
UNCERTAINTY QUANTIFICATION IN MULTI-PHYSICS MODEL FOR WIND TURBINE ASSET MANAGEMENT

Elias FEKHARI

ÉLECTRICITÉ DE FRANCE R&D

Chatou, France

&

CÔTE D'AZUR UNIVERSITY

Nice, France

This dissertation is submitted for the degree of

Doctor of Philosophy

publicly defended on January xx, 2024 in front of the following jury:

Pr. Mireille BOSSY,	INRIA, Sophia-Antipolis	Examiner
Dr. Vincent CHABRIDON	EDF R&D, Chatou	Co-advisor
Dr. Sébastien DA VEIGA	ENSAI, Rennes	Examiner
Dr. Bertrand IOOSS	EDF R&D, Chatou	Thesis director
Dr. Anaïs LOVERA	EDF R&D, Saclay	Invite
Dr. Joseph MURÉ	EDF R&D, Chatou	Co-advisor
Pr. Franck SCHOEFS	Nantes Université, Nantes	Reviewer
Pr. Daniel STRAUB	TUM, Munich	Reviewer
Pr. Bruno SUDRET	ETH, Zürich	Examiner

Table of contents

List of figures	vii
List of tables	xii
Introduction	1
I Introduction to uncertainty quantification and wind energy	7
1 Uncertainty quantification in computer experiments	9
1.1 Introduction	10
1.2 Black-box model specification	10
1.3 Enumerating and modeling the uncertain inputs	11
1.3.1 Sources of the input uncertainties	11
1.3.2 Modeling uncertain inputs with the probabilistic framework	12
1.3.3 Joint input probability distribution	13
1.4 Central tendency uncertainty propagation	16
1.4.1 Numerical integration	16
1.4.2 Numerical design of experiments	22
1.4.3 Summary and discussion	25
1.5 Reliability-oriented uncertainty propagation	26
1.5.1 Problem formalization	27
1.5.2 Rare event estimation methods	29
1.5.3 Summary and discussion	34
1.6 Sensitivity analysis	36
1.6.1 Global sensitivity analysis	36
1.6.2 Reliability-oriented sensitivity analysis	36
1.7 Surrogate modeling	37
1.7.1 Common framework	37
1.7.2 General purposes surrogate model	38
1.7.3 Goal-oriented active surrogate model	40

1.7.4	Summary and discussion	41
1.8	Conclusion	41
2	Introduction to wind turbine modeling and design	43
2.1	Introduction	44
2.2	Wind turbine modeling	44
2.2.1	Synthetic wind generation [TurbSim, Kaimal spectrum]	44
2.2.2	Synthetic wave generation	44
2.2.3	Aerodynamic interactions	44
2.2.4	Servo-Hydro-Aero-Elastic wind turbine simulation [DIEGO]	44
2.2.5	Soil modeling	44
2.2.6	Wake modeling [FarmShadow]	44
2.3	Recommended design practices	44
2.3.1	Design load cases	44
2.3.2	Dynamic response design	44
2.3.3	Fatigue response design	44
2.4	Uncertain inputs	44
2.4.1	Environmental inputs	44
2.4.2	System inputs	44
2.4.3	Probabilistic fatigue assessment	44
2.5	Conclusion	44
II	Contributions to uncertainty quantification and propagation	45
3	Kernel-based uncertainty quantification	47
3.1	Introduction	48
3.2	Nonparametric fit of the environmental inputs	48
3.3	Quantifying and clustering the wake-induced perturbations within a wind farm	48
3.3.1	Uncertainty propagation in a wake model	49
3.3.2	Statistical metric of wake-induced perturbations	50
3.3.3	Conclusion	53
3.4	Conclusion	54
4	Kernel-based central tendency estimation	55
4.1	Introduction	56
4.2	Treatment of uncertainties on the Teesside wind farm	57
4.2.1	Numerical simulation model	58
4.2.2	Measured environmental data	59
4.2.3	Non parametric fit with empirical Bernstein copula	60
4.2.4	Fatigue assessment	62
4.3	Numerical integration procedures for mean damage estimation	63

4.3.1	Quadrature rules and quasi-Monte Carlo methods	63
4.3.2	Kernel discrepancy	65
4.3.3	Kernel herding sampling	68
4.3.4	Bayesian quadrature	71
4.4	Numerical experiments	74
4.4.1	Illustration through analytical toy-cases	74
4.4.2	Application to the Teesside wind turbine fatigue estimation	75
4.5	Conclusion	78
5	Kernel-based metamodel validation	81
5.1	Introduction	82
5.2	Sequential validation design	82
5.3	Computer experiments context	82
5.4	Machine learning given-data context	82
5.5	Application to wind turbine production metamodel	82
5.6	Conclusion	82
III	Contributions to rare event estimation	83
6	Nonparametric rare event estimation	85
6.1	Introduction	86
6.1.1	Background	87
6.2	Bernstein adaptive nonparametric conditional sampling	88
6.2.1	Empirical Bernstein copula	88
6.2.2	Bernstein adaptive nonparametric conditional sampling algorithm	90
6.3	Numerical experiments	91
6.3.1	Results analysis	93
6.4	Application to wind turbine fatigue reliability	94
6.5	Conclusion	94
7	Sequential reliability oriented sensitivity analysis	97
7.1	Conclusion	98
7.2	HSIC for GSA	98
7.3	HSIC for TSA & CSA	98
7.4	Sequential ROSA	98
7.5	Application to wind turbine fatigue reliability	98
7.6	Conclusion	98
Conclusion and perspectives		99
References		101

Appendix A Univariate distribution fitting	109
A.1 Main parametric methods	109
A.2 Main nonparametric methods	110
Appendix B Nonparametric copula estimation	113
B.1 Empirical copula	113
B.2 Empirical Bernstein & Beta copula	113
B.3 Goodness-of-fit	114
Appendix C Advanced rare event estimation algorithms	117
C.1 Subset simulation (SS)	117
C.2 Nonparametric adaptive importance sampling (NAIS)	117
C.3 Parametric adaptive importance sampling using cross-entropy optimization (AIS-CE)	117
Appendix D Résumé étendu de la thèse	119
Appendix E Uncertainty quantification implemented with OpenTURNS	121

