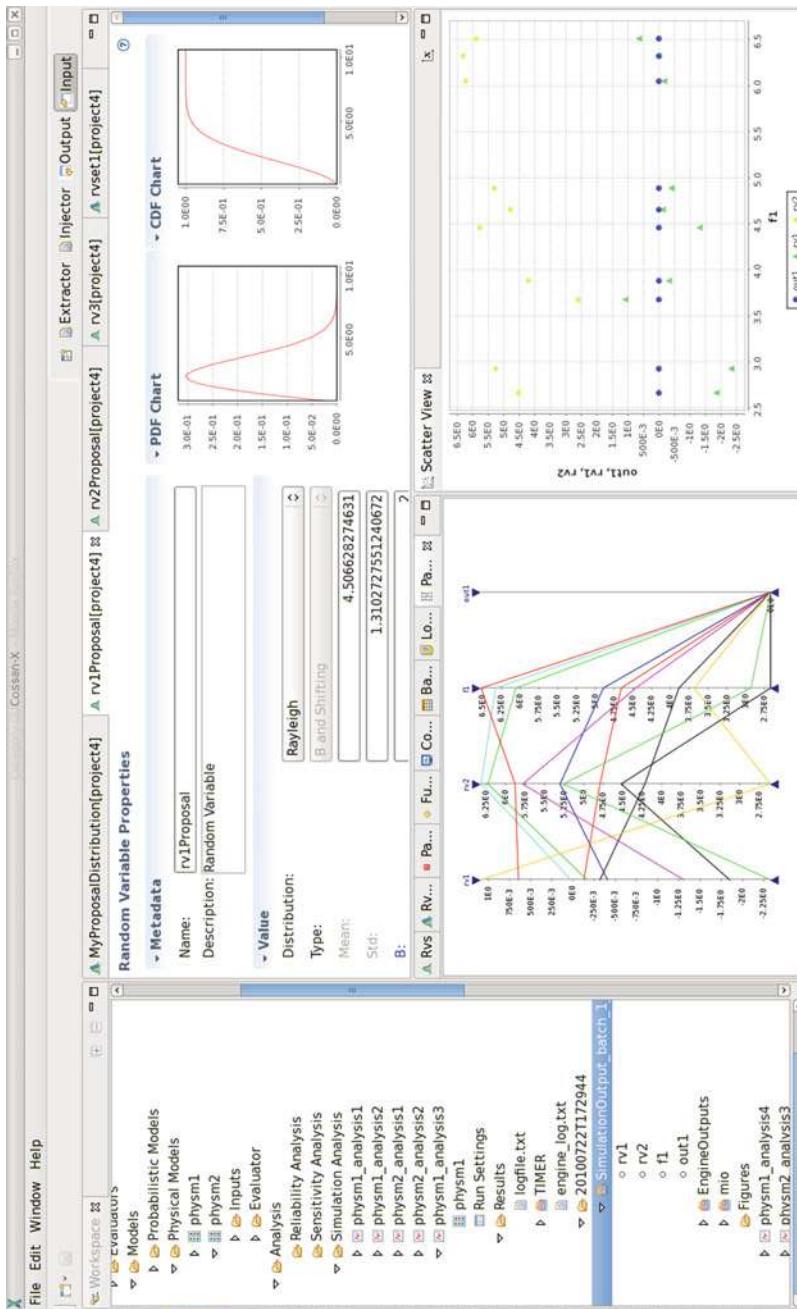


Fig. 56.10 COSSAN-X user interface. The screenshot shows the workbench of COSSAN-X that contains editors to define objects (to define objects, input, and models), wizards to define analysis, visualization tools to show the results, and a workspace to manage and organize the different parts of the analyses

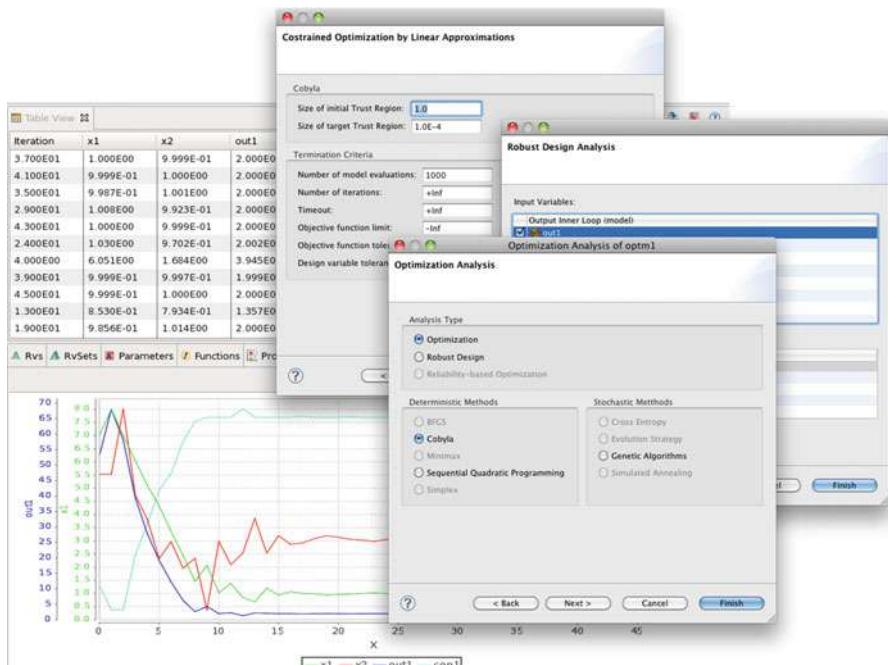


Fig. 56.11 COSSAN-X user interface: wizards and visualization tools. Example of optimization wizard where the available optimization methods are suggested together with the optimization settings. The plot on the *bottom* shows the evolution of the design variables and objective functions during an optimization analysis

independent tool designed for carrying information. The COSSAN-XML tag is self-descriptive and contains the attributes shown in Table 56.1.

The quantity of interest of the analysis is imported in COSSAN by reading ASCII output files generated by the third-party solver. This is done by defining the position of the quantities of interest in the output files. The position can be absolute or relative to some string (called anchor).

The user interface of COSSAN-X offers a very powerful editor, which makes the manipulation of the input/output files for third-party solvers an easy task. It allows to create and include the XLM tag in the input files without the necessity to manually edit the input files as shown in Fig. 56.14. The definition of the position of the quantity of interest can also be defined by means of the user interface as shown in Fig. 56.15.

2.2.3 Core Components

The core components of COSSAN-X are provided by OPENCOSSAN that represents the computational engine of the software. OPENCOSSAN offers the most advanced and recent algorithms for the rational quantification and propagation of uncertainties that have been shown to represent a robust and efficient approach for the uncertainty

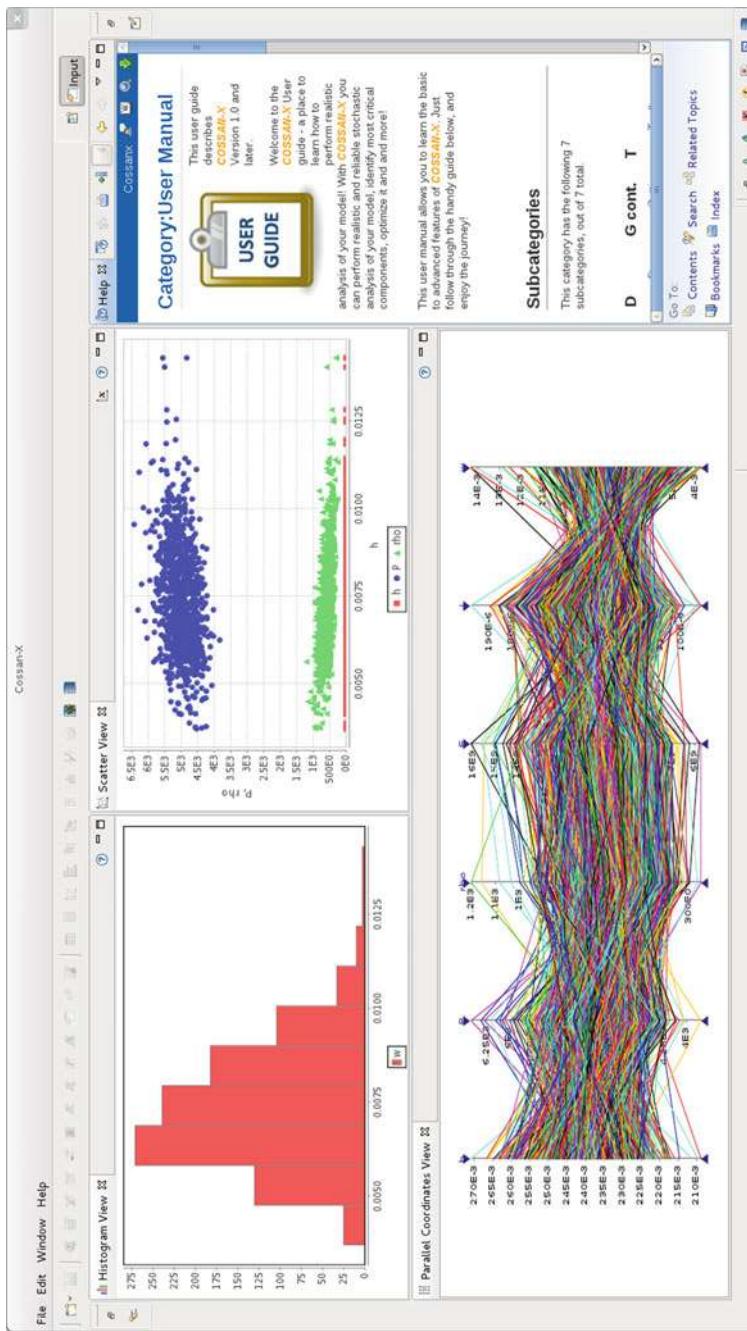


Fig. 56.12 COSSAN-X user interface: visualization tools and embedded help available via the user interface. Example of a histogram (*upper-left window*), scatterplot (*upper-right window*), and parallel coordinate view (*bottom window*). The *right-hand side column* shows the embedded help

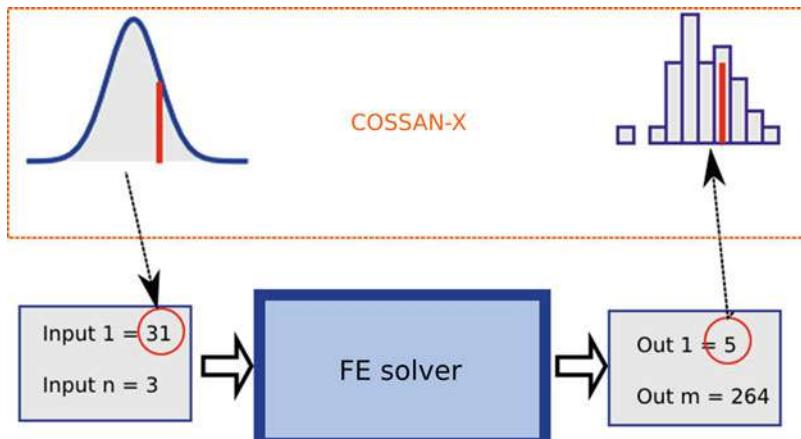


Fig. 56.13 COSSAN-X: interaction with external solvers. Realizations of input variables generated by COSSAN (e.g., Input 1=31) are written into the input file of the FE solver. Then, the FE analysis is performed and the quantity of interest (out 1=5) is retrieved from the output file and returned to COSSAN

Table 56.1 COSSAN-XLM tag used to carry transfer information from COSSAN-X and third-party software

Attribute Name	Description
Name	Name of the variable defined in COSSAN
Index	Indices of variable. The index is used to inject values defined in a vector
Format	Format of the written variable, specified as a string containing formatting operators. A formatting operator starts with a percent sign, %, and ends with a conversion character. Standard formatting string is used
Original	Original text (value) that needs to be replaced

management (see, e.g., [11, 12, 82]). The combination of various algorithms with specific solution sequences permits the analysis of engineering problems. Eventually, these algorithms form the application layer, such as uncertainty quantification, reliability analysis, life cycle management, sensitivity analysis, modal updating, etc.

2.3 Technical Features

The purpose of this section is to summarize the features of COSSAN-X. The reader is referred to the “case studies” session where the techniques and algorithms are used to solve problem of practical interest.

2.3.1 Uncertainty Characterization

Uncertainties can be described within the framework of probability. Probabilistic analysis can be a very powerful tool for the rational treatment of uncertainties.

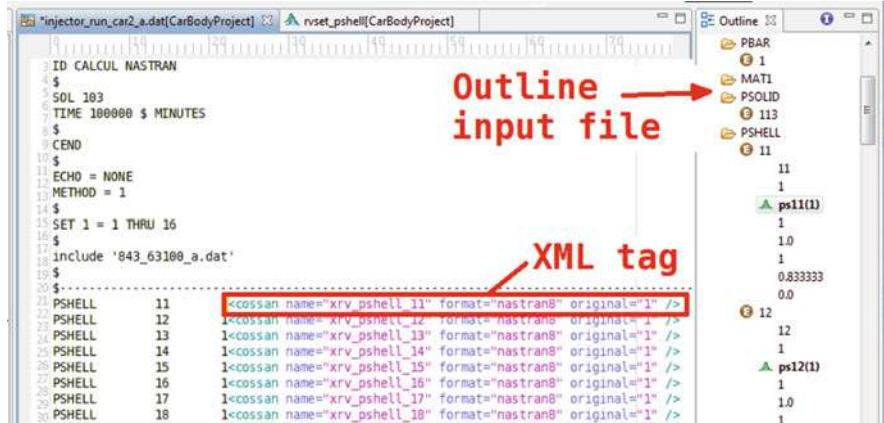


Fig. 56.14 COSSAN-X: manipulation of input file of a commercial FE software. Example of NASTRAN input file, the XLM tags are included in the input file to connect COSSAN variables to FE input parameters

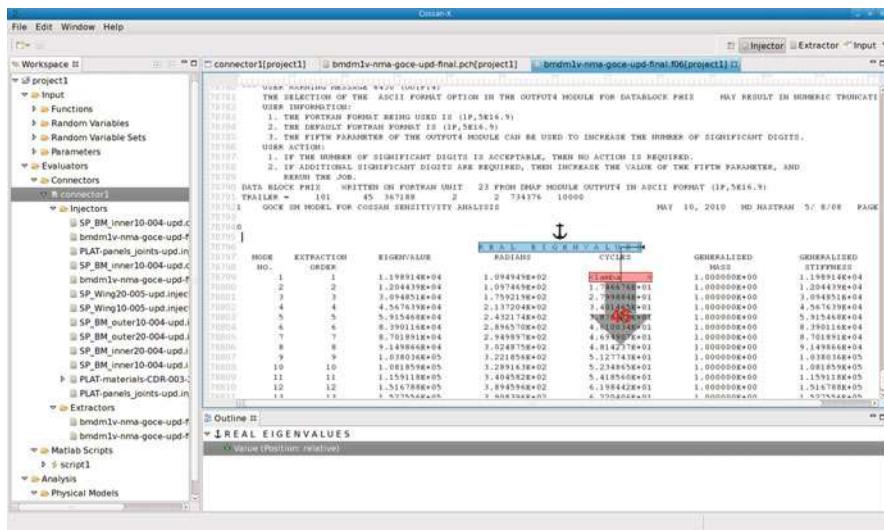


Fig. 56.15 COSSAN-X: definition of the quantities of interest in the output file of a commercial FE software. Example of NASTRAN output file where the quantity of interest is the first 45 values of the eigenvalues

However, traditional probabilistic methods require the representations of uncertainty based on probability density function (PDF) or cumulative density function (CDF). PDF and CDF can be obtained from data sets and used to describe aleatory uncertainties. Scalar values can be modeled using random variables, e.g., static load; time variant quantities can be represented using stochastic processes, e.g.,

wind speed or earthquake excitation; space variant quantities can be described using random fields, e.g., material properties in a solid.

A random variable can be defined by specifying the distribution type, e.g., normal, log-normal, uniform, etc., together with either the parameters of the distribution or its moment(s) (see Fig. 56.16). Multivariate distributions are defined by means of marginal distributions and correlations among them. Alternatively, random variable and multivariate distributions can be constructed starting from a set of realizations. The parameters leading to the best fit are automatically computed using different strategies such as the maximum likelihood estimation, kernel density estimation, and a mixture of one or more multivariate Gaussian distribution components [33] (see Fig. 56.16). Stochastic processes and random fields can be also defined and modeled by defining a functional dependence in a multidimensional continuous space or time dependence (see, e.g., [78, 96]).

When only a very limited number of samples are available, it is not possible to characterize the aleatory uncertainty, and a significant epistemic contribution needs to be taken into account. The epistemic contribution can be so large that an accurate estimation of the PDF/CDF is not possible. In such cases, a more rational treatment of the uncertainty would be, for instance, to use sets or families of PDFs/CDFs that are in agreement with the experimental evidence (i.e., the data set), with a reasonable reliability and robustness. There are different approaches to deal with such scarce data, including statistical tolerance intervals, fitting normal PDF, kernel density estimation techniques, and using nonparametric distribution to fit data samples. A comparison and further details about these techniques can be found in [14, 73].

2.3.2 Uncertainty Quantification and Reliability Analysis

Uncertainty quantification and reliability analysis aim to simulate and ensure the performance of a system or component, i.e., that the envisioned tasks are efficiently performed by the design over its lifetime. In principle, uncertainty quantification can be performed by direct Monte Carlo simulation. In practical cases, such approach is infeasible, due to the number of simulations required. In addition, generalized probabilistic approaches can be extremely demanding in terms of computational costs, and the availability of efficient and flexible computational framework is of paramount importance [12, 58, 79].

COSSAN-X contains the most recent and advanced simulation methods, as summarized in Table 56.2. Approximated methods such as FORM and SORM [27] are available as well. Since the selection of the most appropriate simulation tool and the corresponding settings is in general not an easy task for a nonexpert in stochastic analysis, COSSAN-X provides wizards and predefined setting for supporting users in the selection of the approach for performing efficiently nondeterministic analysis as shown in Fig. 56.17.

2.3.3 Optimization Tools

In today's engineering practice, optimization is an indispensable step of the design cycle of a product. By means of optimization, engineers can reach significant reductions in terms of the manufacturing and operating costs, as well as the

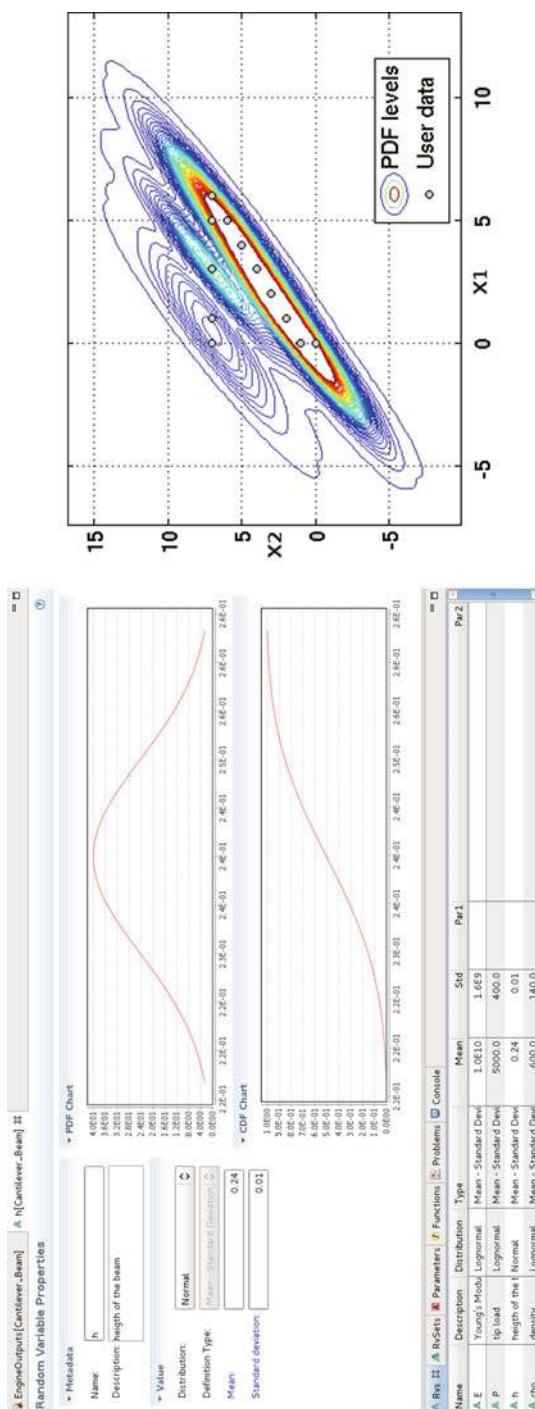


Fig. 56.16 COSSAN-X: definition of the random variable by defining distribution type and moments (on the left) and from a set of realizations (right panel)

Table 56.2 Selection of the uncertainty quantification tools available in COSSAN-X

Simulation methods	References
Monte Carlo simulation	[43, 75]
Latin hypercube sampling	[53]
Quasi-Monte Carlo sampling	[16]
Importance sampling	[62]
Line sampling	[25, 40, 58]
Subset simulation	[4, 56]
Interval Monte Carlo	[99]
Markov Chain Monte Carlo	[21]

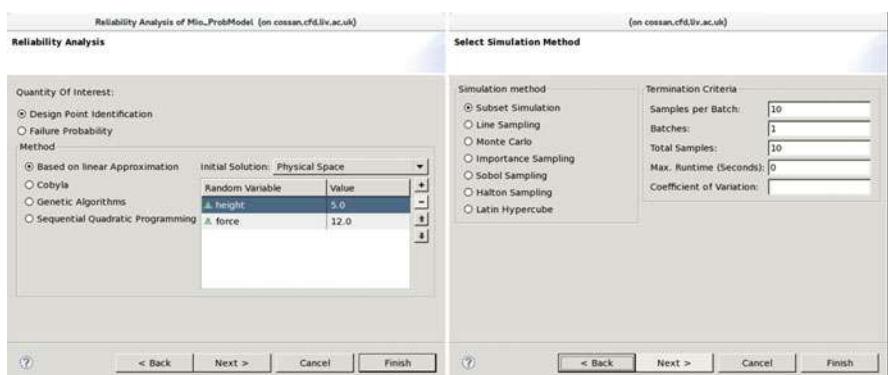


Fig. 56.17 COSSAN-X reliability analysis: identification of the design point (on the left) and the selection of the algorithm for the estimation of the probability of failure (on the right)

improvement in the performance. The optimization toolbox provides a set of widely used gradient-based and gradient-free algorithms both for small- and large-scale analysis, which can be adopted to solve real-life problems involving continuous or discrete design variables, multiple constraints, and objective functions. Again, the toolbox also provides the necessary guidance and assistance to the users for the selection of the most appropriate optimization method (see wizard in Fig. 56.18).

A wide choice of algorithms for dealing with different types of optimization problems is available and summarized in Table 56.3. Specialized strategies to deal explicitly with uncertainties are available as well (see, e.g., [8, 37, 46, 60, 86, 92, 94]).

2.3.4 Meta-modeling

In applications where a costly numerical model is to be evaluated multiple times (such as in the case of stochastic analysis). One way to reduce the analysis time is to use meta-models which approximate the quantities of interest at low computational costs. In other words, meta-models mimic the behavior of the original model (e.g., FE analysis), by means of a mathematical model with negligible computational cost. Using the features of this toolbox, the user can interactively train meta-models to replace their complex models and calibrate them to a desired accuracy. Then,

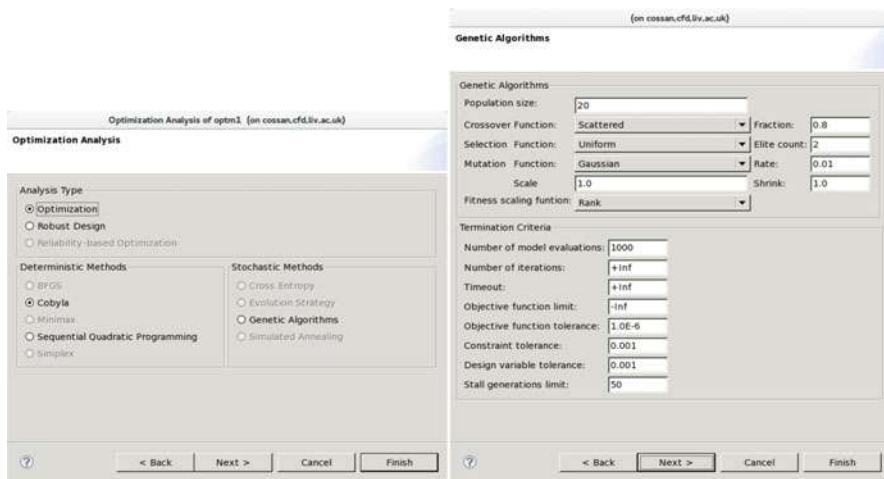


Fig. 56.18 COSSAN-X Optimization analysis: selection of the optimization tool (on the *left*) and setting for the genetic algorithm (on the *right*)

Table 56.3 Overview of the optimization tools available in COSSAN-X

Optimization algorithm	References
Genetic algorithms	[19, 32]
COBYLA and BOBYQA	[71]
SQP	[29]
Simplex	[51]
Simulated annealing	[39, 98]
Evolution strategies	[91]
Alpha-level optimization	[49]

the constructed meta-model can be used for performing uncertainty quantification, sensitivity analysis, optimization, etc. Figure 56.19 shows an example of meta-model in COSSAN-X. The meta-model is created using a set of training (calibration) points and validated using a different set of points.

A number of different meta-modeling techniques are implemented as shown in Table 56.4.

2.3.5 Stochastic Finite Element Toolbox

Stochastic finite element methods (SFEM) extend the capabilities of the classical deterministic finite element analysis, in order to take the structural uncertainties into account and to propagate the unavoidable uncertainties in the structural responses (see e.g., [31, 78, 90]).

Intrusive implementations of SFEM are available for NASTRAN, ABAQUS, and ANSYS, and different and widely used formulations such as perturbation, Neumann expansion, and polynomial chaos expansion are provided.

The main capabilities of the SFEM toolbox are summarized in Table 56.5.

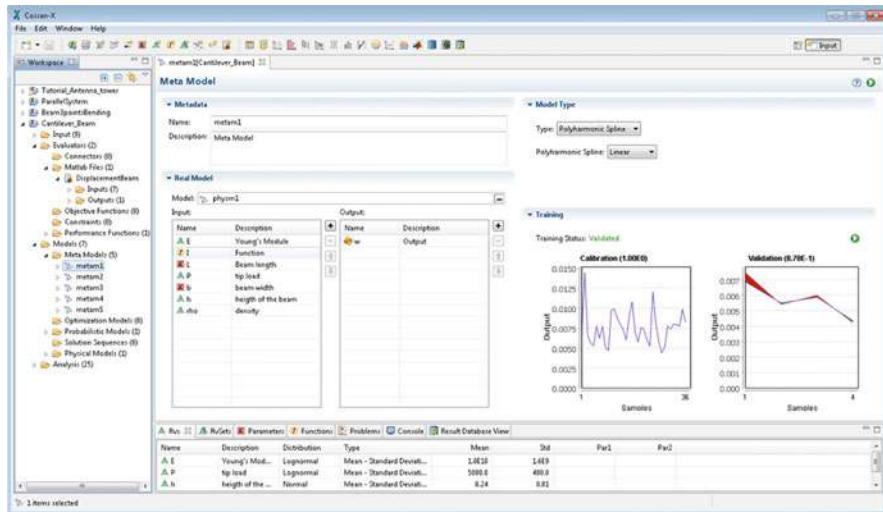


Fig. 56.19 COSSAN-X meta-model: the screenshot shows the input and output of a polyharmonic meta-model with calibration and validation performance

Table 56.4 Overview of the meta-modeling tools available in COSSAN-X

Meta-model	References
Artificial neural networks	[52]
Kriging	[35]
Polyharmonic splines	[34]
Polynomial chaos	[54, 89]
Response surface	[74]

Table 56.5 Overview of the capabilities of the SFEM toolbox

Solvers	NASTRAN, ABAQUS, ANSYS
Random parameters	Young's modulus, density, shell element thickness, beam element cross-sectional dimensions, force
Formulations	Perturbation, Neumann expansion, polynomial chaos expansion
Implementations	Component-wise, solver based, reduced model
Analysis Types	Linear static, modal

2.3.6 Sensitivity Analysis

Sensitivity toolbox allows to study the relationship between the input and output quantities in a model and to identify the most significant variables affecting the response. Consequently, sensitivity analysis is particularly used for model calibration, model validation, and decision-making process purposes, i.e., where it is crucial to identify the parameters which contribute mostly to the output variability.

Sensitivity analysis may be divided into three broad categories: local sensitivity analysis, screening methods, and global sensitivity analysis. Local sensitivity analysis provides information about the system behavior around a selected point in

Table 56.6 Overview of the sensitivity approaches available in COSSAN-X

Sensitivity tool	References
Monte Carlo gradient estimation	[45, 59, 97]
Fourier amplitude sensitivity test	[76, 77]
Sobol sensitivity indices	[61, 77, 88]
Nonspecificity technique	[2]

the input domain, while global sensitivity analysis techniques consider the entire range of the input parameters into account. Screening techniques (or one factor at a time) simply vary one factor at a time and measure the variation in the output.

When epistemic uncertainty is present, the sensitivity analysis can be performed constructing an equivalent model that takes as inputs the values of epistemic uncertainties and returns a scalar quantity. Alternatively, the Hartley-like measure of nonspecificity can be used which does not require the calculation of the probability box associated to the output Dempster-Shafer structure. More details about the approaches to deal with epistemic uncertainty are available in Ref. [68].

COSSAN-X offers various algorithms for the sensitivity analysis, which are summarized in Table 56.6.

2.3.7 High-Performance Computing and Data Management

The computational framework provides transparent access to high-performance computing with any algorithms implemented into the framework.

The analysis of complex systems usually requires the evaluation of different solvers that should run in a specific order. For instance, one solver is preparing the mesh for the FE analysis and another solver is used for post-processing the analysis. The execution of the solvers in the specific order is dealt by COSSAN creating tasks that contain all the commands required to run the solver and some specific commands to copy files among the working directories of the solvers (see Fig. 56.20). Stochastic analyses can also be split in batches (Fig. 56.20). A batch represents a full independent analysis (i.e., execution of tasks). This allows to perform checking the convergence of the analysis or adding more samples in order to refine the analysis.

COSSAN-X performs the parallelization of the stochastic analyses by splitting the analysis on subtasks or jobs and interfacing with industry standard job schedulers, such as GridEngine, Platform/LSF, or OpenLava. These jobs are distributed among the available (remote) resources on a computer cluster and/or grid. The software allows to maximize the use of the available licenses while reducing the execution time (wall clock time) of the analysis task. The software provides different execution and parallelization strategies summarized in Fig. 56.21. In the first parallelization strategy (vertical parallelization), each task (job) performs the analysis of one realization of the input parameters and executes all solvers. Using this strategy, all the solvers run on the same machine (host) selected by the Job Manager. In the second parallelization strategy (horizontal parallelization), first all the executions of

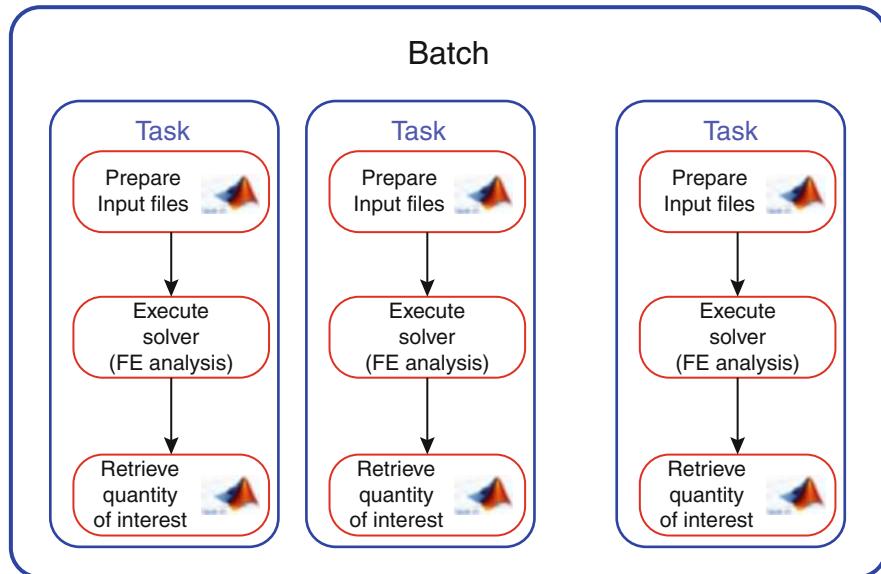


Fig. 56.20 COSSAN-X High-performance computing: splitting of the analysis batches and tasks. The task represents an execution of the solver and it involves the preparation of input files and the collection of the quantity of interest. A batch is a collection of tasks

the first solver are performed, then the second one, and so on. This allows to run each solver on specific machines (hosts) by selecting different queues for different jobs. The third parallelization strategy (granular parallelization) combines the first two methods. Using this approach, dependent jobs are submitted to the Job Manager (e.g., job involving the execution of the second solver starts after the completion of the job involving the first solver).

COSSAN-X High-Performance Computing: Parallelization Strategies. In the vertical parallelization, each job is formed by a full (deterministic) analysis and multiple jobs form the stochastic analysis. In the horizontal parallelization, a job runs only one solver with a specific number of analyses. Then when all the executions of the first solver are completed, job for the analysis of the next solver is submitted. In a granular parallelization, each solver and each (deterministic) analysis form a job. Dependent jobs are submitted to the Job Manager (i.e., a job involving a solver number j is running only after the completion of the corresponding job for the solver $j - 1$).

An important feature is the possibility to use high-performance computing (HPC) resources from machines running different operating systems (i.e., MS Windows, MacOS, and different Linux distributions). COSSAN-X allows the user to define a connection to the head node of a cluster through a secured connection over the Internet (SSH) as shown in Fig. 56.22. By doing so, the user interface is running locally and submits jobs on the HPC when needed. After the jobs are completed, the results are retrieved on the local machine.

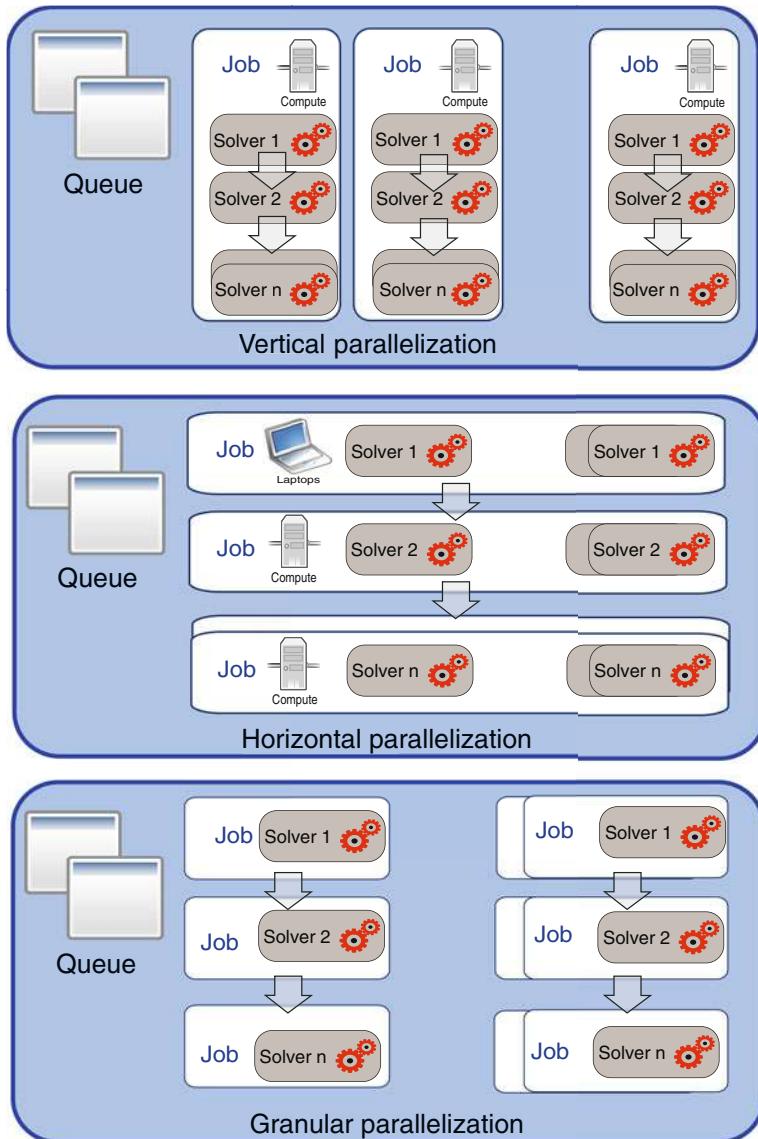


Fig. 56.21 COSSAN-X high-performance computing: parallelization strategies. In the vertical parallelization, each job is formed by a full (deterministic) analysis and multiple jobs form the stochastic analysis. In the horizontal parallelization, a job runs only one solver with a specific number of analyses. Then when all the executions of the first solver are completed, job for the analysis of the next solver is submitted. In a granular parallelization, each solver and each (deterministic) analysis form a job. Dependent jobs are submitted to the Job Manager (i.e., a job involving a solver number j is running only after the completion of the corresponding job for the solver $j - 1$)

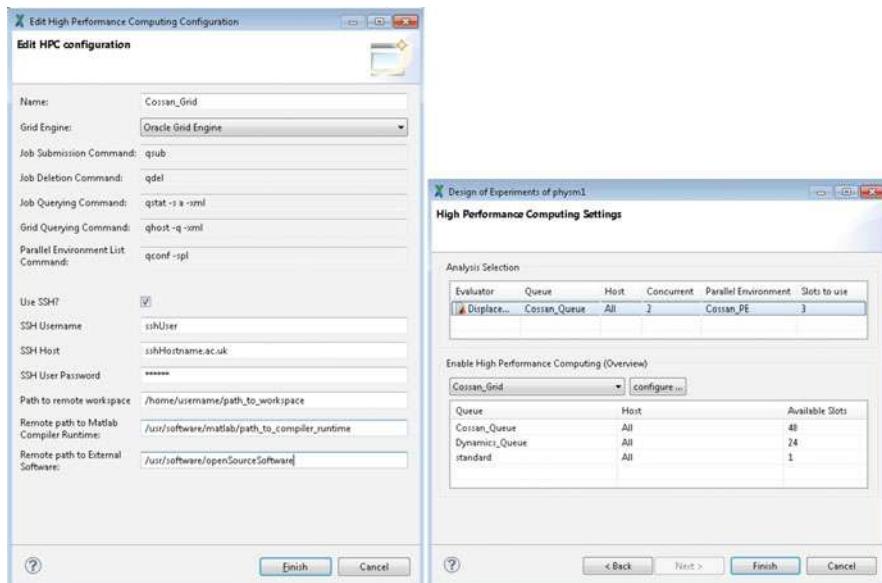


Fig. 56.22 COSSAN-X high-performance computing: definition of setting to access the cluster-/grid computing via Secure Shell (on the *left panel*). Screenshot of the interface for the definition of jobs (i.e., solution of solvers, queues, host, machine, and number of slots (e.g., processors) to be used (on the *right panel*)

Performing a stochastic analysis requires the multiple executions of the solvers. Hence a huge amount of data are generated and they need to be easily accessible to the analyst and properly stored. COSSAN-X allows to store the results of the analysis locally (using an auto-configured SQLite database) or remotely (in a centralized MySQL-like database). The database management is completely transparent to users and data are automatically stored in the user-preferred location that can be accessed directly from COSSAN-X using a dedicated interface shown in Fig. 56.23.

2.3.8 Open and Collaborative Documentation

The availability of an extended documentation, tutorials, and examples is of paramount importance for the usability of a software. Often, such documentation is written in the form of relatively long pages/tutorials. Typically users do not have the inclination/time to read long help articles. Instead, they simply want to know how to get to the next step of whatever process they are carrying out. For this reason the documentation of COSSAN-X is context sensitive. Tooltips and direct access to specific documentation related to the current task performed by the user (e.g., defining random variable, selecting solution strategy, visualizing results) are available without overwhelming the user with unnecessary information.

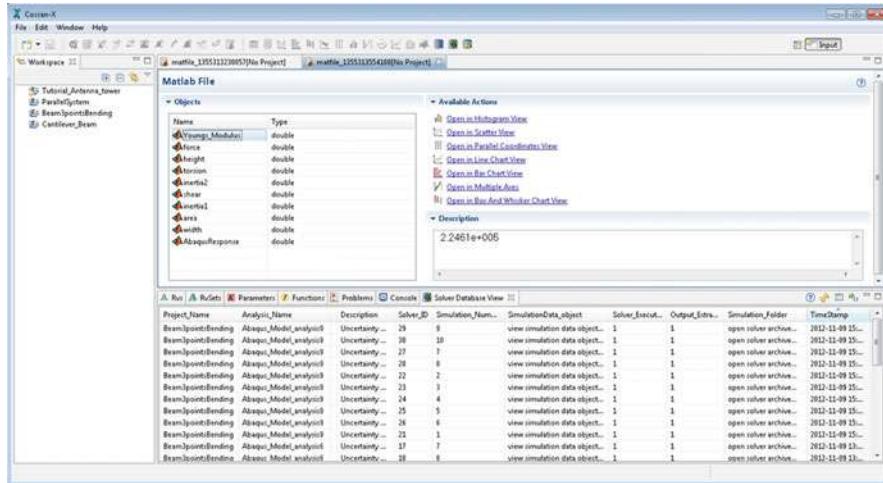


Fig. 56.23 COSSAN-X data management: the analyses can be stored in a local (e.g., SQLite) or centralized (e.g., SQL) database and all the analyses are accessible using the provided interface

In addition, theory manual, tutorial, and examples are maintained using MediaWiki tool to provide open access allowing collaborative development of the documentation (see Fig. 56.24). In addition, industrial users need to create private extensions to the help system, specific to their requirements and work processes.

2.4 OPENCOSSAN: An Open-Source Matlab Toolbox

OPENCOSSAN represents the computational core of the COSSAN project and contains a collection of open-source algorithms, methods, and tools under continuous development at the Institute for Risk and Uncertainty, University of Liverpool, UK [67]. Released under the terms of the GNU Lesser General Public License [30], OPENCOSSAN can be used for free, redistributed, and/or modified. The source code is available upon request at the web address <http://www.cossan.co.uk> or it can be downloaded automatically via the OPENCOSSAN Matlab App (as shown in the following section).

OPENCOSSAN is coded exploiting the object-oriented Matlab® programming environment, where it is possible to define specialized solution sequences, which include reliability methods, sensitivity analysis, optimization strategies, surrogate models, and parallel computing strategies. The computational framework is organized in packages. A package is a namespace for organizing classes and interfaces in a logical manner, which makes large software project OPENCOSSAN easier to manage. A class describes a set of objects with common characteristics such as data structures and methods. Objects that are instances of classes can be aggregated

Collaborative Documentation WIKI

Category Discussion Read Edit View history Search

Category:Getting Started

This section provides the links to basic step-by-step guides, detailing the processes required to install **COSSAN-X** and **OpenCOSSAN** onto your computer. The OpenCOSSAN Engine is an invaluable tool for uncertainty quantification and management representing the core of the COSSAN software, while COSSAN-X is a software package developed to make the concepts and technologies of uncertainty quantification and risk analysis available. The below guides offer a walk through guide to get the software up and running on your computer, and provides detailed guides on installation for operating systems: **Windows**, **Linux** and **MacOS**.

At the bottom of the page, users can also find the links to several further pages detailing information on the running of the software, known bugs within the software packages and some basic information on the importing of tutorials. The below pages provide users with the basic information needed to get started using the software packages.

COSSAN-X [edit]

The page [Installation of COSSAN-X](#) describes how to install a version of **COSSAN-X** on your machine. To access **COSSAN-X** from a remote machine using **NoMachine** connection tools, see [Run COSSAN-X remotely](#). Users are able to access an easy to follow guide, providing links to the required downloads, and screenshots to take users through the download process.

More information can be found for Cossan-X through the following link: [Category:COSSAN-X](#)

OpenCOSSAN [edit]

The page [Installation of OpenCOSSAN Engine](#) provides users with an easy to follow guide, taking you through, in detail, the steps required to get **OpenCOSSAN** installed and up and running on your computer.

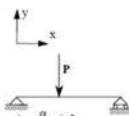
Beam 3-point bending (overview)

This tutorial is also available for OpenCossan in:

- echodemo TutorialBeam3PointBendingMatlab
- echodemo TutorialBeam3PointBendingNastran
- echodemo TutorialBeam3PointBendingAnsys
- echodemo TutorialBeam3PointBendingAbaqus

Tutorial for COSSAN-X
File -> New -> Tutorial -> Tutorial 2: Beam 3-point bending (see also Import tutorial)

This example considers a beam in three points bending. It will be studied using several third-party software and various toolboxes from COSSAN-X.