List of figures

1	General uncertainty quantification framework (adapted from Ajenjo (2023))	3
1.1	Samples of three joint distributions with identical marginals and different dependence structures	14
1.2	Ranked samples represented in the Fig. 1.1	14
1.3	Univariate quadratures nodes ($1 \leq n \leq 15$)	18
1.4	Two univariate Gauss-Legendre quadratures combined as a tensor product and a Smolyak sparse grid	18
1.5	Nested Monte Carlo and quasi-Monte Carlo designs ($n = 256$)	21
1.6	Latin hypercube designs with poor and optimized space-filling properties ($n = 8$)	24
1.7	One-dimensional reliability analysis example	27
1.8	FORM and SORM approximation on a two-dimensional example	31
1.9	Multi-FORM approximation on an example with two MPFPs	31
1.10	Illustration of a rare event estimation	34
1.11	Illustration of a k -fold cross-validation (with $k = 4$)	38
1.12	Illustration of a ordinary kriging model fitted on a limited set of observations ($n = 7$)	40
3.1	South Brittany wind farm layout (left). South Brittany wind rose from the ANEMOC data (right, source: ?)	49
3.2	Joint perturbation at WT 13, 19, and 25	51
3.3	Ambient and wake-disturbed distributions of the wind speed	51
3.4	Ambient and wake-disturbed distributions of the turbulence intensity	51
3.5	Kernel mean embedding of two probability distributions P and Q mapped in the RKHS H_k	52
3.6	South Brittany layout and wake effects measured by the squared MMD on wind conditions. Note that the vertical direction on this plot does not represent the north direction.	53
3.7	Convergence of the squared MMD estimation	54

4.1	General uncertainty quantification and propagation framework (adapted from Ajenjo (2023))	56
4.2	Diagram of the chained OWT simulation model	58
4.3	Teesside wind farm layout (left). Monopile OWT diagram (Chen et al., 2018) (right)	59
4.4	Copulogram of the Teesside measured data ($N = 10^4$ in grey), kernel herding subsample ($n = 500$ in orange). Marginals are represented by univariate kernel density estimation plots (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space	61
4.5	Angular distribution of the wind and waves with a horizontal cross-section of the OWT structure and the mudline	64
4.6	Histogram of the log-damage, at mudline, azimuth 45 deg. (Monte Carlo reference sample)	64
4.7	Kernel mean embedding of a continuous and discrete probability distribution .	66
4.8	Kernel illustrations (left to right: energy-distance, squared exponential, and Matérn 5/2)	70
4.9	Sequential kernel herding for increasing design sizes ($n \in \{10, 20, 40\}$) built on a candidate set of $N = 8196$ points drawn from a complex Gaussian mixture μ .	70
4.10	Bayesian quadrature on a one-dimensional case	72
4.11	Analytical benchmark results on the toy-case #1	76
4.12	Analytical benchmark results on the toy-case #2	77
4.13	Sampling techniques used for the industrial use case	78
4.14	Copulogram of the kernel herding design of experiments with corresponding outputs in color (log-scale) on the Teesside case ($n = 10^3$). The highest values are in red while the lowest values are in blue. Marginals are represented by histograms (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.	79
4.15	Mean estimation convergence (at the mudline, azimuth $\theta = 45$ deg.) on the Teesside case	80
6.1	Evolution of m_{IMSE} for different dimensions and sample sizes.	90
6.2	BANCS on toy-case #1: illustration of conditional sampling and nonparametric fit at the first and second iterations.	90
6.3	QQ-plot for KDE of marginals of the conditional distribution from Fig. 6.2. .	92
6.4	Kendall plot for EBC on the copula of a conditional distribution from Fig. 6.2. .	92
6.5	BANCS sampling steps on toy-case #1.	94
6.6	BANCS sampling steps on toy-case #2.	95

A.1	Adequation of two different Weibull models using their likelihood with a sample of observations (black crosses).	110
A.2	Fit of a bimodal density by KDE using different tuning parameters.	111
A.3	QQ-plot between the data from Example 2 and a KDE model.	112
B.1	Evolution of m_{IMSE} for different dimensions and sample sizes.	115

List of tables

4.1	Teesside Offshore Wind turbine datasheet	58
4.2	Description of the environmental data.	60
4.3	Kernels considered in the following numerical experiments.	69
4.4	Analytical toy-cases	75
6.1	Results of the numerical experiments (subset sample size $N = 10^4$, $p_0 = 0.1$). . .	94

Introduction

Industrial context and motivation

The shift in wind energy projects from limited onshore resources to the vast potential of offshore locations is a growing trend. Offshore wind energy offers several advantages, including more consistent winds and the ability to install larger turbines. Since the installation of the first offshore wind farm in Vindeby, Denmark, in 1991, the industry has experienced rapid growth, with a total capacity of 56GW exploited worldwide in 2021. Over time, offshore wind technology has matured, resulting in significant achievements such as securing projects in Europe through "zero-subsidy bids," where electricity generated by wind farms is sold at wholesale prices.

However, despite the progress of this sector, scaling limitations emerge and numerous scientific challenges. To meet ambitious national and regional development targets, the wind energy industry must address various scaling issues, including port logistics, the demand for critical natural resources, and sustainable end-of-life processes. Furthermore, the field presents various scientific challenges that often involve coupling data with numerical simulations of physical systems and their surrounding environment. The wind energy community is focused on several objectives, including enhancing the design of floating offshore wind turbines, refining wind resource estimation techniques, and optimizing maintenance operations. Additionally, the design, installation and exploitation of these industrial assets implicate several decision-making steps, considering limited access to information. Therefore, properly modeling and treating the various uncertainties along this process proved to be a key success factor in this highly competitive industry.

Overall, the industry needs methods and techniques for uncertainty management to optimize safety margins and asset management. As a wind farm project developer, the attention is first drawn to refining the wind potential of candidate sites by combining different sources of information and modeling the multivariate distribution of environmental conditions within a wind farm. In floating projects, the probabilistic design helps to define safer and more robust solutions. As a wind farm owner, another significant consideration revolves around end-of-life management. This involves evaluating three possible outcomes: extending the operating assets' lifetime, replacing current turbines with more advanced models, or dismantling and selling the wind farm. The first two solutions require assessing the current reliability of the structure

and its remaining useful life. These quantitative evaluations are studied by certification bodies and insurance providers to issue exploitation permits. To deliver rigorous risk assessments, the generic *uncertainty quantification methodology* may be adopted.

Generic methodology for uncertainty quantification

Uncertainty Quantification ([UQ](#)) aims at modeling and managing uncertainties in complex systems. Over the year, generic UQ frameworks were proposed ([de Rocquigny et al., 2008](#)) to quantify and analyze the relations between uncertain input factors and the systems' outcomes. UQ is particularly relevant in situations where experiments or direct observations are costly, time-consuming, or even impossible to conduct.