The presentation of the mechanical model and its implementation in the different toolboxes are in the pages linked below. The model is analysed using either a [Matlab script](#) or using a third party finite element software ([Nastran](#), [Abaqus](#) and [Ansys](#)). Then simulation analysis is performed to investigate the effects of the uncertainties on the mid-span displacement. Finally the tutorial shows how to perform [reliability analysis](#) using this example. Both the uncertainty quantification and the reliability analysis can be executed on the grid using [High Performance Computing](#).

See Also [edit]

- [Beam 3-point bending \(Presentation\)](#)
- [Beam 3-point bending \(Inputs Preparation\)](#)

Fig. 56.24 COSSAN-X collaborative documentation: examples of Wiki pages available on <http://cossan.co.uk/wiki>. The documentation includes tutorials, advises, and tips as well as references and scientific papers

forming more complex objects and proving solutions for practical problem in a compact, organized, and manageable format. The structure of OPENCOSSAN allows for extensive modularity and efficient code reutilization. Objects (instances of a class) can be aggregated forming more complex objects with methods providing solutions for practical problem in a compact, organized, and manageable format. Hence, different objects and methods can be combined by the users to solve specific problems, including uncertainty quantification, sensitivity analysis, reliability analysis, and robust design. Such problems can be solved by adopting traditional probabilistic approaches as well as by generalized probabilistic methods. Thanks to the modular nature of OPENCOSSAN, it is possible to define specialized solution sequences including any reliability method, optimization strategy, and surrogate model or parallel computing strategy to reduce the overall cost of the computation without loss of accuracy. Figure 56.25 shows a simplified representation of the computational framework and the dependencies among the different toolboxes.

OPENCOSSAN does not provide a dedicated user interface, but it relies on Matlab Desktop framework that provides a high-level language and interactive environment for numerical computation, visualization, and programming as shown in Fig. 56.26. In particular the *current folder* panel is used to access files, while the command windows is used to enter commands and interact with OPENCOSSAN (see Fig. 56.27). Finally, the Matlab editor allows to visualize, debug, extend, or modify any part of OPENCOSSAN.

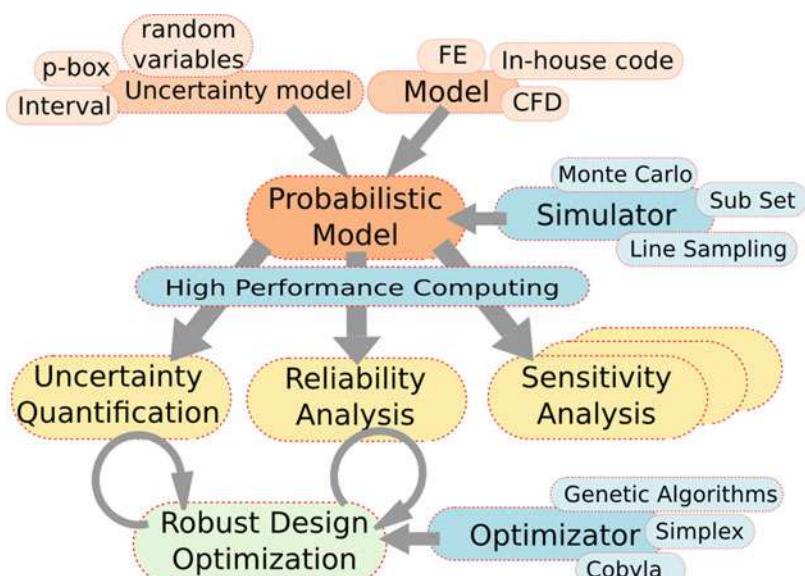


Fig. 56.25 Scheme of the OPENCOSSAN computational framework. The arrows show the relation among components (toolboxes). Circular arrows represent the loops (e.g., robust and reliability-based optimization is using uncertainty quantification and reliability analysis as internal loop of the analysis)

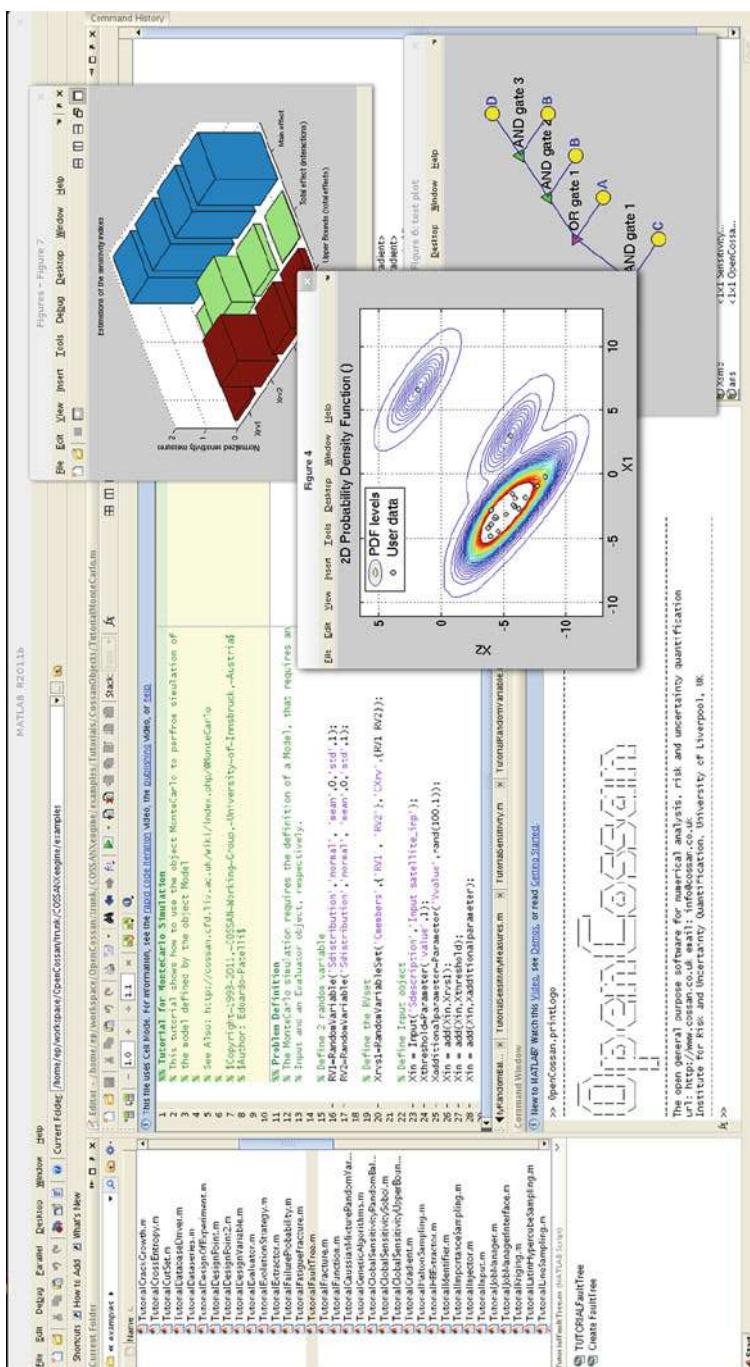


Fig. 56.26 OPENCOSSAN in Matlab environment. The screenshot shows the Matlab Desktop Workspace, the initialization of the OPENCOSSAN toolbox (in the Matlab command line). Some visualization examples show a definition of a Gaussian mixture distribution, a fault tree, and the histogram of sensitivity measures

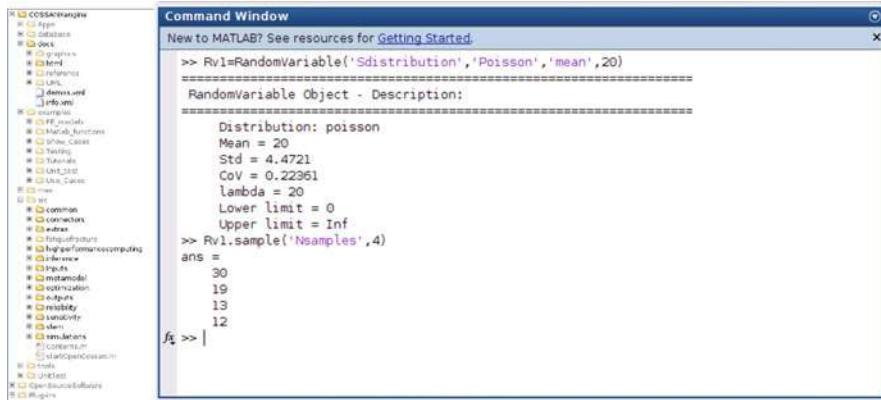


Fig. 56.27 Current folder panel and file organization of OPENCOSSAN in Matlab environment. Example of definition of a random variable and realization of samples

```

% This file is part of openCOSSAN. The open general purpose matlab
% toolbox for numerical analysis, risk and uncertainty quantification.

%% Preparation of the Input
% Definition of the Parameters
maxDisplacement=Parameter('value',0.010,'Sdescription','Maximum allowed displacement');
...
% Definition of the Random Variables
P=RandomVariable('Sdistribution','lognormal','mean',5000,'std',400,'Sdescription','Load');
...
% correlation between rho and E
Mcorrelationeye(4); Mcorrelation(3,4)=0.8; Mcorrelation(4,3)=0.8;
%Set of Random Variable Set
Xrvset=RandomVariableSet('CXrandomVariables',{P h rho E},{'CSmembers',{P 'h' 'rho' 'E}},{'Mcorrelation',Mcorrelation});
% The above prepared object can be added to an Input Object
Xinput=Input('Xmembers',{b Xrvset I maxDisplacement},{'CSmembers',{L 'b' 'Xrvset' 'I' 'maxDisplacement}});
%% Preparation of the Evaluator (this is the model)
%Foldersfileparts(which('TutorialCantileverBeamMatlab.m'));% returns the current folder
Xmio=Mio('Sfile','tipDisplacement.m','Cinputnames',{I 'b' 'L' 'h' 'rho' 'P' 'E} , 'Coutputnames',{w});
% Add the Matlab Input/Output object
Xevaluator=Evaluator('Xmembers',Xmio,'CSmembers',{Xmio});
%% Preparation of the Model (Input + Evaluator)
XmodelBeamMatlab=Model('Xinput',Xinput,'Xevaluator',Xevaluator);
%% Define a Probabilistic Model (Model + Performance Function)
Xperf=PerformanceFunction('Sdemand','w','Scapacity','maxDisplacement','Soutputname','Vg');
% Define a Probabilistic Model
XprobModelBeamMatlab=ProbabilisticModel('Xmodel',XmodelBeamMatlab,'XperformanceFunction',Xperf);

%% Reliability Analysis (by means Monte Carlo Simulation and Subset simulation)
% Monte Carlo simulation
Xmc=MonteCarlo('Nsamples',1e4,'Nbatches',1);
XfailireProbC=Xmc.computeFailureProbability(XprobModelBeamMatlab);
% Subset simulation
Xss=Subset('NinitialSamples',100,'targetFailureProbability',0.1);
XfailireProbSs=Xss.computeFailureProbability(XprobModelBeamMatlab);

```

Fig. 56.28 Example of a OPENCOSSAN script. The script shows the procedure to instantiate objects by calling the corresponding constructors (e.g., an object MonteCarlo is created as follows: “Xmc=MonteCarlo”). Then, the objects can be used accessing their methods (e.g., the failure probability is computed invoking the method computeFailureProbability of the object “Xmc”) and providing a ProbabilisticModel object (e.g., “Xpf=Xmc.computeFailureProbability(XprobModel)”).

OPENCOSSAN provides intuitive, clear, well-documented, and human readable interfaces to the classes. Hence, the OPENCOSSAN code can also be used by users not familiar with Matlab environment. Figure 56.28 shows a Matlab script based on OPENCOSSAN toolbox for solving a reliability problem. The approach adopted

to solve a user-defined problem depends on the representation of the uncertain quantities. In fact, uncertainties can be defined as distributional or free p-boxes, random variables, intervals, fuzzy, etc. Hence, if the uncertain quantities are defined by means of random variables, the framework will estimate the failure probability using Monte Carlo simulation or advanced Monte Carlo methods (subset simulation [4, 5, 56] in the example of Fig. 56.28). On the other hand, if uncertain quantities are defined as intervals or p-boxes, the framework will estimate the bounds of the failure probability. The user can freely control the computational strategies: as an example it is possible to estimate the bounds of the failure probability by means of a double loop Monte Carlo simulation, or by means of tailored solution strategies, e.g., combining an optimization strategy with the line sampling method [24]. Furthermore, the developed numerical methods are highly scalable and parallelizable, thanks to its integration with distributed resource management, such as openlava and GridEngine. These job management tools allow to take advantages of high-performance computing resources.

2.4.1 OPENCOSSAN Matlab App

OPENCOSSAN Matlab App is an interactive application design to assist the end user to obtain easily the OPENCOSSAN source files, access the documentation (help, tutorials, and reference manual), assist the installation phase, and initialize the OPENCOSSAN toolbox. It allows to keep the local version synchronized with the upstream version of the software without the need to configure or install any software versioning and revision control system.

OPENCOSSAN Matlab App is available in www.cossan.co.uk and in Matlab File Exchange and it can be installed in just one click. Then, it will be accessible in the Apps tab of the MATLAB Toolstrip. Figure 56.29 shows an example of the Matlab application tab with the installed OPENCOSSAN Matlab App and a screenshot of the App.

2.5 Engineering Cloud

It is recognized that the application of virtual engineering/prototyping for reducing engineering risks is fundamental to the future success of high-value manufacturing and engineering to bring the most sophisticated technology to market as quickly and cheaply as possible. To meet these challenges, the industry (and in particular aerospace industry) is making increasing use of the so-called virtual prototyping with the aim of physical testing to be eliminated until the final verification phases. However, simulations of virtual prototypes are becoming more comprehensive involving the investigation of multiple variants using different disciplines, thus requiring multi-physics solutions, multidisciplinary optimization, design of experiments, and robustness analyses. This, combined with the need for real-time simulation to support business decision making, is driving the requirement for a new cloud structure which not only requires on-demand computational power but more importantly needs ease of use and tailored application-driven capabilities.

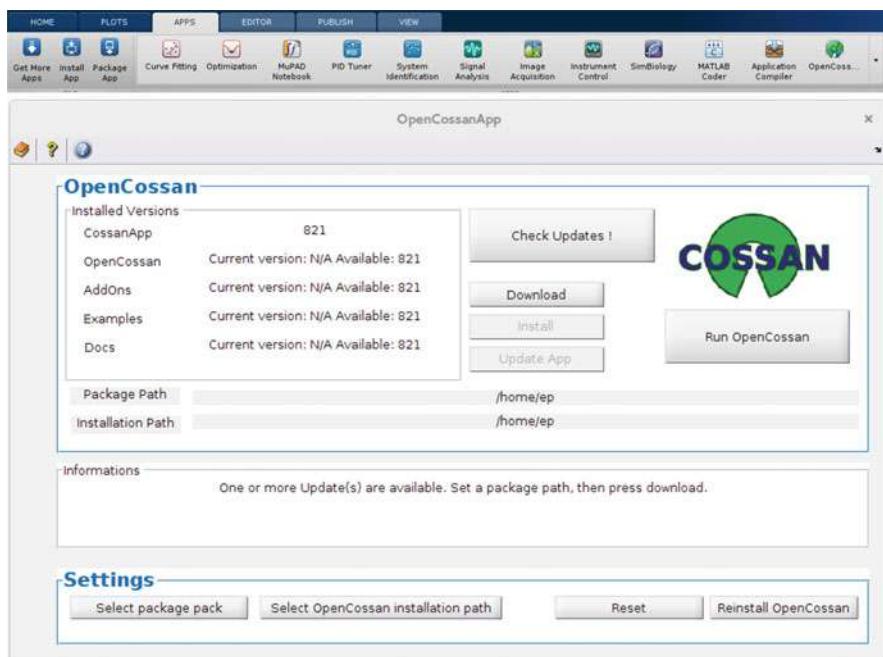


Fig. 56.29 OPENCOSSAN Matlab App. In the upper part of the figure, the Matlab application bar is shown

A combination of technological, social, and economic barriers, including lack of skills, knowledge, flexibility, and affordability of software licensing models and initial costs for computer facilities (including setting up costs and maintenance), is preventing companies from adopting modern software solutions. Hence, in order to meet these needs, COSSAN-X has been integrated into the “Engineering Cloud” project led by the Virtual Engineering Centre (Virtual Engineering Centre, STFC Daresbury Laboratory, Daresbury Science & Innovation Campus, Warrington, WA4 4AD, www.virtualengineeringcentre.com). The “Engineering Cloud” offers “virtual prototypes on demand,” promoting a culture to develop and host “engineering apps,” enabling front-end users to develop locally complex multidisciplinary workflows and upload them in the cloud through a secure web-based interface and perform, e.g., stochastic analysis, robust design, etc. In order to perform such analyses, a range of (deterministic) software tools are combined with COSSAN software and in-house solutions. Thereafter, the execution of the analyses is handled by the cloud taking advantage of the HPC facilities as well as the pool of integrated software as shown schematically in Fig. 56.30.

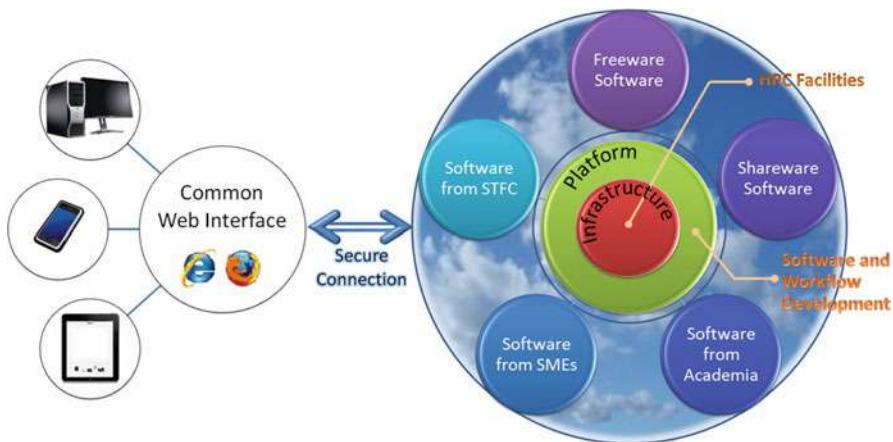


Fig. 56.30 Schematic concept of the Engineering Cloud developed and led by the Virtual Engineering Centre

3 Case Studies

In this section, selected challenging problems are briefly presented in order to demonstrate the applicability and flexibility of the software to solve a wide range of engineering and scientific problems. The first example presents a series of analyses and methodologies that can be used for dealing with aleatory and epistemic uncertainties in a multidisciplinary model. The second example shows the reliability analysis of a large-scale finite element model of a six-story building involving imprecision in the definition of the random variables. In the third example, the robust design of a truss roof is presented. The last example addresses the problem of designing a robust maintenance scheduling for components and systems in the presence of aleatory and epistemic uncertainty.

3.1 Application to the Robust Design of a Twin-Jet Aircraft Control System

NASA Langley Research Center has recently proposed a challenge problem in order to determine limitation and range of applicability of existing uncertainty quantification (UQ) methodologies and to advance the state of the practice in UQ problem. The reader is referred to Ref. [23] for a full description of the NASA UQ challenge problem.

OPENCOSSAN has been used to solve all the tasks proposed in the challenge problem. Here only a small selection of the main findings are reported, and results of the NASA UQ challenge problem are available in Ref. [66].

3.1.1 The Model

A multidisciplinary model that describes the dynamic of a remotely operated twin-jet aircraft has been developed by NASA Langley Research Center to provide an in-flight validation capability for high-risk flight testing beyond the normal flight envelope. The overall aim of the analysis is to identify design points that provided optimal worst-case probability performance in the presence of uncertainty of the control system for the Airborne Subscale Transport Aircraft Research (AirSTAR).

The mathematical model of the AirSTAR test aircraft, S , has been treated as a “Black Box.” The uncertain parameters in the model, p_i , $i = 1, \dots, 21$, are used to describe losses in control effectiveness and time delays resulting from telemetry and communications and to model a spectrum of flying conditions that extend beyond the normal flying envelope. The outputs of S , i.e., the requirements in g_j , $j = 1, \dots, 8$, are used to describe the vehicle stability and performance characteristics in regard to pilot command tracking and handling/riding qualities. Fourteen design parameters, d_k , $k = 1, \dots, 14$, can be tuned to optimize the robustness of the system.

The uncertain parameters are modeled accounting aleatory and epistemic uncertainty. The aleatory is modeled by the use of random variables with fixed function form and known coefficient. The epistemic uncertainty is modeled by the use of interval of fixed but unknown constants. Finally, distributional p-boxes are adopted if the parameters are affected by combined aleatory and epistemic uncertainty. The aim of the analysis is to identify a design point with improved robustness and reliability by minimizing the expectation of the maximum of the output, $J_1 = E[\max_j(g_j)]$, and minimizing the upper bound of the probability of failure $J_2 = P(g_j < 0)$. Figure 56.31 shows a schematic representation of the multidisciplinary model; the sub-models \mathbf{h} , \mathbf{f} ; intermediate variables \mathbf{x} ; and

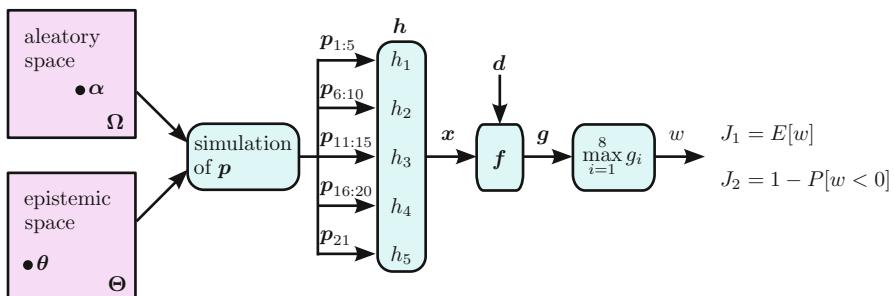


Fig. 56.31 Relationship between the variables and functions of the NASA Langley multidisciplinary uncertainty quantification challenge problem [50]

different performance function g . A detailed description of the problem can be found in [23].

3.1.2 Proposed Approach

Different tools and approaches exist for uncertainty quantification and characterization that can be potentially used in the design and safety-critical systems. Every method is based on some assumptions and hypothesis that often cannot be verified a priori. Moreover, the simulation strategies are able to produce accurate results only if the right set of parameters are selected and this often cannot be verified. Hence, different analyses have been performed using different strategies and hypothesis in order to *cross-validate* the results.

Many of the proposed solutions to the challenge problem make use of the random set theory [3, 47]. Random set theory allows to model under the same framework different representations of the uncertainty (such as cumulative distribution functions, intervals, distribution-free probability boxes, normalized fuzzy sets, and Dempster-Shafer structures) without making any implicit or explicit assumption at all.

3.1.3 Uncertainty Reduction

The aim of model updating is to reduce the epistemic uncertainty on the output of the model $x = h(\alpha; \theta)$ based on the availability of a limited set of data (observations) $\{x_k^e : k = 1, 2, \dots, n_e\}$. These observations of the “true uncertainty model” $\theta^* \in \Theta$ can be used to improve the uncertainty model, i.e., to reduce the original intervals of the epistemic uncertainties by excluding those combinations of parameters that fail to describe the observations as shown in Fig. 56.32. Different approaches can be used for model updating such as nonparametric approaches based on some statistical tests and Bayesian methods. Here, only the Bayesian method is briefly presented.

Bayesian inference is a statistical method in which Bayes’ rule is used to update the probability estimate for a hypothesis as additional information is available.

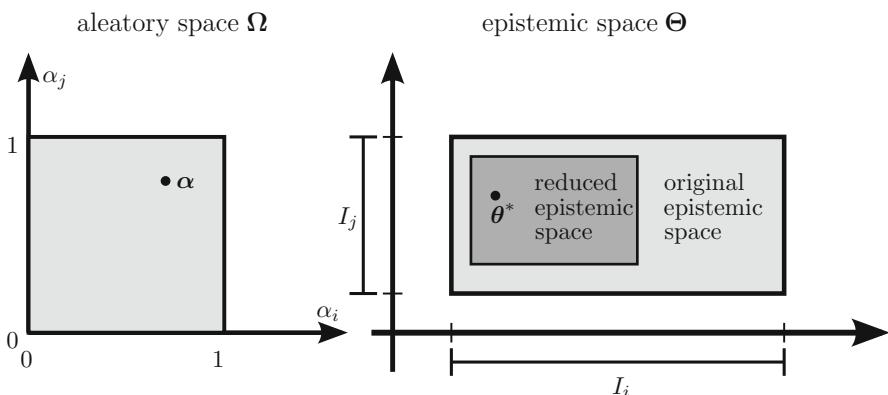


Fig. 56.32 Representation of the uncertainty reduction space for the NASA UQ challenge problem

Suppose we are given a set of observed data points $\mathcal{D}_e := \{x_k^e : k = 1, 2, \dots, n_e\}$ called the *evidence*, which are sampled from a PDF $p(\cdot; \boldsymbol{\theta}^*)$ which belongs to a certain family of PDFs $\{p(\cdot; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ called the *parametric model*. The idea of Bayesian inference is to update our belief about the vector of parameters $\boldsymbol{\theta}^*$ provided that $\boldsymbol{\theta}^*$, the true set of parameters of the PDF, is unknown. Bayes' theorem updates that belief using two antecedents:

- a prior PDF $p(\boldsymbol{\theta})$ which indicates all available knowledge about $\boldsymbol{\theta}^*$ before the evidence \mathcal{D}_e is observed;
- the *likelihood function* $P(\mathcal{D}_e | \boldsymbol{\theta})$, which is a function related to the probability of observing the samples \mathcal{D}_e assuming that the true parameter underlying the model PDF $p(x; \boldsymbol{\theta})$ is $\boldsymbol{\theta}$, is defined as

$$p(\mathcal{D}_e | \boldsymbol{\theta}) = \prod_{k=1}^{n_e} p(x_k^e; \boldsymbol{\theta}), \quad (56.1)$$

when a set of independent and identically distributed observations \mathcal{D}_e is available. Please note that in practice (i.e., for the numerical implementation), the log-likelihood is used instead of the likelihood.

The updated belief about the vector of parameters $\boldsymbol{\theta}^*$ after observing the evidence \mathcal{D}_e , is modeled by the so-called posterior PDF $p(\boldsymbol{\theta} | \mathcal{D}_e)$ which is calculated by

$$p(\boldsymbol{\theta} | \mathcal{D}_e) = \frac{p(\mathcal{D}_e | \boldsymbol{\theta})p(\boldsymbol{\theta})}{P(\mathcal{D}_e)}; \quad (56.2)$$

where the probability of the evidence

$$P(\mathcal{D}_e) = \int_{\boldsymbol{\Theta}} P(\mathcal{D}_e | \boldsymbol{\theta})p(\boldsymbol{\theta}) \cdot \cdot \cdot d\boldsymbol{\theta} \quad (56.3)$$

can be understood as a normalizing constant. We hope that after using the evidence \mathcal{D}_e , the posterior PDF $p(\boldsymbol{\theta} | \mathcal{D}_e)$ is sharply peaked about the true value of $\boldsymbol{\theta}^*$. We will update our belief about the true set of parameters $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$ propagating the evidence through Bayes' equation numerically; this can be performed using an algorithm called Transitional Markov Chain Monte Carlo [21].

Using Laplace's principle of indifference (or more generally, the principle of maximum entropy), we will define a non-informative prior on the space of epistemic uncertainty $\boldsymbol{\Theta}$; in other words, a uniform PDF on $\boldsymbol{\Theta}$, that is, $\boldsymbol{\theta} \sim \text{Unif}(\boldsymbol{\Theta})$, is used to represent the epistemic uncertainty. Different likelihood functions based on different mathematical assumptions can be used. For instance, the likelihood can be estimated through kernel density or approximated using the following expression [7, 20]:

$$p(\mathcal{D}_e | \boldsymbol{\theta}_i) = \prod_{k=1}^{n_e} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \left(\frac{\delta_k}{\sigma}\right)^2\right) \quad (56.4)$$

where $\delta_k = |F^i(x_k^e) - F^e(x_k^e)|$ for $k = 1, 2, \dots, n_e$ and F^i and F^e represent the empirical CDF of the $\{x_j^i : j = 1, 2, \dots, n\}$ and the experimental data \mathcal{D}_e , respectively. The values of the standard deviation are unknown and hence it represents an additional parameter that needs to be estimated during the Bayesian analysis [9].

The Bayesian updating expressed in Eq. (56.1) needs the evaluation of the normalizing factor, i.e., the denominator of Eq. (56.1). Its computation is difficult and computationally expensive. In OPENCOSSAN, to avoid the calculation of the normalizing factor, the Transitional Markov Chain Monte Carlo have been used [21]. This algorithm allows the generation of samples from the complex-shaped unknown posterior distribution through an iterative approach:

$$P_i \propto P(D|\theta, I)^{\beta_i} P(\theta|I) \quad (56.5)$$

where m intermediate distributions P_i are introduced and the contribution of the likelihood is scaled down by an exponent β_i , with $0 = \beta_0 < \dots < \beta_i < \dots < \beta_m = 1$. Thus, the first distribution is the prior distribution, and the last one is the posterior distribution. These intermediate distributions show a more gradual change in the shape from one step to the next when compared with the shape variation from the prior to the posterior.

Each realization obtained from the posterior distribution at the end of the Bayesian updating procedure identifies a possible set of input parameters (i.e., realization of the epistemic uncertainty). The aleatory uncertainty (not reduced by the Bayesian inference method) can be propagated via a Monte Carlo simulation in order to compute the empirical CDF, \hat{F} , of the quantity of interest. The collection of \hat{F}^i obtained using different realizations from the epistemic space describes the p-box of model output. Hence, it is possible to compute the quantiles for each value of the model output and identify confidence bounds of \hat{F} . Confidence bounds are then compared with the empirical distributions of the available experimental realizations, F^e , obtained using Gaussian kernel smoother function. The confidence levels allow to identify the level of refinement of the updated intervals obtained from the posterior distributions.

Figures 56.33 and 56.34 show the posterior distributions obtained for the epistemic parameters. The horizontal red lines represent the cut-off (confidence bound) using the reduced epistemic uncertainty space. For instance, the imprecision of the mean value of p_1 is reduced from $[0.6, 0.8]$ to $[0.608, 0.726]$ and $[0.626, 0.761]$ using 25 and 50 experimental observations, respectively. The results show that the experimental observations do not allow to reduce all the epistemic uncertainties. In fact the data do not contain enough information to improve the knowledge of, e.g., the correlation coefficient between p_4 and p_5 ($\rho(p_1, p_2)$).

3.1.4 Sensitivity Analysis

The aim of sensitivity analysis is to identify and rank the parameters that contribute mostly to the variability of the output of a system h_1 . Two approaches can be used:

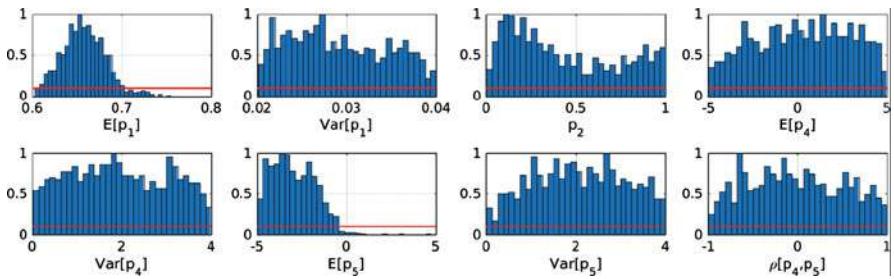


Fig. 56.33 Normalized histogram of $p(\theta | \mathcal{D}_e)$ obtained using approximate Bayesian computational method with (a) 25 experimental observations

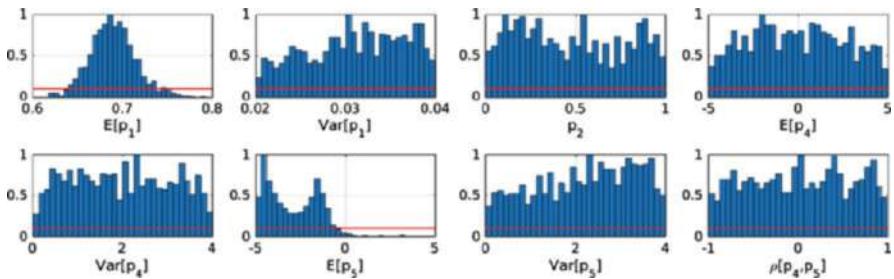


Fig. 56.34 Normalized histogram of $p(\theta | \mathcal{D}_e)$ obtained using approximate Bayesian computational method with 50 experimental observations of x_1

the Hartley-like measure of nonspecificity and the global sensitivity analysis based on the Sobol and total sensitivity measures. Both approaches can be used to deal with sensitivity analysis due to epistemic uncertainty.

The first approach is using the Hartley-like measure of nonspecificity, which is a measure of epistemic uncertainty and which does not require the calculation of the probability box associated to the output Dempster-Shafer structure after the application of the extension principle for random sets. The reader is referred to Ref. [2] for the explanation of the method. The second approach is based on global sensitivity analysis to estimate the Sobol and the total sensitivity indices [87]. The first-order Sobol indices are defined as

$$S_i = \frac{V_{X_i}[E_{X \sim i}(Y | x_i)]}{V[Y]} \quad (56.6)$$

where $V[Y]$ represents the unconditional variance of the quantity of interest and $V_{X_i}[E_{X \sim i}(Y | x_i)]$ the variance of conditional expectation. The total sensitivity index, T_i , measures the contribution to the output variance of x_i of the input factors including all interactions with any other input variables,

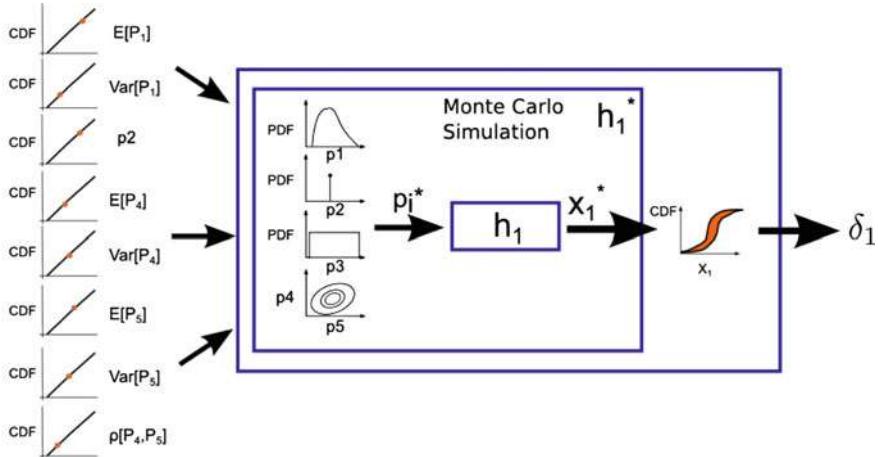


Fig. 56.35 Equivalent model used for performing global sensitivity analysis in the presence of epistemic and aleatory uncertainties. In the example shown here, the model h_1^* is used to perform the sensitivity analysis with respect to the variables x_1

$$T_i = 1 - \frac{V_{X_{\sim i}}(E_{X_i}(Y | x_{\sim i}))}{V(Y)}. \quad (56.7)$$

The global sensitivity approach cannot be applied directly to solve the problems where the uncertainty is described distributional/free p-boxes and intervals. In fact, this method requires the exact knowledge of the PDF of the input variables and the variance of a measurable model output. Global sensitivity analysis can be performed on a equivalent model (h^*), as shown in Fig. 56.35. In the model h^* , the epistemic uncertainties are represented by uniform distributions and they are the only inputs for the model. Then the model uses realizations of the input parameters to define probabilistic distribution for internal variables. The model performs an internal Monte Carlo analysis to compute a CDF of the quantity of interest. Finally, the difference between the estimated CDF and a reference CDF is computed (λ_1). Traditional algorithms (e.g., [41, 61, 77]) can be used to perform the global sensitivity analysis of the model h^* . Figure 56.36 shows an example of global sensitivity analysis obtained by adopting the equivalent model h^* and the Saltelli computational approach [77].

3.1.5 Uncertainty Propagation

For the uncertainty propagation considering aleatory and epistemic uncertainty, two main approaches exist. In the first approach, n samples from the aleatory space are drawn from the product copula $C : [0, 1]^{N_a} \rightarrow [0, 1]$ that models the aleatory dependence between the input variables. Each realization corresponds to a focal element in the theory of random set represented by α . Then using the extension principle (with the optimization method), each input focal element is

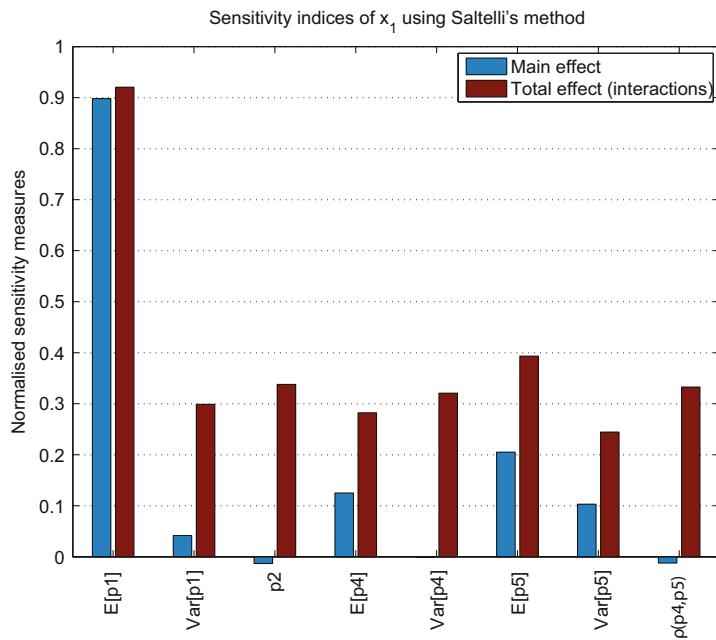


Fig. 56.36 Global sensitivity analysis with respect to the variable x_1 . The figure shows the first-order Sobol sensitivity measure and the total sensitivity indices. The results show that the first variable (mean of p_1) is the inputs that contribute the most to the variability of the output x_1

mapped through the model and a Dempster-Shafer structure with n intervals $[l_i, u_i]$ is finally obtained. Hence, each focal element has a basic mass assignment of $1/n$. It is important to point out that this approach models the probability boxes as distribution-free p-boxes. Furthermore, it requires the calculation of the image of a set through a function using the extension principle and an optimization method (Fig. 56.37).

In the second approach, the quantities of interest (e.g., mean and failure probability estimations) are used as objective functions of a global search in the epistemic space Θ . The global optimizations in the epistemic space $\Theta \equiv \times_{i=1}^{31} I_i$ are performed in order to find the set of parameters that produce the upper and lower bounds of the quantities of the interest (e.g., J_1 and J_2). Hence, for any candidate solution produced by the optimization algorithm $\theta_i \in \Theta$, a set of points $\{\alpha_j, j = 1, 2, \dots, n\}$ is randomly sampled from the aleatory space $\Omega \equiv (0, 1)^{17}$. Then, n realizations are generated according to the uncertainty models of p_1 to p_{21} and propagate to the model for computing the empirical CDF of the outputs of interest. The uncertainty propagation can be performed using simple Monte Carlo method or more advanced and efficient techniques (e.g., [25, 56, 58, 63]). Figure 56.38 shows the uncertainty quantification approach based on the global optimization in the epistemic space.

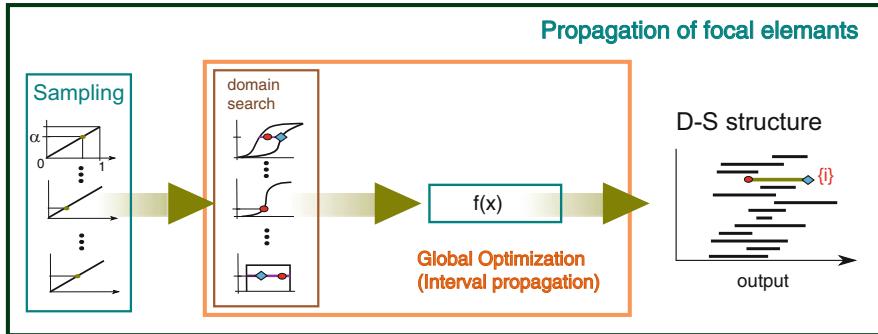


Fig. 56.37 Uncertainty quantification by means of focal element propagation. The approach requires sampling “ α -cut” and then propagates the intervals through the model and computes a Dempster-Shafer structure. The output bounds for each focal element are calculated by means of an optimization procedure

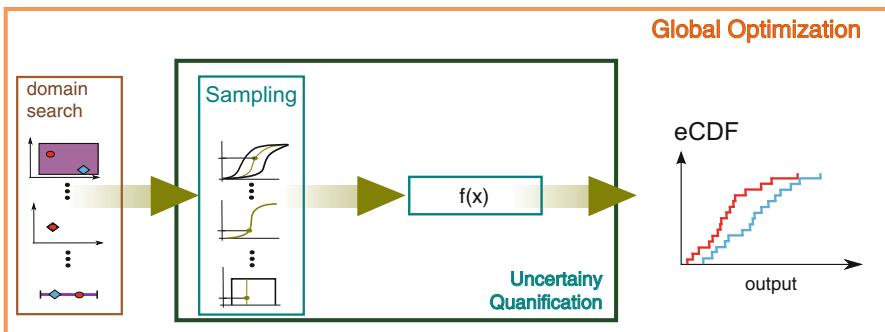


Fig. 56.38 Uncertainty quantification by means of a global optimization in the epistemic space. The inner model is a classical probabilistic problem and can be solved by adopting classical methods for uncertainty quantification

The uncertainty quantification required a double loop approach resulting in a rather challenging task in terms of computational cost. The lower and upper bound of the mean of the model response is obtained as

$$\underline{\mu} = \min_{\theta \in \Theta} \int_{\Omega} h(\alpha; \theta) dC(\alpha) \quad \bar{\mu} = \max_{\theta \in \Theta} \int_{\Omega} h(\alpha; \theta) dC(\alpha) \quad (56.8)$$

while the lower and upper bound of the probability of failure, defined as the exceedance of a critical threshold level h^{crit} of the model response, is obtained as

$$P_f = \min_{\theta \in \Theta} \int_{\Omega} \mathcal{I}[h(\alpha; \theta) > h^{crit}] dC(\alpha) \quad \overline{P_f} = \max_{\theta \in \Theta} \int_{\Omega} \mathcal{I}[h(\alpha; \theta) > h^{crit}] dC(\alpha). \quad (56.9)$$

In Eqs. 56.8–56.9 θ is a vector focal elements, C the copula, and θ the vector of aleatory uncertainty.

3.1.6 Robust Design

The final task in the design of a safety-critical system is to perform a robust design optimization. The main aim of the robust design is to consider explicitly the effects of the uncertainties in the optimization problem. This requires to repeatedly evaluate the performance of the system such as the expected values and probability of failure (the inner loop), which may require considerable numerical efforts, for each candidate solution of the optimization procedure (the outer loop).

Generally, in robust design only one bound is of interest. Nevertheless, the estimation of bounds of the system performance remains a computational challenging task. Thus, the direct solution of the robust design problem is infeasible and surrogate models need to be used. Surrogate models mimic the behavior of the original model, by means of an analytical expression with negligible computational cost. The approximation is constructed by selecting some predefined interpolation points in the design space, at which the failure probability is estimated; then, a surrogate model is adjusted to the data collected in a least square sense. OPENCOSSAN provides the access to different surrogate modes (see Table 56.4) and optimization strategies (see Table 56.3) that allow the analysis to perform efficient robust design.

3.1.7 Final Remarks

The development and design of robust safety-critical systems is a challenging problem since in general quantitative data is either very sparse or prohibitively expensive to collect. OPENCOSSAN provides numerically efficient and scalable tools that have allowed to solve each task required by the NASA Langley UQ challenge problem using two different approaches. Considering different approaches to solve the same engineering problem might be seen as a waste of resources and time. However, all the existing approaches for dealing with epistemic and aleatory uncertainty require fine-tuning of their parameters in order to be efficient and accurate. Hence, it is of paramount importance to be able to verify and cross-validate the results against a different procedure.

3.2 Large-Scale Finite Element Model of a Six-Story Building

In this example the reliability analysis of a six-story building subject to wind load is carried out. Three different models of uncertainty characterization are here considered. Firstly, a standard reliability analysis, where the inputs are modeled by precise probability distribution functions, is performed. Secondly, the structural parameters are modeled as imprecise random variables [14]. In the third analysis both imprecise random variables and intervals are considered for structural parameters.