Computer experiments, also known as numerical experiments or simulations, play an important role in UQ. They involve the use of numerical models to simulate the behavior of a system under various conditions and parameter settings. These virtual experiments provide a cost-effective way to explore the behavior of complex systems and make robust and well-informed decisions. They enable researchers and decision-makers to gain a deeper understanding of the system dynamics, optimize designs, assess risk, and make robust predictions. As a result, uncertainty quantification has become an essential tool in wind energy, benefiting from the multiphysics numerical models simulating the behavior of wind farms interacting with their environment. Nevertheless, numerical models should be calibrated against measured data and pass validation, and verification processes to minimize the residual modeling error. Figure 1 illustrates the UQ methodologies and the standardized usual steps encountered during a study, which are detailed hereafter:

- **Step A – Problem specification:** at this step, it is necessary to establish the system under study and construct a numerical model capable of precisely simulating its behavior. Specifying the problem also involves defining the complete set of parameters inherent to the computer model. This includes the input variables as well as determining the specific output quantity that will be generated by the numerical model;
- **Step B – Uncertainty modeling:** The objective of the second step is to identify all the sources of uncertainty impacting the input variables. Most of the time choosing a probabilistic framework, the modeling methods will depend on the available information (e.g., amount of data, input dimension);
- **Step C – Uncertainty propagation:** This step consists in propagating the uncertain inputs through the computer model, making the output uncertain. Then, the goal becomes the estimation of a quantity of interest (i.e., a statistic on the random output variable of interest). The uncertainty propagation method may differ depending on the quantity of interest targeted (e.g., central tendency, rare event);
- **Step C’ – Inverse analysis:** In this additional step, a sensitivity analysis can be performed to study the role allocated to each uncertain input leading to the uncertain output;

- **Metamodeling:** Since this methodology is frequently used with computationally expensive numerical models, it becomes interesting to emulate these models using statistical models constructed from a limited number of simulations. The uncertainty quantification is then performed on the so-called “metamodel” (or surrogate model) at a reasonable computation cost.

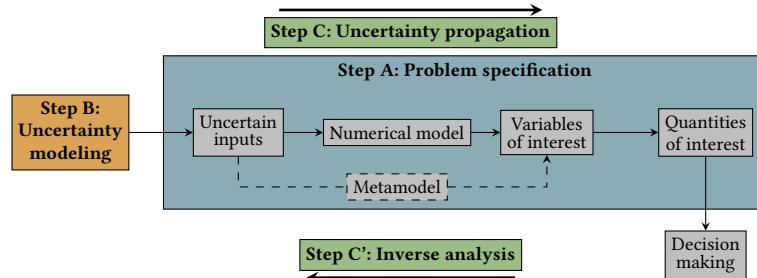


Fig. 1 General uncertainty quantification framework (adapted from [Ajenjo \(2023\)](#))

Problem statement and outline of the thesis

[Rewrite this paragraph] A general topic of research for EDF R&D is to adapt the UQ methodology to offshore wind turbine industrial cases. However, this problem presents various specificities, raising scientific challenges. First, the numerical model studied is composed of a series of three codes, among which one is intrinsically stochastic (i.e., running twice the same numerical model with the same set of inputs results in different outputs). Second, the computational cost of these numerical models quickly requires the use of efficient techniques deployed on high-performance computers to perform UQ. Then, the probabilistic modeling tools available to model the uncertain inputs are challenged by a complex underlying dependence structure. In the presence of large amounts of data describing these complex inputs, different methods to quantify and propagate the uncertainties are needed. Finally, performing a risk assessment on this case study combines all the challenges previously stated. In order to adapt the UQ framework to this industrial case, this thesis aims at answering the following questions:

- Q1** *How to accurately model the complex dependence structure underlying the multivariate distribution of the environmental conditions?*
- Q2** *How to perform an efficient and accurate given-data uncertainty propagation on a costly and stochastic numerical model?*
- Q3** *How to couple rare event estimation with reliability-oriented sensitivity analysis?*

To intend at solving these problems, this thesis is divided into three parts. The first part gathers an introduction to UQ’s state-of-the-art and a specification of the offshore wind turbine problem. The second part presents the contributions to uncertainty quantification and propagation while the third part the contributions to rare event estimation. This manuscript is divided into seven chapters, which are summarized hereafter:

- Chapter 1** Introduction to uncertainty quantification
- Chapter 2** Introduction to wind turbine modeling and design
- Chapter 3** Kernel-based uncertainty quantification
- Chapter 4** Kernel-based central tendency estimation
- Chapter 5** Kernel-based metamodel validation
- Chapter 6** Nonparametric rare event estimation
- Chapter 7** Sequential reliability oriented sensitivity analysis

Numerical developments

In the vain of an open-data approach, this aims at sharing the implementations developed and allows the reader to reproduce numerical results. Along this thesis, the contributions to numerical developments are summarized below:

- This Python package generates designs of experiments based on kernel methods such as Kernel Herding and Support Points. A tensorized implementation of the algorithms was proposed, significantly increasing their performances. Additionally, optimal weights for Bayesian quadrature are provided.

- `otkerneldesign`¹
- This Python package, developed in collaboration with J.Muré, is available on the platform Pypi and fully documented.

-
- `bancs`²
- This Python package proposes an implementation of the “Bernstein Adaptive Non-parametric Conditional Sampling” method for rare event estimation.

-
- `ctbenchmark`³
- This Python package presents a standardized process to benchmark different sampling methods for central tendency estimation.

- This Python package is available on a GitHub repository with analytical benchmarks.

-
- `copulogram`⁴
- This Python package proposes an implementation of a synthetic visualization tool for multivariate distributions.

- This Python package, developed in collaboration with V.Chabridon, is available on the Pypi platform.