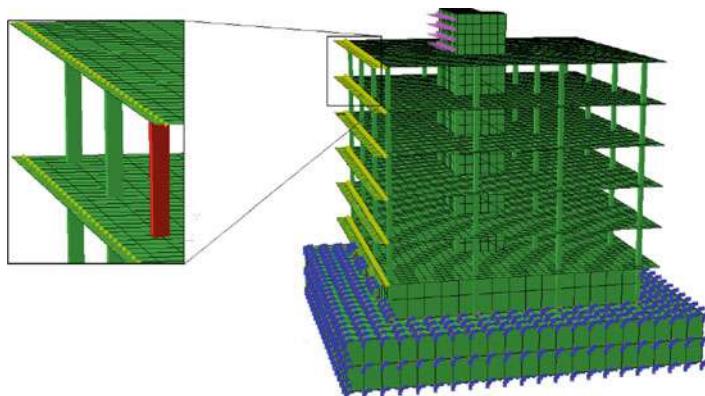


Fig. 56.39 FE model of the six-story building and selected critical component used in this analysis

Table 56.7 Distribution models for the input structural parameters for the six-story building

Parameter ID	Probability distribution	Description	Units
1	Normal (0.1, 0.001)	Column's strength	GPa
2–193	Uniform (0.36, 0.44)	Sections size	m
194–212	Log-normal (35.0, 12.25)	Young's modulus	GPa
213–231	Log-normal (2.5, 0.0625)	Material's density	kg/dm ³
232–244	Log-normal (0.25, 0.000625)	Poisson's ratio	—

3.2.1 The Finite Element Model

An ABAQUS finite element model (FEM) is built for the six-story building, as illustrated in Fig. 56.39, which includes beam, shell, and solid elements. The load is considered as combination of a (simplified) lateral wind load and the self-weight, which are both modeled by deterministic static forces acting on nodes of each floor. The magnitude of the wind load increases with the height of the building. The FEM of the structure involves approximately 8200 elements and 66,300 degrees of freedom. A total of 244 independent random variables are considered to account for the uncertainty of the structural parameters. The material strength (capacity) is represented by a normal distribution, while log-normal distributions are assigned to the Young modulus, the density, and the Poisson ratio. In addition, the cross-sectional width and height of the columns are modeled by independent uniform distributions. A summary of the distribution models is reported in Table 56.7.

Component failure for the columns of the 6th story is considered as failure criterion. The performance function is defined as the difference between the maximum Tresca stress, where $\sigma_{III} \leq \sigma_{II} \leq \sigma_I$ are the principal stresses, and the yield stress σ_y :

$$f(\boldsymbol{\theta}) = |\sigma_I(\boldsymbol{\theta}) - \sigma_{III}(\boldsymbol{\theta})| / 2 - \sigma_y, \quad (56.10)$$

3.2.2 Reliability Analysis

By using the uncertainty model reported in Table 56.7, a reliability analysis is carried out in COSSAN-X and OPENCOSSAN by means of an advanced Monte Carlo method, namely, Advanced Line Sampling [25]. The selected algorithm requires the definition of the so-called important direction. The latter is a direction that points toward the failure region. It can be provided by the analyst or it can be approximated by computing the gradient in the origin of the standard normal space. COSSAN-X can compute automatically the important direction (see Fig. 56.40), while the user needs only to define the total number of lines that will be used to estimate the failure probability.

The Advance Line Sampling procedure implemented in OPENCOSSAN and available via the graphical interface of COSSAN-X is a very efficient method. It allows to estimate accurately small values of failure probability using very limited number of samples. In fact, only 62 model evaluations (i.e., 30 lines) were necessary to estimate a failure probability of $\hat{p}_F = 1.42 \cdot 10^{-4}$ with a coefficient of variation of $CoV = 0.092$.

3.2.3 Imprecision in Distribution Parameters

In this second approach, it has been assumed that insufficient data are available to estimate exactly the parameters of the distributions in Table 56.7. Hence,

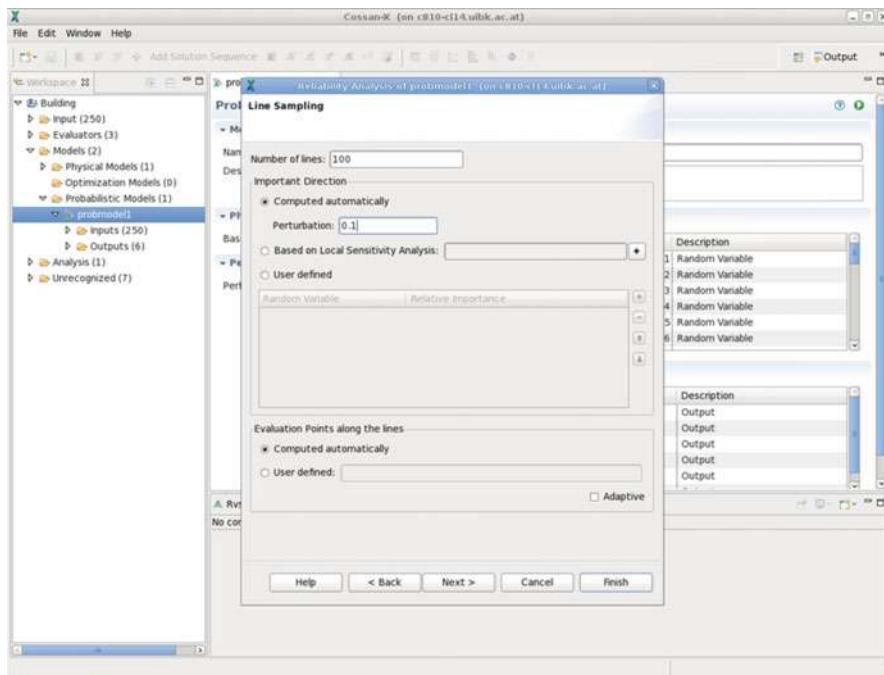


Fig. 56.40 Implementation of the reliability analysis in COSSAN-X. The screenshot shows the wizard page to define the parameters for the line sampling algorithm

intervals are used to model such indetermination. The interval parameters are represented as

$$\underline{p} = p_c (1 - \epsilon), \quad \bar{p} = p_c (1 + \epsilon) \quad (56.11)$$

using the interval center $p_c = (\underline{p} + \bar{p})/2$ and the relative radius of imprecision ϵ . These intervals $[\underline{p}, \bar{p}]$ are defined by a bounded set Q_1 of 488 parameters. In the example, all interval parameters are modeled with the same relative imprecision ϵ . In order to explore the effects of ϵ on the results, a fuzzy set is used to consider a nested set of intervals $\tilde{p} = \{\underline{p}, \bar{p}\}$ for the parameters defined by $\epsilon = \{0, 0.005, 0.01, 0.025, 0.05, 0.075\}$ as shown in Fig. 56.41. The reliability analysis with the generalized model of uncertainty is performed using the important direction determined in the physical space and then remapped in the standard normal space for each realization of the epistemic space.

Since intervals are used to characterize the uncertainty, it is not possible to estimate a single value for the failure probability. Instead, the maximum and minimum of the failure probability are computed. The failure probability is obtained as a fuzzy set, which includes the standard reliability analysis as special case with $\epsilon = 0$. Each interval for the failure probability, p_F , corresponds to the respective interval $\bar{p} = [\underline{p}, \bar{p}]$ in the input for the same membership level, and each membership level is associated with a different value ϵ . The results of the reliability analysis are shown in Fig. 56.41 (right) and summarized in Table 56.8.

OPENCOSSAN implements a very efficient algorithm to estimate the bounds of the failure probability. As shown in Table 56.8, the number of samples required to estimate the bounds of the failure probability is on average 254, which is even less than the total number of model evaluation required by two standard reliability analyses using line sampling (~ 360 samples). This is an astounding result considering that a standard approach to propagate aleatory and epistemic

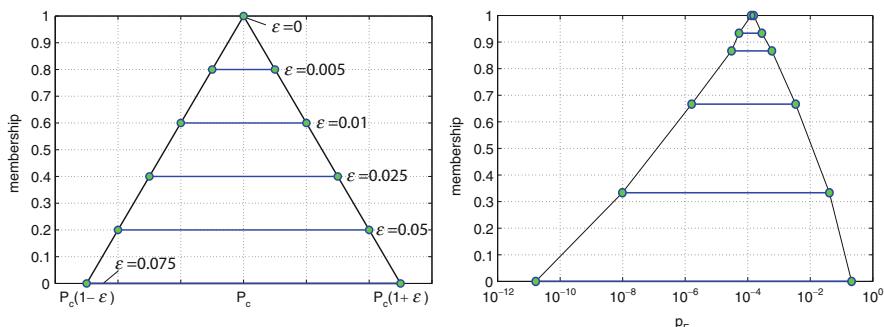


Fig. 56.41 On the left, the fuzzy parameters $\tilde{p} = \{p_c [1 - \epsilon_j, 1 + \epsilon_j]\}_{j=1}^6$ used to model the imprecision in the probabilistic model. On the right, the fuzzy failure probability obtained as a set of results for different levels of imprecision

Table 56.8 Results of the robust reliability analysis of the multistory building from model considering imprecision in distribution parameters. The results are obtained in terms of lower and upper bounds of the failure probability for different values of imprecision ϵ

Imprecision level ϵ	Lower bound		Upper bound		Number of samples N_s
	\underline{p}_F	CoV	\bar{p}_F	CoV	
0.000	$1.42 \cdot 10^{-4}$	$9.2 \cdot 10^{-2}$	$1.42 \cdot 10^{-4}$	$9.2 \cdot 10^{-2}$	126
0.005	$5.75 \cdot 10^{-5}$	$8.7 \cdot 10^{-2}$	$2.63 \cdot 10^{-4}$	$7.1 \cdot 10^{-2}$	257
0.010	$4.57 \cdot 10^{-5}$	$33.6 \cdot 10^{-2}$	$5.30 \cdot 10^{-4}$	$11.5 \cdot 10^{-2}$	250
0.025	$1.75 \cdot 10^{-6}$	$8.8 \cdot 10^{-2}$	$3.22 \cdot 10^{-3}$	$5.3 \cdot 10^{-2}$	253
0.050	$2.27 \cdot 10^{-8}$	$57.0 \cdot 10^{-2}$	$3.88 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$	255
0.075	$1.88 \cdot 10^{-11}$	$12.2 \cdot 10^{-2}$	$2.02 \cdot 10^{-1}$	$3.5 \cdot 10^{-2}$	254

Table 56.9 Input definition from uncertainty model. The relative radius of imprecision is $\epsilon = \{0, 0.01, 0.015, 0.020, 0.025, 0.03\}$

Parameter ID	Uncertainties type		$\bar{p} = p_c [1 - \epsilon, 1 + \epsilon]$, $\bar{x} = [\underline{x}, \bar{x}]$	
1	Distribution	$N(\bar{\mu}, \bar{\sigma}^2)$	$\mu_c = 0.1$	$\sigma_c^2 = 0.001$
2 – 193	Interval	\bar{x}	$\underline{x} = 0.36$	$\bar{x} = 0.44$
194 – 212	Distribution	$LN(\bar{m}, \bar{v})$	$m_c = 35$	$v_c = 12.25$
213 – 231	Distribution	$LN(\bar{m}, \bar{v})$	$m_c = 2.5$	$v_c = 0.0625$
232 – 244	Distribution	$LN(\bar{m}, \bar{v})$	$m_c = 0.25$	$v_c = 0.000625$

uncertainty, driven by two nested loops, would have required several hundreds of thousands of model evaluations (see, e.g., [69]).

3.2.4 Imprecision in Both Distribution Parameters and Structural Parameters

In this example, 192 input parameters $x \in \mathbb{R}^{192}$ are modeled as interval variables, while the 52 remaining structural parameters $\xi \in \mathbb{R}^{52}$ are considered as imprecise random variables (see Table 56.9). The imprecise distribution parameters are modeled using the radius of imprecision ϵ , as in the previous case. The relative radii of imprecision $\epsilon = \{0, 0.01, 0.015, 0.020, 0.025, 0.03\}$ are considered to construct a fuzzy model for all parameters (see Fig. 56.42 (left)).

The results of the reliability analysis are shown in Table 56.10 and in Fig. 56.42 (right). Again, the computational tool is very efficient requiring on average only 254 model evaluations for the estimation of the bounds for each level of imprecision, ϵ . The results show that the level of uncertainty is much larger compared to the previous cases. This is mainly due to the imprecision introduced in the modeling of the cross sections.

3.2.5 Final Remarks

COSSAN-X and the computational framework based on OPENCOSSAN implement very efficient strategies for reliability analysis adopting different representations of the uncertainty. The approaches couple an advanced sampling-based algorithm

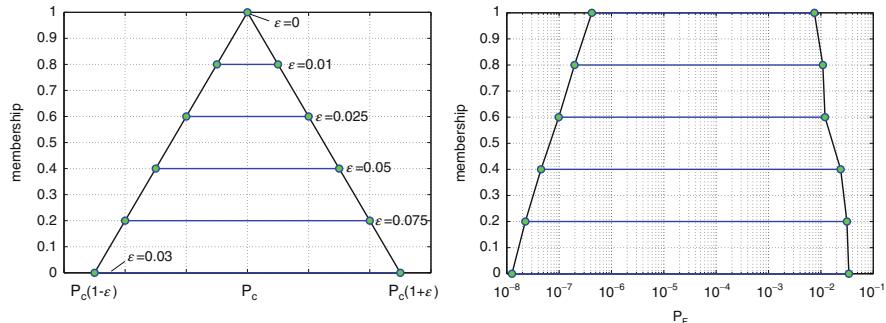


Fig. 56.42 On the left, the fuzzy distribution parameters $\tilde{p} = \{p_c [1 - \epsilon_j, 1 + \epsilon_j]\}_{j=1}^6$. On the right, the estimated fuzzy failure probability for the six-story building

Table 56.10 Results of the robust reliability analysis of the multistory building from model with imprecision in both distribution parameters and structural parameters. The results are obtained in terms of lower and upper bounds of the failure probability

Imprecision level	Lower bound		Upper bound		Number of samples
	\underline{p}_F	CoV	\overline{p}_F	CoV	
0.000	$4.70 \cdot 10^{-7}$	$10.2 \cdot 10^{-2}$	$6.73 \cdot 10^{-3}$	$11.5 \cdot 10^{-2}$	259
0.010	$2.28 \cdot 10^{-7}$	$13.4 \cdot 10^{-2}$	$9.71 \cdot 10^{-3}$	$12.2 \cdot 10^{-2}$	247
0.015	$1.10 \cdot 10^{-7}$	$10.3 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$	$7.6 \cdot 10^{-2}$	255
0.020	$5.19 \cdot 10^{-8}$	$13.1 \cdot 10^{-2}$	$2.08 \cdot 10^{-2}$	$14.6 \cdot 10^{-2}$	255
0.025	$2.51 \cdot 10^{-8}$	$9.97 \cdot 10^{-2}$	$2.72 \cdot 10^{-2}$	$15.3 \cdot 10^{-2}$	249
0.030	$1.40 \cdot 10^{-8}$	$9.94 \cdot 10^{-2}$	$3.21 \cdot 10^{-2}$	$6.5 \cdot 10^{-2}$	254

with optimization procedures. The advanced computational method dramatically reduces the computational costs of the reliability analysis without compromising the accuracy of results and allows to perform the reliability analysis adopting the real FE model without the necessity to train a surrogate model. The advantage of considering explicit imprecision can be fully appreciated in a design context. In fact, the results can be used to identify a tolerable level of imprecision for the inputs given a constraint on the maximum tolerable failure probability. For example, fixing an allowable failure probability of 10^{-3} , the maximum level of imprecision for the distribution parameters is limited to 1% (see Fig. 56.41). Moreover, the outputs show that when the level of imprecision is too large, the results become non-informative.

3.3 Robust Design of a Steel Roof Truss

In this numerical example, the linear static behavior of a steel roof truss is analyzed. The aim is to optimize the total volume of the structure, i.e., the quantity of material required for constructing the steel roof truss taking into account the effect of uncertainties.

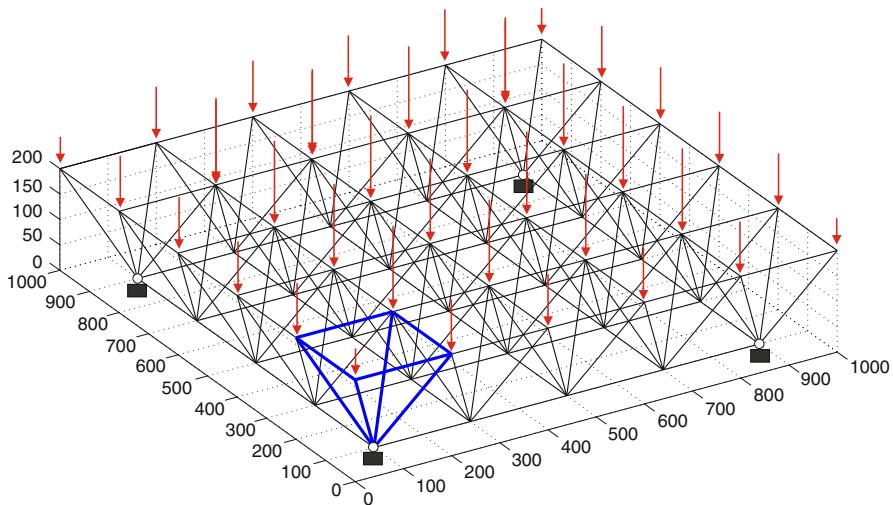


Fig. 56.43 Scheme of the steel roof truss and the load applied. The axes in the figures are expressed in centimeters

Table 56.11 Design variables and parameters of the steel roof truss

Parameter	Description
Design variable (A_1)	60 beams forming the top of the structure
Design variable (A_2)	100 beams connecting the top and the bottom of the structure
Design variable (A_3)	40 beams forming the bottom of the structure
maxDisp	Capacity of the system (10^{-3} [m])
Parameter	Distribution(μ, σ)
Load (L_1)	Normal (12000, 120) [N]
Load (L_2)	Normal (16000, 800) [N]
Load (L_3)	Normal (50000, 20000) [N]
Young's module (E)	Log-normal($2.0 \cdot 10^{11}, 1.05 \cdot 10^{10}$) [Pa]
Density (ρ)	Normal (7500, 150) [kg/m^3]

3.3.1 Description of the Problem

The steel roof truss, as shown in Fig. 56.43, is composed of 200 steel beams with different cross-sectional areas. A total number of three design variables are used to define the cross-sectional area of the structural beams according to the type and location as shown in Table 56.11. The grouping is carried out in order to make the optimization feasible, since an optimization of each single beam might not have been feasible.

It is imposed as a constraint of the optimization problem that the failure probability has to be lower than 10^{-4} . System failure is defined as the exceedance of the maximum allowable nodal displacement defining the performance function. In this context the word failure has to be intended as the occurrence of

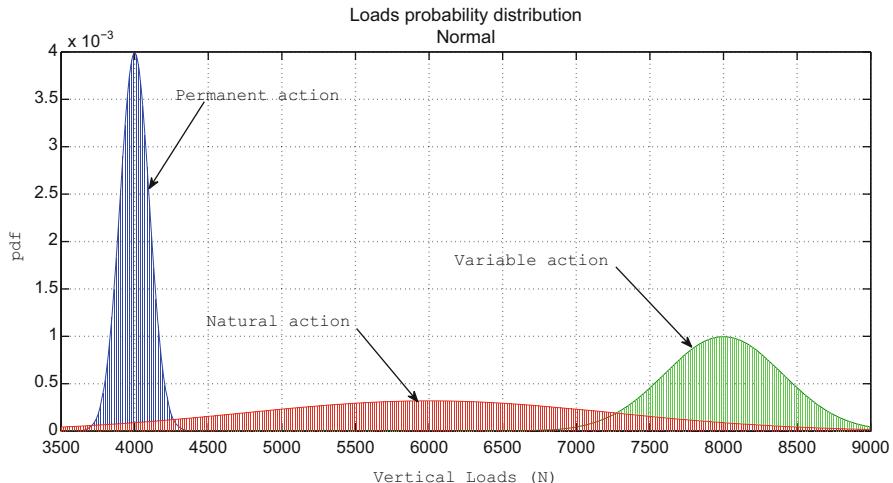


Fig. 56.44 Distribution of the nodal loads representing the permanent action, the natural action, and the variable action, respectively

structural response beyond the target assumption. The uncertainties considered in the numerical example are also summarized in Table 56.11. Nodal loads are modeled as normal independently distributed variables. Each load corresponds to different physical actions applied to the structure. The loads represent permanent, variable, and natural actions and are characterized by an increasing level of uncertainties as shown in Fig. 56.44. The density of the material is also modeled as a random variable.

Note that although the physical quantities such as load and material density are modeled using an unbounded distribution (normal distribution), the probability to sample a negative value is smaller than the maximum numerical precision available in Matlab. Hence this probability is treated as zero.

3.3.2 Analysis

The reliability-based optimization analysis is performed adopting the so-called direct approach. The COBYLA algorithm is used to drive the optimization procedure. Line sampling is used to perform the reliability analysis at each iteration step of the optimization procedure using approximately 60 model evaluations. The evolution of the design variables, objective function, and the constraint during the optimization is shown in Figs. 56.45, 56.46, and 56.47, respectively. The results of the analysis show that the total volume is decreased from the initial value of 6.3 to 5.7 [m³]. The evolution of the design variables shows that the beam section A_3 is larger than the starting design, while the design variables A_1 and A_2 were reduced. The failure probability of the system has been successfully reduced from an initial value of $1.3 \cdot 10^{-2}$ to the prescribed value lower than 10^{-4} .

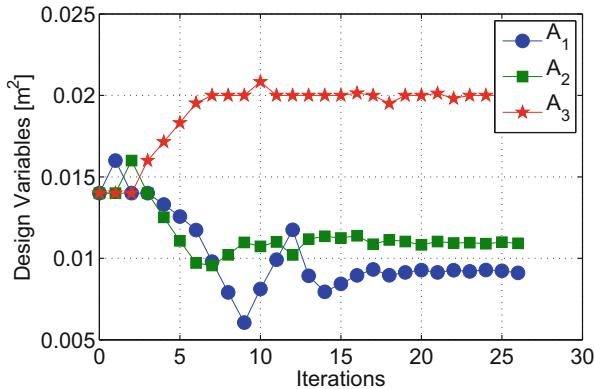


Fig. 56.45 Evolution of the design variables (A_1 , A_2 , and A_3) during the reliability-based optimization analysis

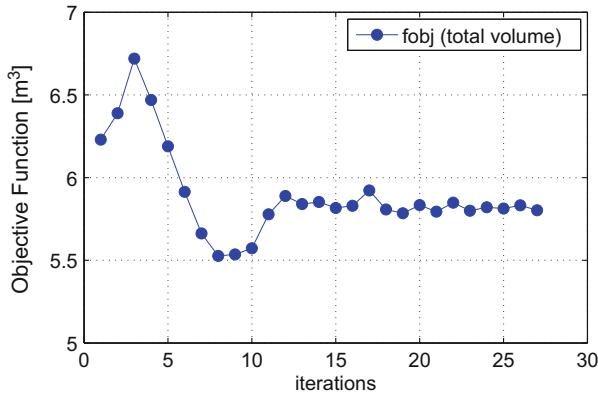


Fig. 56.46 Evolution of the objective function (i.e., total volume) during the reliability-based optimization analysis

3.3.3 Numerical Implementation

This example has been solved in OPENCOSSAN and a brief description of the script shown in Figs. 56.48, 56.49, 56.51, 56.52, and 56.53 is provided here. The script is self-commented and human readable since OPENCOSSAN does not use any acronyms or abbreviations. The script shows the combination of COSSAN objects used to perform the reliability-based optimization analysis that requires the definition of a reliability analysis (the inner loop) used to estimate the probability of failure and an optimization loop (the outer loop) used to identify the values of the design variables that minimize the objective function. First, the inputs of the models are defined. Uncertainties are modeled using *RandomVariable* objects (see lines 38–44 in Fig. 56.48), while the values of the beam sections and the maximum capacity of the system are defined by means of *Parameter* objects (see lines 52–54 and line

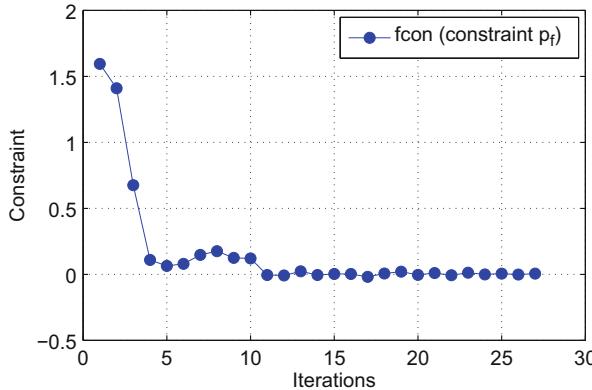


Fig. 56.47 Evolution of the constraint (i.e., max admissible failure probability) during the reliability-based optimization analysis

```

37 % Define random variables
38 - LoadS1 = RandomVariable('Sdistribution','normal','mean',12000,'std',120); % Loads
39 - LoadS2 = RandomVariable('Sdistribution','normal','mean',16000,'std',800); % Loads
40 - LoadS3 = RandomVariable('Sdistribution','normal','mean',50000,'std',20000); % Loads
41 - rho = RandomVariable('Sdistribution','normal','mean',7.5e3,'std',150,'Description','Steel Density');% density
42 - E = RandomVariable('Sdistribution','lognormal','mean',2.1e11,'std',0.05*2.1e11); % Young's moduli
43 - Xrvs = RandomVariableSet('CMembers',{LoadS1 'LoadS2' 'LoadS3' 'rho' 'E'},...
44 - 'Xrv',[LoadS1 LoadS2 LoadS3 E rho]);
45
46 % The starting values of the sections are assigned to parameters.
47 % 3 groups of beams are defined, characterized by the same section
48 % Structure material density is the same for each members
49
50 % The starting values of the sections are assigned to parameters.
51 % 3 groups of beams are defined, characterized by the same section
52 - A1 = Parameter('Value',0.01);
53 - A2 = Parameter('Value',0.01);
54 - A3 = Parameter('Value',0.01);
55
56 - totVolume=Function('Expression','<6A1>*40*2+<6A2>*2.4495*100+<6A3>*60*2;');
57 % Probabilistic model data
58 % Maximum displacement allowed
59 - displacementCapacity=Parameter('Value',0.001);
60 % Create input
61 % Add all the input quantities to an Input object
62 - Xinp = Input('CMembers',{Xrvs,A1,A2,A3,totVolume,displacementCapacity},...
63 - 'CMembers',{Xrvset,'A1','A2','A3','totVolume','displacementCapacity'});
```

Fig. 56.48 OPENCOSSAN script for the steel roof truss problem: definition of inputs

59 in Fig. 56.48, respectively). The total volume of the structure is calculated using a *Function* object (line 56). Finally, these objects are grouped in an *Input* object (lines 62–63).

The model of Fig. 56.43 is solved in Matlab and evaluated by a function defined in a *Mio* object (lines 67–71 of Fig. 56.49). The “solver” is included in an *Evaluator* object that allows to define the parallelization strategy (i.e., using a cluster/grid computing). The combination of the *Evaluator* and the *Input* object defines a *Model* object (line 77 of Fig. 56.49). The uncertainty quantification can be performed defining a simulation method and applied to the defined model. In this example a *MonteCarlo* simulator has been defined with 1000 samples (line 80) and used to evaluate the model (line 82). The results of the analysis are stored in a

```

64 %% Model definition
65 % A matlab function is used to compute the maximum displacement of the
66 % truss structure.
67 - Xmio = Mio('Spath',ExamplePath, ...
68     'Sfile','script_FFastStaticResponse.m', ...
69     'Lfunction',false,'LIOstructure',true, ...
70     'CinputNames',Xinp.Cnames, ...
71     'CoutputNames',{['maxDisp']} );
72 % Define an evaluator
73 - Xeval = Evaluator('CXmembers',{Xmio},{'CSnames', {'Xmio'}});
74 % Preparation of the Physical Model
75 - Xmodel = Model('Xevaluator',Xeval,'Xinput',Xinp);
76 % Run a deterministic analysis to check the model
77 - Xout = Xmodel.deterministicAnalysis;
78 %% Uncertainty Quantification
79 % Define Simulation method
80 - Xmc=MonteCarlo('Nsamples',1000);
81 % preform Analysis
82 - XsimOutMC=Xmc.apply(Xmodel);
83 %% Plot Results
84 - VmaxDisp=XsimOutMC.getValues('Sname','maxDisp');
85 - figure, [nout,xout]=hist(VmaxDisp,100);title('max displacement');

```

Fig. 56.49 OPENCOSSAN script for the steel roof truss problem: uncertainty quantification

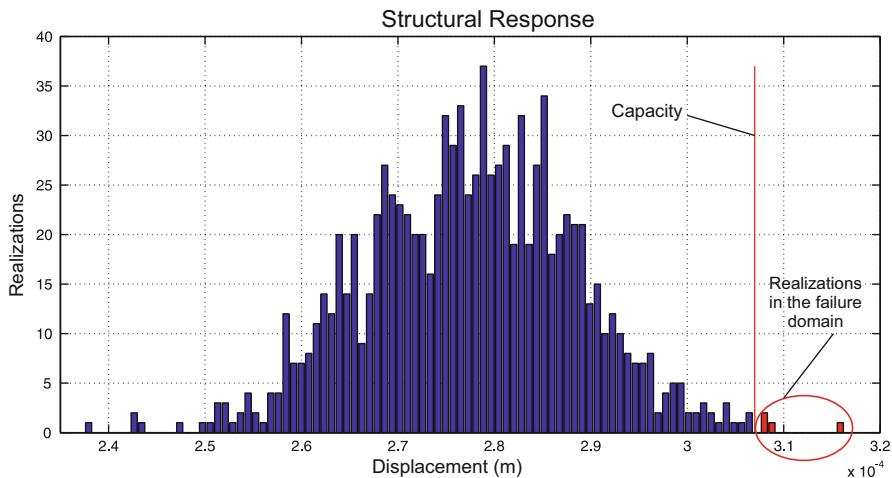


Fig. 56.50 Steel roof truss problem: uncertainty quantification of the maximum displacement

SimulationData object (*XsimOutMC*) and the quantity of interest is extracted (lines 84–85) for the post-processing of the results as shown in Fig. 56.50.

The script in Fig. 56.51 shows the definition reliability analysis. It requires the definition of a *ProbabilisticModel* object that combines a *Model* object with a *PerformanceFunction* (see line 91 of Fig. 56.51). This is defined by creating a *PerformanceFunction* object. In this example the performance function is defined as “capacity minus demand” where the capacity is defined by a parameter defined

```

87 % Reliability analysis
88 % Performance Function
89 - Xperfun = PerformanceFunction('Sdemand','maxDisp','Scapacity','displacementCapacity','Soutputname','Vg');
90 % Define a Probabilistic Model
91 - XprobModel=ProbabilisticModel('Xmodel',xmodel,'XperformanceFunction',Xperfun);
92 % Estimate failure probability adopting Advance Monte Carlo methods
93 - Xls =LineSampling('Nlines',50,'Ladaptive',true);
94 - [XpfLS,XoutLS]=XprobModel.computeFailure(Xls);
95 % Show the results of the reliability analysis
96 - display(XpfLS)
97 % Let have a look at the lines
98 - XoutLS.plotLines('Style','Space Truss Roof: LineSampling + Gradient','Ldistance',false);
99 % XoutLS.plotLines('Ldistance',false);
100 - display(XpfLS.pfhat)

```

Fig. 56.51 OPENCOSSAN script for the steel roof truss problem: reliability analysis

```

120 % Define an Optimization problem
121 % The optimization problem requires at least 1 Design Variable
122 % Define Design Variables
123 - XdvA1=DesignVariable('value',A1.value,'lowerBound',0.005,'upperBound',0.02);
124 - XdvA2=DesignVariable('value',A2.value,'lowerBound',0.005,'upperBound',0.011);
125 - XdvA3=DesignVariable('value',A3.value,'lowerBound',0.005,'upperBound',0.0105);
126
127 - totVolumeDV=Function('Sexpression','<&XdvA1>*40*2+<&XdvA2>*2.4495*100+<&XdvA3>*60*2;');
128 % Target failure probability
129 targetPf=Parameter('value',1e-4');
130
131 % Define Input object for OptimizationProblem
132 - Xdvinput = Input('Sdescription','Test Input','CSmembers',...
133 - {'XdvA1','XdvA2','XdvA3','totVolumeDV','targetPf'},'CXmember',{XdvA1,XdvA2,XdvA3,totVolumeDV,targetPf});
134 % Define the objective function
135 % The objective function is the minimization of the failure probability
136 % associated to the ProbabilisticModel defined above.
137 - XobjFun = ObjectiveFunction('Sdescription','obtaining target Pf',...
138 - 'Script','for n=1:length(Tinput), Toutput(n).fobj=Tinput(n).totVolumeDV; end',...
139 - 'Cinputnames',{''totVolumeDV''},...
140 - 'Coutputnames',{''fobj''});
141
142 - Xconst = Constraint('Sdescription','constraint Pf',...
143 - 'Script','for n=1:length(Tinput), Toutput(n).fcon=Tinput(n).pf-Tinput(n).targetPf; end',...
144 - 'Cinputnames',{''pf'',''targetPf''},...
145 - 'Coutputnames',{''fcon''});

```

Fig. 56.52 OPENCOSSAN script for the steel roof truss problem: optimization problem

in the input object (`displacementCapacity`) and the demand is the output computed by the solver (`MaxDisp`).

The reliability analysis is performed defining a simulation object (e.g., `LineSampling` calling the method `computeFailure` (line 94 in Fig. 56.51).

The optimization problem is defined by creating a new `Input` object containing design variables as shown in lines 123–132 in Fig. 56.52. Objective function and constraint are defined invoking the construction method for `ObjectiveFunction` and `Constraint`, respectively.

A reliability-based optimization problem is defined by combining a `ProbabilisticModel` object, a simulator object to perform the reliability analysis, and `ObjectiveFunction` and `Constraint` objects and the mapping between design variables and quantities defined in the inner loop as shown in lines 148–157 in Fig. 56.53. The COBYLA optimization approach is defined (see line 159) and the reliability-based optimization is performed invoking the method `optimize`. Finally, the results of the RBO analysis are plotted.

Finally an optimization method is selected (`MyOptimizer`) and passed to the method `optimize` of the `RBO` object to perform the analysis. Multiple optimization

```

147 % Define RBO problem
148 - XrboProblem = RBOProblem('Sdescription','Simple RBO problem for truss structure', ...
149   'XprobabilisticModel',XprobModel, ...
150   'Xsimulator',xls, ...
151   'Xinput',Xdvinput, ... % input containing the Design Variable
152   'XobjectiveFunction',XobjFun, ...
153   'Xconstraint',Xconst, ...
154   'SfailureProbabilityName','pf',... % Name of the failure probability
155   'CdesignvariableMapping',{'XdvA1' 'A1' 'parametervalue';...
156     'XdvA2' 'A2' 'parametervalue';...
157     'XdvA3' 'A3' 'parametervalue'}});
158 %% Performing optimization
159 - Xoptimizer=COBYLA('initialTrustRegion',0.0025,'finalTrustRegion',0.0001);
160 - Xoptimum=XrboProblem.optimize('Xoptimizer',Xoptimizer);
161 % Show results
162 - display(Xoptimum)
163 % Plot graphs
164 - Xoptimum.plotDesignVariables;
165 - Xoptimum.plotObjectiveFunctions;
166 - Xoptimum.plotConstraints

```

Fig. 56.53 OPENCOSSAN script for the steel roof truss problem: reliability-based optimization analysis

procedures and efficient approximate algorithms can be chosen to solve the problem. COBYLA method is used in this example.

Reliability-based optimization can also be performed in COSSAN-X. The graphical user interface provides wizards and intuitive problem definition that allows the user to perform robust design and reliability-based optimization (an example of wizard that assists the user in the definition of a reliability-based optimization (RBO) is shown in Fig. 56.54).

3.3.4 Final Remarks

It is important to note that the total computational efforts required by the reliability-based optimization analysis adopting the line sampling and COBYLA (1200 model evaluations) represent only a small fraction of a single direct reliability analysis based on Monte Carlo simulation ($\approx 10^5$ model evaluations). The procedure for the robust design presented here is very general and it can be easily adopted for the robust design of different structures and systems. In addition, the user has the flexibility to select and use different optimization and reliability algorithms. For instance, subset simulation [56] can be used to estimate the failure probability in the inner loop and genetic algorithms can drive the optimization search.

3.4 Robust Maintenance Scheduling Under Aleatory and Epistemic Uncertainty

Maintenance activities are important strategies to prevent loss of serviceability or even collapse of structures. A vast amount of operational costs are associated with inspection and eventual repair, which adds up to a huge cost of failure. Due to the unavoidable uncertainties present in inspection and repair activities as well as in the

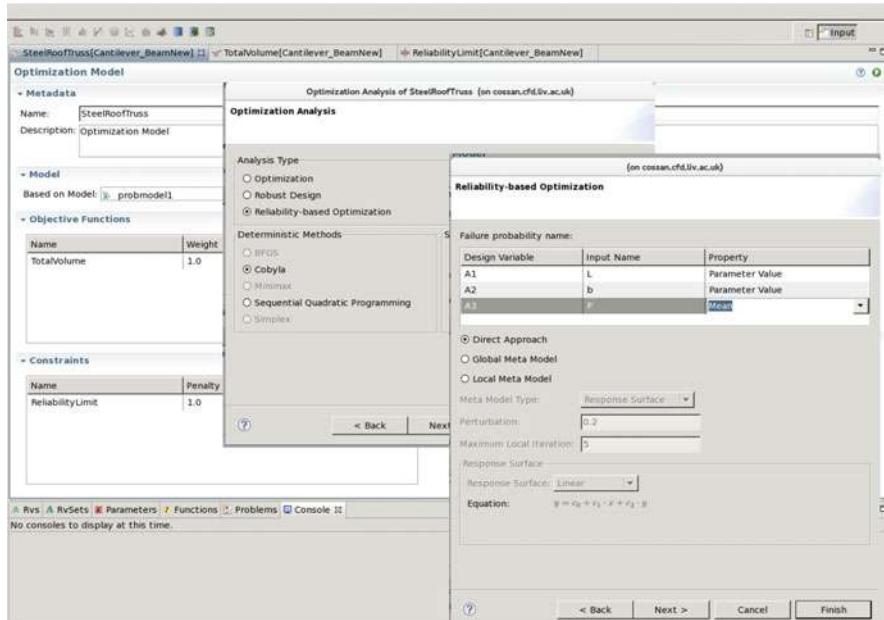


Fig. 56.54 COSSAN-X: definition of a reliability-based optimization problem, the wizard to select optimization algorithm and definition of the mapping between the inner loop (reliability analysis) and the outer loop (optimization analysis)

model performance prediction, scheduling maintenance activities is a challenging task. It requires the availability of efficient maintenance strategies in order to better quantify risk and to devise effective resilience solutions. Reliability-based optimization (RBO) offers a systematic and robust approach for making decisions under uncertainties. The RBO is a subset of classical probability theory and it can be combined with generalized probabilistic methods in order to deal with diverse representations of the uncertainties that includes variability, imprecision, incompleteness, vagueness, ambiguity, indeterminacy, dubiety, subjective experience, and expert knowledge [14, 100]. This extension allows one to obtain robust maintenance strategies since the solution can be obtained without introducing either artificial or unrealistic assumptions.

3.4.1 The Model

The design of robust maintenance strategies for metallic structures subject to cyclic loading is here analyzed. The maintenance activities are performed in order to prevent the failure of a welded connection in a simplified model of a bridge structure (Fig. 56.55). Due to cyclic loading caused by the vehicles, traffic on the bridge deck, metallic components tend to develop fatigue cracks. Fatigue is a localized damage process of a component produced by cyclic loading. As these cracks propagate,

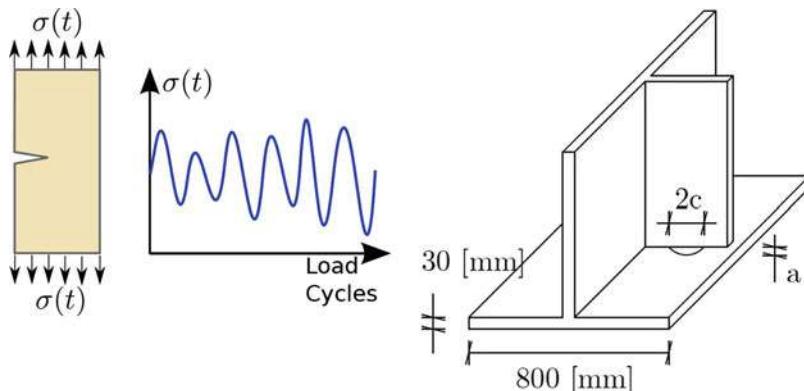


Fig. 56.55 Fatigue crack due to cyclic loading and a simplified model of welded connection of a bridge structure [22]

the structural system accumulates damage that may lead to loss of serviceability or collapse.

One of the most effective and traditional approaches to model crack propagation is the so-called S-N curves approach originally proposed by [55] that describes the relationship between stress amplitude and the number of load cycles. The crack propagation is modeled using the Paris-Erdogan law:

$$\frac{da}{dN} = C(\Delta K)^m \quad (56.12)$$

where N represents the number of load cycles, a represents the crack length, ΔK is the stress intensity factor, and C and m are two parameters that depend on the material properties. The failure condition for the welded component is given as the stress intensity factor that exceeds the material's toughness. This condition can be expressed as the crack length exceeding a critical value as shown in Eq. (56.12). Hence, Eq. (56.12) can be integrated with respect to the number of load cycles until the failure condition is reached. The Paris-Erdogan law is appropriate for characterizing crack growth under constant amplitude cyclic loading, small-scale yielding (i.e., yielding ahead of the crack tip), and long cracks. For those cases where these conditions are not met, the Paris-Erdogan law may not be appropriate and alternative models should be considered. In particular, note that the Paris-Erdogan law cannot model the crack initiation stage. Here, it is assumed that structures possess initial cracks of length. More specifically, the initial crack length, a_0 , is modeled with a log-normal distribution:

$$p(a_0) = \frac{1}{a_0 \sigma \sqrt{2\pi}} e^{-\frac{(\ln a_0 - \mu)^2}{2\sigma^2}}, \quad a_0 > 0. \quad (56.13)$$

The critical crack length is set as 15 mm. The parameters of the Paris-Erdogan law are taken as $m = 2.4$ and $C = 2 \cdot 10^{-10} \text{ mm}/\text{cycle(N/mm}^{1.5}\text{)}^{2.4}$, while the amplitude of the alternating stress applied is 30 MPa. Imprecise probability is adopted to characterize the mean value of the initial crack length μ_{a_0} and modeled as a fuzzy variable with a triangular membership function defined by the triplet {0.5, 1, 1.5} mm and a standard deviation of 0.4 mm. In this model, the maintenance activities consist of a nondestructive inspection and repair. A nondestructive inspection is a procedure that is used to detect cracks in a structure without introducing additional damage. This includes ultrasonic, magnetic-particle, liquid penetrant, radiographic, remote visual inspection. Such inspection activities are not perfect, and the outcome of the inspection (i.e., probability of detection) can be modeled as follows:

$$POD = (1 - p)(1 - e^{-\lambda a}) \quad (56.14)$$

where λ represents the quality of the inspection, a crack length at the time of the inspection and p the probability of non-detection of a very large crack that depends on the quality of the inspection. If the crack is detected, the crack will be repaired and it is assumed that the reparation will be perfect (i.e., no crack after a reparation activity).

During the target lifetime of the structure, the following events may occur: the crack length reaches a critical value and fracture occurs at a time before inspection takes place; the structure survives until a nondestructive inspection is carried out. The inspection may not detect any cracks and hence no repair is carried out. In case one or more cracks are detected, the structure is repaired.

Despite the unavoidable uncertainties, the selection of appropriate time of inspection, t_I (and eventual repair), is of fundamental importance for scheduling effective maintenance activities. A robust maintenance scheduling is defined as the maintenance strategy that minimizes the total costs for inspection, repair, and failure. These costs are affected by uncertainties. For instance, the initial crack length, a_0 , affects the probability of failure of the system, and hence the expected failure cost and the quality of inspection λ affect the cost of inspection. The maintenance problem can be generally formulated as a constrained optimization problem, where the constraint represents the limit state safety level that the system has to comply with.

Given a system that evolves in time, $S(t)$, a mission time, T_M , which is the time when the system is required to function as specified, and a number of inspections, N , performed at times, $t^{\text{insp}} \in \mathbb{R}^N$, the maintenance problem is formulated as an optimization task, where both objective and constraints require the evaluation of the reliability, $r(t)$. Three main different costs can be identified: the costs due to inspection C_I and repair C_R and the cost due to the failure (and their consequences), C_F . It is assumed that manufacturing costs are deterministic as they are linked to construction and usage of materials. Note that, as pointed out in [93], the costs of

```
% =====
% OPENCOSSAN (http://www.cossan.co.uk) General purpose matlab toolbox for risk and uncertainty quantification
% Authors: Edoardo Patelli and Marco de Angelis
% =====
%% Definition of the Input
% Definition of the Design Variable (load cycles)
timeInspection = DesignVariable('Sdescription','Time of Inspection','lowerBound', 0.4e6,'upperBound',1.6e6);
% Definition of the Fuzzy variable
meanInitialCrack = FuzzyVariable('VsupportPoints',[0.5 1 1.5],'ValphaLevels',[0 0.5 1]);
% Collect DesignVariable and Fuzzy Variable in the Input Object
MyInput = Input('XdesignVariable',timeInspection,'XfuzzyVariable',meanInitialCrack);
%% Definition of the Probabilistic Model
% MyEvaluator provides the link to the deterministic model
MyModel = Model('Xinput',MyInput,'Xevaluator',MyEvaluator);
% Definition of the Performance Function
MyPerformanceFunction = PerformanceFunction('Sdemand','crackLength','Scapacity','criticalValue','Soutputname','Vg');
% Definition of the Probabilistic Model
MyProbabilisticModel = ProbabilisticModel('Xmodel',MyModel,'XperformanceFunction',MyPerformanceFunction);
%% Definition of the Reliability Based Optimization
% Definition of the Reliability solver
MySimulator = ImportanceSampling('Nsamples',100);
% Definition of the Objective function (Expected value of cost)
MyObjectiveFunction = ObjectiveFunction('Sscript','expectedCost=pf*Cr+pr*Cr');
% Definition of the RBO problem and consequent mapping in design space
MyRBOProblem = RBOProblem('XProbabilisticModel',MyProbabilisticModel,'Xsimulator',MySimulator, ...
'XobjectiveFunction',MyObjectiveFunction,'StailureProbabilityName','pf','alphaCut',0.5,'intervalMeasure','centralValue');
% Definition of the Optimization solver
MyOptimizer = Cobyla('NmaxModelEvaluations',60);
%% Performing RBO analysis
Xoptimum = MyRBOProblem.optimize('Xoptimizer',MyOptimizer);
```

Fig. 56.56 Example of a OPENCOSSAN script used to solve the robust optimization problem including the definition of imprecise variables (fuzzy variables)

repair and failure are obtained as expected values, $E[\cdot]$, as they are obtained from the estimation of repair and failure probability, respectively.