¹Documentation: <https://efekhari27.github.io/otkerneldesign/master/>

²Repository: <https://github.com/efekhari27/bancs>

³Repository: <https://github.com/efekhari27/ctbenchmark>

⁴Repository: <https://github.com/efekhari27/copulogram>

Publications and communications

The research contributions in this manuscript are based on the following publications:

Book Chap.	<u>E. Fekhari</u> , B. Iooss, J. Muré, L. Pronzato and M.J. Rendas (2023). “Model predictivity assessment: incremental test-set selection and accuracy evaluation”. In: <i>Studies in Theoretical and Applied Statistics</i> , pages 315–347. Springer.
Jour Pap.	<u>E. Fekhari</u> , V. Chabridon, J. Muré and B. Iooss (2023). “Fast given-data uncertainty propagation in offshore wind turbine simulator using Bayesian quadrature”. In: <i>Data-Centric Engineering</i> . [<u>E. Fekhari</u> , V. Chabridon, J. Muré and B. Iooss (2023). “TO DO: Bernstein adaptive nonparametric conditional sampling”. In: <i>Special Issue in Honor of Professor Armen Der Kiureghian. Reliability Engineering & System Safety.</i>]
Int. Conf	<u>E. Fekhari</u> , B. Iooss, V. Chabridon, J. Muré (2022). “Numerical Studies of Bayesian Quadrature Applied to Offshore Wind Turbine Load Estimation”. In: <i>SIAM Conference on Uncertainty Quantification (SIAM UQ22)</i> , Atlanta, USA. (Talk) <u>E. Fekhari</u> , B. Iooss, V. Chabridon, J. Muré (2022). “Model predictivity assessment: incremental test-set selection and accuracy evaluation”. In: <i>22nd Annual Conference of the European Network for Business and Industrial Statistics (ENBIS 2022)</i> , Trondheim, Norway. (Talk) <u>E. Fekhari</u> , B. Iooss, V. Chabridon, J. Muré (2022). “Efficient techniques for fast uncertainty propagation in an offshore wind turbine multi-physics simulation tool”. In: <i>Proceedings of the 5th International Conference on Renewable Energies Offshore (RENEW 2022)</i> , Lisbon, Portugal. (Paper & Talk) <u>E. Fekhari</u> , V. Chabridon, J. Muré and B. Iooss (2023). “Bernstein adaptive nonparametric conditional sampling: a new method for rare event probability estimation” ⁵ . In: <i>Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)</i> , Dublin, Ireland. (Paper & Talk) <u>E. Vanem</u> , <u>E. Fekhari</u> , N. Dimitrov, M. Kelly, A. Cousin and M. Guiton (2023). “A joint probability distribution model for multivariate wind and wave conditions”. In: <i>Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2023)</i> , Melbourne, Australia. (Paper) <u>A. Lovera</u> , <u>E. Fekhari</u> , B. Jézéquel, M. Dupoirion, M. Guiton and E. Ardillon (2023). “Quantifying and clustering the wake-induced perturbations within a wind farm for load analysis”. In: <i>Journal of Physics: Conference Series (WAKE 2023)</i> , Visby, Sweden (Paper)
Nat. Conf.	<u>E. Fekhari</u> , B. Iooss, V. Chabridon, J. Muré (2022). “Kernel-based quadrature applied to offshore wind turbine damage estimation”. In: <i>Proceedings of the Mascot-Num 2022 Annual Conference (MASCOT NUM 2022)</i> , Clermont-Ferrand, France (Poster) <u>E. Fekhari</u> , B. Iooss, V. Chabridon, J. Muré (2023). “Rare event estimation using nonparametric Bernstein adaptive sampling”. In: <i>Proceedings of the Mascot-Num 2023 Annual Conference (MASCOT-NUM 2023)</i> , Le Croisic, France (Talk)

⁵This contribution was rewarded by the “CERRA Student Recognition Award”

PART I:

INTRODUCTION TO UNCERTAINTY QUANTIFICATION AND WIND ENERGY

Toute pensée émet un coup de dé.

S. MALLARMÉ

Chapter **1**

Uncertainty quantification in computer experiments

1.1	Introduction	10
1.2	Black-box model specification	10
1.3	Enumerating and modeling the uncertain inputs	11
1.3.1	Sources of the input uncertainties	11
1.3.2	Modeling uncertain inputs with the probabilistic framework	12
1.3.3	Joint input probability distribution	13
1.4	Central tendency uncertainty propagation	16
1.4.1	Numerical integration	16
1.4.2	Numerical design of experiments	22
1.4.3	Summary and discussion	25
1.5	Reliability-oriented uncertainty propagation	26
1.5.1	Problem formalization	27
1.5.2	Rare event estimation methods	29
1.5.3	Summary and discussion	34
1.6	Sensitivity analysis	36
1.6.1	Global sensitivity analysis	36
1.6.2	Reliability-oriented sensitivity analysis	36
1.7	Surrogate modeling	37
1.7.1	Common framework	37
1.7.2	General purposes surrogate model	38
1.7.3	Goal-oriented active surrogate model	40
1.7.4	Summary and discussion	41
1.8	Conclusion	41

1.1 Introduction

The progress of computer simulation gradually allows the virtual resolution of more complex problems in scientific fields such as physics, astrophysics, engineering, climatology, chemistry, or biology. This domain often provides a deterministic solution to complex problems depending on several inputs. Associating a UQ analysis with these possibly nonlinear numerical models is a key element to improving the understanding of the phenomena studied. A wide panel of UQ methods has been developed over the years to pursue these studies with a reasonable computational cost.

This chapter presents the standard tools and methods from the generic UQ framework [Sullivan \(2015\)](#), exploited later in this thesis. It is structured as follows: Section [add pointer] describes the context of the model specification step; Section [add pointer] presents a classification of the inputs uncertainties and the probabilistic framework to model them; Section [add pointer] and [add pointer] introduce various methods to propagate the input uncertainties through the numerical model for different purposes; finally, Section [add pointer] presents the main inverse methods to perform sensitivity analysis in our framework.

OpenTURNS¹. This high-performance Python library is dedicated to UQ ([Baudin et al., 2017](#)). OpenTURNS (“Open source initiative for the Treatment of Uncertainties, Risks’N Statistics”) is developed by an industrial researchers from EDF R&D, Airbus Group, PHIMECA Engineering, IMACS and ONERA. It combines high-performance using C++ programming with high-accessibility through a Python API. Overall, this opensource library provides tools for various steps of the UQ framework (e.g., uncertainty quantification, uncertainty propagation, surrogate modeling, reliability, sensitivity analysis and calibration). To guaranty the software quality, the development follows robust processes such as unit testing and multiplatform continuous integration. An active community hosted on a dedicated forum helps new users and discusses areas of improvement. Finally, no-code users can benefit from OpenTURNS’s Graphical User Interface software, named [Persalys²](#). In this chapter, minimal OpenTURNS implementations of the methodological concepts will be presented.

1.2 Black-box model specification

The uncertainty quantification studies in our framework are performed around an input-output numerical simulation model. This numerical model, or code, is hereafter considered as *black-box* since the knowledge of the underlying physics doesn’t inform the UQ methods. Alternatively, one could consider *intrusive* UQ methods, introducing uncertainties within the resolution of computer simulation (see e.g., [Le Maître and Knio \(2010\)](#)). In practice, the numerical model might be a sequence of codes executed in series to obtain a variable of interest.