The cost due to inspections depends on inspection quality, λ , and on the inspection times, t_I . In this example the dependence of the time of inspection is not considered and $C_I(\lambda, t_I) \approx C_I(\lambda)$. Costs due to repair occur only if a crack is detected. Hence they depend on the probability of repair and in turn to the inspection quality and the crack length, a . The cost of failure is also function of the inspection quality as well as on the state of damage (i.e., crack length a). The total cost of maintenance is (the objective function)

$$E[C_T] = C_I(\lambda) + E[C_R(\lambda)] + E[C_F(\lambda, t)]. \quad (56.15)$$

It is important to notice that the evaluation of the objective function (Eq. (56.15)) requires solving a reliability problem which significantly increases the computational costs of the analysis. Furthermore, since some input variables are modeled with imprecise probability, the outputs are also affected by imprecision (i.e., only bounds of the expected costs can be obtained).

The maintenance problem can be solved by using advance simulation techniques available in OPENCOSSAN. An example of the script required to perform the robust optimization is shown in Fig. 56.56. The solution of this problem requires the definition of an inner loop to solve the reliability problem. In this example,

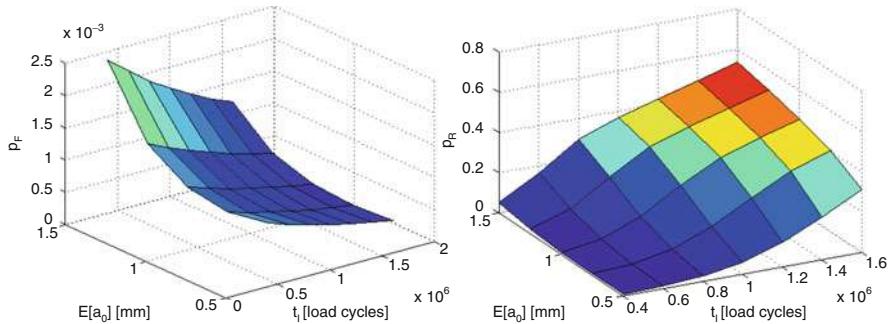


Fig. 56.57 Probability of failure (left plot) and repair (right plot) as a function of time (i.e., load cycles) and the membership function (representing the imprecision level) of the initial crack length

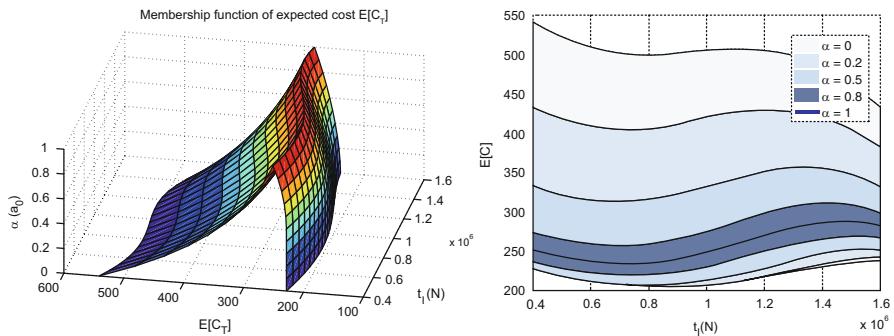


Fig. 56.58 Expected total cost as a function of the time of inspection for different values of the membership function (α) of the initial crack length

importance sampling procedure [38, 62] has been used to estimate the probability of failure of the metallic component. The gradient-free COBYLA algorithm [70] is selected to drive the optimization procedure.

3.4.2 Results

The results of the robust scheduling maintenance are shown in Figs. 56.57 and 56.58. Figure 56.57 shows the estimated probability of failure and probability of repair of the metallic structures subject to cyclic loading. It shows that when the inspection is performed too early, the cost tends to increase. This is due to the fact that by the time inspection takes place, the crack is still too small to be detected. As a consequence, there is a small probability of detecting (and repair) the crack. As time increases, the inspection becomes more effective in detecting the crack. As a consequence the probability of failure is reduced. This implies the total expected costs to become minimal. If the inspection is performed too late, the total expected costs increase again due to a nonnegligible probability of failure before the inspection. The optimal inspection time identified is equal to 0.74 million load cycles. Figure 56.58 shows

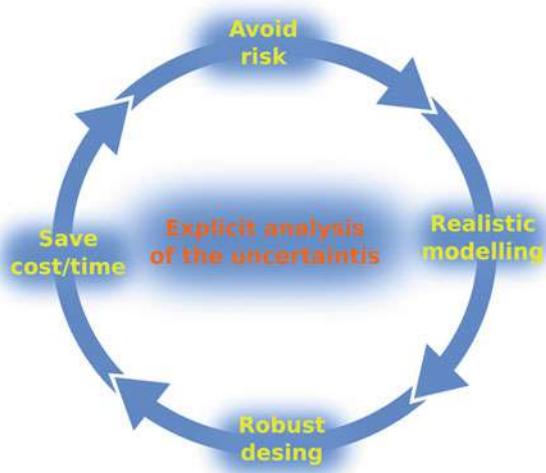
the expected total costs $E[C_T]$ as a function of the time of inspection t_I for different levels of imprecision α in the initial crack length $\mu(a_0)$. $\alpha = 1$ correspond to a traditional probabilist analysis where no imprecision in the model parameters is present.

The results obtained in this numerical example have shown the relevance of considering uncertainty in the scheduling of maintenance activities for fatigue-prone metallic components. In fact, the optimal time for performing maintenance is a compromise between repair activities and the negative consequences of failure. Considering explicitly epistemic uncertainties allows to identify the robustness of the results as a function of the imprecision of the input parameters. The numerical strategy adopted computes the upper and lower values of the expected costs for different alpha levels by means of a global optimization strategy combined with an efficient reliability approach. This allows creating models capable of more rationally processing the uncertainties that are no longer based only on traditional probabilistic approaches and providing to the final user self-contained measures of robustness.

4 Conclusions

Stochastic analysis is the basis for designing more competitive, reliable, and resilient products on different engineering fields such as automotive and nuclear industry, aerospace, mechanical and civil engineering, and more. It supports sustainable developments and economical and safety relevant decisions in our society (Fig. 56.59). To ensure a faultless life of complex technological installations, engineering systems, and products and to provide decision margins, the explicit consideration of all the uncertainties and threats needs to be considered. Uncertainty and imprecision are unavoidable since they are, e.g., inherent within manufacturing

Fig. 56.59 Advantages of the explicit consideration of the uncertainties



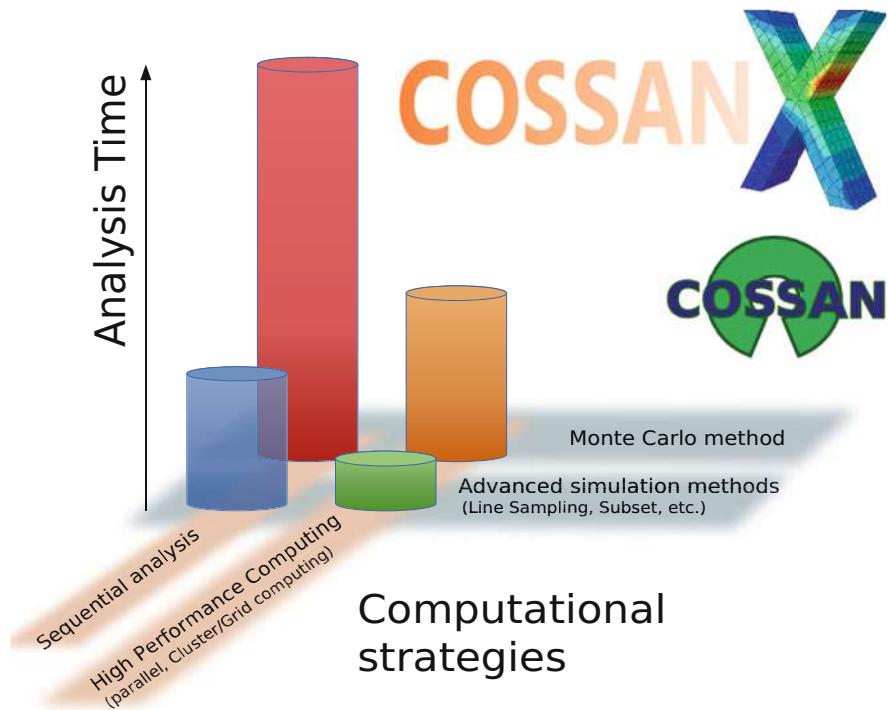


Fig. 56.60 Computational costs required by a nondeterministic analysis and advantages of using advanced simulation methods and high-performance computing implemented in the COSSAN software

process, fatigue and corrosion, human errors, and extreme load conditions (e.g., wind, wave, earthquake). In consequence, realistic consideration and treatment of uncertainties of various nature and scales is a key issue in the development of sustainable, durable, cost-effective, and feasible engineering solutions.

The merits of considering uncertainties are manifold and industry is fully aware that stochastic methods offer a much more realistic approach for analysis and design. However, the utilization of such approaches in practical applications remains quite limited. One common limitation is that the computational cost of stochastic analysis is often by orders of magnitude higher than the deterministic analysis. These computational costs can be significantly reduced combining efficient numerical strategies with high-performance computing (see Fig. 56.60). In this way nondeterministic analysis can be included as a common practice in computational models and numerical simulations allowing engineers to design products faster and cope with risk and uncertainty.

The general-purpose stand-alone application COSSAN-X is an easy-to-use yet powerful software for uncertainty quantification and risk management. Its user-friendly graphical interface allows to create a bridge between the academic research

and the industrial practice. The reason is that COSSAN-X allows nonexpert users in programming and in stochastic analysis to account for the uncertainty in their models in a straightforward manner and without an excessive learning curve. It also represents an indispensable tool for training professionals and students. This is because stochastic analysis can be taught and learned without the necessity to write ad hoc programs or scripts, and moreover, stochastic analyses are performed using the same (deterministic) models that the users are already familiar with.

The open-source model adopted for the computational engine (i.e., OPENCOSSAN) encourages the cross-discipline utilization of stochastic analysis. The open-source approach makes the software development more sustainable, continuously updated to make the cutting-edge technologies available to a large number of developers, researchers, and academics resulting in a reduction of code duplication, an increasing of the software reliability, and, finally, enable world-class research.

References

1. Alvarez, D.A.: Infinite random sets and applications in uncertainty analysis. PhD thesis, Arbeitsbereich für Technische Mathematik am Institut für Grundlagen der Bauingenieurwissenschaften. Leopold-Franzens-Universität Innsbruck, Innsbruck. Available at <https://sites.google.com/site/Diegoandresalvarezmarin/R斯thesis.pdf> (2007)
2. Alvarez, D.A.: Reduction of uncertainty using sensitivity analysis methods for infinite random sets of indexable type. *Int. J. Approx. Reason.* **50**(5), 750–762 (2009)
3. Alvarez, D.A., Hurtado, J.E.: An efficient method for the estimation of structural reliability intervals with random sets, dependence modelling and uncertain inputs. *Comput. Struct.* **142**, 54–63 (2014)
4. Au, S.K., Beck, J.: Estimation of small failure probabilities in high dimensions by subset simulation. *Probab. Eng. Mech.* **16**(4), 263–277 (2001)
5. Au, S.K., Patelli, E.: Subset Simulation in finite-infinite dimensional space. *Reliab. Eng. Syst. Saf.* 2016, 148, 66–77
6. Aven, T., Zio, E.: Some considerations on the treatment of uncertainties in risk assessment for practical decision making. *Reliab. Eng. Syst. Saf.* **96**, 64–74 (2011)
7. Barber, S., Voss, J., Webster, M.: The rate of convergence for approximate Bayesian computation. arXiv preprint, arXiv:13112038 (2013)
8. Beaurepaire, P., Valdebenito, M., Schüller, G.I., Jensen, H.: Reliability-based optimization of maintenance scheduling of mechanical components under fatigue. *CMAME* **221–222**, 24–40 (2012)
9. Beck, J.L., Katafygiotis, L.S.: Updating models and their uncertainties. I: Bayesian statistical framework. *J. Eng. Mech. ASCE* **124**(4), 455–461 (1998)
10. Beer, M., Ferson, S.: Fuzzy probability in engineering analyses. In: Ayyub, B. (ed.) *Proceedings of the First International Conference on Vulnerability and Risk Analysis and Management (ICVRAM 2011) and the Fifth International Symposium on Uncertainty Modeling and Analysis (ISUMA)*, pp. 53–61, 11–13 Apr 2011, University of Maryland, ASCE, Reston (2011)
11. Beer, M., Ferson, S.: Special issue of mechanical systems and signal processing “imprecise probabilities—what can they add to engineering analyses?”. *Mech. Syst. Signal Process.* **37**(1–2), 1–3 (2013). doi:<http://dx.doi.org/10.1016/j.ymssp.2013.03.018>, <http://www.sciencedirect.com/science/article/pii/S0888327013001180>

12. Beer, M., Patelli, E.: Editorial: engineering analysis with vague and imprecise information. *Struct. Saf.* **52**, Part B, 143 (2015). doi:<http://dx.doi.org/10.1016/j.strusafe.2014.11.001>, <http://www.sciencedirect.com/science/article/pii/S0167473014001106>. Special Issue: Engineering Analyses with Vague and Imprecise Information.
13. Beer, M., Phoon, K.K., Quek, S.T. (eds.): Special issue: Modeling and analysis of rare and imprecise information. *Struct. Saf.* **32** (2010)
14. Beer, M., Zhang, Y., Quek, S.T., Phoon, K.K.: Reliability analysis with scarce information: Comparing alternative approaches in a geotechnical engineering context. *Struct. Saf.* **41**(6), 1–10 (2013). doi:<http://dx.doi.org/10.1016/j.strusafe.2012.10.003>, <http://www.sciencedirect.com/science/article/pii/S0167473012000689>
15. Benjamin, J., Schuëller, G., Wittmann, F. (eds.): Proceedings of the second international seminar on structural reliability of mechanical components and subassemblies of nuclear power plants, special volume. *J. Nucl. Eng. Des.* **59**, 1–168 (1989)
16. Bratley, P., Fox, B.L.: Algorithm 659: implementing Sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.* **14**(1), 88–100 (1988). doi:<http://doi.acm.org/10.1145/42288.214372>
17. Bucher, C., Pradlwarter, H.J., Schuëller, G.I.: Computational stochastic structural analysis (COSSAN). In: Schuëller, G.I. (ed.) *Structural Dynamics – Recent Advances*, pp. 301–316. Springer, Berlin/Heidelberg (1991)
18. Bucher, C., Pradlwarter, H.J., Schuëller, G.I.: COSSAN – (Computational stochastic structural analysis) – Perspectives of software developments. In: Schuëller, G.I., et al. (ed.) *Proceedings of the 6th International Conference on Structural Safety and Reliability (ICOS-SAR'93)*, pp. 1733–1740. A.A. Balkema Publications, Rotterdam/Innsbruck (1994)
19. Busacca, P.G., Marseguerra, M., Zio, E.: Multiobjective optimization by genetic algorithms: application to safety systems. *Reliab. Eng. Syst. Saf.* **72**(1), 59–74 (2001). <http://www.sciencedirect.com/science/article/B6V4T-42G751J-7/2/f0bf8189c921c1d6029d1f9b56524094>
20. Chiachio, M., Beck, J.L., Chiachio, J., Rus, G.: Approximate Bayesian computation by subset simulation. arXiv preprint, arXiv:14046225 (2014)
21. Ching, J., Chen, Y.: Transitional Markov Chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging. *J. Eng. Mech.* **133**(7), 816–832 (2007). doi:10.1061/(ASCE)0733-9399(2007)133:7(816), <http://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9399%282007%29133%3A7%28816%29>
22. Crémona, C., Lukić, M.: Probability-based assessment and maintenance of welded joints damaged by fatigue. *Nucl. Eng. Des.* **182**(3), 253–266 (1998)
23. Crespo, L.G., Kenny, S.P., Giesy, D.P.: The NASA langley multidisciplinary uncertainty quantification challenge. In: 16th AIAA Non-Deterministic Approaches Conference – AIAA SciTech, American Institute of Aeronautics and Astronautics (2014). doi:10.2514/6.2014-1347, <http://dx.doi.org/10.2514/6.2014-1347>
24. de Angelis, M., Patelli, E., Beer, M.: An efficient strategy for interval computations in risk-based optimization. In: ICOSSAR, 16–20 June 2013. Columbia University, New York (2013)
25. de Angelis, M., Patelli, E., Beer, M.: Advanced line sampling for efficient robust reliability analysis. *Struct. Saf.* **52**, 170–182 (2015). doi:10.1016/j.strusafe.2014.10.002, <http://www.sciencedirect.com/science/article/pii/S0167473014000927>
26. DeFinetti, B.: Theory of Probability: A Critical Introductory Treatment. Wiley, Chichester (1990)
27. Der Kiureghian, A., Dakessian, T.: Multiple design points in first and second-order reliability. *Struct. Saf.* **20**(1), 37–49, doi:10.1016/S0167-4730(97)00026-X, <http://www.sciencedirect.com/science/article/B6V54-3T2H6KD-3/2/241e203d3372ca22a2cc463c44cc98ca> (1998)
28. Ditlevsen, O., Madsen, H.O.: Structural Reliability Methods, Internet edition. Wiley, Chichester (2005)
29. Exler, O., Schittkowski, K.: A trust region SQP algorithm for mixed-integer nonlinear programming. *Optim. Lett.* (2007). doi:10.1007/s11590-006-0026-1
30. Free Software Foundation: Free software foundation, GNU lesser general public license, version 3. <http://www.gnu.org/licenses/lgpl.html> (2007)

31. Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach. Springer, New York/Berlin/Heidelberg. Revised edition 2003, Dover Publications, Mineola/New York (1991)
32. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
33. Goller, B., Pradlwarter, H.J., Schuëller, G.I.: Robust modal updating with insufficient data. *Comput. Methods Appl. Mech. Eng.* **198**(37–40), 3096–3104 (2009). doi:10.1016/j.cma.2009.05.009
34. Harder, R., Desmarais, R.: Interpolation using surface splines. *J. Aircr.* **2**, 189–191 (1972)
35. Hoshiya, M.: Kriging and conditional simulation of gaussian field. *J. Eng. Mech. ASCE* **121**(2), 181–186 (1995)
36. Jensen, H., Catalan, M.: On the effects of non-linear elements in the reliability-based optimal design of stochastic dynamical systems. *Int. J. Nonlinear Mech.* **42**(5), 802–816 (2007)
37. Jensen, H., Valdebenito, M., Schuëller, G.: An efficient reliability-based optimization scheme for uncertain linear systems subject to general gaussian excitation. *Comput. Methods Appl. Mech. Eng.* **198**(1), 72–87 (2008)
38. Kijawatworawet, W.: Reliability of structural systems using adaptive importance directional sampling. PhD thesis, Institute of Engineering Mechanics, Leopold-Franzens University, Innsbruck, EU (1992)
39. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science*, Number 4598, 13 May 1983 220, **4598**, 671–680 (1983). citesearc.ist.psu.edu/kirkpatrick83optimization.html
40. Koutsourelakis, P.S., Pradlwarter, H.J., Schuëller, G.I.: Reliability of structures in high dimensions, part I: algorithms and applications. *Probab. Eng. Mech.* **19**(4), 409–417 (2004). doi:10.1016/j.probengmech.2004.05.001
41. Kucherenko, S., Delpuech, B., Iooss, B., Tarantola, S.: Application of the control variate technique to estimation of total sensitivity indices. *Reliab. Eng. Syst. Saf.* **134**, 251–259 (2015). doi:10.1016/j.ress.2014.07.008
42. Laplace, P.S.: A Philosophical Essay on Probabilities. Dover Publications, New York (1814)
43. Liu, J.: Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics. Springer, New York (2001)
44. Melchers, R.E.: Structural reliability: analysis and prediction. Wiley, Chichester (2002)
45. Melchers, R.E., Ahammed, M.: Gradient estimation for applied Monte Carlo analyses. *Reliab. Eng. Syst. Saf.* **78**(3), 283–288 (2002). <http://www.sciencedirect.com/science/article/B6V4T-475R7RS-8/2/8eaa29f83ddacc51937b7005aed69481>
46. Mitsas, I., Kougioumtzoglou, I., Beer, M., Patelli, E., Mottershead, J.: Robust design optimization of structural systems under evolutionary stochastic seismic excitation. In: Vulnerability, Uncertainty, and Risk, American Society of Civil Engineers, pp. 215–224 (2014). doi:10.1061/9780784413609.022, <http://dx.doi.org/10.1061/9780784413609.022>
47. Molchanov, I.: Theory of Random Sets. Springer, London (2005)
48. Möller, B., Beer, M.: Fuzzy-Randomness – Uncertainty in Civil Engineering and Computational Mechanics. Springer, Berlin/New York (2004)
49. Müller, B., Graf, W., Beer, M.: Fuzzy structural analysis using alpha-level optimization. *Comput. Mech.* **26**, 547–565 (2000)
50. NASA Standard for Models and Simulations: Tech. Rep. NASA-STD-7009, National Aeronautics and Space Administration (NASA) (2013)
51. Nelder, J., Mead, R.: A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965)
52. Nissen, S.: Implementation of a fast artificial neural network library (fann). Tech. rep., Department of Computer Science University of Copenhagen (DIKU), <http://fann.sf.net> (2003)
53. Olsson, A., Sandberg, G., Dahlblom, O.: On Latin hypercube sampling for structural reliability analysis. *Struct. Saf.* **25**, 47–68(22) (2003). doi:10.1016/S0167-4730(02)00039-5, <http://www.ingentaconnect.com/content/els/01674730/2003/00000025/00000001/art00039>

54. Panayirci, H.M.: Efficient solution for Galerkin based polynomial chaos expansion systems. *Adv. Eng. Softw.* **41**(412), 1277–1286 (2010). doi:10.1016/j.advengsoft.2010.09.004
55. Paris, P., Erdogan, F.: A critical analysis of crack propagation laws. *J. Basic Eng. Trans. ASME* **85**, 528–534 (1963)
56. Patelli, E., Au, I.: Efficient Monte Carlo algorithm for rare failure event simulation. In: 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12), Vancouver, 12–15 July 2015, <http://hdl.handle.net/2429/53247> (2015)
57. Patelli, E., Broggi, M.: On general purpose software for the efficient uncertainty management of large finite element models. In: NAFEMS World Congress, 9–12 June 2013, Salzburg, NAFEMS, http://academia.edu/attachments/31544367/download_file (2013)
58. Patelli, E., de Angelis, M.: Line sampling approach for extreme case analysis in presence of aleatory and epistemic uncertainties. In: European Safety and Reliability Conference – ESREL – 7–10 Sept 2015. CRC Press/Balkema (2015)
59. Patelli, E., Pradlwarter, H.: Monte Carlo gradient estimation in high dimensions. *Int. J. Numer. Methods Eng.* **81**(2), 172–188 (2010). doi:10.1002/nme.2687
60. Patelli, E., Schuëller, G.I.: Computational optimization strategies for the simulation of random media and components. *Comput. Optim. Appl.* 1–29 (2012). doi:10.1007/s10589-012-9463-1, <http://dx.medra.org/10.1007/s10589-012-9463-1>
61. Patelli, E., Pradlwarter, H.J., Schuëller, G.I.: Global sensitivity of structural variability by random sampling. *Comput. Phys. Commun.* **181**, 2072–2081 (2010). doi:10.1016/j.cpc.2010.08.007
62. Patelli, E., Pradlwarter, H., Schuëller, G.: On multinormal integrals by importance sampling for parallel system reliability. *Struct. Saf.* **33**, 1–7 (2011). doi:10.1016/j.strusafe.2010.04.002
63. Patelli, E., Pradlwarter, H.J., Schuëller, G.I.: On multinormal integrals by importance sampling for parallel system reliability. *Struct. Saf.* **33**, 1–7 (2011). doi:10.1016/j.strusafe.2010.04.002
64. Patelli, E., Valdebenito, M.A., Schuëller, G.I.: General purpose stochastic analysis software for optimal maintenance scheduling: application to a fatigue-prone structural component. *Int. J. Reliab. Saf.* **5**, 211–228 (2011). Special Issue on: “Robust Design – Coping with Hazards Risk and Uncertainty”
65. Patelli, E., Panayirci, H.M., Broggi, M., Goller, B., Pradlwarter, P.B.H.J., Schuëller, G.I.: General purpose software for efficient uncertainty management of large finite element models. *Finite Elem. Anal. Des.* **51**, 31–48 (2012). doi:10.1016/j.finel.2011.11.003, <http://dx.medra.org/10.1016/j.finel.2011.11.003>
66. Patelli, E., Alvarez, D.A., Broggi, M., de Angelis, M.: An integrated and efficient numerical framework for uncertainty quantification: application to the NASA Langley multidisciplinary uncertainty quantification challenge. In: 16th AIAA Non-Deterministic Approaches Conference (SciTech 2014), American Institute of Aeronautics and Astronautics, AIAA SciTech (2014). doi:10.2514/6.2014-1501
67. Patelli, E., Broggi, M., de Angelis, M., Beer, M.: Opencossan: an efficient open tool for dealing with epistemic and aleatory uncertainties. In: Vulnerability, Uncertainty, and Risk, American Society of Civil Engineers, pp. 2564–2573 (2014). doi:10.1061/9780784413609.258, <http://dx.doi.org/10.1061/9780784413609.258>
68. Patelli, E., Alvarez, D.A., Broggi, M., de Angelis, M.: Uncertainty management in multidisciplinary design of critical safety systems. *J. Aerosp. Inf. Syst.* **12**, 140–169 (2015). doi:10.2514/1.I010273
69. Pedroni, N., Zio, E., Ferrario, E., Pasanisi, A., Couplet, M.: Propagation of aleatory and epistemic uncertainties in the model for the design of a food protection dike. In: PSAM 11 & ESREL, Jun 2012, Helsinki, pp. 1–10 (2012)
70. Powell, M.: Direct search algorithms for optimization calculations. *Acta Numer.* **7**, 287–336 (1998)
71. Powell, M.J.D.: The BOBYQA algorithm for bound constrained optimization without derivatives. Tech. rep., Department of Applied Mathematics and Theoretical Physics, Cambridge, <http://fann.sf.net> (2009)

72. Pradlwarter, H., Schuëller, G.: Reliability assessment of uncertain linear systems in structural dynamics. In: Belyaev, A.K., Langley, R.S. (eds.) IUTAM Symposium on the Vibration Analysis of Structures with Uncertainties, Saint Petersburg, pp. 363–378 (2011)
73. Romero, V., Mullins, J., Swiler, L., Urbina, A.: A comparison of methods for representing and aggregating uncertainties involving sparsely sampled random variables – more results. *SAE Int. J. Mater. Manuf.* **6**(3) (2013). <http://www.scopus.com/inward/record.url?eid=2-s2.0-84876425264&partnerID=40&md5=72ea116c4e8d25c856e55d3d07af890>
74. Roux, W.J., Stander, N., Haftka, R.T.: Response surface approximation for structural optimization. *Int. J. Numer. Methods Eng.* **42**, 517–534 (1998)
75. Rubinstein, R.: Simulation and the Monte Carlo Method. John Wiley & Sons, New York/Chichester/Brisbane/Toronto (1981)
76. Saltelli, A., Bolado, R.: An alternative way to compute fourier amplitude sensitivity test (fast). *Comput. Stat. Data Anal.* **26**(4), 445–460 (1998). doi:10.1016/S0167-9473(97)00043-1, <http://www.sciencedirect.com/science/article/B6V8V-3SX829Y-5/2-1147936f52dc9461d1f69aa319bb117>
77. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Salsana, M., Tarantola, S.: Global Sensitivity Analysis: The Primer. Wiley, Chichester (2008)
78. Schenk, C.A., Schuëller, G.I.: Uncertainty Assessment of Large Finite Element Systems, Lecture Notes in Applied and Computational Mechanics, vol 24. Springer, Berlin/Heidelberg/New York (2005). <http://www.springer.com/materials/mechanics/book/978-3-540-25343-3>, ISBN:978-3-540-25343-3
79. Schuëller, G.: Efficient Monte Carlo simulation procedures in structural uncertainty and reliability analysis – recent advances. *J. Struct. Eng. Mech.* **32**(1), 1–20 (2009)
80. Schuëller, G.I.: On procedures for reliability assessment of mechanical systems and structures. *J. Struct. Eng. Mech.* **25**(3), 275–289 (2007)
81. Schuëller, G.I., Pradlwarter, H.J.: Computational stochastic structural analysis(COSSAN) – a software tool. *Struct. Saf.* **28**(1–2), 68–82 (2006). doi:10.1016/j.strusafe.2005.03.005
82. Schuëller, G.I., Pradlwarter, H.J.: Uncertainty analysis of complex structural systems. *Int. J. Numer. Methods Eng.* **80**(6–7), 881–913 (2009). doi:10.1002/nme.2549
83. Schuëller, G.I., Stix, R.: A critical appraisal of methods to determine failure probabilities. *J. Struct. Saf.* **4**(4), 293–309 (1987)
84. Schuëller, G.I. (ed.): GI Uncertainties in structural mechanics and analysis – computational methods. *Comput. Struct. – Special Issue* **83**(14), 1031–1149 (2005). doi:10.1016/j.compstruc.2005.01.004
85. Schuëller, G.I. (ed.): GI Structural reliability software. *Struct. Saf. – Special Issue* **28**(1–2), 1–216 (2006). doi:10.1016/j.strusafe.2005.03.001
86. Schuëller, G., Jensen, H.: Computational methods in optimization considering uncertainties – an overview. *Comput. Methods Appl. Mech. Eng.* **198**(1), 2–13 (2008)
87. Sobol', I.: Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.* **1**(4), 407–414 (1993)
88. Sobol', I.: Global sensitivity indices for nonlinear mathematical modes and their Monte Carlo estimates. *Math. Comput. Simul.* **55**, 217–280 (2001)
89. Sudret, B.: Meta-models for structural reliability and uncertainty quantification. ArXiv e-prints 1203.2062 (2012)
90. Sudret, B., Der Kiureghian, A.: Stochastic finite element methods and reliability a state-of-the-art report. Tech. rep., Department of Civil and Environmental Engineering, University of California, Berkeley (2000)
91. Thomas, B.: Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York (1996). doi:10.19-509971-0
92. Valdebenito, M.: Reliability-based optimization: Efficient strategies for high dimensional reliability problems. PhD thesis, Institute of Engineering Mechanics, University of Innsbruck, Innsbruck (2010)

93. Valdebenito, M., Schuëller, G.: Design of maintenance schedules for fatigue-prone metallic components using reliability-based optimization. *Comput. Methods Appl. Mech. Eng.* **199**, 2305–2318 (2010)
94. Valdebenito, M., Patelli, E., Schuëller, G.: A general purpose software for reliability-based optimal design. In: Muhanna, M.B.R., Mullen, R. (eds.) 4th International Workshop on Reliable Engineering Computing: Robust Design – Coping with Hazards, Risk and Uncertainty, Research Publishing Services, Singapore, pp. 3–22 (2010). doi:10.3850/978-981-08-5118-7_plenary-1
95. Valdebenito, M., Pradlwarter, H., Schuëller, G.: The role of the design point for calculating failure probabilities in view of dimensionality and structural non linearities. *Struct. Saf.* **32**(2), 101–111 (2010). doi:10.1016/j.strusafe.2009.08.004
96. Vanmarcke, E.: Random fields: analysis and synthesis. Published by MIT Press, Cambridge, MA (1983); Web Edition by Rare Book Services, Princeton University. Princeton, Cambridge, MA (1998)
97. Wang, P., Lu, Z., Tang, Z.: A derivative based sensitivity measure of failure probability in the presence of epistemic and aleatory uncertainties. *Comput. & Math. Appl.* **65**(1), 89–101 (2013). doi:10.1016/j.camwa.2012.08.017, <http://www.sciencedirect.com/science/article/pii/S0898122112006438>
98. Youssef, H., Sait, S.M., Adiche, H.: Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Eng. Appl. Artif. Intell.* **14**(2), 167–181 (2001). doi:10.1016/S0952-1976(00)00065-8, <http://www.sciencedirect.com/science/article/B6V2M-42JRD52-6/2/a02150bf476eef0d9f64652698ddea7>
99. Zhang, H., Mullen, R.L., Muhanna, R.L.: Interval Monte Carlo methods for structural reliability. *Struct. Saf.* **32**(3), 183–190 (2010)
100. Zhang, M., Beer, M., Quek, S.T., Choo, Y.S.: Comparison of uncertainty models in reliability analysis of offshore structures under marine corrosion. *Struct. Saf.* **32**(6), 425–432 (2010)

Stefano Tarantola and William Becker

Abstract

SIMLAB 4.0 is a comprehensive stand-alone software package for performing global sensitivity analysis. Several sampling strategies and sensitivity measures are available. SIMLAB includes the most recent variance-based formulas for first-order and total-order sensitivity indices, graphical methods, as well as more classical methods. The peculiarity of SIMLAB, in contrast to previous versions of the package, is the possibility to run sequential sensitivity analysis, which allows updating the sensitivity measures at each run, or group of runs, of the model. The techniques can be accessed through the R environment as well as through a graphical user interface. The user can also add new techniques by simply adding the corresponding R code to the core layer. SIMLAB can be downloaded for free from the Joint Research Centre's website.

Keywords

Sensitivity analysis • Software • Monte Carlo • Uncertainty • Computer models

Contents

1	Introduction	1980
2	Installation	1981
3	Data Workflow	1982
4	Overview of Using SIMLAB	1983

S. Tarantola (✉)

Statistical Indicators for Policy Assessment, Joint Research Centre of the European Commission, Ispra (VA), Italy

Institute for Energy and Transport, European Commission Joint Research Centre, Ispra (VA), Italy

e-mail: stefano.tarantola@jrc.ec.europa.eu

W. Becker

European Commission Joint Research Centre, Ispra (VA), Italy

e-mail: william.becker@jrc.ec.europa.eu

5 Available Techniques	1984
5.1 Sampling	1984
5.2 Model Execution	1990
5.3 Post-processing	1994
6 Extending SIMLAB with R	1996
7 Conclusions	1998
References	1998

1 Introduction

SIMLAB 4.0 is a newly implemented version of a software framework for uncertainty and sensitivity analysis. SIMLAB is the property of the Joint Research Centre (JRC) of the European Commission, which has financed its design and development since its first version, made available back in 1999. With this product, the JRC aims to employ up-to-date tools and practices of global sensitivity analysis in order to disseminate the culture of sensitivity analysis to an ever-increasing number of customers. In agreement with the EC dissemination policy, SIMLAB is publicly available for use by any person, company, or organization that downloads, installs, and uses the software, according to an end user software license agreement.

SIMLAB is available on the website of the *Econometrics and Applied Statistics* research group of the European Commission's Joint Research Centre [3].

SIMLAB contains a set of techniques to execute global sensitivity analysis (GSA), primarily by Monte Carlo and sampling-based methods (as opposed to emulator/metamodel-based methods). No local sensitivity analysis tools are present in the software. The conceptual framework of GSA and the techniques available to perform GSA are described in Sobol' (variance-based) sensitivity indices: theory and estimation algorithms. In summary, GSA is based on performing multiple model evaluations with a probabilistically-selected model input and then using the results of these evaluations to quantify the relative importance of the inputs in determining the uncertainty of the model output.

The tool, running in a 64-bit environment under Windows, is delivered as a self-installing setup. It provides a set of transparent, well-commented, and self-maintainable functions, coded in the R environment, that implement GSA techniques. The user can work both in the R environment by calling the R functions and, on the pre-compiled functions, through a graphical user interface (GUI), developed in C# for the .NET framework, which facilitates the use of the application and allows to visually present the results of the sensitivity analysis.

The R package acts as the core layer of the application and contains all the algorithms and methods of global sensitivity analysis, with source codes available to the user. SIMLAB can easily be extended with new GSA techniques by adding the corresponding R code to the core layer. Modifications and maintenance of the core algorithms can be carried out by the user.

The GUI is built on top of a control layer, developed in C++, which links the core layer in R with the GUI itself. The control layer is responsible for the correct

management of the calls to R, given the instructions provided by the user through the GUI. The control layer also handles warning and error messages.

The system requires the installation of R. SIMLAB uses two R packages through which R and SIMLAB communicate: *Rcpp* and *RInside*. Other R packages are used for statistical operations: *stats*, *pspearman*, *sensitivity*, *lhs*, *randtoolbox*, and *rngWELL*. Those packages are automatically downloaded from the Comprehensive R Archive Network (<https://cran.r-project.org/>) during SIMLAB installation.

SIMLAB gathers a variety of routines, written by different authors in Matlab or R, that implement various GSA techniques. These routines are heterogeneously programmed and are harmonized and integrated in SIMLAB.

SIMLAB offers a test suite of three commonly used analytical functions that the user can employ to test the methods implemented in SIMLAB. These functions are described in this chapter.

With respect to previous versions, SIMLAB 4.0 offers a new functionality that allows the user to perform sequential sensitivity analysis. The sequential process is entirely managed by the control layer. Specifically, the user can run a number of iterations of the sequence composed by sample generation, model execution, and sensitivity estimation, thus obtaining continuous updates and convergence monitoring of the sensitivity analysis results in real time. In the sequential sensitivity analysis, the iteration is repeated until the user is satisfied with the results of the sensitivity analysis. The user does not need to specify *ex ante* how many sample points to use in the analysis. A visualization tool assists the user to monitoring the convergence of the sensitivity results. This approach has the advantage of stopping the analysis when the user is satisfied with the level of precision of the GSA results, avoiding useless, and often expensive, extra model executions.

2 Installation

Requirements to install SIMLAB:

- A PC running Windows
- An internet connection
- R already installed

SIMLAB is currently only available for Windows. SIMLAB is written in C++, but uses packages from R. Therefore in order to use SIMLAB, the user must have both SIMLAB and R installed. It is recommended to install R prior to installing SIMLAB. R can be downloaded from <https://cran.r-project.org/bin/windows/base/>. Throughout the installation process, an internet connection is required.

To install SIMLAB, first unzip the Simlab_v4.rar file. In the folder SIMLAB Installation Files, run the setup.exe file. At this point, you may be asked to install Visual C++ Runtime Libraries. Since SIMLAB is dependent on these components, you must choose “Install” to continue. The SIMLAB Setup Wizard will now open automatically. Follow the instructions of the Wizard, including choosing a suitable

directory for the installation. You will also be asked to nominate a folder which will be used for automatically depositing files to exchange information with R. Next, the Wizard will download and install the required R packages onto your R installation. This can also be done manually if desired. The following R packages are required to run SIMLAB:

- stats
- pspearman
- sensitivity
- lhs
- randtoolbox
- e1071
- mc2d
- rngWELL

Additional to these packages, SIMLAB installs its own package “SimLab4R” on R.

After installing the packages, automatically or otherwise, SIMLAB is installed and ready to run on your computer.

3 Data Workflow

Figure 57.1 below shows how data are processed by SIMLAB. The files for data exchange are represented by green boxes and contain ASCII text with an easy to understand format. These files contain information on the input distributions and their correlation structure, the generated sample (sample file), and the results of the model execution (simulated output). The setup of the input variables (i.e., the choice of the probability density functions and their parameters and the correlation matrix for the statistical characterization of input uncertainty) and the computer code implementing the simulation model are provided by the user (the blue boxes). The red boxes are the tools offered by SIMLAB: the sample generation step, the

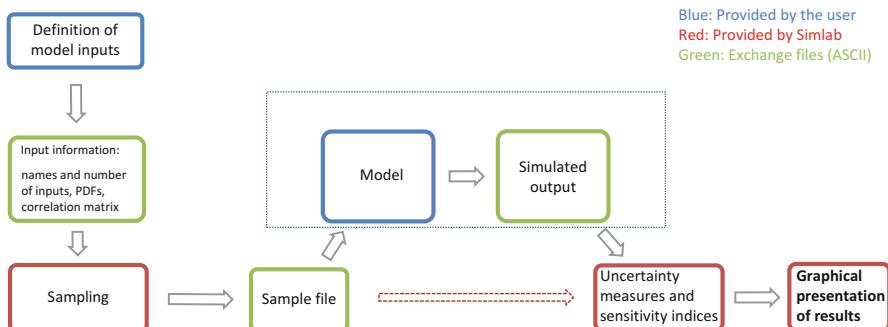


Fig. 57.1 SIMLAB data workflow (Courtesy of Federico Ferretti, Joint Research Centre, European Commission)

uncertainty analysis, and the GSA technique to compute the sensitivity indices. This latter is usually intimately related to the type of sampling used. SIMLAB also offers graphical tools to visualize the results of the GSA.

The core layer in R contains different functions that can be considered as building blocks for the GSA. Those functions allow the user to select among different types of sampling methods, to impose a desired correlation between inputs, to generate random samples from a set of assigned distributions, to evaluate the GSA measures according to the method selected, and to choose the stopping criterion for the GSA evaluation process. For example, the R command used to generate a log-uniformly distributed sample of size N with lower bound a and upper bound b is:

```
    sam <- random(N, 1)
    qloguniform(sam, a, b)
```

The GUI facilitates the interaction of the user with the core layer. The GUI allows the user to save the working configuration for subsequent analyses to avoid the need to rerun the entire process. In particular, four working configurations can be saved: the definition of the model inputs (their number, their names, their probability distributions, and their correlation), the generated sample of inputs (with information on the type of sampling strategy used), the generated model output, and the evaluated sensitivity indices. A specific GUI functionality is dedicated to monitor the convergence of the sensitivity indices through a visualization tool.

The next version of SIMLAB will foresee the possibility to pause the sequential analysis and then restart it at a later stage.

4 Overview of Using SIMLAB

The first step of a GSA is to select ranges and probability density functions for the model inputs that are the object of the analysis. This is accomplished by the “Factors definition” function of the GUI. Here all the properties of the inputs are specified. It is possible to add new inputs, modifying or removing existing ones. The configuration of the model input can be saved to file and an existing configuration can be loaded into SIMLAB. Once the inputs have been defined it is also possible to specify a correlation matrix. See the next section for a summary of the probability density functions handled by SIMLAB.

The second step of the GSA is the sample generation. Four different sampling methods can be set: simple random, quasi-random, Latin hypercube, and the Sobol’ design. This latter is required if one is interested to compute Sobol’ sensitivity indices. As the GSA is executed sequentially in blocks of samples, the user is required to insert three values: the size of the first subsample, the size of the subsequent subsamples, and the maximum allowed sample size which, when reached, would terminate the GSA if not previously terminated. The generated sample can be saved to a file in ASCII format or as a csv file. The entire configuration composed by information on the input and generated sample can also be saved. The generated sample can be visualized as multiple histograms (one

per input), cumulative histograms, 2D scatterplots (between pairs of inputs), and cobwebs (all inputs together).

In the third step “Output variables selection,” model output variable names are assigned. Multiple output variables and time-dependent variables can be handled.

Further, the user has to define a threshold value for the convergence criterion. The convergence criterion is based on the calculation of the absolute difference between each sensitivity index at step $n - 1$ and step n and on the calculation of the maximum of these absolute differences across all inputs and all outputs. The process will stop when this maximum difference gets smaller than the threshold value chosen by the user. Other convergence criteria will be added in the next release of SIMLAB.

The “model execution” step allows the user to link a model to SIMLAB. The user has to specify the name of an executable (the code of the model) that returns the model output variables with the names specified in the “output variables selection step.” SIMLAB foresees the possibility for the user to add a set of arguments needed by the model when launching the executable. The user has also to specify the name of the sample file from where the sample generated at the i th iteration will be read and the name of the output file where the results of the model output at the i th iteration will be saved. This latter specification is important in order not to lose the results of the model runs, which might be computationally intensive. The executable (the code of the model) has to foresee the reading from (writing to) the sample (output) file specified by SIMLAB. As such, SIMLAB will be able to run sequentially, generating a first subset of samples, running the model on this subset, and producing a subset of model outputs, which are saved to the output file and are used to obtain estimates of sensitivity indices of various nature.

The final two steps are “sensitivity evaluation” and “uncertainty evaluation,” the first being much more developed in SIMLAB. See the next section for an overview of the sensitivity analysis methods and the options offered by the uncertainty evaluation.

5 Available Techniques

Referring back to Fig. 57.1, the workflow of a sensitivity analysis consists of a number of steps. Fundamentally, SIMLAB performs two basic tasks of this workflow: generating the sample design that is used to specify the input values to each model run and post-processing the results of the model runs specified by the sample. For both of these operations, SIMLAB has a number of features which are outlined in this section.

5.1 Sampling

The selection of the sample design for a sensitivity analysis must be given some prior consideration, because the type of design dictates which measures of sensitivity can be estimated. The sensitivity measures that are associated with each

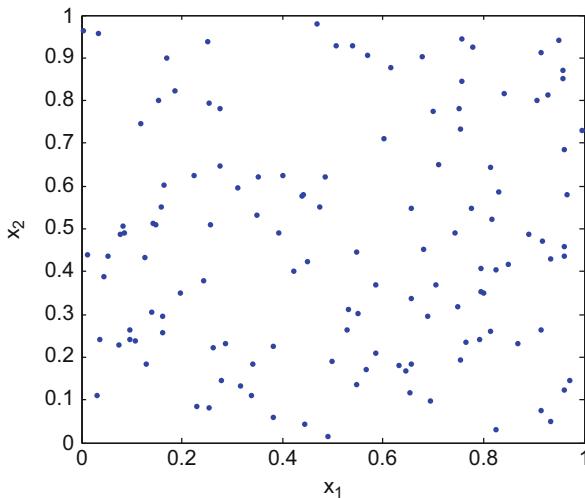


Fig. 57.2 Simple random sampling in two dimensions with 128 points

design are described in the following section. The sample designs available in SIMLAB are as follows:

Simple random: this is simply a sequence of (pseudo-)random points taken from the input space—see Fig. 57.2. Random sampling has the advantage that sample replications can be taken, in order to estimate the error due to the sample size; however, random sampling is characterized by clustering of points which can reduce the efficiency of the design in terms of convergence in Monte Carlo applications.

Latin hypercube: this is the well-known sampling design proposed by [4], which has the property that sample points are quite evenly distributed with respect to each dimension – see Fig. 57.3. For a sample of n points, the range of each variable is divided into n equally spaced intervals. A sample point is taken by randomly selecting an interval from each dimension and then taking a random sample point from within the resulting hypercube. Intervals are sampled without replacement to ensure even distribution of points with respect to each variable. A drawback of Latin hypercube sampling is that an existing design cannot be extended to higher sample sizes without repositioning all sample points.

Quasi-random: a so-called *low-discrepancy sequence* which has many of the properties of random numbers but ensures that points are well spaced in order to improve the rate of convergence of sensitivity estimates – see Fig. 57.4. Typically, quasi-random numbers will result in more accurate estimates of sensitivity at a given sample size. SIMLAB uses the Sobol' (LP- τ) sequence [9], which also has the advantage that points can be added sequentially without restructuring the whole design. Note that the LP- τ sequence has particularly low discrepancy at sample values that are positive integer powers of 2, i.e., $n = 2^i$, $i = 1, 2, \dots$

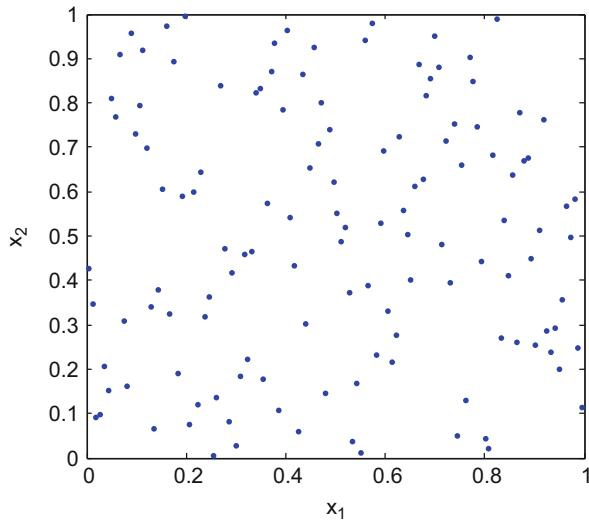


Fig. 57.3 Latin hypercube sampling in two dimensions with 128 points

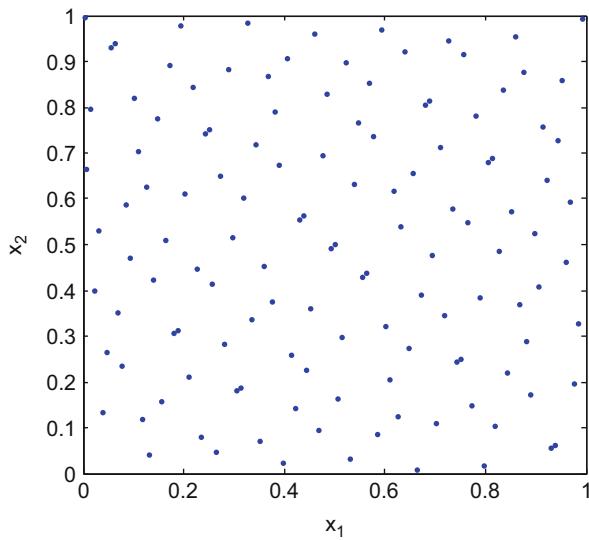
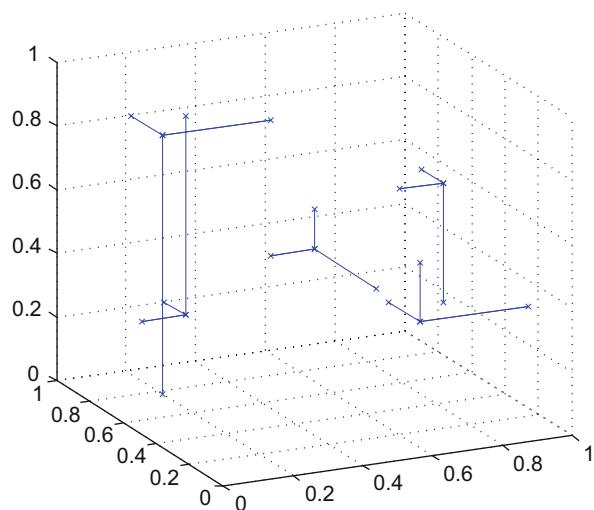


Fig. 57.4 Sobol' LP- τ sampling in two dimensions with 128 points

Sobol' design: this is a structured design based on quasi-random sampling, structured in such a way as to allow estimation of variance-based sensitivity indices (first order and total order), via Monte Carlo integration – see Chapter 5 – and [10]. It is sometimes called “radial” sampling, since the design consists of a number n_r of smaller designs that each have $k + 1$ points (where k is

Fig. 57.5 Sobol' radial design in three dimensions



the number of inputs of the model/function) that “radiate” from a single starting point – see Fig. 57.5. In fact, each of these smaller designs is equivalent to a set of k one-at-a-time designs, where each variable is perturbed individually. The total sample size (number of model runs) is therefore $n = n_r(k + 1)$.

FAST and Extended FAST design: these designs are also structured in such a way as to allow estimation of variance-based sensitivity indices: first-order indices for FAST [2], and both first-order and total-order indices for extended FAST [7]. FAST and extended FAST methods are not implemented sequentially in SIMLAB.

The sampling in all of the above designs can be performed with respect to the following distributions: discrete, uniform, piecewise uniform, log uniform, piecewise log uniform, normal, log normal, triangular, exponential, beta, gamma, and Weibull.

Before generating samples, the first step is to define the probability distributions of the input factors of the model. SIMLAB’s navigation pane is organized in steps which follow the logical workflow of a sensitivity analysis. Correspondingly, the first step is to go to the “Factors definition” pane (Fig. 57.6). This gives an overview of each input factor. To start defining the distributions of input factors, click “Add new factor” and click “Edit” on the new factor that appears in the table. This opens a window as shown in Fig. 57.7. From here, the distribution type and parameters can be set, with information on the distribution given at the bottom of the window. Click “Save params” to save the distribution parameters to SIMLAB’s database. This process can be repeated for each input factor. Notice that every time a new factor is added, it appears as a subheading in the Factors definition heading of the navigation pane, allowing it to be edited at any time. A summary of all factors can be found by returning to the “Factors definition” window – see Fig. 57.8. Here the

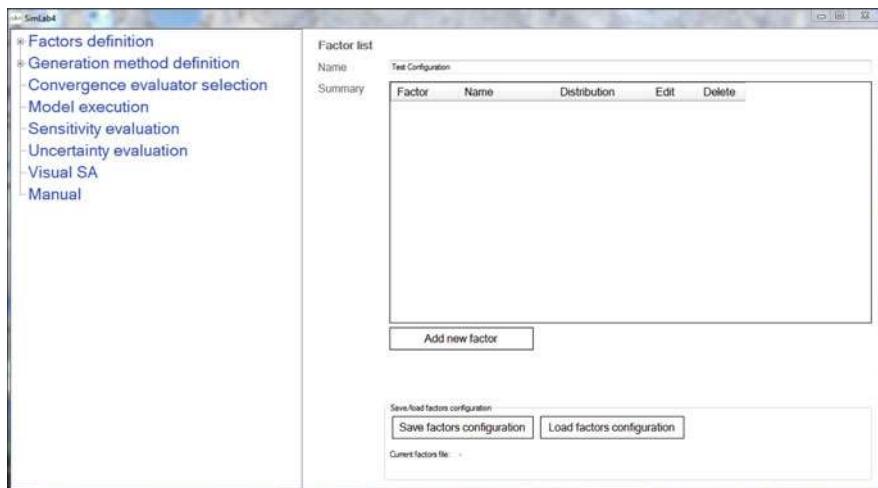


Fig. 57.6 The “Factors definition” pane

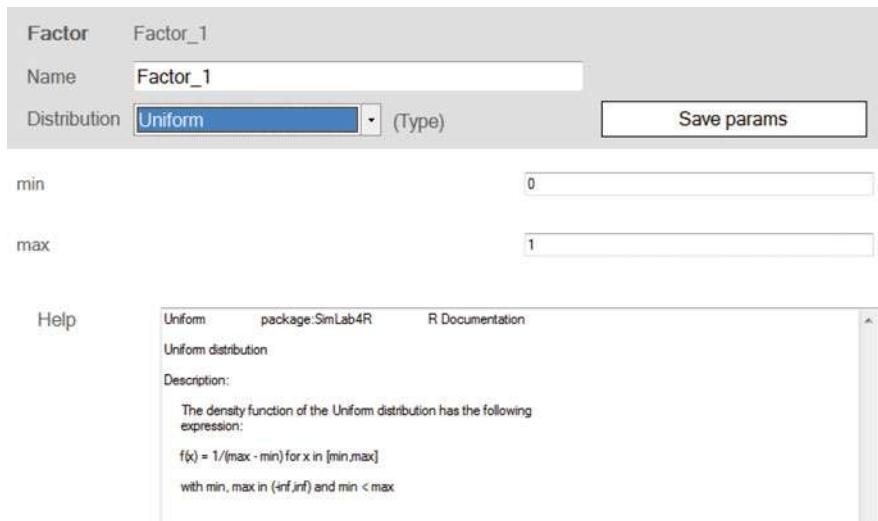


Fig. 57.7 Input distribution editing pane

configuration of all factors can also be saved for use in other sessions or projects and similarly loaded from earlier work.

To actually generate a sample, go to the “Method” subheading of the “Generation method definition” heading in the navigation pane. Here, the options for the sample generation can be set (see Fig. 57.9). The user must choose the method of sample generation using the “Method” and “Aux function” drop-down menus. Below, parameters can be set which control the sample size and optionally the step size

Factor list

Name	Test Configuration				
Summary	Factor	Name	Distribution	Edit	Delete
	Factor_1	Factor_1	Uniform	Edit	Delete
	Factor_2	Factor_2	Normal	Edit	Delete
	Factor_3	Factor_3	Uniform	Edit	Delete

Add new factor

Save/load factors configuration

Save factors configuration Load factors configuration

Current factors file: -

Fig. 57.8 Input parameter summary pane

in the sample generation, if a sequential sampling strategy is to be used. To perform an ordinary batch-mode sensitivity analysis (as opposed to sequential), set the “first subset size” option to the same number as the “maximum sample set size,” i.e., the total number of sample points. The “next subsets size” option should be set to zero. In the case of sequential sensitivity analysis, these settings can be used to control the step sizes for each iteration of sensitivity analysis.

Returning to the “Generation method definition” page, a summary of the variables and the sampling method is now displayed – see Fig. 57.10. To generate the sample, click “Generate and save whole sample.” At this point, the sample can be viewed in a number of ways by clicking the “View generated sample” button. A window will appear which allows the selection of factor, which can be selected by moving them to the “selected factors list” using the buttons – see Fig. 57.11. Histograms, cobwebs, and scatterplots can be generated depending on the number of variables selected. For example, with two factors selected, a scatterplot can be generated (Fig. 57.12), and with two or more variables, a cobweb plot can be created (Fig. 57.13). For the latter, values can be normalized or not by checking the box in the bottom right corner.

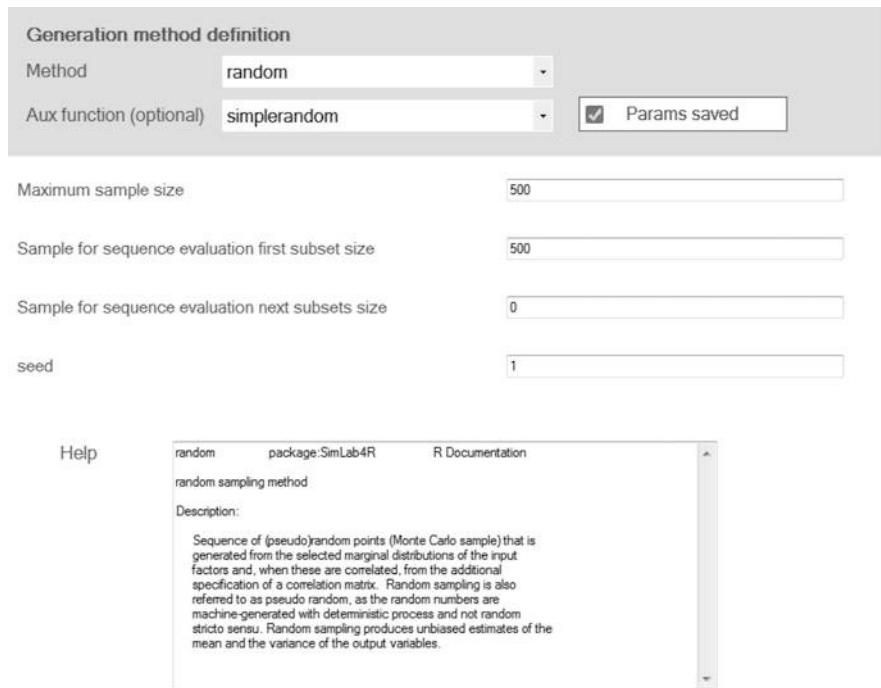


Fig. 57.9 Sample method settings

5.2 Model Execution

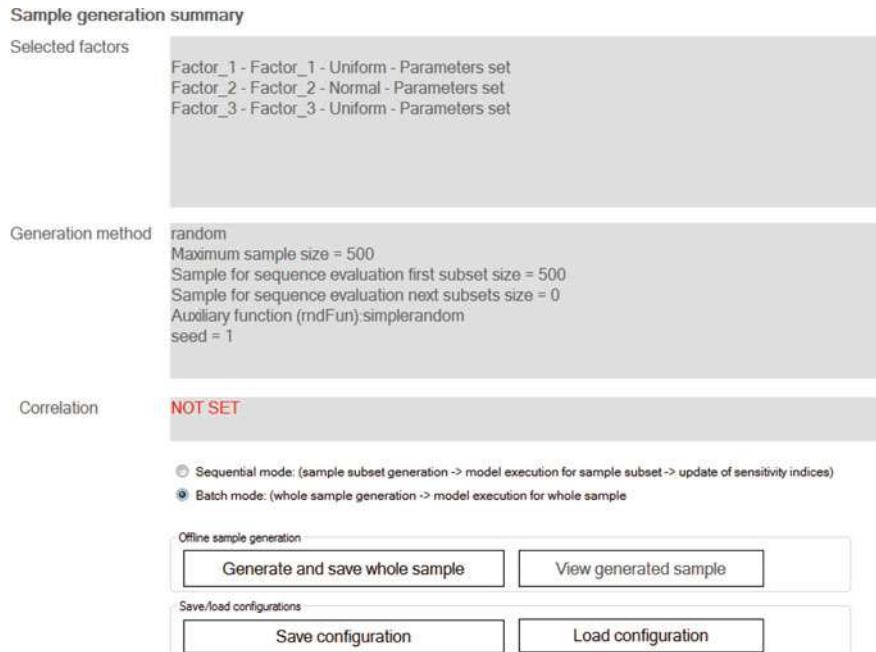
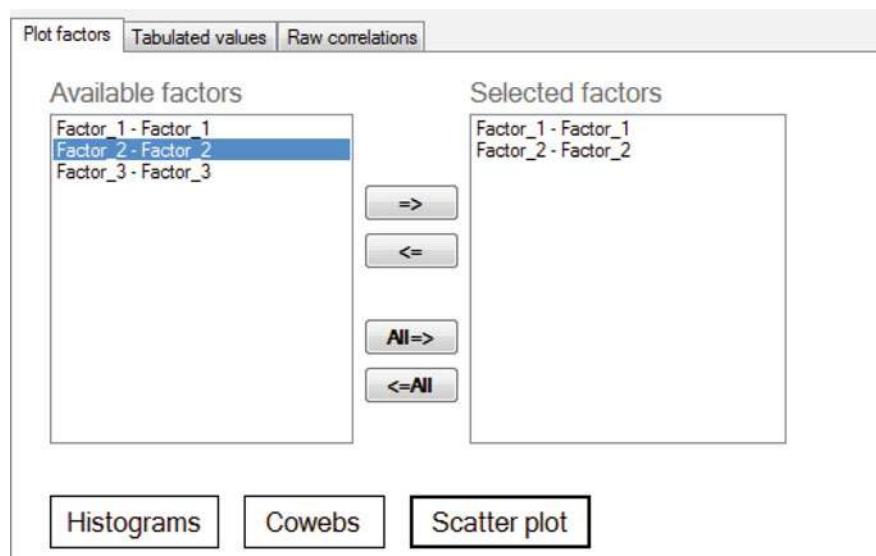
The “Model Execution” pane is presented in Fig. 57.14. SIMLAB is offered with three simple test models that can be found in the installation files. They allow the user to play and learn how to use the package. The first test model is the linear function of the form:

$$Y = \sum_{j=1}^d a_j X_j$$

The a_j coefficients are selected by the user to decide upon the relative importance of the inputs. The marginal distributions of the X_j and their correlation matrix are selected by the user.

The second test model is the so-called Sobol’ g -function, a classical test function for which an analytical expression of the Sobol’ indices is available. $Y = f_1(X_1) \times \dots \times f_k(X_k)$ with $(X_1, \dots, X_k) \sim \mathcal{U}([0, 1]^k)$ and

$$f_j(X_j) = \frac{|4X_j - 2| + a_j}{1 + a_j}, \quad a_j \geq 0, \quad j = 1, \dots, k.$$

**Fig. 57.10** Summary of sample and sample generation**Fig. 57.11** Displaying sample plots

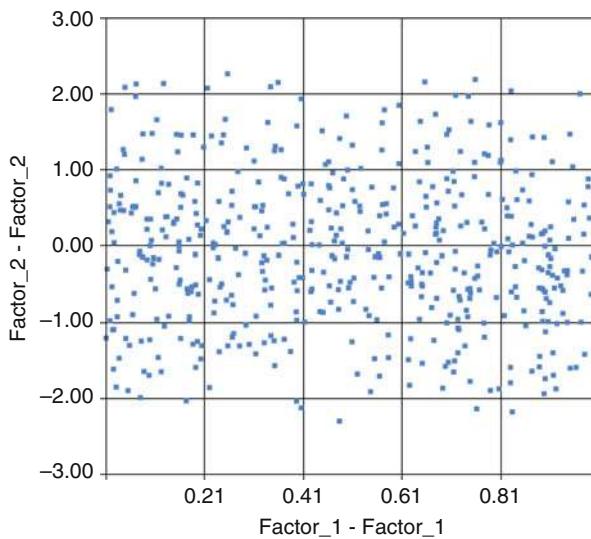


Fig. 57.12 Scatterplot of one input variable against another

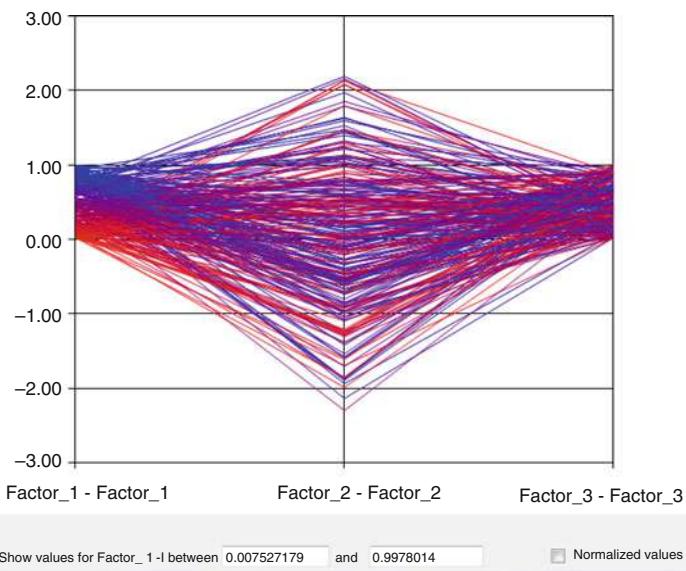


Fig. 57.13 Cobweb plot of three input variables without normalization

The a_j coefficients can be set by the user to decide upon the relative importance of the inputs. $a_j = 0$ corresponds to a very important input. As a_j increases, the relative importance of the input decreases.

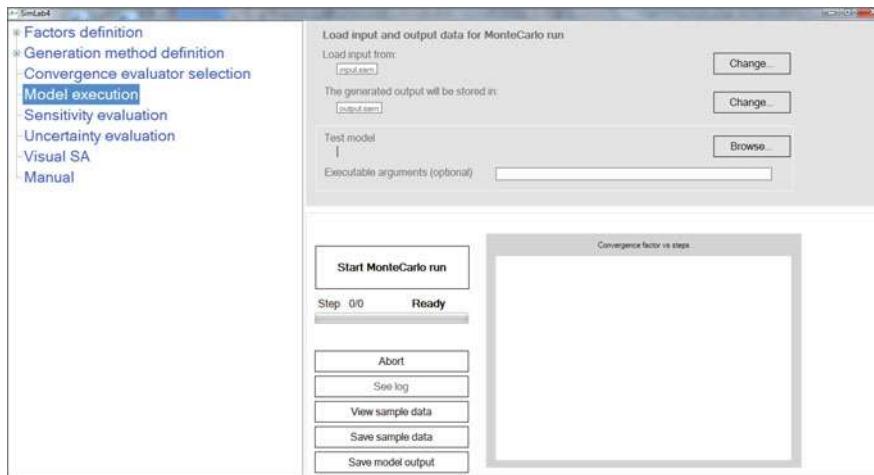


Fig. 57.14 The aspect of the “Model Execution” pane



Fig. 57.15 How to set up the “Model Execution” pane for the linear model

The third test model is the so-called Ishigami function for which analytical Sobol’ indices are also available. The Ishigami function has the form: $Y = \sin(X_1) + 7 \sin^2(X_2) + 0.1 X_3^4 \sin(X_1)$ with the three inputs uniformly distributed in the range $(-\pi, \pi)$.

In the “Model Execution” pane, the test model is selected by clicking on “Browse” and choosing the executable file that implements the model. The executable file must also contain the instructions to read from the sample file and to write to the output file. For example, the executable file “LinearModel.exe” requires some arguments to be supplied by the user. These are the name of the input file, the name of the output file, and the coefficients a_i . If the user wants to run the sensitivity analysis of the linear model with coefficients $a_1 = 2, a_2 = 7, a_3 = 5$, he has to prepare the “Model Execution” pane as depicted in Fig. 57.15.

During the execution of a sequential sensitivity analysis, the sensitivity estimates obtained at a given iteration are compared with those obtained at the previous

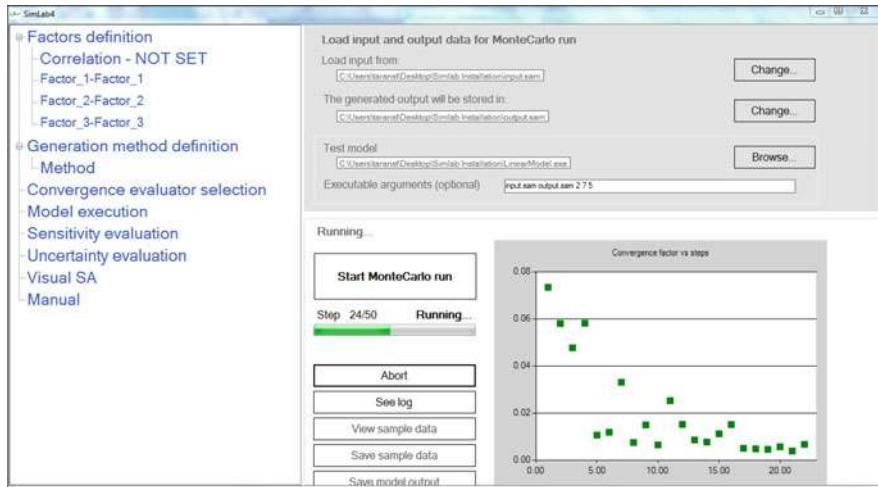


Fig. 57.16 Sensitivity analysis in progress

iteration. The convergence criterion is applied and, if convergence is not reached yet, the analysis proceeds. During this phase, the “Model Execution” pane looks as in Fig. 57.16.

5.3 Post-processing

The model must be run a number of times using one of the designs described in the previous section: this will return a vector of model output values. A number of measures of sensitivity can be estimated by SIMLAB based on the resulting model output values. Which measures can be estimated is dependent on the type of sampling design – see Fig. 57.17 for a summary.

The methods supported by SIMLAB that are compatible with simple random, Latin hypercube, and quasi-random sampling are as follows (detailed descriptions of the methods are left to the references provided):

Standardized (rank) regression coefficients: this involves fitting a simple linear regression to the sample data and then using the coefficients, standardized by their respective variances, as measures of sensitivity (Saltelli et al. 2000). This approach is appealingly simple and can work with a large number of input variables. However, when the output of the model is nonlinear with respect to its inputs, sensitivity may be misrepresented.

Kolmogorov-Smirnov test: sensitivity is measured by a statistical test which, after ordering with respect to a given variable and dividing the sample into two subsets, compares the distributions of each subset to see whether there

	Stand. Reg. Coeffs	K-S Test	Cont. to Samp. Mean	Cont. to Samp. Var.	CUSUNORO	EASI	Sens. Indices
Simple random	X	X	X	X	X	X	
Latin hypercube	X	X	X	X	X	X	
Quasi MC	X	X	X	X	X	X	
Sobol' structured							XC

Fig. 57.17 Compatibility of sampling methods with sensitivity measures

is a significant difference. The degree of difference is taken as a measure of sensitivity [8].

Contribution to the sample mean plot: a plot which shows how each input variable contributes to the sample mean by estimating the mean as the sample points are successively added, in order of the value of each input variable. The plots can be used as measures of sensitivity and give further information about the effects of specific quantiles of the input distributions [1].

Contribution to the sample variance plot: these plots are the same concept as the contribution to the sample mean plots, but applied to the sample variance instead of the mean [11].

CUSUNORO: a similar approach to the contribution to the sample mean, which proceeds by standardizing the ordered data to have zero mean and unit variance [6].

EASI: an algorithm that estimates first-order sensitivity indices using the fast Fourier transform [5].

The Sobol' design is more specialized than the random or quasi-random sampling and is specifically intended for use in estimating variance-based sensitivity indices, in particular first-order and total-order sensitivity indices (see Chapter 5). Note that the output from the model can be in the form of a time series or a multidimensional variable (vector of outputs). SIMLAB can provide sensitivity indices for each time point or for each element of the vector of outputs.

In order to estimate the sensitivity measures discussed in this section, model output values must be available, corresponding to the value of the model output at each of the input points in the sample matrix. If the model is available as an executable file or perhaps reachable via the command line (e.g., via Matlab), SIMLAB can run the model automatically at each of the sample points and perform a sequential sensitivity analysis by gradually increasing the sample size until convergence has been reached. Alternatively, SIMLAB can read model output values in an ASCII format – this might be easier if the model is already set up to quickly read a matrix of input values. In this case, the file containing the model

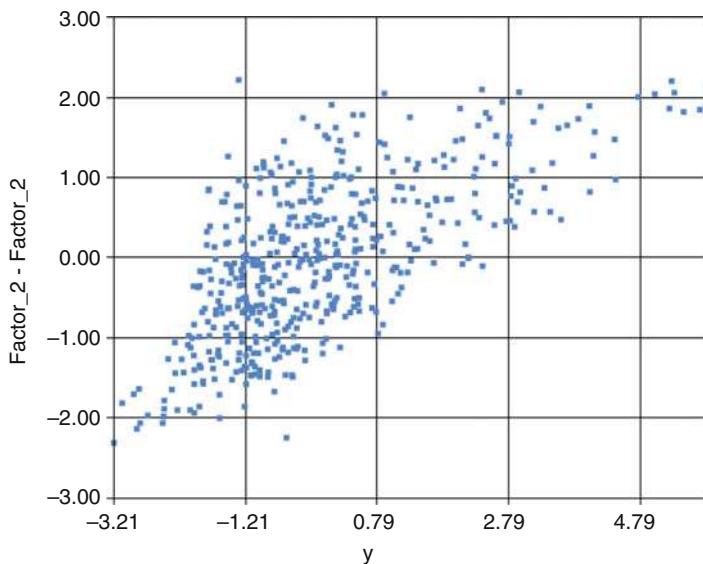


Fig. 57.18 An example of a scatterplot of the model output against one of its inputs

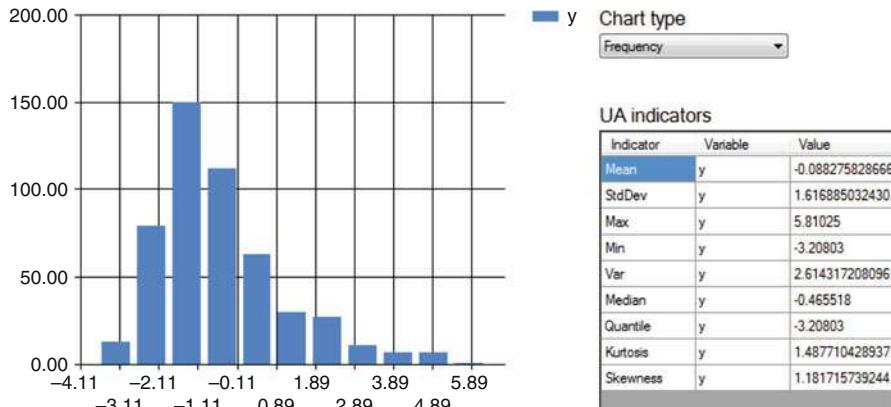
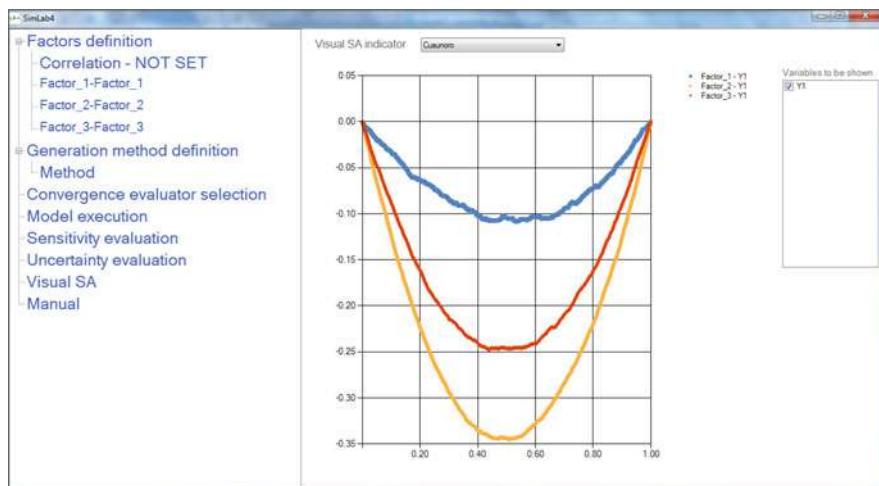
output values should be selected in the “Model execution” pane. Be sure that the file path is also specified. By clicking “Start Monte Carlo run,” SIMLAB uses the input and output samples to estimate all sensitivity measures.

To view the sensitivity measures, go to the “Sensitivity evaluation” heading in the navigation pane. The type of sensitivity measure can be selected from the drop-down menu at the top. The measures can be displayed either on a chart or in a table, although if the output is not time dependent, a table is usually more appropriate. Scatterplots can also be generated by clicking the “Visualise SA” button, for example, Fig. 57.18 shows a scatterplot of the output against one input variable.

The “Uncertainty evaluation” heading also gives information about the uncertainty in the model output, displaying a histogram, as well as giving measures of mean, variance, measures of skewness, and so on (see Fig. 57.19). Finally, by going to the “Visual SA” window, a range of visual indications of sensitivity can be plotted. Figure 57.20 shows a CUSUNORO plot for the three model inputs of the linear model shown previously. Other options that are available with a random or quasi-random sample are “contribution to sample mean” plots and “contribution to sample variance” plots.

6 Extending SIMLAB with R

SIMLAB may also be extended to use other functions and packages within R. This includes adding other types of distributions of input variables, as well as

**Fig. 57.19** Histogram and statistics of output distribution**Fig. 57.20** Contribution to sample variance plot

adding different convergence criteria, other sensitivity methods, and uncertainty indicators.

In order to make such extensions, it is necessary to edit the R project source file “SimLab4R,” which forms the technical basis of the SIMLAB installation. This must be done via RStudio. This file will be available either as part of the SIMLAB installation or separately via the website.

The process of adding extensions is best illustrated by an example. Imagine that a new type of input distribution is required – the Cauchy distribution. The package that must be edited here is “simLabDistributions.R.” In this case, the following steps must be followed:

-
1. Add the name “Cauchy” in the list of distribution names in the `getDistributions` function.
 2. Add the probability density function via the “`dcauchy`” function in the `stats` package. This is done by adding a call to “`dcauchy`” in the “`getDensityFunction`” function.
 3. Add the inverse cumulative density function in the same way, by adding a line to call the “`qcauchy`” function from the “`getInverseCDF`” function.
 4. Signal discrete and/or piecewise distribution if required, by adding lines to “`isDensityPiecewise`” and “`isDensityDiscrete`” (in this example not).
 5. Define required parameters by adding a line to “`getParameters`.”

All the steps here are explained in more detail in the manual. Further possible steps include specifying default values and validity for parameters and specifying whether distributions allow truncation.

7 Conclusions

SIMLAB is a comprehensive stand-alone program for performing global sensitivity analysis, with a number of diverse sampling strategies and sensitivity measures available for estimation, both from classical methods and more recent research. It has the capacity for sequential analysis, and its GSA techniques can be accessed through the R environment, as well as through the graphical user interface. The user can add new GSA techniques by simply adding the corresponding R code to the core layer. SIMLAB can be downloaded for free from [3].

References

1. Bolado-Lavin, R., Castaings, W., Tarantola, S.: Contribution to the sample mean plot for graphical and numerical sensitivity analysis. *Reliab. Eng. Syst. Saf.* **94**(6), 1041–1049 (2009)
2. Cukier, R., Levine, H., Shuler, K.: Nonlinear sensitivity analysis of multiparameter model systems. *J. Comput. Phys.* **26**(1), 1–42 (1978)
3. European Commission SIMLAB: Sensitivity analysis software – Joint Research Centre. <https://ec.europa.eu/jrc/en/samo/simlab> (2015)
4. McKay, M., Beckman, R., Conover, W.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245 (1979)
5. Plischke, E.: An effective algorithm for computing global sensitivity indices (EASI). *Reliab. Eng. Syst. Saf.* **95**, 354–360 (2010)
6. Plischke, E.: An adaptive correlation ratio method using the cumulative sum of the reordered output. *Reliab. Eng. Syst. Saf.* **107**, 149–156 (2012)
7. Saltelli, A., Tarantola, S., Chan, K.: Quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* **41**(1), 39–56 (1999)
8. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: *Sensitivity Analysis in Practice. A Guide to Assessing Scientific Models*. John Wiley and Sons, Chichester (2004)
9. Sobol', I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* **7**(4), 86–112 (1967)

10. Sobol', I.M.: Sensitivity estimates for nonlinear mathematical models. *Math. Model. Computat. Exp.* **1**(4), 407–414 (1993)
11. Tarantola, S., Kopustinskas, V., Bolado-Lavin, R., Kaliatka, A., Ušpuras, E., Vaišnoras, M.: Sensitivity analysis using contribution to sample variance plot: application to a water hammer model. *Reliab. Eng. Syst. Saf.* **99**, 62–73 (2012)

Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin

Abstract

The needs to assess robust performances for complex systems and to answer tighter regulatory processes (security, safety, environmental control, health impacts, etc.) have led to the emergence of a new industrial simulation challenge: to take uncertainties into account when dealing with complex numerical simulation frameworks. Therefore, a generic methodology has emerged from the joint effort of several industrial companies and academic institutions. EDF R&D, Airbus Group, and Phimeca Engineering started a collaboration at the beginning of 2005, joined by IMACS in 2014, for the development of an open-source software platform dedicated to uncertainty propagation by probabilistic methods, named OpenTURNS for open-source treatment of uncertainty, Risk 'N Statistics. OpenTURNS addresses the specific industrial challenges attached to uncertainties, which are transparency, genericity, modularity, and multi-accessibility. This paper focuses on OpenTURNS and presents its main features: OpenTURNS is an open-source software under the LGPL license that presents itself as a C++ library and a Python TUI and which works under Linux and Windows environment. All the methodological tools are described in the different sections of this paper: uncertainty quantification, uncertainty propagation, sensitivity analysis, and metamodeling. A section also explains the generic wrappers' way to link OpenTURNS to any external code. The paper illustrates

M. Baudin (✉) • A.-L. Popelin

Industrial Risk Management Department, EDF R&D France, Chatou, France
e-mail: michael.baudin@edf.fr; anne-laure.popelin@edf.fr

A. Dutfoy

Industrial Risk Management Department, EDF R&D France, Saclay, France
e-mail: anne.dutfoy@edf.fr

B. Iooss

Industrial Risk Management Department, EDF R&D, Chatou, France

Institut de Mathématiques de Toulouse, Université Paul Sabatier, Toulouse, France
e-mail: bertrand.iooss@edf.fr

as much as possible the methodological tools on an educational example that simulates the height of a river and compares it to the height of a dike that protects industrial facilities. At last, it gives an overview of the main developments planned for the next few years.

Keywords

OpenTURNS • Uncertainty • Quantification • Propagation • Estimation • Sensitivity • Simulation • Probability • Statistics • Random vectors • Multivariate distribution • Open source • Python module • C++ library • Transparency • Genericity

Contents

1	Introduction	2002
1.1	Presentation of OpenTURNS	2003
1.2	The Uncertainty Management Methodology	2004
1.3	Main Originality of OpenTURNS	2005
1.4	The Flooding Model	2006
2	Uncertainty Quantification	2007
2.1	Modeling of a Random Vector	2007
2.2	Stochastic Processes	2009
2.3	Statistics Estimation	2012
2.4	Conditioned Distributions	2014
2.5	Bayesian Calibration	2015
3	Uncertainty Propagation	2015
3.1	Min-Max Approach	2016
3.2	Central Tendency	2018
3.3	Failure Probability Estimation	2019
4	Sensitivity Analysis	2023
4.1	Graphical Tools	2024
4.2	Sampling-Based Methods	2025
5	Metamodels	2027
5.1	Polynomial Chaos Expansion	2027
5.2	The Kriging Approximation	2030
6	The External Simulator	2032
6.1	Fast Evaluations of G	2032
6.2	Evaluation of the Derivatives	2034
6.3	High-Performance Computing	2035
7	Conclusions	2035
	References	2036

1 Introduction

The needs to assess robust performances for complex systems and to answer tighter regulatory processes (security, safety, environmental control, health impacts, etc.) have led to the emergence of a new industrial simulation challenge: to take uncertainties into account when dealing with complex numerical simulation

frameworks. Many attempts at treating uncertainty in large industrial applications have involved domain-specific approaches or standards: metrology, reliability, differential-based approaches, variance decomposition, etc. However, facing the questioning of their certification authorities in an increasing number of different domains, these domain-specific approaches are no more appropriate. Therefore, a generic methodology has emerged from the joint effort of several industrial companies and academic institutions: [28] reviews these past developments. The specific industrial challenges attached to the recent uncertainty concerns are:

- Transparency: open consensus that can be understood by outside authorities and experts
- Genericity: multi-domain issue that involves various actors along the study
- Modularity: easy integration of innovations from the open-source community
- Multi-accessibility: different levels of use (simple computation, detailed quantitative results, and deep graphical analyses) and different types of end users (graphical interface, Python interpreter, and C++ sources)
- Industrial computing capabilities: to secure the challenging number of simulations required by uncertainty treatment

As no software was fully answering the challenges mentioned above, EDF R&D, Airbus Group, and Phimeca Engineering started a collaboration at the beginning of 2005, joined by IMACS in 2014, for the development of an open-source software platform dedicated to uncertainty propagation by probabilistic methods, named OpenTURNS for open-source treatment of uncertainty, Risk 'N Statistics [10, 29]. OpenTURNS is actively supported by its core team of four industrial partners (IMACS joined the consortium in 2014) and its industrial and academic user community that meet through the website www.openturns.org and annually during the OpenTURNS User's Day. At EDF, OpenTURNS is the repository of all scientific developments on this subject, to ensure their dissemination within the several business units of the company. The software has also been distributed for several years via the integrating platform Salome [27].

1.1 Presentation of OpenTURNS

OpenTURNS is an open-source software under the LGPL license that presents itself as a C++ library and a Python TUI and which works under Linux and Windows environment, with the following key features:

- Open-source initiative to secure the transparency of the approach and its openness to ongoing research and development (R&D) and expert challenging
- Generic to the physical or industrial domains for treating of multi-physical problems
- Structured in a *practitioner guidance* methodological approach

- With advanced industrial computing capabilities, enabling the use of massive distribution and high-performance computing, various engineering environments, large data models, etc.
- Includes the largest variety of qualified algorithms in order to manage uncertainties in several situations
- Contains complete documentation (reference guide, use cases guide, user manual, examples guide, and developers' guide)

All the methodological tools are described after this introduction in the different sections of this paper: uncertainty quantification, uncertainty propagation, sensitivity analysis, and metamodeling. Before the conclusion, a section also explains the generic wrappers' way to link OpenTURNS to any external code.

OpenTURNS can be downloaded from its dedicated website www.openturns.org which offers different pre-compiled packages specific to several Windows and Linux environments. It is also possible to download the source files from the *SourceForge* server (www.sourceforge.net) and to compile them within another environment: the OpenTURNS developer's guide provides advice to help compiling the source files. At last, OpenTURNS has been integrated for more than 5 years in the major Linux distributions (e.g., debian, ubuntu, redhat, and suze).

1.2 The Uncertainty Management Methodology

The uncertainty management generic methodology [29] is schematized in Fig. 58.1. It consists of the following steps:

- Step A: specify the random inputs X , the deterministic inputs d , the model G (analytical, complex computer code or experimental process), the variable of interest (model output) Y , and the quantity of interest on the output (central dispersion, its distribution, probability to exceed a threshold, etc.). The fundamental relation writes:

$$Y = G(X, d) = G(X), \quad (58.1)$$

with $X = (X_1, \dots, X_d)$.

- Step B: quantify the sources of uncertainty. This step consists in modeling the joint probability density function (pdf) of the random input vector by direct methods (e.g., statistical fitting, expert judgment) [15].
- Step B': quantify the sources of uncertainty by indirect methods using some real observations of the model outputs [39]. The calibration process aims to estimate the values or the pdf of the inputs, while the validation process aims to model the bias between the model and the real system.
- Step C: propagate uncertainties to estimate the quantity of interest. With respect to this quantity, the computational resources, and the CPU time cost of a single model run, various methods will be applied: analytical formula, geometrical

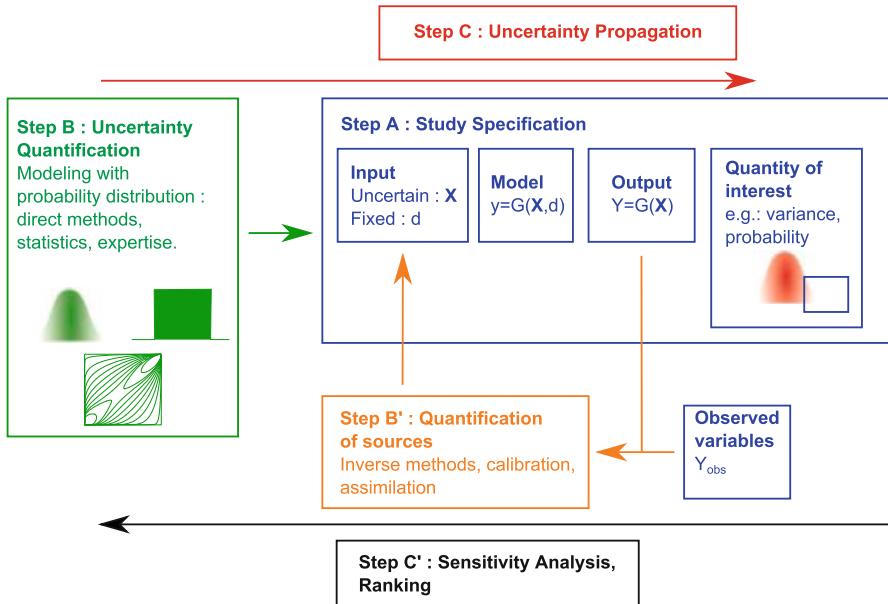


Fig. 58.1 The uncertainty management methodology

approximations, Monte Carlo sampling strategies, metamodel-based techniques, etc. [11, 21].