¹OpenTURNS installation guide and documentation are available at <https://openturns.github.io/www/>

²Persalys is a free-download software available at <https://www.persalys.fr/obtenir.php>

Moreover, the simulation model is in most cases deterministic, otherwise, it is qualified as intrinsically stochastic (i.e., two runs of the same model taking the same inputs return different outputs). Then, most numerical simulation presents modeling errors. In the following, it will be assumed that the numerical models passed a *validation & verification* phase, to quantify their confidence and predictive accuracy.

Formally, part of the problem specification is the definition of the set of d input variables $\mathbf{x} = (x_1, \dots, x_d)^\top$ considered uncertain (e.g., wind speed, wave period, etc.). In this thesis, the models considered will only present scalar outputs. UQ methods dedicated to other types of outputs exist (see e.g., for time series outputs [Lataniotis \(2019\)](#), for functional outputs [Auder et al. \(2012\)](#); [Rollón de Pinedo et al. \(2021\)](#)). Let us then define the following numerical model:

$$\mathcal{M} : \left| \begin{array}{l} \mathcal{D}_x \subseteq \mathbb{R}^d \longrightarrow \mathcal{D}_y \subseteq \mathbb{R} \\ \mathbf{x} \longmapsto y. \end{array} \right. \quad (1.1)$$

Unlike the typical machine learning input-output dataset framework, the UQ analyst can simulate the output image of any inputs (in the input domain), using the numerical model. However, numerical simulations often come with an important computational cost. Therefore UQ methods should be efficient and require as few simulations as possible. In this context, metamodels (or surrogate models) are statistical approximations of the costly numerical model, that can be used to perform tractable UQ. Metamodels are only built and validated on a limited number of simulations (in a *supervised learning* framework). In practice, the model specification step is often associated with the development of a *wrapper* of the code. The wrapper of a numerical model is an overlay of code allowing its execution in a parametric way, which is often associated with a *high-performance computer* (HPC) deployment. Once the model is specified, a critical step of uncertainty quantification is enumerating the input uncertainties and building an associated mathematical model.

1.3 Enumerating and modeling the uncertain inputs

1.3.1 Sources of the input uncertainties

To ensure a complete risk assessment (e.g., associated with the exploitation of a wind turbine throughout its life span), the analyst should construct a list of uncertain inputs as exhaustive as possible. Even if these uncertainties might have different origins, they should all be considered jointly in the UQ study. The authors proposed to classify them for practical purposes into two groups:

- **aleatory uncertainty** regroups the uncertainties that arise from natural randomness (e.g., wind turbulence). From a risk management point of view, these uncertainties are qualified as *irreducible* since the industrials facing them will not be able to acquire additional information to reduce them (e.g., additional measures).

- **epistemic uncertainty** gathers the uncertainties resulting from a lack of knowledge. Contrarily to the aleatory ones, epistemic uncertainties might be reduced by investigating their origin.

Der Kiureghian and Ditlevsen (2009) offers a discussion on the relevance of this classification. They affirm that this split is practical for decision-makers to identify possible ways to reduce their uncertainties. However, this distinction should not affect the way of modeling or propagating uncertainties. [To illustrate the limits of this split, add examples of uncertainties presenting both an aleatory and epistemic aspect.] In the following, the probabilistic framework is introduced to deal with uncertainties.

1.3.2 Modeling uncertain inputs with the probabilistic framework

Uncertainties are traditionally modeled with objects from the probability theory. In this thesis, the *probabilistic framework* is adopted. Alternative theories exist to mathematically model uncertainties. For example, imprecise probability theory allows more general modeling of the uncertainties. It becomes useful when dealing with very limited and possibly contradictory information (e.g., expert elicitation). The core probabilistic tools and objects are introduced hereafter.

The *probability space* (i.e., a measure space with its total measure summing to one), also called probability triple and denoted $(\Omega, \mathcal{A}, \mu)$. This mathematical concept first includes a sample space Ω , which contains a set of outcomes $\omega \in \Omega$. An *event* is defined as a set of outcomes in the sample space. Then, a σ -algebra \mathcal{A} (also called event space) is a set of events. Finally, a probability function $\mu : \mathcal{A} \rightarrow [0, 1]$, is a positive probability measure associated with an event. Most often, the choice of the probability space will not be specified. The main object will be functions defined over this probability space: random variables.

The *random vector* \mathbf{X} (i.e., multivariate random variable) is a measurable function defined as:

$$\mathbf{X} : \begin{cases} \Omega & \longrightarrow \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d \\ \omega & \longmapsto \mathbf{X}(\omega) = \mathbf{x}. \end{cases} \quad (1.2)$$

In the following, the random vector \mathbf{X} will be considered to be a squared-integrable function against the measure μ (i.e., $\int_{\Omega} |\mathbf{X}(\omega)|^2 d\mu(\omega) < \infty$). Moreover, this work will focus on continuous random variables.

The *probability distribution* of the random vector \mathbf{X} is the pushforward measure of μ by \mathbf{X} . Which is a probability measure on $(\mathcal{D}_{\mathbf{x}}, \mathcal{A})$, denoted $\mu_{\mathbf{X}}$ and defined by:

$$\mu_{\mathbf{X}}(B) = \mu(\mathbf{X} \in B) = \mu(\omega \in \Omega : \mathbf{X}(\omega) \in B), \quad \forall B \in \mathcal{A}. \quad (1.3)$$

The *cumulative distribution function* (CDF) is a common tool to manipulate random variables. It is a function $F_{\mathbf{X}} : \mathcal{D}_{\mathbf{x}} \rightarrow [0, 1]$ defined for all $\mathbf{x} \in \mathcal{D}_{\mathbf{x}}$ as:

$$F_{\mathbf{X}}(\mathbf{x}) = \mu(\mathbf{X} \leq \mathbf{x}) = \mu(X_1 \leq x_1, \dots, X_d \leq x_d) = \mu_{\mathbf{X}}([-\infty, x_1] \times \dots \times [-\infty, x_d]). \quad (1.4)$$

The CDF is a positive, increasing, right-continuous function, which tends to 0 as x tends to $-\infty$ and to 1 as x tends to $+\infty$. In the continuous case, one can also define a corresponding *probability density function* (PDF) $f_X : \mathcal{D}_x \rightarrow \mathbb{R}_+$ with $f_X(x) = \frac{\partial^d F_X(x)}{\partial x_1 \dots \partial x_d}$.