- Step C': analyze the sensitivity of the quantity of interest to the inputs in order to rank uncertainty sources [14, 36].

For each of these steps, OpenTURNS offers a large number of different methods whose applicability depend on the specificity of the problem (dimension of inputs, model complexity, CPU time cost for a model run, quantity of interest, etc.).

1.3 Main Originality of OpenTURNS

OpenTURNS is innovative in several aspects. Its input data model is based on the multivariate cumulative distribution function (CDF). This enables the usual sampling approach, as would be appropriate for statistical manipulation of large datasets, but also facilitates analytical approaches. Distributions are classified (continuous, discrete, elliptic, etc.) in order to take the best benefit of their properties in algorithms. If possible, the exact final cumulative density function is determined (thanks to characteristic functions implemented for each distribution, the Poisson summation formula, the Cauchy integral formula, etc.).

OpenTURNS explicitly models the dependence with copulas, using the Sklar theorem. Furthermore, different sophisticated analytical treatments may be explored:

aggregation of copulas, composition of functions from R^n into R^d , extraction of copula, and marginals from any distribution.

OpenTURNS defines a domain-specific oriented object language for probability modeling and uncertainty management. This way, the objects correspond to mathematical concepts and their interrelations map the relations between these mathematical concepts. Each object proposes sophisticated treatments in a very simple interface.

OpenTURNS implements up-to-date and efficient sampling algorithms (Mersenne-Twister algorithm, Ziggurat method, the Sequential Rejection Method, etc.). Exact Kolmogorov statistics are evaluated with the Marsaglia method and the noncentral Student and noncentral χ^2 distribution with the Benton and Krishnamoorthy method.

OpenTURNS is the repository of recent results of PhD research carried out at EDF R&D: for instance, the sparse polynomial chaos expansion method based on the LARS method [7], the adaptive directional stratification method [25] which is an accelerated Monte Carlo sampling technique, and the maximum entropy order statistics copulas [20].

1.4 The Flooding Model

Throughout this paper, the discussion is illustrated with a simple application model that simulates the height of a river and compares it to the height of a dike that protects industrial facilities as illustrated in Fig. 58.2. When the river height exceeds that of the dike, flooding occurs. This academic model is used as a pedagogical example in [14]. The model is based on a crude simplification of the 1D hydrodynamical equations of Saint-Venant under the assumptions of uniform

Fig. 58.2 The flood example: simplified model of a river

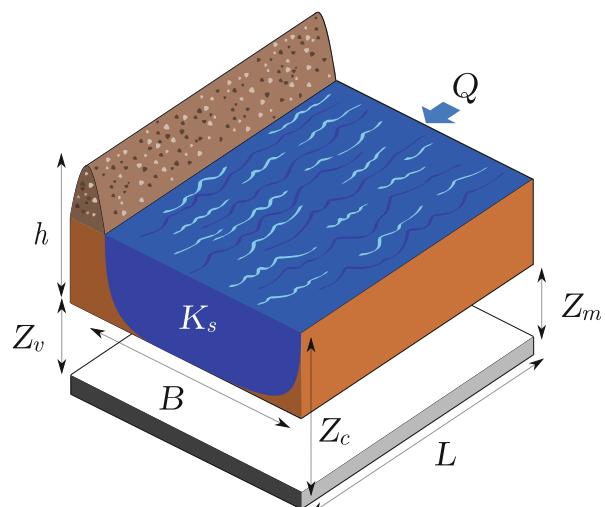


Table 58.1 Input variables of the flood model and their probability distributions

Input	Description	Unit	Probability distribution
Q	Maximal annual flow rate	m^3/s	Gumbel $\mathcal{G}(1.8e^{-3}, 1014)$
K_s	Strickler coefficient	—	Normal $\mathcal{N}(30, 7.5)$
Z_v	River downstream level	m	Triangular $\mathcal{T}(47.6, 50.5, 52.4)$
Z_m	River upstream level	m	Triangular $\mathcal{T}(52.5, 54.9, 57.7)$

and constant flow rate and large rectangular sections. It consists of an equation that involves the characteristics of the river stretch:

$$H = \left(\frac{Q}{BK_s \sqrt{\frac{Z_m - Z_v}{L}}} \right)^{0.6}, \quad (58.2)$$

where the output variable H is the maximal annual height of the river, B is the river width, and L is the length of the river stretch. The four random input variables Q , K_s , Z_v , and Z_m are defined in Table 58.1 with their probability distribution. The randomness of these variables is due to their spatiotemporal variability, our ignorance of their true value, or some inaccuracies of their estimation.

2 Uncertainty Quantification

2.1 Modeling of a Random Vector

OpenTURNS implements more than 40 parametric distributions which are continuous (more than 30 families) and discrete (more than 10 families), with several sets of parameters for each one. Some are multivariate, such as the Student distribution or the normal one.

Moreover, OpenTURNS enables the building of a wide variety of multivariate distributions, thanks to the combination of the univariate margins and a dependence structure, the copula, according to the Sklar theorem: $F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$ where F_i is the CDF of the margin X_i and $C : [0, 1]^d \rightarrow [0, 1]$ the copula.

OpenTURNS proposes more than 10 parametric families of copula: Clayton, Frank, Gumbel, Farlie-Morgenstein, etc. These copula can be aggregated to build the copula of a random vector whose components are dependent by blocks. Using the inverse relation of the Sklar theorem, OpenTURNS can extract the copula of any multivariate distribution, whatever the way it has been set up: for example, from a multivariate distribution estimated from a sample with the kernel smoothing technique.

All the distributions can be truncated in their lower and/or upper area. In addition to these models, OpenTURNS proposes other specific constructions. Among them,

note the random vector which writes as a linear combination of a finite set of independent variables: $\mathbf{X} = a_0 + a_1 \mathbf{X}_1 + \dots + a_N \mathbf{X}_N$, thanks to the Python command, written for $N = 2$ with explicit notations:

```
>>>myX= RandomMixture ([ distX1 , distX2 ] , [ a1 , a2 ] , a0 )
```

In that case, the distribution of X is *exactly* determined, using the characteristic functions of the X_i distributions and the Poisson summation formula.

OpenTURNS also easily models the random vector whose probability density function (pdf) is a linear combination of a finite set of independent pdf: $f_{\mathbf{X}} = a_1 f_{\mathbf{X}_1} + \dots + a_N f_{\mathbf{X}_N}$, thanks to the Python command, with the same notations as previously (the weights are automatically normalized):

```
>>>mypdfX= Mixture ([ distX1 , distX2 ] , [ a1 , a2 ])
```

Moreover, OpenTURNS implements a random vector that writes as the random sum of univariate independent and identically distributed variables, this randomness being distributed according to a Poisson distribution: $\mathbf{X} = \sum_{i=1}^N \mathbf{X}_i$, $N \sim \mathcal{P}(\lambda)$, thanks to the Python command:

```
>>>d= CompoundDistribution (lambda , distX )
```

where all the variables X_i are identically distributed according to $distX$. In that case, the distribution of X is *exactly* determined, using the characteristic functions of the X_i distributions and the Poisson summation formula.

In the univariate case, OpenTURNS exactly determines the pushforward distribution \mathcal{D} of any distribution \mathcal{D}_0 through the function $f : \mathbb{R} \rightarrow \mathbb{R}$, thanks to the Python command (with straight notations):

```
>>>d= CompositeDistribution (f , d0 )
```

Finally, OpenTURNS enables the modeling of a random vector (X_1, \dots, X_d) which almost surely verifies the constraint $X = X_1 \leq \dots \leq X_d$, proposing a copula adapted to the ordering constraint [8]. OpenTURNS verifies the compatibility of the margins F_i with respect to the ordering constraint and then builds the associated distribution, thanks to the Python command, written in dimension 2:

```
>>>d=MaximumEntropyOrderStatisticsDistribution ([ distX1 , distX2 ])
```

Figure 58.3 illustrates the copula of such a distribution, built as the ordinal sum of some maximum entropy order statistics copulas.

The OpenTURNS Python script to model the input random vector of the tutorial presented previously is as follows:

```
#Margin distributions:
>>>dist_Q = Gumbel(1.8e-3, 1014)
>>>dist_Q = TruncatedDistribution(dist_Q,0.0, TruncatedDistribution.LOWER)
>>>dist_K = Normal(30.0, 7.5)
>>>dist_K = TruncatedDistribution(dist_K,0., TruncatedDistribution.LOWER)
>>>dist_Zv = Triangular(47.6,50.5,52.4)
>>>dist_Zm = Triangular(52.5,54.9,57.7)
# Copula in dimension 4 for (Q,K,Zv,Zm)
>>>R=CorrelationMatrix (2)
```

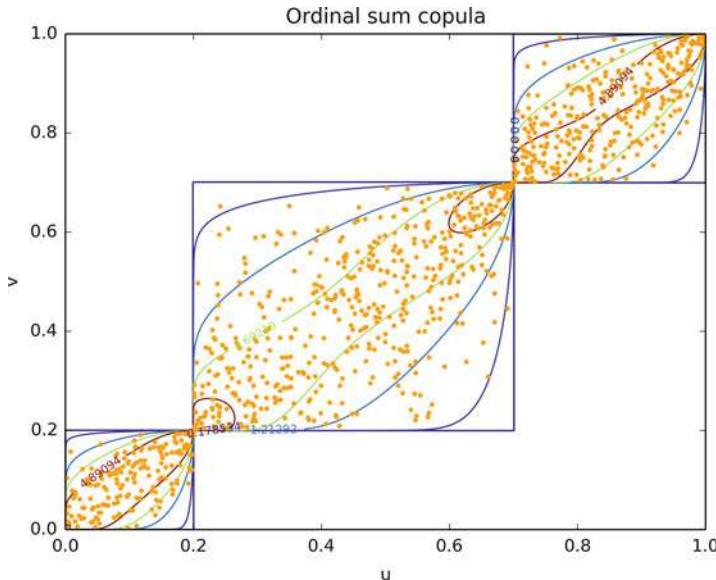


Fig. 58.3 An example of maximum entropy copula which almost surely satisfies the ordering constraint: $X_1 \leq X_2$

```
>>>R[0,1]=0.7
>>>copula = ComposedCopula([IndependentCopula(2), NormalCopula(R)])
# Final distribution for (Q,K,Zv,Zm)
>>>distInput=ComposedDistribution([loi_Q, loi_K, loi_Zv, loi_Zm], copula)
# Final random vector (Q,K,Zv,Zm)
>>>inputVector=RandomVector(distInput)
```

Note that OpenTURNS can truncate any distribution to a lower, an upper bound, or a given interval. Furthermore, a normal copula models the dependence between the variables Z_v and Z_m , with a correlation of 0.7. The variables (Q, K) are independent. Both blocks (Q, K) and (Z_v, Z_m) are independent.

2.2 Stochastic Processes

OpenTURNS implements some multivariate random fields $X : \Omega \times \mathcal{D} \rightarrow \mathbb{R}^d$ where $\mathcal{D} \in \mathbb{R}^s$ is discretized on a mesh. The user can easily build and simulate a random walk, a white noise as illustrated in Figs. 58.4 and 58.5. The Python commands write:

```
>>>myWN = WhiteNoise(myDist, myMesh)
>>>myRW = RandomWalk(myOrigin, myDist, myTimeGrid)
```

Any field can be exported into the VTK format which allows it to be visualized using, e.g., ParaView (www.paraview.org).

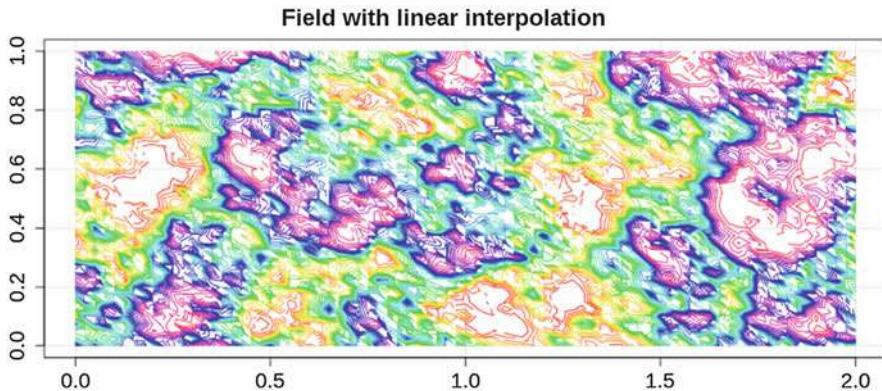


Fig. 58.4 A normal bivariate white noise

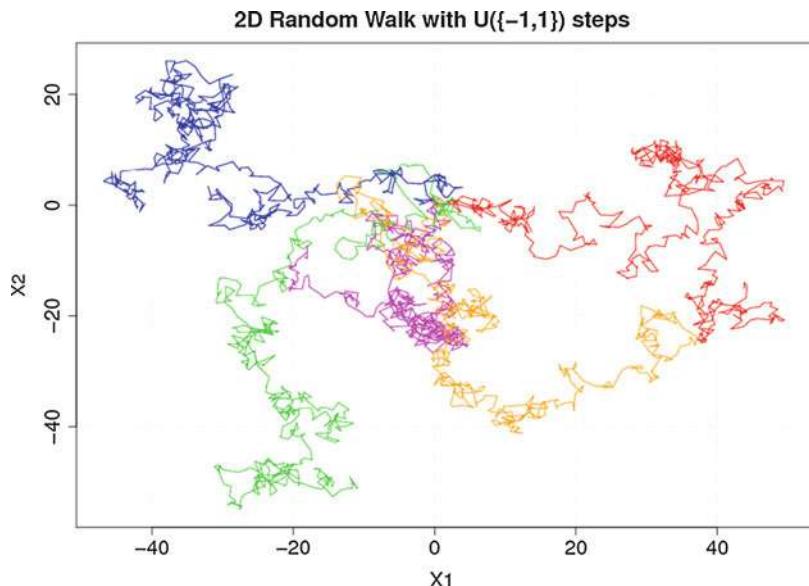


Fig. 58.5 A normal bivariate random walk

Multivariate ARMA stochastic processes $X : \Omega \times [0, T] \rightarrow \mathbb{R}^d$ are implemented in OpenTURNS which enables some manipulations on times series such as the Box Cox transformation or the addition/removal of a trend. Note that the parameters of the Box Cox transformation can be estimated from given fields of the process.

OpenTURNS models normal processes, whose covariance function is a parametric model (e.g., the multivariate exponential model) as well as defined by the user as illustrated in Fig. 58.6. Stationary processes can be defined by its spectral density function (e.g., the Cauchy model).

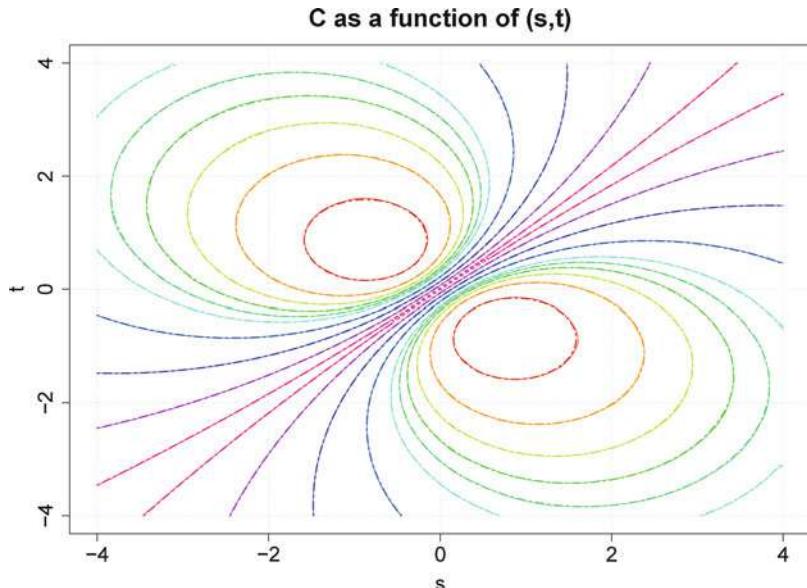


Fig. 58.6 A user-defined nonstationary covariance function and its estimation from several given fields

With explicit notations, the following Python commands create a stationary normal process defined by its covariance function, discretized on a mesh, with an additional trend:

```
>>>myNormalProcess=TemporalNormalProcess (myTrend, myCovarianceModel, myMesh)
```

Note that OpenTURNS enables the mapping of any stochastic processes X into a process Y through a function $f: Y = f(X)$ where the function f can consist, for example, of adding or removing a trend, applying a Box Cox transformation in order to stabilize the variance of X . The Python command is, with explicit notations:

```
>>>myYprocess=CompositeProcess (f, myXprocess)
```

Finally, OpenTURNS implements multivariate processes defined as a linear combination of K deterministic functions $(\phi_i)_{i=1,\dots,K}: \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_2}$:

$$X(\omega, \mathbf{x}) = \sum_{i=1}^K A_i(\omega) \phi_i(\mathbf{x})$$

where (A_1, \dots, A_K) is a random vector of dimension K . The Python command writes:

```
>>>myX = FunctionalBasisProcess (myRandomCoeff, myBasis, myMesh)
```

2.3 Statistics Estimation

OpenTURNS enables the user to estimate a model from data, in the univariate as well as in the multivariate framework, using the maximum likelihood principle or the moment-based estimation.

Some tests, such as the Kolmogorov-Smirnov test, the chi-square test, and the Anderson-Darling test (for normal distributions), are implemented and can help to select a model among others, from a sample of data. The Python command to build a model and test it writes:

```
>>> estimatedBeta = BetaFactory(sample)
>>> testResult = FittingTest.Kolmogorov(sample, estimatedBeta)
```

OpenTURNS also implements the kernel smoothing technique which is a nonparametric technique to fit a model to data: any distribution can be used as kernel. In the multivariate case, OpenTURNS uses the product kernel. It also implements an optimized strategy to select the bandwidth, depending on the number of data in the sample, which is a mix between the Silverman rule and the plugin one. Note that OpenTURNS proposes a special treatment when the data are bounded, thanks to the mirroring technique. The Python command to build the nonparametric model and to draw its pdf is as simple as the following one:

```
>>> estimatedDist = KernelSmoothing().build(sample)
>>> pdfGraph = estimatedDist.drawPDF()
```

Figure 58.7 illustrates the resulting estimated distributions from a sample of size 500 issued from a beta distribution: the kernel smoothing method takes into account

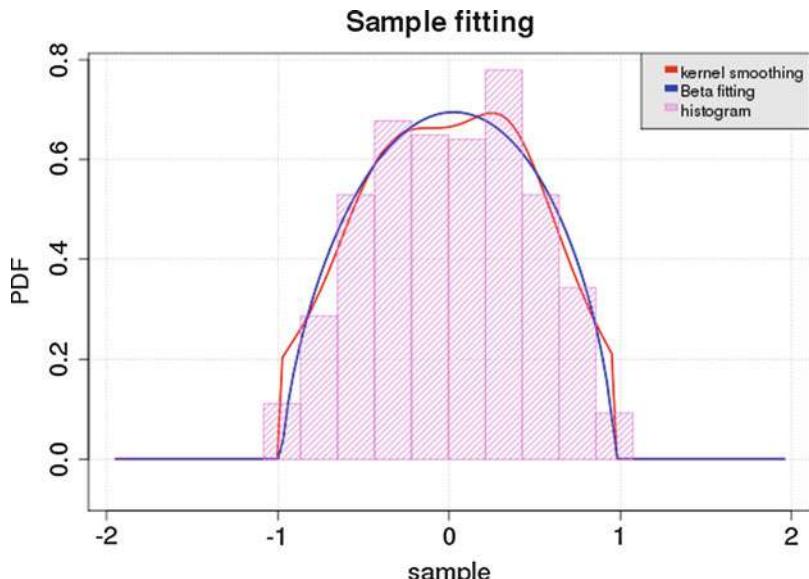


Fig. 58.7 Beta distribution estimation from a sample of size 500: parametric estimation versus kernel smoothing technique

the fact that data are bounded by 0 and 1. The histogram of the data is drawn to enable comparison.

Several visual tests are also implemented to help select models: among them, the QQ plot test and the Henry line test which writes (in the case of a beta distribution for the QQ plot test):

```
>>>graphQQplot = VisualTest.DrawQQplot(sample, Beta())
>>>graphHenryLine = VisualTest.DrawHenryLine(sample)
```

Figure 58.8 illustrates the QQ plot test on a sample of size 500 issued from a beta distribution: the adequation seems satisfying!

Stochastic processes also have estimation procedures from sample of fields or, if the ergodic hypothesis is verified, from just one field. Multivariate ARMA processes are estimated according to the BIC and AIC criteria and the Whittle estimator, which is based on the maximization of the likelihood function in the frequency domain. The Python command to estimate an $ARMA(p, q)$ process of dimension d , based on a sample of time series, writes:

```
>>>estimatedARMA = ARMALikelihood(p, q, d).build(sampleTimeSeries)
```

Moreover, OpenTURNS can estimate the covariance function and the spectral density function of normal processes from given fields. For example, the Python command to estimate a stationary covariance model from a sample of realizations of the process writes:

```
>>>myCovFunc = StationaryCovarianceModelFactory().build(sampleProcess)
```

This estimation is illustrated in Fig. 58.6.

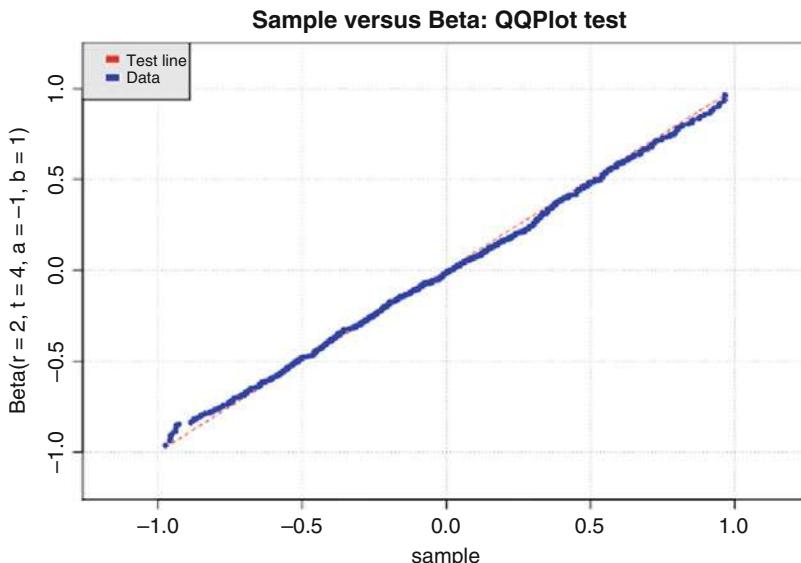


Fig. 58.8 QQ plot test: theoretical model beta versus the sample of size 500

2.4 Conditioned Distributions

OpenTURNS enables the modeling of multivariate distributions by conditioning. Several types of conditioning are implemented.

At first, OpenTURNS enables the creation of a random vector X whose distribution $\mathcal{D}_{X|\Theta}$ whose parameters Θ form a random vector distributed according to the distribution \mathcal{D}_Θ . The Python command writes:

```
>>>myXrandVect = ConditionalRandomVector(distXgivenTheta, distTheta)
```

Figure 58.9 illustrates a random variable X distributed according to a normal distribution, $\mathcal{D}_{X|\Theta=(M,\Sigma)} = \text{Normal}(M, \Sigma)$, whose parameters are defined by $M \sim \text{Uniform}([0, 1])$ and $\Sigma \sim \text{Exponential}(\lambda = 4)$. The probability density function of X has been fitted with the kernel smoothing technique from $n = 10^6$ realizations of X with the normal kernel. It also draws, for comparison needs, the probability density function of X in the case where the parameters are fixed to their mean value.

Furthermore, when the random vector Θ is defined as $\Theta = g(Y)$ where the random vector follows a known distribution \mathcal{D}_Y and g is a given function, OpenTURNS creates the distribution of X with the Python command:

```
>>>finalDist = ConditionalDistribution(distXGgivenTheta, distY, g)
```

Figure 58.10 illustrates the distribution of X that follows a $\text{Uniform}(A, B)$ distribution, with $(A, B) = g(Y)$, $g : \mathbb{R} \rightarrow \mathbb{R}^2$, $g(Y) = (Y, 1 + Y^2)$ and Y follows a $\text{Uniform}(-1, 1)$ distribution.

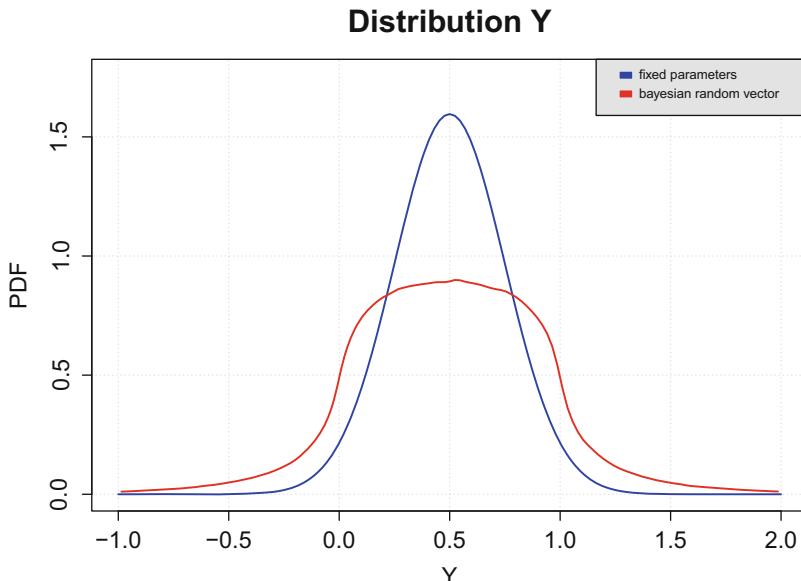


Fig. 58.9 Normal distribution with random or fixed parameters

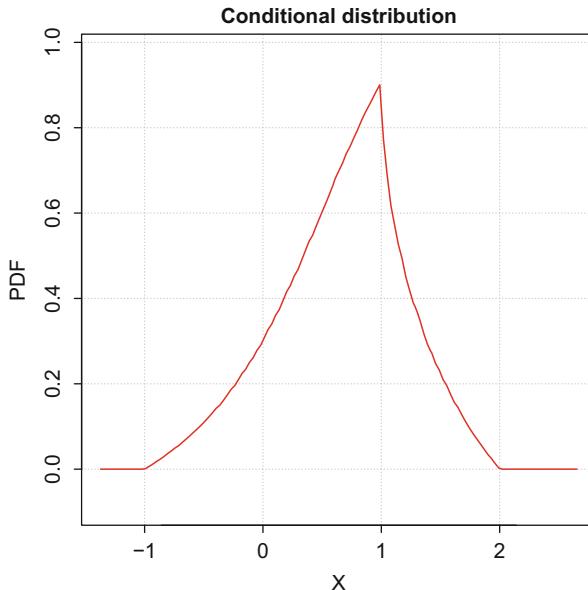


Fig. 58.10 $Uniform(Y, 1 + Y^2)$, with $Y \sim Uniform(-1, 1)$

2.5 Bayesian Calibration

Finally, OpenTURNS enables the calibration of a model (which can be a computer code), thanks to the Bayesian estimation, which is the evaluation of the model's parameters. More formally, let's consider a model G that writes: $y = G(x, \theta)$ where $x \in \mathbb{R}^{d_1}$, $y \in \mathbb{R}^{d_3}$ and $\theta \in \mathbb{R}^{d_2}$ is the vector of unknown parameters to calibrate. The Bayesian calibration consists in estimating θ , based on a certain set of n inputs (x^1, \dots, x^n) (an experimental design) and some associated observations (z^1, \dots, z^n) which are regarded as the realizations of some random vectors (Z^1, \dots, Z^n) , such that, for all i , the distribution of Z^i depends on $y^i = g(x^i, \theta)$. Typically, $Z^i = Y^i + \epsilon^i$ where ϵ^i is a random measurement error. Once the user has defined the prior distribution of θ , OpenTURNS maximizes the likelihood of the observations and determines the posterior distribution of θ , given the observations, using the Metropolis-Hastings algorithm [5, 23].

3 Uncertainty Propagation

Once the input multivariate distribution has been satisfactorily chosen, these uncertainties can be propagated through the G model to the output vector Y . Depending on the final goal of the study (min-max approach, central tendency, or reliability), several methods can be used to estimate the corresponding quantity

of interest, tending to respect the best compromise between the accuracy of the estimator and the number of calls to the numerical, and potentially costly, model.

3.1 Min-Max Approach

The aim here is to determine the extreme (minimum and maximum) values of the components of Y for the set of all possible values of X . Several techniques enable it to be done:

- Techniques based on design of experiments: the extreme values of Y are sought for only a finite set of combinations $(\mathbf{x}^1, \dots, \mathbf{x}^n)$.
- Techniques using optimization algorithms.

3.1.1 Techniques Based on Design of Experiments

In this case, the min-max approach consists of three steps:

- Choice of experiment design used to determine the combinations $(\mathbf{x}^1, \dots, \mathbf{x}^n)$ of the input random variables
- Evaluation of $\mathbf{y}^i = G(\mathbf{x}^i)$ for $i = 1, \dots, N$
- Evaluation of $\min_{1 \leq i \leq N} y_i^k$ and of $\max_{1 \leq i \leq N} y_i^k$, together with the combinations related to these extreme values: $\mathbf{x}_{k,\min} = \operatorname{argmin}_{1 \leq i \leq N} y_i^k$ and $\mathbf{x}_{k,\max} = \operatorname{argmax}_{1 \leq i \leq N} y_i^k$

The type of design of experiments impacts the quality of the metamodel and then on the evaluation of its extreme values. OpenTURNS gives access to two usual families of design of experiments for a min-max study:

- Some stratified patterns (axial, composite, factorial, or box patterns). Here are the two command lines that generate a sample from a two-level factorial pattern:

```
>>>myCenteredReducedGrid = Factorial(2, levels)
>>>mySample = myCenteredReducedGrid.generate()
```

- Some weighted patterns that include, on the one hand, random patterns (Monte Carlo, LHS) and, on the other hand, low-discrepancy sequences (Sobol, Faure, Halton, Reverse Halton, and Haselgrave, in dimension $n > 1$). The following lines illustrate the creation of a Faure sequence in dimension 2 or a Monte Carlo design experiment from a bidimensional normal (0,1) distribution:

```
# Sobol Sequence Sampling
>>>mySobolSample = FaureSequence(2).generate(1000)
# Monte Carlo Sampling
>>>myMCSSample = MonteCarloExperiment(Normal(2), 100)
```

Figures 58.11 and 58.12, respectively, illustrate a design of experiments issued from a Faure sequence and a normal distribution in dimension 2.

Fig. 58.11 The first 1000 points according to a Faure sequence of dimension 2

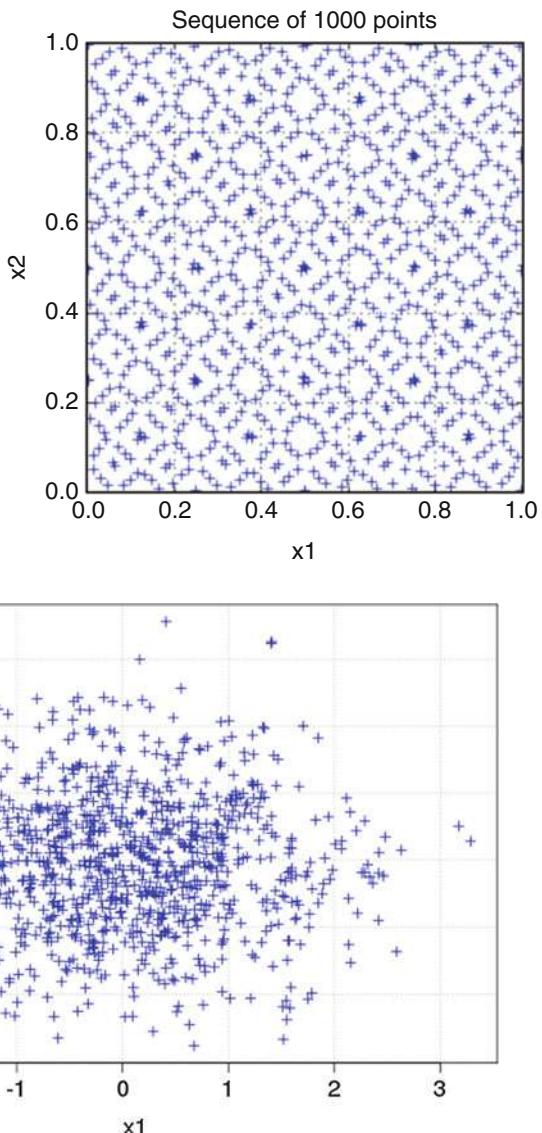


Fig. 58.12 Sample of 1000 points according to a normal $(0,1)$ distribution of dimension 2

3.1.2 Techniques Based on Optimization Algorithm

In this kind of approach, the min or max value of the output variable is sought, thanks to an optimization algorithm. OpenTURNS offers several optimization algorithms for the several steps of the global methodology. Here, the Truncated Newton Constrained (TNC) is often used, which minimizes a function with variables subject to bounds, using gradient information. More details may be found in [26].

```

# For the research of the min value
>>>myAlgoTNC = TNC(TNCSpecificParameters() , limitStateFunction ,
                     intervalOptim , startingPoint , TNC.MINIMIZATION)
# For the research of the max value
>>>myAlgoTNC = TNC(TNCSpecificParameters() , limitStateFunction ,
                     intervalOptim , startingPoint , TNC.MAXIMIZATION)
# Run the research and extract the results
>>>myAlgoTNC.run()
>>>myAlgoTNCResult = BoundConstrainedAlgorithm(myAlgoTNC).getResult()
>>>optimalValue = myAlgoTNCResult.getOptimalValue()

```

3.2 Central Tendency

A central tendency evaluation aims at evaluating a reference value for the variable of interest, here the water level, H , and an indicator of the dispersion of the variable around the reference. To address this problem, mean $\mu_Y = e(Y)$ and the standard deviation $\sigma_Y = \sqrt{V(Y)}$ of Y are here evaluated using two different methods.

First, following the usual method within the measurement science community [12], μ_Y and σ_Y have been computed under a Taylor first-order approximation of the function $Y = G(\mathbf{X})$ (notice that the explicit dependence on the deterministic variable \mathbf{d} is here omitted for simplifying notations):

$$\mu_Y \simeq G(\mathbb{E}(\mathbf{X})) \quad (58.3)$$

$$\sigma_Y \approx \sum_{i=1}^d \sum_{j=1}^d \frac{\partial G}{\partial X_i} \Big|_{e(X)} \frac{\partial G}{\partial X_j} \Big|_{e(X)} \rho_{ij} \sigma_i \sigma_j, \quad (58.4)$$

σ_i and σ_j being the standard deviation of the i th and j th components X_i and X_j of the vector \mathbf{X} and ρ_{ij} their correlation coefficient. Thanks to the formulas above, the mean and the standard deviation of H are evaluated as 52.75m and 1.15m, respectively:

```

>>>myQuadCum = QuadraticCumul(outputVariable)
# First order Mean
>>>meanFirstOrder = myQuadCum.getMeanFirstOrder()[0]
# Second order Mean
>>>meanSecondOrder = myQuadCum.getMeanSecondOrder()[0]
# First order Variance
>>>varFirstOrder = myQuadCum.getCovariance()[0,0]

```

Then, the same quantities have been evaluated by a Monte Carlo evaluation: a set of 10000 samples of the vector \mathbf{X} is generated and the function $G(\mathbf{X})$ is evaluated, thus giving a sample of H . The empirical mean and standard deviation of this sample are 52.75 and 1.42 m, respectively. Figure 58.13 shows the empirical histogram of the generated sample of H .

```

# Create a random sample of the output variable of interest of size 10000
>>>outputSample = outputVariable.getNumericalSample(10000)

```

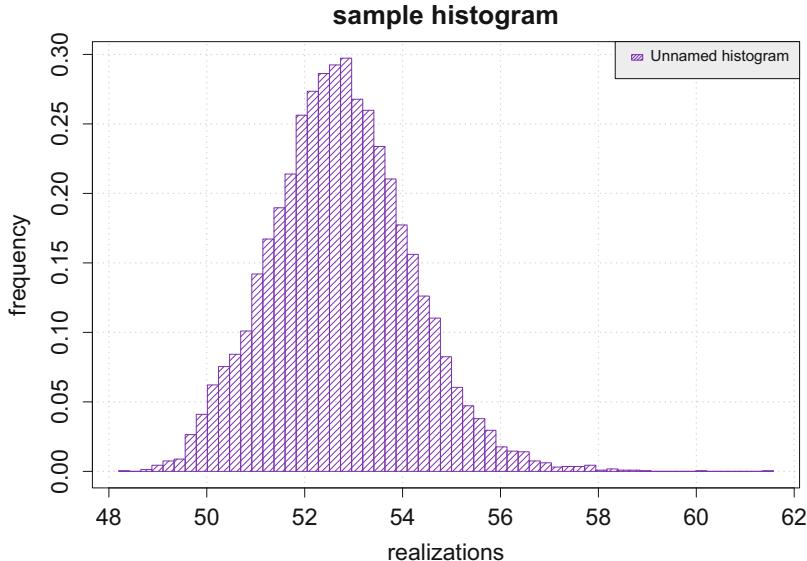


Fig. 58.13 Empirical histogram of 10000 samples of H

```
# Get the empirical mean
>>> empiricalMean = outputSample.computeMean()
# Get the empirical covariance matrix
>>> empiricalCovarianceMatrix = outputSample.computeCovariance()
```

3.3 Failure Probability Estimation

This section focuses on the estimation of the probability for the output Y to exceed a certain threshold s , noted P_f in the following. If s is the altitude of a flood protection dike, then the above excess probability, P_f , can be interpreted as the probability of an overflow of the dike, i.e., a failure probability.

Note that an equivalent way of formulating this reliability problem would be to estimate the $(1 - p)$ -th quantile of the output's distribution. This quantile can be interpreted as the flood height q_p which is attained with probability p each year. $T = 1/p$ is then seen to be a return period, i.e., a flood as high as $q_{1/T}$ occurs on average every T years.

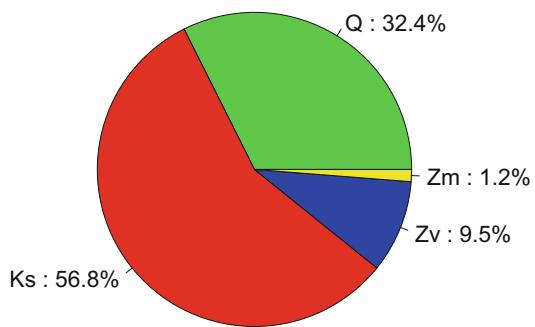
Hence, the probability of overflowing a dike with height s is less than p (where p , for instance, could be set according to safety regulations) if and only if $s \geq q_p$, i.e., if the dike's altitude is higher than the flood with return period equal to $T = 1/p$.

3.3.1 FORM

A way to evaluate such failure probabilities is through the so-called first-order reliability method (FORM) [9]. This approach allows, by using an equiprobabilistic transformation and an approximation of the limit state function, the evaluation with

Fig. 58.14 FORM importance factors

FORM Importance Factors – Event Zc > 58.0



a much reduced number of model evaluations, of some low probability as required in the reliability field. Note that OpenTURNS implements the Nataf transformation where the input vector X has a normal copula, the generalized Nataf transformation when X has an elliptical copula, and the Rosenblatt transformation for any other cases [16–19].

The probability that the yearly maximal water height H exceeds $s=58$ m is evaluated using FORM. The Hasofer-Lind reliability index was found to be equal to $\beta_{HL} = 3.04$, yielding a final estimate of:

$$\hat{P}_{f, \text{FORM}} = 1.19 \times 10^{-3}.$$

The method gives also some importance factors that measure the weight of each input variable in the probability of exceedance, as shown on Fig. 58.14.

```
>>>myFORM = FORM(Cobyla(), myEvent, meanInputVector)
>>>myFORM.run()
>>>FormResult = myFORM.getResult()
>>>pFORM = FormResult.getEventProbability()
>>>HasoferIndex = FormResult.getHasoferReliabilityIndex()
# Importance factors
>>>importanceFactorsGraph = FormResult.drawImportanceFactors()
```

3.3.2 Monte Carlo

Whereas the FORM approximation relies on strong assumptions, the Monte Carlo method is always valid, independently from the regularity of the model. It is nevertheless much more computationally intensive, covering all the input domain to evaluate the probability of exceeding a threshold. It consists in sampling many input values $(X^{(i)})_{1 \leq i \leq N}$ from the input vector joint distribution, then computing the corresponding output values $Y^{(i)} = g(X^{(i)})$. The excess probability P_f is then estimated by the proportion of sampled values $Y^{(i)}$ that exceed t :

$$\hat{P}_{f, MC} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{Y^{(i)} > s\}}. \quad (58.5)$$

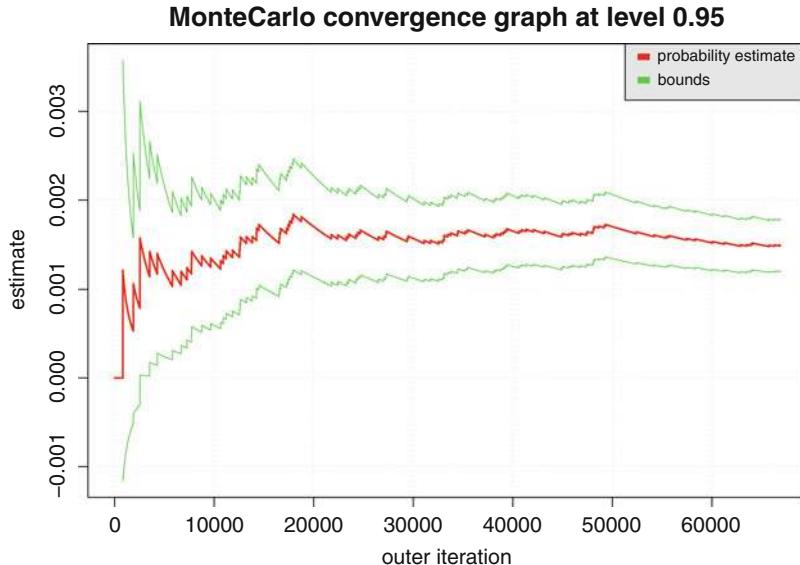


Fig. 58.15 Monte Carlo convergence graph

The sample average of the estimation error $\hat{P}_{f,MC} - P_f$ decreases as $1/\sqrt{N}$ and can be precisely quantified by a confidence interval derived from the central limit theorem. In the present case, the results are:

$$\hat{P}_{f,MC} = 1.50 \times 10^{-3},$$

with the following 95% confidence interval:

$$I_{P_f,MC} = [1.20 \times 10^{-3}, 1.79 \times 10^{-3}].$$

These results are coherent with those of the FORM approximation, confirming that the assumptions underlying the latter are correct. Figure 58.15 shows the convergence of the estimate depending on the size of the sample, obtained with OpenTURNS.

```
>>>myEvent = Event(outputVariable, Greater(), threshold)
>>>myMonteCarlo = MonteCarlo(myEvent)
# Specify the maximum number of simulations
>>>numberMaxSimulation = 100000
>>>myMonteCarlo.setMaximumOuterSampling(numberMaxSimulation)
# Perform the algorithm
>>>myMonteCarlo.run()
# Get the convergence graph
>>>convergenceGraph = myMonteCarlo.drawProbabilityConvergence()
>>>convergenceGraph.draw("MonteCarloCovergenceGraph")
```

3.3.3 Importance Sampling

An even more precise estimate can be obtained through importance sampling [31], using the Gaussian distribution with identity covariance matrix and mean equal to the design point u^* as the proposal distribution. Many values $(U^{(i)})_{1 \leq i \leq N}$ are sampled from this proposal. Because $\phi_n(u - u^*)$ is the proposal density from which the $U^{(i)}$ have been sampled, the failure probability can be estimated without bias by:

$$\hat{P}_{f,IS} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{G \circ T^{-1} U^{(i)} > s\}} \frac{\phi_n(U^{(i)})}{\phi_n(U^{(i)} - u^*)} \quad (58.6)$$

The rationale of this approach is that, by sampling in the vicinity of the failure domain boundary, a larger proportion of values fall within the failure domain than by sampling around the origin, leading to a better evaluation of the failure probability and a reduction in the estimation variance. Using this approach, the results are:

$$\hat{P}_{f,IS} = 1.40 \times 10^{-3}$$

As in the simple Monte Carlo approach, a 95%-level confidence interval can be derived from the output of the importance sampling algorithm. In the present case, this is equal to:

$$I_{P_f,IS} = [1.26 \times 10^{-3}, 1.53 \times 10^{-3}],$$

and indeed provides tighter confidence bounds for P_f .

```
# Specify the starting point from FORM algorithm
>>> standardPoint = FormResult.getStandardSpaceDesignPoint()
# Define the importance distribution
>>> sigma = [1.0, 1.0, 1.0, 1.0]
>>> importanceDistrib = Normal(standardPoint, sigma, CorrelationMatrix(4))
# Define the IS algorithm : event, distribution, criteria of convergence
>>> myAlgoImportanceSampling = ImportanceSampling(
    (myStandardEvent, importanceDistrib))
>>> myAlgoImportanceSampling.setMaximumOuterSampling(maximumOuterSampling_IS)
>>> myAlgoImportanceSampling.setMaximumCoefficientOfVariation(0.05)
```

3.3.4 Directional Sampling

The directional simulation method is an accelerated sampling method that involves as a first step a preliminary iso-probabilistic transformation as in the FORM method. The basic idea is to explore the space by sampling in several directions in the standard space. The final estimate of the probability P_f after N simulations is the following:

$$\hat{P}_{f,DS} = \frac{1}{N} \sum_{i=1}^N q_i$$

where q_i is the probability obtained in each explored direction. A central limit theorem allows to access to some confidence interval on this estimate. More details on this specific method can be found in [32].

In practice in OpenTURNS, the directional sampling simulation requires the choice of several parameters in the methodology: a sampling strategy to choose the explored directions, a “root strategy” corresponding to the way to seek the limit state function (i.e., a sign change) along the explored direction, and a nonlinear solver to estimate the root. A default setting of these parameters allows the user to test the method in one command line:

```
>>>myAlgo = DirectionalSampling(myEvent)
```

3.3.5 Subset Sampling

The subset sampling is a method for estimating rare event probability, based on the idea of replacing rare failure event by a sequence of more frequent events F_i :

$$F_1 \supset F_2 \supset \cdots \supset F_m = F$$

The original probability is obtained conditionally to the more frequent events:

$$P_f = P(F_m) = P\left(\bigcap_{i=1}^m F_i\right) = P(F_1) \prod_{i=2}^m P(F_i | F_{i-1})$$

In practice, the subset simulation shows a substantial improvement ($N_T \sim \log P_f$) compared to crude Monte Carlo ($N_T \sim \frac{1}{P_f}$) sampling when estimating rare events. More details on this specific method can be found in [2].

OpenTURNS provides this method through a dedicated module. Here also, some parameters of the methods have to be chosen by the user: a few command lines allows the algorithm to be set up before its launch.

```
>>>mySSAlgo=SubsetSampling(myEvent)
# Change the target conditional probability of each subset domain
>>>mySSAlgo.setTargetProbability(0.9)
# Set the width of the MCMC random walk uniform distribution
>>>mySSAlgo.setProposalRange(1.8)
# This allows to control the number of samples per step
>>>mySSAlgo.setMaximumOuterSampling(10000)
# Run the algorithm
>>>mySSAlgo.run()
```

4 Sensitivity Analysis

The sensitivity analysis aims to investigate how a given computational model answers to variations in its inputs. Such knowledge is useful for determining the degree of resemblance of a model and a real system, distinguishing the factors that

mostly influence the output variability and those that are insignificant, revealing interactions among input parameters and correlations among output variables, etc. A detailed description of sensitivity analysis methods can be found in [36] and in the *Sensitivity analysis* chapter of the Springer Handbook. In the global sensitivity analysis strategy, the emphasis is put on apportioning the output uncertainty to the uncertainty in the input factors, given by their uncertainty ranges and probability distributions.

4.1 Graphical Tools

In sensitivity analysis, graphical techniques have to be used first. With all the scatterplots between each input variable and the model output, one can immediately detect some trends in their functional relation. The following instructions allow scatterplots of Fig. 58.16 to be obtained from a Monte Carlo sample of size $N = 1000$ of the flooding model.

```
>>> inputSample = inputRandomVector.getNumericalSample(1000)
>>> inputSample.setDescription(['Q', 'K', 'Zv', 'Zm'])
>>> outputSample = finalModelCrue(inputSample)
>>> outputSample.setDescription(['H'])
# Here, stack both samples in one
>>> inputSample.stack(outputSample)
>>> myPairs = Pairs(inputSample)
>>> myGraph = Graph()
>>> myGraph.add(myPairs)
```

In the right column of Fig. 58.16, it is clear that the strong and rather linear effects of Q and Z_v on the output variable H . In the plot of the third line and fourth column, it is also clear that the dependence between Z_v and Z_m comes from the large correlation coefficient introduced in the probabilistic model.

However scatterplots do not capture some interaction effects between the inputs. Cobweb plots are then used to visualize the simulations as a set of trajectories. The following instructions allow the cobweb plots of Fig. 58.17 to be obtained where the simulations leading to the largest values of the model output H have been colored in red.

```
>>> inputSample = inputRandomVector.getNumericalSample(1000)
>>> outputSample = finalModelCrue(inputSample)
# Graph 1 : value based scale to describe the Y range
>>> minValue = outputSample.computeQuantilePerComponent(0.05)[0]
>>> maxValue = outputSample.computeQuantilePerComponent(0.95)[0]
>>> myCobweb = VisualTest.DrawCobWeb(inputSample, outputSample,
                                         minValue, maxValue, 'red', False)
```

The cobweb plot allows us to immediately understand that these simulations correspond to large values of the flow rate Q and small values of the Strickler coefficient K_s .

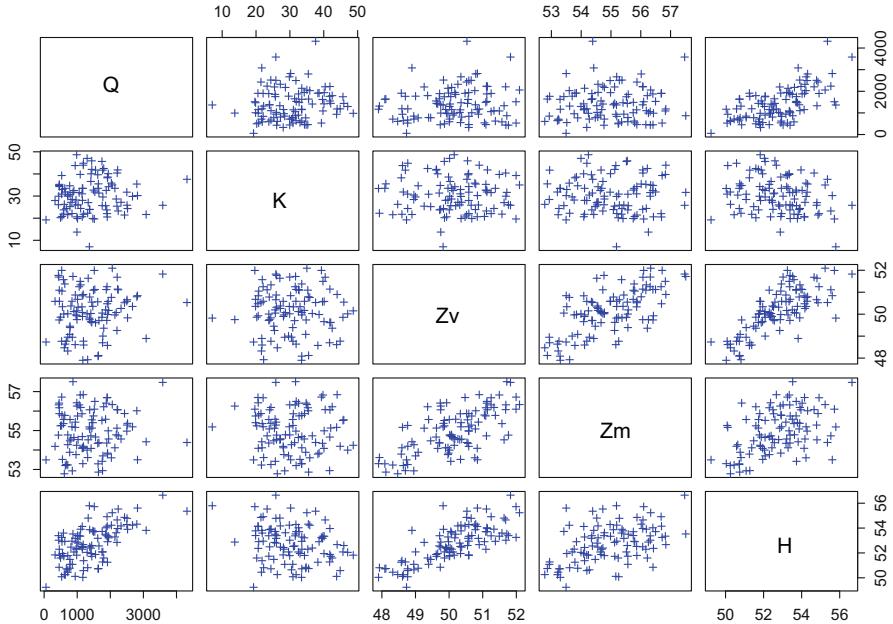


Fig. 58.16 Scatterplots between the inputs and the output of the flooding model: each combination (input i, input j) and (input i, output) is drawn, which enables to exhibit some correlation patterns

4.2 Sampling-Based Methods

In order to obtain quantitative sensitivity indices rather than qualitative information, one may use some sampling-based methods which often suppose that the input variables are independent. The section illustrates some of these methods on the flooding model with independence between its input variables.

If the behavior of the output Y compared to the input vector X is overall linear, it is possible to obtain quantitative measurements of the inputs influences from the regression coefficients α_i of the linear regression connecting Y to the $X = (X_1, \dots, X_p)$. The standard regression coefficient (SRC), defined by:

$$\text{SRC}_i = \alpha_i \frac{\sigma_i}{\sigma_Y} \quad (\text{for } i = 1 \dots p), \quad (58.7)$$

with σ_i (resp. σ_Y), the standard deviation of X_i (resp. Y), measures the variation of the response for a given variation of the parameter X_i . In practice, the coefficient R^2 (the variance percentage of the output variable Y explained by the regression model) also helps to check the linearity: if R^2 is close to one, the relation connecting Y to all the parameters X_i is almost linear and the SRC sensitivity indices make sense.

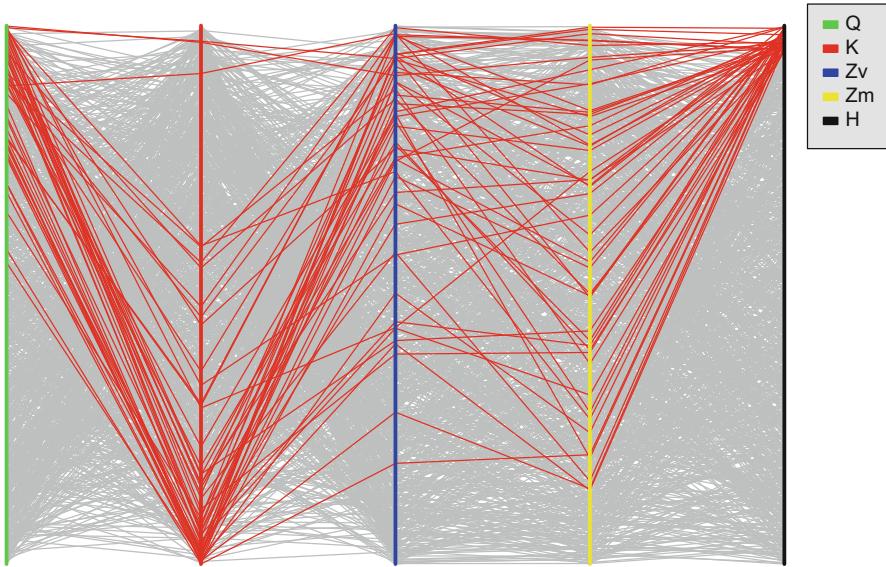


Fig. 58.17 Cobweb plot for the flooding model: each simulation is drawn. The input marginal values are linked to the output value (last column). All the simulations that led to a high quantile of the output are drawn in red: the cobweb plot enables to detect all the combinations of the inputs they come from

Table 58.2 Regression coefficients and SRC of the flood model inputs ($\alpha_0 = -0.1675$ and $R^2 = 0.97$)

	Q	K_s	Z_v	Z_m
α_i	3.2640	0.0012	-0.0556	1.1720
SRC _i	0.3462	0.0851	0.6814	0.0149

The following instructions allow the results of Table 58.2 to be obtained from a Monte Carlo sample of size $N = 1000$.

```
>>>inputSample = inputRandomVector.getNumericalSample(1000)
>>>outputSample = finalModelCrue(inputSample)
>>>SRCCoefficient = CorrelationAnalysis.SRC
        (inputSample, outputSample)
>>>linRegModel=LinearModelFactory().build
        (inputSample, outputSample, 0.90)
>>>Rsquared = LinearModelTest.LinearModelRSquared(inputSample,
        outputSample, linRegModel, 0.90)
```

The SRC values confirm our first conclusions drawn from the scatterplots visual analysis. As $R^2 = 0.97$ is very close to one, the model is quasi-linear. The SRC coefficients are sufficient to perform a global sensitivity analysis.

Several other estimation methods are available in OpenTURNS for a sensitivity analysis purpose:

- Derivatives and Pearson correlation coefficients (linearity hypothesis between output and inputs).
- Spearman correlation coefficients and standard rank regression coefficients (monotonicity hypothesis between output and inputs).
- Reliability importance factors with the FORM/SORM importance measures presented previously (Sect. 3).
- Variance-based sensitivity indices (no hypothesis on the model). These last indices, often known as Sobol indices and defined by:

$$S_i = \frac{\text{Var}[\mathbb{E}(Y|X_i)]}{\text{Var}(Y)} \text{ (first order index) and} \quad (58.8)$$

$$S_{T_i} = \sum_{i=1}^p S_i + \sum_{i < j} S_{ij} + \dots \text{ (total index),}$$

are estimated in OpenTURNS with the classic pick-freeze method based on two independent Monte Carlo samples [34]. In OpenTURNS, other ways to compute the Sobol indices are the extended FAST method [35] and the coefficients of the polynomial chaos expansion [38].

5 Metamodels

When each model evaluation is time consuming, it is usual to build a surrogate model which is a good approximation of the initial model and which can be evaluated at negligible cost. OpenTURNS proposes some usual polynomial approximations: the linear or quadratic Taylor approximations of the model at a particular point, or a linear approximation based on the least squares method. Two recent techniques implemented in OpenTURNS are detailed here: the polynomial chaos expansion and the Kriging approximation.

5.1 Polynomial Chaos Expansion

The polynomial chaos expansion enables the approximation of the output random variable of interest $Y = G(X)$ with $g : \mathbb{R}^d \rightarrow \mathbb{R}^p$ by the surrogate model:

$$\tilde{Y} = \sum_{k \in K} \alpha_k \Psi_k \circ T(X)$$

where $\alpha_k \in \mathbb{R}^p$, T is an iso-probabilistic transformation (e.g., the Rosenblatt transformation) which maps the multivariate distribution of X into the multivariate distribution $\mu = \prod_{i=1}^d \mu_i$, and $(\Psi_k)_{k \in \mathbb{N}}$ is a multivariate polynomial basis of

$\mathcal{L}_\mu^2(\mathbb{R}^d, \mathbb{R}^p)$ which is orthonormal according to the distribution μ . K is a finite subset of \mathbb{N} . \mathbf{Y} is supposed to be of finite second moment.

OpenTURNS proposes the building of the multivariate orthonormal basis $(\Psi_k(\mathbf{x}))_{k \in \mathbb{N}}$ as the Cartesian product of orthonormal univariate polynomial family $(\Psi_l^i(z_i))_{l \in \mathbb{N}}$:

$$\Psi_k(\mathbf{z}) = \Psi_{k_1}^1(z_1) * \Psi_{k_2}^2(z_2) * \cdots * \Psi_{k_d}^d(z_d)$$

The possible univariate polynomial families associated to continuous measures are:

- Hermite, which is orthonormal with respect to the *normal*(0, 1) distribution
- Jacobi($\alpha, \beta, param$), which is orthonormal with respect to the *Beta*($\beta + 1, \alpha + \beta + 2, -1, 1$) distribution if $param = 0$ (default value) or to the *Beta*($\alpha, \beta, -1, 1$) distribution if $param = 1$
- Laguerre(k), which is orthonormal with respect to the *Gamma*($k + 1, 1, 0$) distribution
- Legendre, which is orthonormal with respect to the *Uniform*(-1, 1) distribution

OpenTURNS proposes three strategies to truncate the multivariate orthonormal basis to the finite set K : these strategies select different terms from the multivariate basis, based on a convergence criterion of the surrogate model and the cleaning of the less significant coefficients.

The coefficients of the polynomial decomposition writes:

$$\boldsymbol{\alpha} = \operatorname{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^K} E_\mu \left[\left(g \circ T^{-1}(\mathbf{Z}) - \sum_{k \in K} \alpha_k \Psi_k(\mathbf{Z}) \right)^2 \right] \quad (58.9)$$

as well as:

$$\boldsymbol{\alpha} = (E_\mu [g \circ T^{-1}(\mathbf{Z}) \Psi_k(\mathbf{Z})])_k \quad (58.10)$$

where $\mathbf{Z} = T(\mathbf{X})$ is distributed according to μ .

It corresponds to two points of view implemented by OpenTURNS: the relation (58.9) means that the coefficients $(\alpha_k)_{k \in K}$ minimize the mean quadratic error between the model and the polynomial approximation; the relation (58.10) means that α_k is the scalar product of the model with the $k-th$ element of the orthonormal basis $(\Psi_k)_{k \in K}$. In both cases, the expectation E_μ is approximated by a linear quadrature formula that writes, in the general case:

$$E_\mu [f(\mathbf{Z})] \simeq \sum_{i \in I} \omega_i f(\mathcal{E}_i) \quad (58.11)$$

where f is a function $L_1(\mu)$. The set I , the points $(\mathcal{E}_i)_{i \in I}$, and the weights $(\omega_i)_{i \in I}$ are evaluated from weighted designs of experiments which can be random (Monte Carlo experiments and importance sampling experiments) or deterministic (low-discrepancy experiments, user-given experiments, and Gaussian product experiments).

At last, OpenTURNS gives access to:

- The composed model $h : \mathbf{Z} \mapsto \mathbf{Y} = G \circ T^{-1}(\mathbf{Z})$, which is the model of the reduced variables \mathbf{Z} . Then, $h = \sum_{k \in \mathbb{N}} \alpha_k \Psi_k$.
- The coefficients of the polynomial approximation: $(\alpha_k)_{k \in K}$.
- The composed metamodel: \hat{h} , which is the model of the reduced variables reduced to the truncated multivariate basis $(\Psi_k)_{k \in K}$. Then, $\hat{h} = \sum_{k \in K} \alpha_k \Psi_k$.
- The metamodel: $\hat{g} : \mathbf{X} \mapsto Y = \hat{h} \circ T(\mathbf{X})$ which is the polynomial chaos approximation as a NumericalMathFunction. Then, $\hat{g} = \sum_{k \in K} \alpha_k \Psi_k \circ T$.

When the model is very expensive to evaluate, it is necessary to optimize the number of coefficients of the polynomial chaos expansion to be calculated. Some specific strategies have been proposed by [6] for enumerating the infinite polynomial chaos series: OpenTURNS implements the hyperbolic enumeration strategy which is inspired by the so-called sparsity-of-effects principle. This strategy states that most models are principally governed by main effects and low-order interactions. This enumeration strategy selects the multi-indices related to main effects.

The following lines illustrate the case where the model $\mathcal{G} : \mathbf{x} \mapsto x \sin x$ and where the input random vector follows a uniform distribution on $[-1, 1]$:

```
# Define the model
>>> model = NumericalMathFunction(['x'], ['x*sin(y)'])
# Create the input distribution
>>> distribution = Uniform()
# Construction of the orthonormal basis
>>> polyColl = [0.]
>>> polyColl[0] = StandardDistributionPolynomialFactory
                                (distribution)
>>> enumerateFunction = LinearEnumerateFunction(1)
>>> productBasis = OrthogonalProductPolynomialFactory
                                (polyColl, enumerateFunction)
# Truncature strategy of the multivariate orthonormal basis
# Choose all the polynomials of degree <= 4
>>> degree = 4
>>> indexMax = enumerateFunction.getStrataCumulatedCardinal(degree)
# Keep all the polynomials of degree <= 4
# which corresponds to the 5 first ones
>>> adaptiveStrategy = FixedStrategy(productBasis, indexMax)
# Evaluation strategy of the approximation coefficients
>>> samplingSize = 50
>>> experiment = MonteCarloExperiment(samplingSize)
```

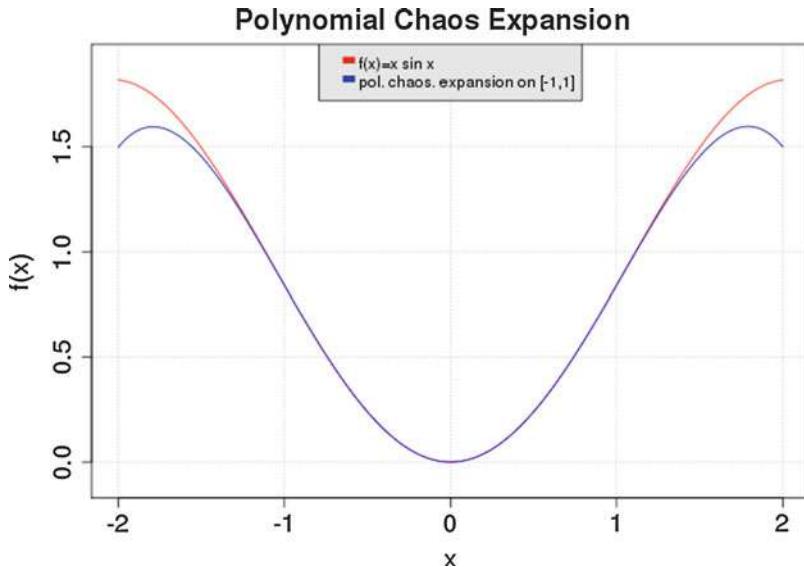


Fig. 58.18 An example of a polynomial chaos expansion: the *blue line* is the reference function $\mathcal{G} : x \mapsto x \sin x$ and the *red one* its approximation only valid on $[-1, 1]$

```
>>> projectionStrategy = LeastSquaresStrategy(experiment)
# Creation of the Functional Chaos Algorithm
>>> algo = FunctionalChaosAlgorithm(model, distribution,
                                     adaptiveStrategy, ..., projectionStrategy)
>>> algo.run()
# Get the result
>>> functionalChaosResult = algo.getResult()
>>> metamodel = functionalChaosResult.getMetaModel()
```

Figure 58.18 illustrates the result.

5.2 The Kriging Approximation

Kriging (also known as Gaussian process regression) [24, 30, 33, 37] is a Bayesian technique that aims at approximating functions (most often in order to surrogate them because they are expensive to evaluate). In the following, it is assumed that the aim is to surrogate a scalar-valued model $G : x \mapsto y$. Note the OpenTURNS implementation of Kriging can deal with vector-valued functions ($G : x \mapsto y$), with simple loops over each output. It is also assumed that the model was run over a design of experiments in order to produce a set of observations gathered in the following dataset: $((x^i, y^i), i = 1, \dots, n)$. Ultimately Kriging aims at producing a predictor (also known as a response surface or metamodel) denoted as \tilde{G} .

It is assumed that the model G is a realization of the normal process $Y : \Omega \times \mathbb{R}^d \rightarrow \mathbb{R}$ defined by:

$$Y(\omega, \mathbf{x}) = m(\mathbf{x}) + Z(\omega, \mathbf{x}) \quad (58.12)$$

where $m(\mathbf{x})$ is the trend and $Z(\mathbf{x})$ is a zero-mean Gaussian process with a covariance function $c_\theta : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ which depends on the vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$:

$$\mathbb{E}[Z(\mathbf{x}), Z(\mathbf{y})] = c_\theta(\mathbf{x}, \mathbf{y}) \quad (58.13)$$

The trend is generally taken equal to the generalized linear model:

$$m(\mathbf{x}) = (\mathbf{f}(\mathbf{x}))^t \boldsymbol{\beta} \quad (58.14)$$

where $(\mathbf{f}(\mathbf{x}))^t = (f_1, \dots, f_p)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$. Then, the Kriging method approximates the model f by the mean of the Y given that:

$$Y(\omega, \mathbf{x}^{(i)}) = y^{(i)} \quad \forall i = 1, \dots, n \quad (58.15)$$

The Kriging metamodel \tilde{G} of G writes:

$$\tilde{G}(\mathbf{x}) = \mathbb{E}[Y(\omega, \mathbf{x}) | Y(\omega, \mathbf{x}^{(i)}) = y^{(i)}, \forall i = 1, \dots, n] \quad (58.16)$$

The metamodel is then defined by:

$$\tilde{G}(\mathbf{x}) = (\mathbf{f}(\mathbf{x}))^t \tilde{\boldsymbol{\beta}} + (\mathbf{c}_\theta(\mathbf{x}))^t C_\theta^{-1} (\mathbf{y} - F \tilde{\boldsymbol{\beta}}) \quad (58.17)$$

where $\tilde{\boldsymbol{\beta}}$ is the least squares estimator for $\boldsymbol{\beta}$ defined by:

$$\tilde{\boldsymbol{\beta}} = (F^t C_\theta^{-1} F)^{-1} F^t C_\theta^{-1} \mathbf{y} \quad (58.18)$$

and $C_\theta = [c_\theta(x_i, x_j)]_{i,j=1\dots n}$, $F = [f(x_i)^t]_{i=1\dots n}$ and $\mathbf{c}_\theta^t(\mathbf{x}) = [c_\theta(x, x_i)]_{i=1\dots n}$. The line command writes:

```
>>> algo = KrigingAlgorithm(inputSample, outputSample, basis,
                           covarianceModel)
>>> algo.run()
>>> result = algo.getResult()
>>> metamodel = result.getMetaModel()
>>> graph = metamodel.draw()
```

Figure 58.19 approximates the previously defined model $\mathcal{G} : \mathbf{x} \mapsto x \sin x$ with a realization of a Gaussian process based on six observations.

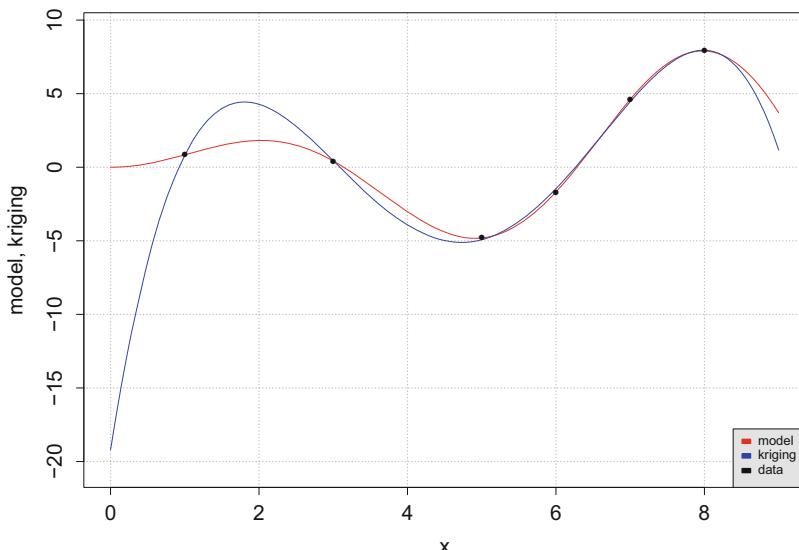


Fig. 58.19 An example of Kriging approximation based on six observations: the *blue line* is the reference function $G : x \mapsto x \sin x$ and the *red one* its approximation by a realization of a Gaussian process

6 The External Simulator

6.1 Fast Evaluations of G

On the practical side, the OpenTURNS software provides features which make the connection to the simulator G easy and make its evaluation generally fast. Within the OpenTURNS framework, the method to connect to G is called “wrapping.”

In the simplest situations, the function G is analytical and the formulas can be provided to OpenTURNS with a character string. Here, the Muparser C++ library [4] is used to evaluate the value of the mathematical function. In this case, the evaluation of G by OpenTURNS is quite fast.

In the following Python script, consider the function $G : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, where $G_1(\mathbf{x}) = x_1 + x_2 + x_3$ and $G_2(\mathbf{x}) = x_1 - x_2 x_3$, for any real numbers x_1 , x_2 , and x_3 . The input argument of the NumericalMathFunction class is a Python tuple, where the first item describes the three input variables, the second item describes the two output variables, and the last item describes the two functions G_1 and G_2 .

```
>>>G = NumericalMathFunction(
    ("x0", "x1", "x2"),
    ("y0", "y1"),
    ("x0+x1+x2", "x0-x1*x2"))
```

Once created, the function G can be used as a regular Python function or can be passed as an input argument of other OpenTURNS classes.

In most cases, the function G is provided as a Python function, which can be connected to OpenTURNS with the `PythonFunction` class. This task is easy (for those who are familiar with this language) and allows the scientific packages already available in Python to be combined. For example, if the computational code uses XML files on input or output, it is easy to make use of the XML features of Python (e.g., the `minidom` package). Moreover, if the function evaluation can be vectorized (e.g., with the `numpy` package), then the `func_sample` option of the `PythonFunction` class can improve the performance a lot.

The following Python script creates the function G associated with the flooding model. The `flood` function is first defined with the `def` Python statement. This function takes the variable X as input argument, which is an array with four components, Q , K_s , Z_v , and Z_m , which correspond to the input random variables in the model. The body of the `flood` function is a regular Python script, so that all Python functions can be used at this point (e.g., the `numpy` or `scipy` functions). The last statement of the function returns the overflow S . Then the `PythonFunction` class is used in order to convert this Python function into an object that OpenTURNS can use. This class takes as input arguments the number of input variables (in this case, 4), the number of outputs (in this case, 1), and the function and returns the object G .

```
>>>from openturns import PythonFunction
>>>def flood(X) :
    L = 5.0e3;          B = 300.0
    Q, K_s, Z_v, Z_m = X
    alpha = (Z_m - Z_v)/L
    H = (Q/(K_s*B*sqrt(alpha)))**0.6
    return H
>>>G = PythonFunction(4, 1, flood)
```

If, as many of the computational codes commonly used, the data exchange is based on text files, OpenTURNS provides a component (`coupling_tools`) which is able to read and write structured text files based, for example, on line indices and perhaps containing tables (using line and column indices). Moreover, OpenTURNS provides a component which can evaluate such a Python function using the multi-thread capabilities that most computers have.

Finally, when the computational code G is provided as a C or Fortran library, OpenTURNS provides a generic component to exchange data by memory, which is much faster than with files. In this case, the evaluation of G is automatically multi-thread. This component can be configured by Python, based on a XML file. If this method is not flexible enough, then the connection can be done with the C++ library.

The previous techniques are documented in the OpenTURNS developer's guide [1].

Figure 58.20 compares the performance of three methods to connect to the function G : the `PythonFunction` class, the `PythonFunction` class with

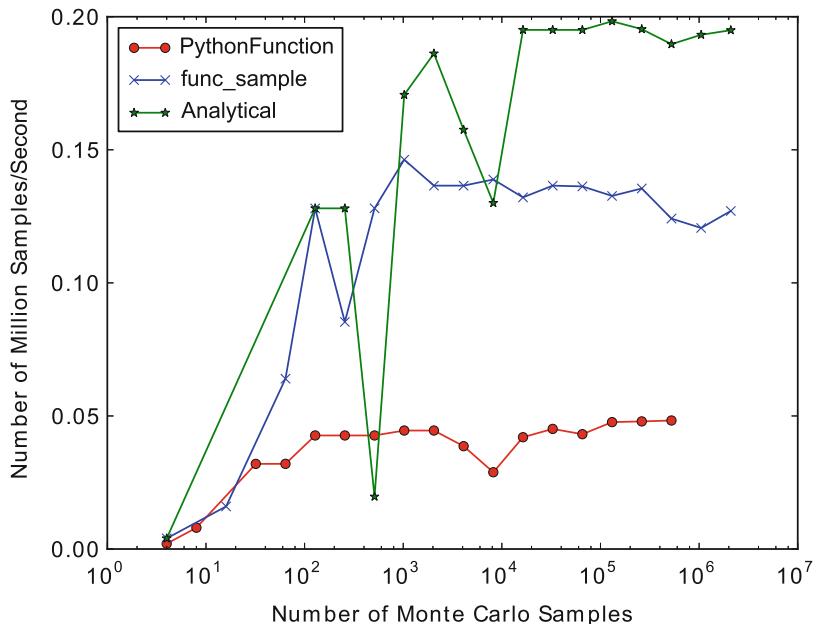


Fig. 58.20 Performance of various connection methods in OpenTURNS

the `func_sample` option, and the analytical function. This test was performed with a basic MS Windows laptop computer. Obviously, the fastest method is the analytical function, which can provide as many as 0.2 million samples per second, a performance which is four times the performance of the `PythonFunction` class.

6.2 Evaluation of the Derivatives

When the algorithm involves an optimization step (e.g., in the FORM-SORM method) or a local approximation of G (e.g., in the Taylor development used to approximate the expectation and variance), the derivatives of G are required.

When the computer code can compute the gradient and Hessian matrix of G , this information can be used by OpenTURNS. This happens sometimes, for example, when the computer code has been differentiated with automatic differentiation methods, such as forward or adjoint techniques.

In the case where the function is analytical and is provided as a character string, OpenTURNS is able to compute the exact derivatives of G . In order to do this, the software uses the Ev3 C++ library [22] to perform the symbolic computation of the derivatives and MuParser [4] to evaluate it.

In most common situations, however, the code does not compute its derivatives. In this case, OpenTURNS provides a method to compute the derivatives based on

finite difference formulas. By default, a centered finite difference formula for the gradient and a centered formula for the Hessian matrix are used.

6.3 High-Performance Computing

For most common engineering practices, OpenTURNS can evaluate G with the multi-thread capabilities of most laptop and scientific workstations. However, when the evaluation of G is more CPU consuming or when the number of evaluations required is larger, these features are not sufficient by themselves, and it is necessary to use a high-performance computer such as the Zumbrota, Athos, or Ivanhoe supercomputers available at EDF R&D which have from 16,000 to 65,000 cores [40].

In this case, two solutions are commonly used. The first one is to use a feature which can execute a Python function on remote processors, connected on the network with ssh. Here, the data flow is based on files, located in automatically generated directories, which prevents the loss of intermediate data. This feature (`DistributedPythonFunction`) allows each remote processor to use its multi-thread capabilities, providing two different levels of parallel computing.

The second solution is to use the OpenTURNS component integrated in the Salome platform. This component, along with a graphical user interface, called “Eficas,” makes use of a software, called “YACS,” which can call a Python script. The YACS module allows calculation schemes in Salome to be built, edited, and executed. It provides both a graphical user interface to chain the computations by linking the inputs and outputs of computer codes and then to execute these computations on remote machines.

Several studies have been conducted at EDF based on the OpenTURNS component of Salome. For example, an uncertainty propagation study (the thermal evaluation of the storage of high-level nuclear waste) was making use of a computer code where one single run required approximately 10 min on the 8 cores of a workstation (with shared memory). Within Salome, the OpenTURNS simulation involving 6000 unitary evaluations of the function G required 8000 CPU hours on 32 nodes [3].

7 Conclusions

This educational example has shown a number of questions and problems that can be addressed by UQ methods, uncertainty quantification, central tendency evaluation, excess probability assessment, and sensitivity analysis, that can require the use of a metamodel.

Different numerical methods have been used for solving these three classes of problems, leading substantially to the same (or very similar) results. In the industrial practice of UQ, the main issue (which actually motivates the choice of one mathematical method instead of another) is the computational budget, which is actually given by the number of allowed runs of the deterministic model G .

When the computer code implementing G is computationally expensive, one needs specifically designed mathematical and software tools.

OpenTURNS is specially intended to meet these issues: (i) it includes a set of efficient mathematical methods for UQ and (ii) it can be easily connected to any external black box model G . Thanks to these two main features, OpenTURNS is a software that can address many different physics problems and thus help to solve industrial problems. From this perspective, the partnership around OpenTURNS focuses efforts on the integration of the most efficient and innovative methods required by the industrial applications that takes into account both the need of genericity and of ease to communicate. The main projects for 2015 concern the improvement of the Kriging implementation to integrate some very smart methods of optimization. Around this theme, some other classical optimization methods will also be generalized or newly implemented.

A growing need for model exploration and analysis of uncertainty problem in industrial applications is to better visualize the information provided by such a volume of data. In this area, specific visualization software, such as ParaView, can provide very efficient and interactive features. Taking the advantage of the integration of OpenTURNS in the Salome platform, EDF is working on a better link between the ParaView module in Salome (called ParaVIS) and the uncertainty analysis with OpenTURNS: in 2012, functional boxplot [13] has been implemented. Some recent work around in situ visualization for uncertainty analysis should also be developed and implemented and so benefit very computationally expensive model physics that generate an extremely high volume of data.

Part of this work has been backed by French National Research Agency (ANR) through the Chorus project (no. ANR-13-MONU-0005-08). We are grateful to the OpenTURNS Consortium members. We also thank Regis Lebrun, Mathieu Couplet, and Merlin Keller for their help.

References

1. Airbus, EDF, Phimeca: Developer's guide, OpenTURNS 1.4 (2014). <http://openturns.org>
2. Au, S., Beck, J.L.: Estimation of small failure probabilities in high dimensions by subset simulation. *Probab. Eng. Mech.* **16**, 263–277 (2001)
3. Barate, R.: Calcul haute performance avec OpenTURNS, workshop du GdR MASCOT-NUM, Quantification d'incertitude et calcul intensif. <http://www.gdr-mascotnum.fr/media/openturns-hpc-2013-03-28.pdf> (2013)
4. Berg, I.: muparser, <http://muparser.beltoforion.de>, fast Math Parser Library (2014)
5. Berger, J. (ed.): Statistical Decision Theory and Bayesian Analysis. Springer, New York (1985)
6. Blatman, G.: Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis. PhD thesis, Clermont University (2009)
7. Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Comput. Phys.* **230**, 2345–2367 (2011)
8. Butucea, C., Delmas, J., Dutfoy, A., Fischer, R.: Maximum entropy copula with given diagonal section. *J. Multivar. Anal.* **137**, 61–81 (2015)
9. Ditlevsen, O., Madsen, H.: Structural Reliability Methods. Wiley, Chichester/New York (1996)

10. Dutfoy, A., Dutka-Malen, I., Pasanisi, A., Lebrun, R., Mangeant, F., Gupta, J.S., Pendola, M., Yalamas, T.: OpenTURNS, an open source initiative to treat uncertainties, Risks'N statistics in a structured industrial approach. In: Proceedings of 41èmes Journées de Statistique, Bordeaux (2009)
11. Fang, K.T., Li, R., Sudjianto, A.: Design and Modeling for Computer Experiments. Chapman & Hall/CRC, Boca Raton (2006)
12. gum08: JCGM 100-2008 – Evaluation of measurement data – guide to the expression of uncertainty in measurement. JCGM (2008)
13. Hyndman, R., Shang, H.: Rainbow plots, bagplots, and boxplots for functional data. *J. Comput. Graph. Stat.* **19**, 29–45 (2010)
14. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Meloni, C., Dellino, G. (eds.) *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Springer, New York (2015)
15. Kurowicka, D., Cooke, R.: *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley, Chichester/Hoboken (2006)
16. Lebrun, R., Dutfoy, A.: Do rosenblatt and nataf isoprobabilistic transformations really differ? *Probab. Eng. Mech.* **24**, 577–584 (2009)
17. Lebrun, R., Dutfoy, A.: A generalization of the nataf transformation to distributions with elliptical copula. *Probab. Eng. Mech.* **24**, 172–178 (2009)
18. Lebrun, R., Dutfoy, A.: An innovating analysis of the nataf transformation from the viewpoint of copula. *Probab. Eng. Mech.* **24**, 312–320 (2009)
19. Lebrun, R., Dutfoy, A.: A practical approach to dependence modelling using copulas. *J. Risk Reliab.* **223**(04), 347–361 (2009)
20. Lebrun, R., Dutfoy, A.: Copulas for order statistics with prescribed margins. *J. Multivar. Anal.* **128**, 120–133 (2014)
21. Lemaire, M.: *Structural Reliability*. Wiley, Hoboken (2009)
22. Liberty, L.: Ev3: a library for symbolic computation in c++ using n-ary trees, <http://www.lix.polytechnique.fr/~liberti/Ev3.pdf> (2003)
23. Marin, J.M., Robert, C. (eds.): *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer, New York (2007)
24. Marrel, A., Iooss, B., Van Dorpe, F., Volkova, E.: An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput. Stat. Data Anal.* **52**, 4731–4744 (2008)
25. Munoz-Zuniga, M., Garnier, J., Remy, E.: Adaptive directional stratification for controlled estimation of the probability of a rare event. *Reliab. Eng. Syst. Saf.* **96**, 1691–1712 (2011)
26. Nash, S.: A survey of truncated-newton methods. *J. Comput. Appl. Math.* **124**, 45–59 (2000)
27. OPEN CASCADE S.: Salome: the open source integration platform for numerical simulation. <http://www.salome-platform.org> (2006)
28. Pasanisi, A.: Uncertainty analysis and decision-aid: methodological, technical and managerial contributions to engineering and R&D studies. Habilitation Thesis of Université de Technologie de Compiègne, France <https://tel.archives-ouvertes.fr/tel-01002915> (2014)
29. Pasanisi, A., Dutfoy, A.: An industrial viewpoint on uncertainty quantification in simulation: stakes, methods, tools, examples. In: Dienstfrey, A., Boisvert, R. (eds.) *Uncertainty Quantification in Scientific Computing – 10th IFIP WG 2.5 Working Conference, WoCoUQ 2011, Boulder, 1–4 Aug 2011. IFIP Advances in Information and Communication Technology*, vol. 377, pp. 27–45. Springer, Berlin (2012)
30. Rasmussen, C., Williams, C., Dietterich, T.: *Gaussian Processes for Machine Learning*. MIT, Cambridge (2006)
31. Robert, C.P., Casella, G.: *Monte Carlo Statistical Methods*. Springer, New York (2004)
32. Rubinstein, R.: *Simulation and the Monte-Carlo Methods*. Wiley, New York (1981)
33. Sacks, J., Welch, W., Mitchell, T., Wynn, H.: Design and analysis of computer experiments. *Stat. Sci.* **4**, 409–435 (1989)
34. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* **145**, 280–297 (2002)

35. Saltelli, A., Tarantola, S., Chan, K.: A quantitative, model-independent method for global sensitivity analysis of model output. *Technometrics* **41**, 39–56 (1999)
36. Saltelli, A., Chan, K., Scott, E. (eds.): *Sensitivity Analysis*. Wiley Series in Probability and Statistics. Wiley, Chichester/New York (2000)
37. Santner, T., Williams, B., Notz, W.: *The Design and Analysis of Computer Experiments*. Springer, New York (2003)
38. Sudret, B.: Global sensitivity analysis using polynomial chaos expansion. *Reliab. Eng. Syst. Saf.* **93**, 964–979 (2008)
39. Tarantola, A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia (2005)
40. Top 500 Supercomputer Sites: Zumbrota <http://www.top500.org/system/177726>, BlueGene/Q, Power BQC 16C 1.60GHz, Custom (2014)

Index

A

- Adaptive computations, 665–668
Adaptive cross approximation algorithm, 877
Adaptive learning, 1481, 1491
Adaptive methods, 1691
Adaptivity tools, 653–659
Adjoint model, 1126–1129
Advanced computational method, 1957
Advanced mean value (AMV), 1737, 1752, 1753, 1761
Advanced simulation and computing (ASC) program, 1590
Aggregated indices, 1345–1348, 1353
Akaike information criterion (AIC), 683
Aleatory-epistemic, 1696, 1699, 1714, 1716
Aleatory uncertainty, 1371, 1505, 1912, 1927, 1943, 1947, 1949, 1964–1970, 1572
characterization of, 1515
effects of, 1518–1530
representation of, 1506–1508
Algebraic system, applications, 49–56
Algorithmic structure, 44–48
Allowable stress design (ASD), 1548
Ambiguity, 1367
Anisotropic enrichment, 655–656
A-posteriori error estimates, 957–959, 975
Applications, UQ, 1434–1435
Approximate Bayesian computation (ABC), 788, 797, 817
methods, 36, 43–44
A-priori error estimate, 957–959, 962
Askey Scheme, 830–831
Asymptotic random fluctuations, in one dimension, 500–502
Automatic differentiation, 1132
tools, 1132, 1139, 1140

B

- Basis function, 206, 210, 211
Basis Pursuit Denoising (BPDN), 834
Bayes factor, 442–444
Bayes' formula, 315, 316, 343
Bayesian approach, 195, 560
Bayesian calibration, 1624–1628, 1658, 1684, 1686, 1687, 1690
Bayesian compressive sensing (BCS), 687
Bayesian computational method, 1948
Bayesian formulation, 200–201
Bayesian inference, 788, 790, 808, 811, 813, 816, 1723, 1825
adversarial model error and Bayesian error, 174
complete class theorem, 175–177
methods, 34, 36–39
problem, 1870
statistical error, 173–174
QUESO (*see* Quantification of Uncertainty for Estimation, Simulation and Optimization, (QUESO))
Bayesian information criterion (BIC), 683
Bayesian inversion, 313, 314–316, 941, 979, 980, 986
common features, 349–364
continuous time Markov processes, 379–380
elliptic inverse problem, 319–320, 345–349
finite dimensional Langevin equation, 380–384
infinite dimensional Langevin equation, 384–392
inverse heat equation, 316–318
Metropolis-Hastings methods, 367–371
posterior distribution, 340–349