The expected value of a random vector $\mathbb{E}[X]$, also called the first moment, is a vector defined as:

$$\mathbb{E}[X] = \int_{\Omega} X(\omega) d\mu(\omega) = \int_{\mathcal{D}_x} x f_X(x) dx = (\mathbb{E}[X_1], \dots, \mathbb{E}[X_d])^\top. \quad (1.5)$$

In addition, considering two random variables X_i and X_j , with $i, j \in \{1, \dots, d\}$, one can write their respective variance:

$$\text{Var}(X_i) = \mathbb{E}[X_i - \mathbb{E}[X_i]]^2, \quad (1.6)$$

and a covariance describing their joint variability:

$$\text{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]. \quad (1.7)$$

The standard deviation $\sigma_{X_j} = \sqrt{\text{Var}(X_j)}$ and coefficient of variation $\delta_{X_j} = \frac{\text{Var}(X_j)}{\|\mathbb{E}[X_j]\|}$ are two quantities directly associated to the two first moments.

1.3.3 Joint input probability distribution

This section aims at presenting various techniques to model and infer a joint probability distribution (or multivariate distribution). It will first introduce the *copula*, a universal mathematical tool to model the dependence structure of a joint distribution. Then, a few methods to fit a joint distribution over a dataset will be mentioned. And finally, a panel of tools to evaluate the goodness of fit between a probabilistic model and a dataset will be recalled.

From a practical point of view, people tend to properly model the single effects of their input uncertainties. However, modeling the dependence structure underlying in a joint distribution is often overlooked. To illustrate the importance of this step, Fig. 1.1 represents three i.i.d samples from three bivariate distributions sharing the same single effects (e.g., here two exponential distributions) but different dependence structures. One can assume that the joint distribution is the composition of the single effects, also called marginals, and an application governing the dependence between them.

An empirical way of isolating the three dependence structures from this example is to transform the samples in the ranked space. Let us consider a n -sized sample $X_n = \{x^{(1)}, \dots, x^{(n)}\} \in \mathcal{D}_x^n$. The corresponding ranked sample is defined as: $R_n = \{r^{(1)}, \dots, r^{(n)}\}$, where³ $r_j^{(l)} = \sum_{i=1}^n \mathbb{1}_{\{x_j^{(i)} \leq x_j^{(l)}\}}$, $\forall j \in \{1, \dots, d\}$. Ranking a multivariate dataset allows us to isolate the dependence structure witnessed empirically. Fig. 1.2 shows the same three samples from Fig. 1.1 in the ranked space. One can first notice that the marginals are uniform since each rank is uniformly distributed. Then, the scatter plot from the distribution with independent copula (left plot) is uniform while the two others present different patterns.

³The *indicator function* is defined such that $\mathbb{1}_{\{\mathcal{A}\}}(x) = 1$ if $x \in \mathcal{A}$ and is equal to zero otherwise.

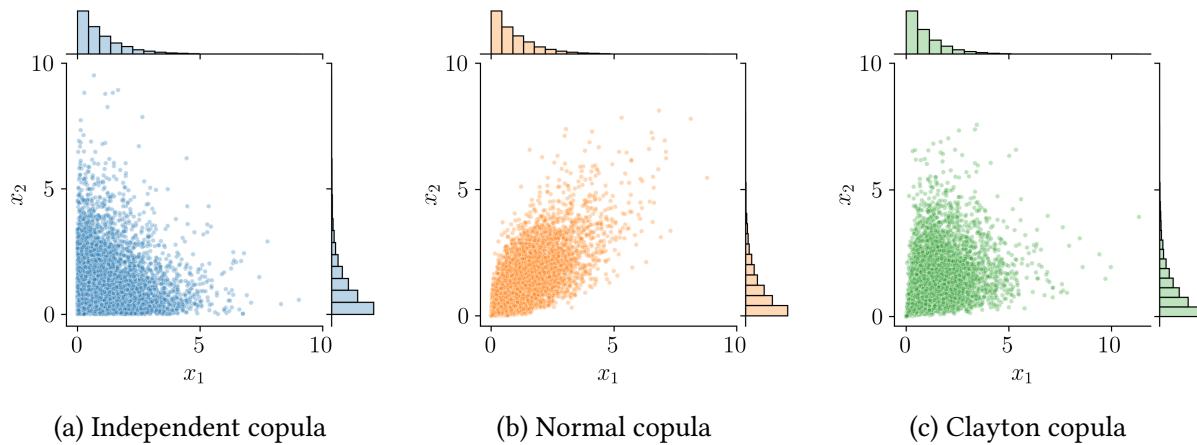


Fig. 1.1 Samples of three joint distributions with identical marginals and different dependence structures

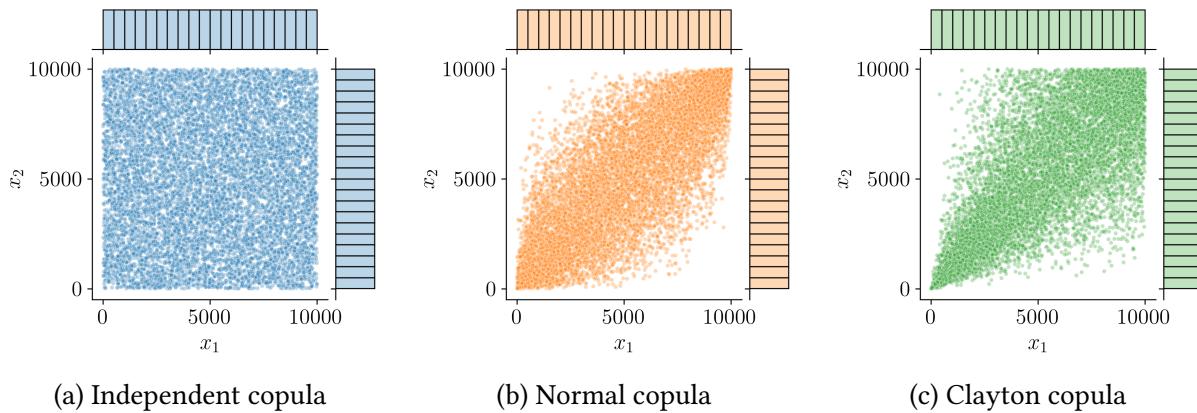


Fig. 1.2 Ranked samples represented in the Fig. 1.1

A theorem states that the multivariate distribution of any random vector can be broken down into two objects (Joe, 1997). First, a set of univariate marginal distributions describing the behavior of the individual variables; Second, a function describing the dependence structure between all variables: a copula.