- Bayesian inversion (*cont.*)
 prior modeling
 Besov prior, 324–330
 Gaussian prior, 330–335
 i.i.d. random sequence, 321
 i.i.d. sequences, 321
 mean function, 321
 random function, 321
 uniform prior, 322–324
 random field perspective, 335–338
 sequential Monte Carlo methods, 371–379
- Bayesian method, 892, 893, 913
- Bayesian networks (BN), 431, 432, 451, 453, 456, 458, 4140
 for cardiovascular disease, 1470–1471
 causal mechanisms, 1474–1475
 causal *vs.* noncausal BNs, 1473–1474
 descriptive analytics, 1488
 evaluation and learning analytics, 1489
 posterior inference in, 1475–1477
 predictive analytics, 1488
 probabilistic dependencies, 1471–1473
 risk management decision-making, 1482–1485
 structure discovery problem, 1477
 uncertainty in, 1479–1482
 VOI, DBN and sequential experiments, 1485–1488
- Bayesian regression, 681–682, 684
- Bayesian robustness, 180
- Bayesian surrogate, 560
- Bayes linear, 10, 14
- Bayes' rule, 41, 180
- Bayes' sampling distribution, 183
- Bayes' theorem, 315, 340, 342–343, 349, 1079, 1372
- BBGKY hierarchies, 1044–1045
- Best approximation, 738, 741, 742
- Best rank-m approximation, 860
- Beta prior probability distribution, 1374
- BHM, 195, 201, 208, 209, 213
- Bias correction, 71
- Bioterrorism Risk Assessment (BTRA) model, 1414
- Black-box, 1655, 1662, 1664
- Block indices, 1345, 1348–1351
- Borel-Kolmogorov paradox, 179–180
- Borel probability, 160, 161
- Borel structure, 170
- Bounded orthonormal systems, 837–838
- Bounded random potentials perturbation theory for, 506–509
- Brownian motion, 502, 524, 1039
- C**
- C++
 library, 2001, 2003
 templates, 1795, 1804
- Calibration, 22, 23, 71, 73, 440–441, 448, 451–452, 458, 458, 1651
 parameters, 73–74
- Cameron-Martin norm, 415
- Cameron-Martin space, 415, 417
- Cameron-Martin theorem, 344, 416, 423, 526
- Causal analytics, 1453, 1488–1491
- Causal graph, 1453, 1477, 1479, 1481, 1490, 1492, 1493
- Causal laws, 1442, 1465
- Cavity flow problem, 846–851
- Čebyšev inequalities, 164
- Change of support, 1349–1351
- Change-point analysis (CPA), 1449, 1452, 1456, 1489
- Chemical system parameter estimation in, 56–65
- Clairvoyant's information, 1393
- Classical group screening, 1159
- Closure problem, 1050–1057
- CLT method, 506
- Coarse-grained dynamics in phase space, 1046
- Coarse-grained models of particle system, 1065
- Coarse graining particle systems, 1047
- Coarsening
 procedure, 654–655
 thresholding error, 654
- Code scaling assessment and uncertainty (CSAU), 1593, 1620
- Coefficient of variation, 1080
- Colored noise, 765, 766
- Column weighting, 842–843
- Common structural rules (CSR), 1571
- Composite indicator(s), 1188–1214
 construction of, 1188
 estimation main effects, 1196
 importance measures, 1192–1195
 local polynomial regression, 1198–1199
 penalised splines, 1197–1198
 polynomial regression, 1196–1197
 sensitivity analysis, 1195
 transformations and weighting, 1191–1192
- Composite materials, 931
- Compressed sensing, 1003–1005, 1009–1011, 1017, 1019–1021, 1024–1026
- Computational Markov chains, 1858
- Computational method
 advanced, 1957
 Bayesian, 1948

- Computer emulator, 558–559
Computer experiments, 298, 1108, 1109
Computer models, 1980–1982
Computer simulators, 10, 12–13
Consistency, 1366
Contraction principle, 504–505
Convergence and computational time analysis, 667–670
Convex optimization, 797–801, 809, 820
Corrosion approach, 1575
COSSAN project, 1916, 1919
COSSAN-X, 1920, 1937
COSSAN-X high performance computing, 1933, 1934, 1935
Counterfactual analyses, 1461
Counter-terrorism models, 1408
Covariance function, 85, 336, 338, 420
Covariance operator, 330, 332, 336–339, 367, 405, 406, 410, 415–418, 421
Cox’s theorem, 1365, 1366
Credibility, 523
Credibility assessment scale (CAS), 1594
Cross validation, 1709, 1710
Cumulative distribution function (CDF), 770–771
Curse of dimensionality, 685, 859
CUSUNORO, 1995
Cylindrical Wiener process, 419
Cynefin model, 1425
- D**
- DAG model, 1455. *See also* Directed acyclic graph (DAG)
Dakota software, 1652–1661
Data free inference (DFI), 36, 38, 48
Data map, 170
Data workflow SIMLAB, 1982–1983
Decision analysis, 1377
 hurricane modification, 1384
Decision analysis, 1379
Decision theory, 177, 181
Defender event trees, 1410
Defense Modeling and Simulation Organization (DMSO), 1404
Density estimation, 797, 809, 819
Department of Homeland Security (DHS), 1411, 1414
Derivative
 defined, 1124
 Gâteau, 1126–1128
 Gâteaux, 1127
 partial, 1126, 1130
- Derivative based global sensitivity measures (DGSM), 1242–1261
groups of variables, 1253–1255
Morris method, 1244–1245
normally distributed variables, 1251–1253
randomly distributed variables, 1245
uniformly distributed variables, 1245, 1249–1251
upper bounds in general case, 1255–1257
- Derivative-based methods, 1124
Desideratum, 1373, 1376
Design of experiments (DoE), 296
 methods, 1430
Detail spaces, 643–645
Deterministic sampling method, 701
Di Finetti’s theorem, 1371
Dimension reduction, 1335, 1337, 1340, 1696
Dirac masses, 162
Directed acyclic graph (DAG), 1455, 1467, 1469, 1470, 1473, 1477, 1478, 1487
Directional enrichment criteria, 658–659
Discrepancy functions, 73, 89–90
 posterior distributions, 80
Discrepancy model spectral stochastic, 993–1034
Discrepancy modeling, 1723, 1725, 1728
Distanced-based dissimilarity measure, 1339
Distributionally robust optimization (DRO), 164
Dittus-Boelter correlation, 1627
Domain decomposition approach, 638
Duffing equation, 1040
Dynamical system, 558, 582–583
Dynamic Bayesian networks (DBNs), 1474, 1486, 1488, 1489
- E**
- Effective algorithm for computing sensitivity indices (EASI), 1232, 1235, 1995
Effective dimension g-Sobol’ function and, 1232–1234
Effective medium models application to, 515
Efficient Global Reliability Analysis (EGRA) method, 1737, 1763
Efron-Stein inequalities, 515
Elliptic equations stochasticity for, 499–500
Embedded model, 146, 152
Embedded sampling methods, 1773, 1804
Empirical interpolation method, 863, 970–972
Emulation, 10, 14–20
Emulator adequacy, 1887–1890
Engineering design, 1422

- Enhanced stochastic evolutionary algorithm, 290, 293
- Ensemble learning algorithms, 1463, 1490
- Entropic inference, 39–42
- Entropy, 1376
- Epistemic uncertainty, 1529, 1572, 1912, 1932, 1943–1945, 1948, 1949, 1964–1970
applies, 1371
effects of, 1513–1517
LHS and propagation of, 1517–1518
representation of, 1509–1513
- Epistemic UQ, 1683
- Epistemic versus aleatory uncertainties, 1370–1372
- Error estimation, 1295–1298
- Estimation, 2027
failure probability, 2019
statistics, 2012–2013
- Evaluation analytics, 1489
- Exchangeable sequences, 1371
- Experimental uncertainty, 71, 73
- Expert judgment, 1368
- Expert opinion distributions, 1628
- F**
- Factorial group screening, 1159–1161
- Factorial screening design, 1148, 1159
issues with, 1157
- Factors definition pane, 1987
- Factor sparsity, 1144
- FANOVA, 1218–1219, 1225, 1228
- FAST and Extended FAST design, 1987
- Fatigue
crack propagation, 1552
definition, 1542, 1552
design guidelines and specifications, 1552
fracture mechanics approach, 1553–1557
- Federal Air Marshals Service (FAMS), 1413
- Feldman-Hájek Theorem, 349, 418, 423
- Fernique theorem, 415
- Financial Secrecy Index (FSI), 1191, 1204–1209
- Fine-grained parallelism, 1771, 1773, 1774
- Finite element (FE) method, 1547
- Finite element analysis (FEA), 1734, 1748, 1753, 1756, 1761
- Finite element model (FEM), 76, 1953
- Finite variance, 404, 406
- First-order sensitivities, 690
- Fixed-TM preposterior analysis, 104
- Fokker-Planck equation (FPE), 765, 766, 780
- Forecasting, 10, 28–29
- Forward problem, 1834
- Fourier amplitude sensitivity test (FAST), 1229–1232
- Fourier transform, 405
- Fractional Brownian motion, 1060
- Fractional factorial design
iterated, 1162–1163
nonregular, 1151–1154
regular, 1149–1151
- Fragmented uncertainty approach, 1568
- Frobenius norm, 418
- G**
- Galerkin method, 700
- Galerkin projection(s), 621, 622, 863–864
scheme, 525
- Gamma-Laguerre PCEs, 619
- Gartner-Ellis, 504
- Gâteau derivative, 1126–1128
- Gauss-Dantzig Selector, 1170
- Gauss-Hermite PCE type, 619
- Gaussian probability measure, 415
- Gaussian process (GP), 74, 81, 478, 479, 501, 568–569, 638, 1291, 1302, 1737, 1738, 1741, 1758, 1760, 1763
analytic formulae, 1309
 γ -exponential covariance function, 1307
group screening for, 1163
main effects visualization, 1308
model, 478, 480, 483, 492, 1147–1148, 1869, 1874–1876, 1893–1894
Monte Carlo integration, 1310
Monte-Carlo integration, 1310
predictive distribution, 1303–1305
sensitivity analysis, 1308
sequential design, 1305
Sobol' indices, 1310–1312
squared exponential covariance
function, 1306
 v -Matérn covariance function, 1306–1307
variable selection for, 1170–1171
- Gaussian process models for simulation
analysis (GPMsA), 1867–1907
analysis problem, 1868–1870
core functions of, 1870
data preparation, 1876–1882, 1894–1897
in Matlab environment, 1870
parameter in, 1906
statistical model, 1871–1873
- Gaussian process regression, 561–568.
See also Gaussian processes (GPs)
- Gaussian random field, 336
- Gauss-Markov theorem, 166

- Generalized linear model, 563
Generalized polynomial chaos (gPC), 526, 556, 564
Generation method definition, 1988
Generator, 223, 225, 226, 233, 236, 238, 240, 894, 905, 906, 910, 916–919, 926–929
Genericity, 2001, 2003, 2036
Geophysical applications, VSA, 1135
Glaciology, 1124, 1138
Global sensitivity analysis (SA), 690–693, 1218, 1242, 1266, 1700, 1717, 1817–1818
Global surrogates, 676
Global terrorism database (GTD), 1412
Good country index (GCI), 1191, 1209–1212
Gradient code validation, 1133–1134
Gradient computation, 1130–1134
Granger causality tests, 1459–1460
Graphical user interface (GUI), 1631, 1980
Greedy algorithm, 866, 961–963
Green's function, 336
G-Sobol' function, 1314–1316
and effective dimension, 1232, 1234
- H**
Hellinger distance, 313, 349, 351, 353, 354, 357, 358, 364, 386, 406–411, 423
Heterogeneous microstructure, 931
Hierarchical basis, 747–749, 752, 754–758
Hierarchical methods, 610
High-dimensional approximation, 729, 750
High-dimensional model representation (HDMR), 686
High dimension, random field. *See* Random field
High fidelity approximation, 956–958
High performance computing, 1916, 1933, 1971
COSSAN-X, 1933–1935
and data management, 1932–1935
High stochastic dimension, random field.
See Random field
Hilbert scale, 317, 330, 344, 396–398, 400, 401
Hilbert space, 184, 422
Hilbert-Schmidt norm, 418
Hilbert-Schmidt operator, 533
Hilbertian basis, 640
History matching, 10, 19, 22–26
Hölder continuous function, 396
Homogenization, 505
Homogenization effect, 1041
Homogenization theory, for large random potentials, 510–512
Human computers, 159
Human teams, 159
Hydrodynamic experimental community, 1583
Hyperparameters, 74
- I**
Identifiability, 71, 81–85
improvement, 94
lack of, 73–76, 79
multiple responses, 93–95
posterior distribution, 75
single responses, 91
Identification, 225, 239, 276, 887–889, 892, 906–908, 912–914
i.i.d. sequence, 321, 339, 402–403, 413
Impact assessment, 1116, 1117
Importance sampling (IS), 1081–1084, 1092, 1657, 1662, 1670, 1671, 1689
density, 1082
weight, 1082
Importance sampling using elementary events (ISEE), 1083
Imprecise probability, 1914, 1920, 1967
Incremental analysis, 1549
Inference given summary statistics, 33–66
Infinite-dimensional inverse problems
likelihood, 1851
posterior, 1852
prior, 1849–1851
Influence diagram (ID), 1482–1485
Information based complexity, 181
Institute of Electrical and Electronics Engineers (IEEE), 1404
Integro-difference equation (IDE), 203, 206
Intelligent adversary models, 1408–1409
comparative analysis, 1410
components, 1409
experimental data, 1411–1412
historical data, 1412–1413
proxy model, 1413
risk assessment, 1409
sensitivity analysis, 1410
simulation validation, 1410–1411
Stackelberg games, 1414
subject matter expert, 1414–1415
use of analogy, 1409
use of uncertainty, 1409
Internal discrepancy, 20–22, 24, 26, 28
Interrupted time series analysis, 1456, 1466
Interval estimation, 1676
Intervention analysis, 1456–1459, 1491

Intrusive spectral projection (ISP), 618, 629
 Intrusive UQ, 632
 Inverse problem(s), 225, 888, 1834–1835
 Bayesian approach. *See* Bayesian inversion definition, 1834
 infinite-dimensional inverse problems.
 see Infinite-dimensional inverse problems
 statistical inverse problem,
 1840–1841, 1847
 Ishigami function, 1312–1314
 Iterative solution algorithms, 605

J

Joint distributions, 1629
 Joint metamodeling, 1338–1339, 1341

K

Karhunen-Loëve approach, 574–575
 Karhunen-Loëve Expansion (KLE), 525, 533,
 534, 572, 621
 Kendall's equivalence, 162
 Kernel Density Estimation (KDE), 629
 Kernel smoothing, 1198
 Key-uncertainty drivers, 1266
 Knothe-Rosenblatt map, 790, 792, 798,
 809, 812
 Kolmogorov
 axioms, 1365
 continuity criterion, 324, 329, 333,
 338–340, 412–414, 417, 420–423
 extension theorem, 340, 403, 412, 419, 422
 Kolmogorov-Smirnov test, 1994
 Kriging, 479–481, 483
 conditioning method, 1310, 1311
 Kronecker separable design, 1901–1903
 Kullback-Leibler divergence, 406, 410, 684

L

Lack of identifiability, 73–76, 79
 Lagrange interpolation approach, 703
 Lagrange polynomial interpolation, 675
 Laguerre polynomials, 524
 Langevin equation, 393, 765
 finite dimensional Langevin equation,
 380–384
 infinite dimensional Langevin equation,
 384–392
 non-Markovian, 765
 Langevin stochastic partial differential equations. *See* Langevin equation

Large deviations principle (LDP), 503
 Large random potentials homogenization theory for, 510–512
 Latent process, 201, 204–206, 210, 213
 Latin hypercube, 1985
 design, 111
 Latin hypercube sampling (LHS), 1163–1166,
 1226, 1233, 1517–1518
 Lax-Milgram theorem, 539
 Learning analytics, 1489
 Le Cam's approach, 176
 Legendre-Gauss quadrature, 78
 Legendre-Uniform PCEs, 619
 Likelihood, 315, 316, 340, 341, 344, 348, 370,
 372, 379
 definition, 347
 log, 315, 342
 Linear regression models, 1146–1147, 1147
 Linear solvers, 1802–1804
 Load and resistance factor design (LRFD),
 1542, 1548, 1571
 Local polynomial regression, 1198–1199
 Local sensitivity analysis, 1124,
 1140, 1141
 Local surrogates, 676
 Log-prior function, 1904
 logThetaPrior, 1904
 Long-range random potentials convergence to stochastic limits for, 512–514
 Lorenz-96 system, 1045
 Low-rank approximation, 859
 of multivariate functions, 869–873
 of order-two tensors, 860–862
 from samples of the function, 873–875
 Low-rank manifolds, 859
 Low-rank tensor methods for parameter-dependent equations, 875–879

M

Macro-parameter, 1334–1335, 1340, 1341
 Map labelling, 1336–1337, 1341, 1343
 Marginal distributions, 1628
 Marginal structural models (MSMs),
 1462, 1463
 Marine risk-based design methods, 1571
 Markov chain Monte Carlo (MCMC), 24, 103,
 482, 484, 485, 493, 569, 1084, 1173,
 1626, 1869
 algorithm, 46
 method, 44, 195, 200, 214, 682, 1722,
 1725, 1726, 1819–1822, 1832, 1835,
 1853, 1863, 1864
 methods and SMC methods, 372–379

- continuous time Markov processes, 379–380
Metropolis-Hastings methods, 367–371
model initialization and, 1882–1885, 1898–1899
simulation, 1556
Markov models, 1377
Mathematical software, 1833, 1854–1855
Matlab, 1916, 1959, 1961
 App, 1941
 toolbox, 1936–1941
Maximin design, 293, 308
Maximum a posteriori (MAP), 569
 estimate, 1079
 estimators, 350, 358–364
Maximum Entropy (MaxEnt)
 density, 36
 principle, 36, 253, 890, 906, 917, 925–926
Maximum likelihood estimator (MLE), 88, 569
Maximum likelihood method, 906, 908, 912
MCMC. *See* Markov chain Monte Carlo (MCMC)
Mean-based preconditioning, 608–610
Mean function, 321, 329, 330, 332, 336
Measure transport, 788, 813
MediaWiki tool, 1936
Metamodel, 1929–1931
 dimension reduction, 1337
 distance-based metamodel, 1339
 joint metamodeling, 1338–1339
 use of, 1348
Meteorology, 1124
Miner’s rule, 1558
Model derivative, 1244
Model discrepancy, 20–22, 70
Model ensembles, 1441, 1463, 1481, 1489
Model execution pane, 1990–1994
Model execution step, 1984
Modeling and simulation (M&S), 1592
Model order reduction, 859
Model reliability metric, 442, 444
Model selection, 1305–1307
Model uncertainty quantification
 formulation, 73
Model Utilization Risk Management (MURM), 1594
Model verification and validation (V&V), 1402. *See also* Verification and validation (V&V)
Modified Metropolis algorithm (MMA), 1085
Modular Bayesian approach, 74
Moment independent importance measures, 1266–1285
Monte Carlo (MC), 1980, 1985, 1986
 approach, 556
 methods, 793, 794
 sampling, 629, 701, 1221–1222, 1226–1227
Monte Carlo simulation (MCS), 98, 1078–1081, 1092, 1550, 1917, 1918, 1927, 1941, 1947, 1964
Mori-Zwanzig approach to uncertainty quantification, 1038–1066
Mori-Zwanzig equation, 1058–1060
 time-convolutionless form of, 1048–1049
Mori-Zwanzig projection operator framework, 1045–1050
Morris function, 1317–1319
Morris method, 1166–1169, 1242–1246, 1257
Multi-agent influence diagram (MAID), 1489, 1491
Multi-attribute utility (MAU), 1379
 models, 1413
Multicore architectures, 1773, 1774
Multidimensional enrichment criteria, 656–658
Multidimensional extension, 648–651
Multi-fidelity, 1679, 1680, 1683, 1691
 extensions, 1004–1012
 moments for stochastic collocation, 1011–1012
PCE with additive discrepancy, 1010–1011
PCE with multiplicative discrepancy, 1011
 sparse grid results, 1030–1031
Multigrid approach, 606
Multilevel approach, 606
Multi-level coarse-graining, 1049–1050
Multi-level system, 431–435
 with type-I interaction, 450–455
 with type-II interaction, 457–464
Multi-output Gaussian process, 570
Multi-response Gaussian process, 85
Multi-response modular Bayesian approach, 86–91
Multiresolution analysis, 640, 641
Multiresolution system (MRS) one-dimensional, 639–646
Multiresolution space, 640–641
Multiscale operators, 652
Multivariate, 197, 204–205, 209
Multiwavelet (MW)
 basis, 643–646
 expansions, 643
 mother functions, 645–646
Mutual coherence, 838–839

N

- National Research Council, 1411
Navier-Stokes equations, 1600

Neutronics, 1639
 Non-compact sets, 162
 Non-Gaussian, 221, 886, 887, 889–892,
 899, 902
 Non-Gaussian random field, 889, 912–913
 BVP, 892
 general properties, 915
 OAPSM, 906–907
 parameterized representation, 904
 Non-Gaussian Rosenblatt process, 502
 (Non) identifiability, 71
 Non-informative prior distributions, 85
 Non-intrusive spectral projection (NISP),
 618, 629
 Nonlinear analysis tools, 524
 Nonlinear AutoRegressive Moving
 Average with eXogenous input
 (NARMAX), 1043
 Non-Markovian Langevin equations, 765
 Nonparametric uncertainty, 221, 222
 Non-spatial covariance, 88
 Notch stress approach, 1560
 Nugget effect, 483, 484
 Nugget parameter, 483–484
 Numerical evaluation of stochastic structures
 under stress (NESSUS), 1738
 Abaqus model, 1749
 AMV+ method, 1737, 1752–1754, 1761
 CDF, 1736, 1737
 deterministic parameter variation, 1743
 EGRA method, 1737
 full cumulative distribution
 analysis, 1744
 global sensitivity analysis, 1744
 GP model, 1737
 graphical mapping, 1750
 Graphical user interface (GUI),
 1734–1749
 mean value probabilistic analysis, 1752
 methods and capabilities in, 1734
 problem statement, 1739, 1746, 1747, 1754
 random variable input, 1740
 response model, 1741–1742
 results visualization, 1744, 1745
 solution strategy, 1754, 1756,
 1758–1761
 specified performance levels, 1744
 specified probability levels, 1744
 system-level reliability analysis, 1736
 uncertainty propagation methods, 1735
 variable mapping, 1750
 Numerical optimization, 1702, 1720
 Numerical uncertainty, 1635
 Nyquist-Shannon sampling theorem, 1231

O

Observed Fisher information, 105
 Ocean, 195, 207–209
 Oceanography, 1124
 Ocean wave climate, 1577
 O'Hagan approach, 576–577
 One dimension
 asymptotic random fluctuations in,
 500–502
 large deviations in, 502–505
 One-dimensional multiresolution system,
 639–646
 OPENCOSSAN Matlab App, 1936–1941
 Open source, 2001, 2003
 Open-source Matlab toolbox, 1936–1941
 Open-source model, 1972
 Open source software, 1654
 OpenTURNS, 2003
 main originality, 2005–2007
 presentation, 2005
 Operational validation, 1406
 Operator overloading, 1770, 1771, 1804
 Optimal subspaces characterization, 861
 Optimal transport, 788–790, 808
 Optimal uncertainty quantification (OUQ),
 161–163
 Optimization, 1652–1655
 tools, 1927–1930
 Optimization under uncertainty, 1696, 1730
 Ordinary differential equations (ODEs), 619,
 625–631
 problem, 659–662
 Orthogonal arrays (OA), 1226, 1230, 1233
 Orthogonal matching pursuit (OMP), 835
 Orthogonal polynomials, 829

P

Parallel algorithms, 1831, 1854
 Parallel computation, 1831, 1832, 1854
 Parallel computing, 1654
 Parameter estimation, 34
 in chemical system, 56–65
 Parameter-dependent equation, 858
 low-rank tensor methods for,
 875–879
 Parameters' posterior distribution, 1885–1887,
 1899–1900
 Parameter uncertainty, 70, 73
 Parametric noise, 525
 Parametric-nonparametric uncertainties,
 275–280
 Partial correlation coefficients
 (PRCCs), 1524

- Partial derivative, 1126, 1130
Partial differential equation (PDE), 498, 558, 587–593
Partial rank correlation coefficients (PRCCs), 1531
Particle system, coarse-grained models of, 1065
Path analysis, 1467–1470
PCE-MC framework concentration inequalities and coupled, 515–517
PDF equation, 1043
Pearson's *correlation ratio*, 1189
Performance assessments (PAs)
 aleatory uncertainty (*see* Aleatory uncertainty)
 characterization of uncertainty, 1504–1506
 epistemic uncertainty, 1509–1513
 propagation and display of uncertainty, 1513–1530
 sensitivity analysis, 1530–1533
 WIPP PA, 1508
Performance indices, 1188
Perturbation theory for bounded random potentials, 506–509
Petrov-Galerkin (PG) discretization, 956
Phase transition diagrams, 835–836
Phenomena Identification and Ranking Table (PIRT), 1601–1603, 1605, 1631, 1632, 1634, 1636, 1638
Physical model, 557–558
Piecemeal approach, 1569
Plausibility, 1365
Poincaré inequality, 1255, 1256
Polynomial chaos (PC), 527–532, 604, 1766, 1768–1770, 1773, 1774, 1776–1778, 1812–1815, 1820
surrogates, 695
 Bayesian regression, 681–682
 global sensitivity analysis, 690–693
 high dimensionality, 684–687
 input PC specification, 677–679
 model selection and validation, 683–684
 moment evaluation, 688–689
 nonlinear/nonsmooth/discontinuous forward model, 688
 projection, 679–680
 non-intrusive, 997–999
 with random coefficients, 534–537
Polynomial chaos decompositions (PCE), 529, 534, 539
 adapted representations of, 537–539
Stochastic Galerkin implementation of, 539–542
Polynomial chaos expansion (PCE), 498, 499, 523, 527, 528, 529, 534, 539, 618–619, 828, 889–891, 902–903, 908–913, 1232 1662, 1664, 1665, 1667, 1668, 1679, 1680, 1686
error estimation, 1297–1298
experimental design, 1296
Hermite polynomials, 1293
information matrix, 1296
intrusive approach, 1295
least-angle regression algorithm, 1297
Legendre polynomials, 1293
mathematical setup, 1292
multivariate polynomials, 1294
non intrusive techniques, 1295
non standard variables and truncation scheme, 1294–1295
orthogonal polynomials, 1293
sobol' decomposition and indices, 1299–1300
statistical moments, 1298
univariate polynomials, 1294
Polynomial norms, 677
Polynomial order, 831–832
Polynomial regression, 1196–1197
Porous media, 931
Positive-definite random matrix, 234–241
Posterior, 313, 315, 316, 340–350, 354, 355, 365, 371, 376, 393, 406
Posterior covariance function, 76
Pout object, 1892
Precision operator, 336, 417, 418
Preconditioning, 1770, 1804
Prediction
 calibration and validation, 151–152
 embedded model, 147, 148, 152
 model error, 149–150
 operator, 653
 reliable theory, 146
 state variables, 147
 sufficiency of, 153
Predictive analytics, 1480, 1488
Predictive capability maturity model (PCMM), 1590–1611, 1614
 application specific, 1600–1601
 aspects of, 1598
 classification guidance, 1591
 calibration, 1615
 code bugs, 1615
 code verification, 1599, 1615
 computational simulation, 1600
 in credibility, 1605–1610
 decomposition of validation, 1599
 elements, 1596

- Predictive capability maturity model (PCMM)
(cont.)
- foundational, 1599–1600
 - kiviat plots for, 1592
 - low-level validation, 1600
 - neutronics, 1639
 - original, 1596–1597
 - and PIRT, 1605
 - QASPR, 1603
 - QPRT, 1631–1632
 - quality levels, 1591
 - role of, 1592
 - scores, 1640
 - software quality, 1599
 - solution verification, 1600, 1615
 - thermal hydraulics, 1639–1640
 - validation, 1615
 - variations on a theme, 1597–1599
 - validation pyramid, 1632
- Predictive maturity index (PMI), 1594
- Preposterior analysis, 98–103
- Preposterior covariance, 97, 103
- vs.* posterior covariance, 109–117
- Preposterior standard deviation, 117
- Prescriptive analytics, 1489
- Principal component analysis (PCA), 567, 865
- Principle of maximum entropy, 1079
- Prior, 313, 315, 316, 318–321, 338, 340, 343–345, 347–350, 364, 369, 370, 372, 373, 376, 393
- Besov, 321, 324–330, 338, 339, 349
 - Gaussian, 321, 330–335, 338, 348–349, 358, 360, 364, 371, 379, 393
 - knowledge, 561
 - model, 891, 906, 907, 913
 - uniform, 321–324, 334, 339, 346–349
- Probabilistic analysis, NESSUS, 1734.
- See also* Numerical evaluation of stochastic structures under stress (NESSUS)
- Probabilistic risk analysis (PRA), 1410
- Probabilistic risk assessments (PRAs), 1504, 1514
- Probability, 828, 2019
- theory, 34, 1378
- Probability density function (PDF), 34, 630, 677, 678, 682, 689, 764, 767–769
- distribution methods for, 771–780
 - nonlinear, 774–776
 - systems of, 776–780
 - with random coefficients, 773–774
 - weakly nonlinear, 772–774
- Processed data products (PDF), 51–55
- pseudo-metric, 52–55
- Product design, 1422
- Product development process (PDP), 1423
- Projection-based model order reduction methods, 859, 862–869
- Propagation and display of uncertainty, 1513–1530
- Propagation uncertainty, 2015–2023
- Propensity score, 1462
- Proper generalized decomposition, 859
- Proper orthogonal decomposition, 859, 865
- Pseudo-spectral operation, 623
- Python module, 2008, 2009, 2011
- Q**
- Quadratic nonlinearity, 203
- Quadrature based sampling, 842
- Qualification of Alternatives to the SPUR Reactor (QASPR), 1603–1610
- Qualitative validation method, 1406
- Quantification uncertainty, 2007–2015
- Quantification of Uncertainty for Estimation, Simulation and Optimization (QUESO), 1831, 1835–1839, 1843–1846, 1861–1863
- API considerations, 1863–1864
 - BaseJointPdf, 1855
 - BaseScalarFunction, 1855
 - BaseVectorRealizer, 1856
 - custom priors, 1852
 - DAKOTA, 1833
 - data post-processing and visualization, 1847
 - DRAM algorithm, 1857
 - emulators, 1863
 - exascale, 1864
 - forward problem, 1834
 - input file, 1858, 1861
 - inverse problem, 1835
 - inverse problems, 1839–1840
 - (see also* Inverse problems likelihood distribution)
- motivation, 1831
 - mpirun command, 1846
 - Multi-level Monte Carlo, 1857–1858
 - PDF objects, 1853–1854
 - pre-conditioned Crank-Nicolson proposal, 1858
 - prediction, 1835
 - PyMC, 1833
 - random variable objects, 1854
 - realizer objects, 1854

- software engineering, 1854–1855
YOURAPP, 1846
WinBUGS, 1833
- Quantified Parameter Ranking Table (QPRT),
1631–1632, 1635–1636, 1638
- Quantitative margins and uncertainty
(QMU), 1594
- Quantitative model validation, 1407
- Quantity of interest (QoI), 149–152, 828, 1616,
1620, 1623, 1631, 1632, 1843
- Quasi-experiments (QEs), 1453, 1466, 1491
- Quasi-Monte Carlo (QMC) sampling, 1224
- Quasi random, 1985
- Quasi-random balance design (QRBD), 1231
- R**
- Radioactive waste, performance assessments,
1230. *See also* Performance
assessments (PAs)
- Random Balance Design (RBD)
- Random balance design, quasi, 1231
- Random coefficients, 764
uncertainty quantification in PDEs with,
765–766
- Random elastic medium, 242
- Random field, 335
algebraic prior stochastic models, 914–931
boundary value problem, 888, 890, 892
Euclidean space, 894
exponential-type representation, 900,
902, 904
Gaussian, 336
isotropic, 336
Kronecker symbol, unit matrix and
indicator function, 894
lower-bounded random fields, 899–900
methodology, 893
non-Gaussian second-order random
field, 889
norms and usual operators, 894
polynomial chaos expansion, 902–903
probability space, mathematical
expectation, 895
random observation model, 905
sets of matrices, 894
square-type representation,
900–902, 904
stationary, 336
statistical identification, 887–888
statistical inverse problem
(*see also* Statistical inverse
problem Stiefel manifold)
Stiefel manifold, 904
- stochastic dimension, 886–887
tensor-valued random fields, 931
- Random function, 321, 322, 324, 335, 338,
339, 413
- Random matrix, 221–223, 900, 901, 913,
917–919, 922–925
- Random media, 896, 897
- Random vector, 228, 2015
Gaussian second-order random vector, 889
independent realizations, 908
MaxEnt principle, 925–926
model observation, 887
non-Gaussian second-order random vector,
890–891
probability density function, 906
probability space, mathematical
expectation, 895
stochastic dimension, 886–887
- Randomness in mathematical models, 764–765
- Rate functions, 503
- Recovery measures, 835
- Reduced basis, 859, 941
acceleration of MCMC, 982–983
compression, 958–960, 964–970
construction, 960–963
- Reduced order models (ROM) models, 1428
- Reduced rank representation, 210, 212, 214
- Reduction calculus, 178
- Redundancy
definition, 1543, 1544
load multipliers, 1544
probabilistic measure, 1544
reserve strength factor, 1543
structural component design, 1548–1552
time-variant, 1545
- Reference distribution variable selection
(RDVS), 1171
- Regression function, 87
- Relevance, 434, 435, 459
- Reliability, 1654, 1655, 1661, 1662,
1667–1670
analysis, 1952–1957
based optimization, 1938, 1959–1961,
1963–1965
importance measures, 1280
- Reliability-based approach, 1571
- Reliable theory, 146
- Replication design, 1226
- Representation of random fields, 899, 900, 918
exponential-type, 900
polynomial chaos expansion, 908–912
square-type, 900–902
- Reproducing kernel Hilbert space (RKHS),
423, 532

- Reserve strength factor, 1543
 Residual stresses, 1574, 1575
 Resource Governance Index (RGI), 1191, 1200–1205
 Response function, 161
 Response surface, 1735, 1737, 1758, 1759, 1763
 Response surface method (RSM), 1547
 Response surface methodology, 1698
 Restricted isometry constant (RIC), 836–837
 Restriction operator, 652
 Reynolds number, 1627
 Risk analysis, 1909, 1916
 Risk assessment, 1104, 1106
 Risk management analysis tool (RMAT), 1411
 Robust optimization, 1968
 Rolls-Royce Engineering tool guide, 1427
- S**
- Sampling, 1654, 1655, 1657–1659
 design, 1226–1227
 Scalar conservation law, 662–670
 Scatterplots, 1532
 Screening
 classical group, 1159
 defined, 1144
 design, issues, 1157
 factorial group, 1159–1161
 groups of variables, 1159–1163
 sequential bifurcation, 1161–1162
 supersaturated designs for, 1154–1157
 without surrogate model, 1148
 Seaway loading models, 1581–1583
 Sensitivity, 2004, 2005, 2023–2024
 auditing, 1114–1118
 indices with given data, 1226–1227
 measures, common rationale, 1276–1278
 Sensitivity analysis (SA), 676, 1104–1119,
 1242, 1245, 1258, 1523, 1530–1533,
 1652, 1654, 1663, 1912, 1931–1932,
 1947–1950, 1980, 1981, 1984, 1987
 methods, 1108
 principles, 1105–1108
 R software packages, 1112–1113
 Sensitivity analysis on model output (SAMO), 1105
 Separability, 322, 339, 394–397
 Separable space, 320–322
 Sequential importance sampling, 1084
 Sequential Monte Carlo (SMC), 570
 methods, 371–379
 Shakedown, 1574
 Shared memory parallelism, 1775, 1781
 SIMLAB 4.0 , 1980, 1981
 SIMLAB data workflow, 1982–1983
 Simple random, 1985
 Simply supported beam
 identifiability problem, 76–81
 multiple-response approach to, 91–95
 preposterior and surrogate preposterior analyses, 112–117
 schematic of, 77
 single response study, 106, 117
 Simulation, 1467, 1468, 1493, 2002, 2022, 2023, 2026
 Simulation verification & validation, 1426
 Single exponential decay model, 50–51
 Singular value decomposition, 861–862
 Site indices, 1344, 1350, 1351
 S-N approach, 1557
 description, 1557
 fatigue life estimation, 1559
 fatigue reliability index, 1561
 Miner's damage accumulation index, 1558
 nominal stress approach, 1559
 notch stress approach, 1560
 reliability index, 1560
 structural hot spot stress approach, 1559
 Sobolev
 spaces, 399
 embeddings, 400–402
 fractional Sobolev norms, 400
 spaces, 396
 Sobol' design, 1986
 Sobol' indices, 435, 460, 460, 1243,
 1249, 1257, 1260, 1299–1301,
 1309–1312, 1320
 global sensitivity, 1246–1249, 1290
 Sobol' sensitivity indices, 1218, 1220–1221,
 1223–1227
 Software, 1763, 1980, 1981
 Sparse approximation methods, 859
 Sparse grid, 940, 1019–1020, 1777
 adaptive, 1006–1008, 1022–1024
 construction, 732–735, 1001–1004
 gCSM error, 739–743
 generalized, 1001–1002
 interpolation, 748–751, 954–955
 isotropic, 1001, 1015–1017
 multifidelity, 1030–1031
 with pre-defined offset, 1006
 RB construction, 975–976
 stochastic collocation, 752–753
 Sparsity, PCE, 832–833
 Spatial correlation, 88

- Spatio-temporal inputs, 1344
aggregated sensitivity indices, 1345–1348, 1353
block sensitivity indices, 1345, 1349–1351
computational cost, 1341
dimension reduction, 1335
macro-parameter, 1334–1335
map labelling, 1336–1337
metamodels, 1337–1340, 1348
NOE test, 1342–1343
sensitivity index maps, 1344, 1349, 1352–1353
simulating random samples, 1333
spatio-temporal data handling, 1341
switch/trigger input, 1336
- Spatio-temporal process, 195, 197, 199, 201–202
- Spectral methods, 1227–1232, 1234
- Splitting, 1085, 1089–1091
- Stability analysis, 1124, 1134–1135, 1140
- Stackelberg games, 1414
- Statistical decision theory, 170
- Statistical forward problem (SFP), 1835, 1842–1843
- Statistical inference methods, 34
- Statistical inverse problem (SIP), 225, 891, 1843, 1847
- APSM, 905–906
- experimental data, 1840–1841
- experimental data sets, 898
- methods for, 888–889
- non-Gaussian random field, 891, 907, 912–913
- OAPSM, 906–907
- optimization process, 892
- polynomial chaos expansion, 908–912
- posterior stochastic model, 913
- prior, likelihood and posterior, 1841
- stochastic elliptic operator and boundary value problem, 895–897
- stochastic finite element approximation, 897
- Statistics, 2006, 2008
- Steel roof truss, robust design, 1957–1962
- Stepwise Gaussian Process Variable Selection (SGPVS), 1170
- Stepwise rank regressions, 1516, 1531
- Stochastic advection-reaction, 1061–1063
- Stochastic analysis, 1913–1915
- Stochastic burgers equation, 1063–1065
- Stochastic collocation (SC), 700–714, 999–1001, 1662, 1680, 1686
definition, 701–702
interpolation approach, 702–709
- pseudo projection approach, 712–713
regression type, 709–712
- Stochastic collocation methods (SCMs)
adaptive hierarchical, 753–755
global, 731
hierarchical, 746–753
local, 746
- Stochastic coupled physics embedded quadratures for, 542–545
- Stochastic design, 1654, 1655, 1662
- Stochastic drift, 768
- Stochastic element (SE) bases, 641–643
- Stochastic fatigue crack growth analysis, 1554
- Stochastic finite element methods (SFEM), 766
toolbox, 1930–1931
- Stochastic Galerkin and collocation methods, 766
- Stochastic Galerkin discretization, 602–615
- Stochastic Galerkin equation, 541–542
- Stochastic Galerkin methods, 1765–1768, 1771
- Stochastic Galerkin projection method, 660
- Stochasticity for elliptic equations, 499–500
- Stochastic linear corrosion model, 1575
- Stochastic ordinary-differential equation (sODE), 764, 1041
- Stochastic partial differential equation (SPDE), 719, 720, 1039, 1041
- Stochastic process, 532–534
construction, 545–546
- Stochastic resonance, 1057–1061
- Stochastic system analysis, 638
- Stone-Weierstrass theorem, 526
- Structural equation model (SEM), 1443, 1444, 1467, 1469, 1470
- Structural fatigue cracking, 1576
- Structural hot spot stress approach, 1559
- Structural reliability, 1280–1284
- Structure discovery, 1477
- Subject matter expert (SME), 1414–1415
- Subset Simulation (SS), 1084–1089, 1095
- Sufficient statistics, 42–43
- Surrogate models, 1653, 1671, 1672, 1675, 1680, 1683, 1686, 1688, 1689, 1735, 1813. *See also* Response surface
global surrogates, 676
interpolation methods, 675
local surrogates, 676
moment evaluation, 676
nonparametric, 675
parametric, 675

- Surrogate models (*cont.*)
 PC surrogates (*see* Polynomial chaos (PC) surrogates)
 regression approaches, 675
 sensitivity analysis, 676
- Surrogate preposterior analysis, 105–106
- Symmetric random matrix, 232
- Synthetic one-dimensional example, 580–582
- System of interest, 159
- System verification and validation (V&V), 1426
- Systematic fractional replicate design, 1157–1159
- Systems engineering, 1425
- T**
- Take tons off sensibly (TOTS), 1570
- Tangent model, 1126–1129
- Tensor-structured equations, 875–876
- Terrestrial microbes, Mars, 1379–1384
- Tikhonov regularization, 358–364
- Time-convolutionless form of Mori-Zwanzig equation, 1048
- Time-variant redundancy, 1545
- Tonelli's theorem, 184
- Total variation distance, 375, 406, 407, 409, 411, 423
- Trans-Alaskan pipeline service (TAPS), 1569
- Transfer entropy (TE), 1453, 1459, 1461
- Transparency, 2001, 2003
- Transportation Security Administration (TSA), 1411
- Trigger input, 1336, 1341
- Twin-jet aircraft control system, 1943–1952
- U**
- Uncertainty, 1980, 1983, 1984, 1996, 1997, 2003
 analysis, 1105
 analytics, 1489
 aleatory, 1280
 epistemic, 1280
 management methodology, 2004–2005
 propagation, 2015–2023
 quantification, 2007
- Uncertainty approach
 aleatory, 1572
 basic design parameters data, 1573–1578
 epistemic, 1572
 fragmented, 1568
 handling in design codes, 1571–1572
 integrated works, 1583
- seaway loading models, 1581–1583
 structural strength models, 1579–1581
- Uncertainty Management framework, 1428
 decision analysis, 1431–1434
 modelling & simulation framework, 1430
 propagation of uncertainty, 1431
 quantification of input uncertainty, 1429–1430
- Uncertainty in simulation models, 1431
- Uncertainty propagation (UP) problem, 556, 559
- Uncertainty quantification (UQ), 4, 447, 449, 452–453, 462–464, 522, 523, 828, 938–986, 993–1034, 1696, 1714, 1869, 1910–1972
 analysis, 1700
 applications to, 514–515
 bottom-up parameter exposure, 1621
 Čebyšev inequalities and optimization theory, 164
 considerations in, 5
 characterization, 1698
 game theory to decision theory, 165–166
 general parameter exposure, 1622
 generalization, 170
 identification, 1698
 maximum fuel temperature, 1641
 maximum rod power, 1641
 mean squared error, variance and bias, 171
 mixing models, 172
 model error and optimal models, 170
 Mori-Zwanzig approach to, 1038–1066
 multidisciplinary nature, 4
 objectives, 1698
 optimal interval of confidence, 172
 optimization, 1728–1730
 optimization approach to statistics, 166–167
- OUQ, 161–163
- parameter distributions, 1623–1630
- process, 1697
- propagation, 1699
- PSUADE, 1703–1728
- QoI extraction, 1623
- reactivity, 1641
- and reliability analysis, 1927–1929
- scaling, 1620
- space of experiments, 172
- stochastic and robust optimization, 163–164
- top-down parameter exposure, 1621
- total uncertainty, 1616–1618
- user effect, 1630, 1630
- variety of sources, 4

- Wilks formula, 1618–1620
worst case analysis, 163
- Uncertainty quantification in linear structural dynamics, 268–274
- Uncertainty quantification in nonlinear structural dynamics, 275–280
- United States Coast Guard (USCG), 1413
- UQ Toolkit (UQTk), 1811, 1813
- Urban Area Security Initiative (UASI), 1413
- V**
- Validation, 441–448, 451–452, 458
prediction, models for, 145
(*see also* Prediction)
- Value of information-based sensitivity measure, 1274–1276
- Variational methods, 1124–1141
- Variational sensitivity analysis (VSA), 1124, 1135
- Verification, 430, 436–439, 448, 451–452, 458, 1533
- Verification and validation (V&V)
definition, 1402–1404
intelligent adversary models, 1407
(*see also* Intelligent adversary models)
- in model development process, 1405–1406
model validation techniques, 1403
quantitative model, 1406
- Vertical launch system (VLS), 1570
- Viking Lander contamination model, 1381
- W**
- Wald, Abraham, 167–169
- Waste Isolation Pilot Plant (WIPP), 1504, 1506, 1508, 1523
- White noise, 765, 776, 777
- Wiener's Gaussian space, 524
- Wiener measure, 341, 381, 390, 391
- Wiener process, 366, 367, 382, 388, 419–422
- Wiener-based approach, 525
- Wiener-Itô-Segal isomorphism, 524
- Wilks formula, 1618–1620
- Wind, 195, 196, 198, 199, 208–209
- Wind turbine vertical axis, 1025–1031
- Winkler's generalization, 162
- Worst case analysis, 163
- Y**
- Young's modulus, 77
- Yucca Mountain (YM), 1507, 1508, 1510, 1514, 1518, 1526