Theorem 1 (Sklar's theorem). *Let $\mathbf{X} \in \mathbb{R}^d$ be a random vector and its joint CDF $F_{\mathbf{X}}$ with marginals $\{F_{X_j}\}_{j=1}^d$, there exists a copula $C : [0, 1]^d \rightarrow [0, 1]$, such that:*

$$F_{\mathbf{X}}(x_1, \dots, x_d) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d) = C(F_{X_1}(x_1), \dots, F_{X_d}(x_d)). \quad (1.8)$$

If the marginals F_{X_i} are continuous, then this copula is unique. If the multivariate distribution has a PDF $f_{\mathbf{X}}$, it can also be expressed:

$$f_{\mathbf{X}}(x_1, \dots, x_d) = c(F_{X_1}(x_1), \dots, F_{X_d}(x_d)) \times f_{X_1}(x_1) \times \dots \times f_{X_d}(x_d), \quad (1.9)$$

where c is the density of the copula, sometimes also called copula by misuse of language. The reader might refer to [cite proof] for further mathematical proof.

Theorem 1 expresses the joint CDF by combining marginal CDFs and a copula, which is practical for sampling joint distributions. Conversely, the copula can be defined by using the joint CDF and the marginal CDFs:

$$C(u_1, \dots, u_d) = F_X(F_{X_1}^{-1}(u_1), \dots, F_{X_d}^{-1}(u_d)) \quad (1.10)$$

This equation allows us to extract a copula from a joint distribution by knowing its marginals. Additionally, copulas are invariant under increasing transformations. This property is important to understand the use of rank transformation to display the copula without the marginal effects.

Identically to the univariate continuous distributions, a large catalog of families of copulas exists (e.g., independent, Normal, Clayton, Frank, Gumbel copula, etc.). Note that the independent copula implies that the distribution is fully defined by the product of its marginals. To infer a joint distribution, this theorem divides the fitting problem into two independent problems: fitting the marginals and fitting the copula. Provided a dataset, this framework allows the combination of a parametric (or nonparametric) fit of marginals with a parametric (or nonparametric) fit of the copula.

To infer a joint distribution over a dataset, the analyst should determine a fitting strategy. Appropriate data visualization helps to choose the fitting methods susceptible to be relevant to the problem. The following points can be checked at this early stage: [formalize bullets]

- Is the distribution unimodal? If not, mixtures methods or nonparametric models might be required;
- Is the validity domain restrictive? If so, specific families of parametric distributions can be chosen or truncations can be applied;
- Is the dimension medium or high? In our context, a problem [with dimension $d > 10$] might be qualified as medium dimension problem. If the dimension is too high: tensorize the distribution as much as possible.
- Is the dependence structure complex? Graphically, the dataset in the ranked space gives an empirical description but some independence tests exist as well.

Appendix A details the main techniques to estimate marginal distributions. Then, Appendix B introduces different nonparametric methods to infer a copula, including the empirical Bernstein copula and the Beta copula. The adequation between a fitted probabilistic model and a dataset should be validated, therefore, Appendices A and B respectively present visual and quantitative tools for goodness-of-fit evaluation.

OpenTURNS 1 (Bivariate distribution). The following Python code proposes a minimalist OpenTURNS implementation of a probabilistic uncertainty modeling.

1.4 Central tendency uncertainty propagation

The previous section aimed at building a probabilistic model of the uncertainties considering the knowledge available. This one will introduce diverse forward propagation of uncertainty through a numerical model. This step is hereafter qualified as “global” because the analysis of the resulting output random variable will particularly focus on its central tendency (i.e., expected value and variance). This approach contrasts with the uncertainty propagation dedicated to rare event estimation, which will be introduced in the next section (e.g., for a reliability or certification problem).

The difficulties related to any uncertainty propagation mostly arise from the practical properties of the numerical model. Its potential high dimension, low regularity and nonlinearities each represent a challenge. These studies rely on a finite number of observations which depends on the computational budget the analyst can afford. This forward propagation might be a finality of the uncertainty quantification, but keep in mind that it fully stands on an accurate uncertainty modeling. Uncertainty propagation should be perceived as a standardized process with modular bricks, on which the “garbage in, garbage out” concept fully applies.

This section introduces the main methods of global uncertainty propagation. Outlining the strong links between numerical integration (i.e., Lebesgue integration or central tendency estimation) and numerical design of experiments.

1.4.1 Numerical integration

Forward uncertainty propagation aims at integrating a measurable function $g : \mathcal{D}_X \rightarrow \mathbb{R}$ with respect to a probability measure μ . Numerical integration brings algorithmic tools to help the resolution of this probabilistic integration (i.e., Lebesgue integration).

In practice, this integral is approximated by summing a finite n -sized set of realizations $\mathbf{y}_n = \{g(\mathbf{x}^{(1)}), \dots, g(\mathbf{x}^{(n)})\}$ from a set of input samples $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$. A *quadrature* establishes a rule to select the input samples \mathbf{X}_n (also called nodes), and an associated set of weights $\mathbf{w}_n = \{w_1, \dots, w_n\} \in \mathbb{R}^n$. The approximation given by a quadrature rule is defined as a weighted arithmetic mean of the realizations:

$$I_\mu(g) := \int_{\mathcal{D}_X} g(\mathbf{x}) d\mu(\mathbf{x}) \approx \sum_{i=1}^n w_i g(\mathbf{x}^{(i)}). \quad (1.11)$$

For a given sample size n , our goal is to find a set of tuples $\{\mathbf{x}^{(i)}, w_i\}_{i=1}^n$ (i.e., quadrature rule), giving the best approximation of our quantity. Ideally, the approximation quality should be fulfilled for a wide class of integrands. Most quadrature rules only depend on the measure space $(\Omega, \mathcal{A}, \mu)$, regardless of the integrand values. In the context of a costly numerical model, this property allows the analyst to massively distribute the calls to the numerical model.

This section aims at presenting the main multivariate numerical integration techniques. These methods have very different properties: some are deterministic and some are aleatory;

some are sequential (or nested) some are not; some are victims of the curse of dimensionality and some are not. [A summary table of the different methods and their respective properties is proposed ADD REF TO TABLE].

Classical multivariate deterministic quadrature

Historically, quadrature methods have been developed for univariate integrals. The Gaussian rule and the Fejér-Clebschaw-Curtis rule are two univariate deterministic quadratures that will be briefly introduced.

Gaussian quadrature is a powerful univariate quadrature building together a set of irregular nodes and a set of weights. The computed weights are positive, which ensures a numerically stable rule even for large sample sizes.

Different variants of rules exist, the most famous being the Gauss-Legendre quadrature. In this case, the function g to be integrated with respect to the uniform measure on $[-1, 1]$ is approximated by Legendre polynomials. Considering the Legendre polynomial of order n , denoted l_n , the quadrature nodes $x^{(i)}_{i=1}^n$ are given by the polynomial roots. The respective weights are given by the following formula:

$$w_i = \frac{2}{\left(1 - (x^{(i)})^2\right) (l'_n(x^{(i)}))^2}. \quad (1.12)$$

This rule guarantees a very precise approximation provided that the integrand is well-approximated by a polynomial of degree $2n - 1$ or less on $[-1, 1]$. This rule is deterministic but not sequential, meaning that two rules with sizes n_1 and n_2 , $n_1 < n_2$ will not be nested. However, a sequential extension is proposed by the Gauss-Kronrod rule (Laurie, 1997), offering lower accuracy.

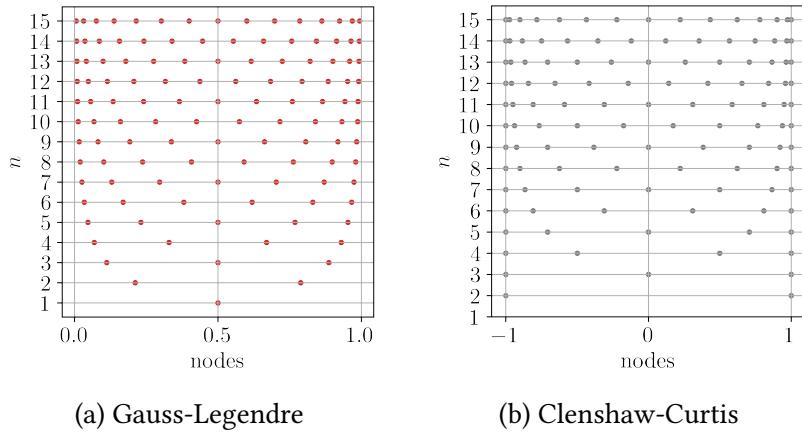
To overcome this practical drawback, Fejér then Clebschaw with Curtis proposed a nested rule with mostly equivalent accuracy as Gaussian quadrature. This method is usually presented to integrate a function with respect to the uniform measure on $[-1, 1]$ and starts with a change of variables:

$$\int_{-1}^1 g(x) dx = \int_0^\pi g(\cos(\theta)) \sin(\theta) d\theta \quad (1.13)$$

This expression can be written as an expansion of the integrand using cosine series. Moreover, cosine series are closely related to the Chebyshev polynomials of the first kind. Fejér's "first rule" (Trefethen, 2008) relies on the Chebyshev polynomials roots as nodes $x^{(i)} = \cos(\theta^{(i+1/2)})$, and the following weights:

$$w_i = \frac{2}{n} \left(1 - 2 \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{1}{4j^2 - 1} \cos(j\theta^{(2i+1)}) \right) \quad (1.14)$$

These two univariate integration schemes are both very efficient on a wide panel of functions. Yet, Fejér-Clebschaw-Curtis is sequential and offers easy implementations, benefitting from powerful algorithms such as the *fast Fourier transform*.

Fig. 1.3 Univariate quadratures nodes ($1 \leq n \leq 15$)

Uncertainty quantification problems are rarely unidimensional, but one can build a multivariate quadrature rule by defining the tensor product (also called full grids) of univariate rules. This exhaustive approach quickly shows its practical limits as the problem's dimension increases. Alternatively, sparse multivariate quadratures (i.e., Smolyak sparse grid) explore the joint domain more efficiently. [Introduce the recurrent Smolyak formula?]

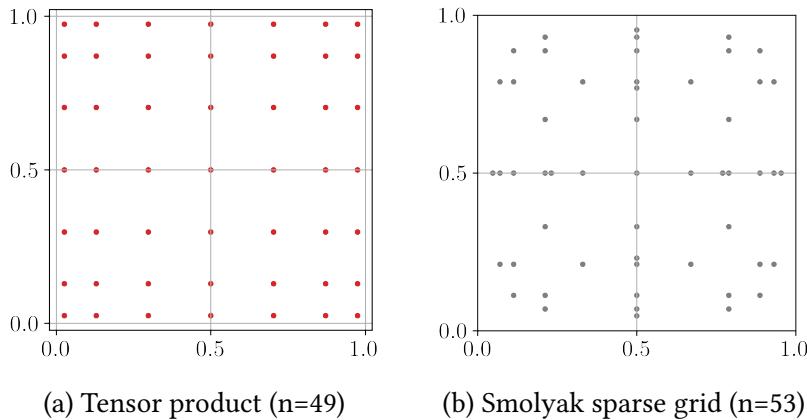


Fig. 1.4 Two univariate Gauss-Legendre quadratures combined as a tensor product and a Smolyak sparse grid

Monte Carlo methods

Monte Carlo methods were initially developed in the 1940s to solve problems in neutronics. Ever since this frequentist techniques have been applied to the resolution of the Lebesgue integral. To integrate a function g against a measure μ , it randomly generates points following the input measure. The integral is estimated by taking the uniform arithmetic mean of the images of these nodes obtained by this random process.

This aleatory method requires to be able to generate points following a given distribution. To do so, the most common approach is to first generate a sequence of random points uniformly on $[0, 1]$. These sequences mimic actual uniform randomness but are in fact generated

by deterministic algorithms (also called pseudorandom number generators). Pseudorandom algorithms generate a sequence of numbers with a very large, but finite length. This sequence can be exactly repeated by fixing the same initial point, also called *pseudorandom seed*. Most programming languages use the Mersenne Twister pseudorandom generator (Matsumoto and Nishimura, 1998), offering a very long period (around 4.3×10^{6001} iterations).

Formally, the “Vanilla” Monte Carlo (sometimes called “crude” Monte Carlo) method uses a set of i.i.d samples $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$ following the joint distribution of μ . The Monte Carlo estimator of the integral is given by:

$$I_\mu(g) \approx \bar{y}_n^{\text{MC}} = \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}^{(i)}). \quad (1.15)$$

By construction, the law of large numbers makes this estimator unbiased, however, it converges relatively slowly. Considering the images of the sample \mathbf{X}_n , one can also estimate the variance of the output random variable $\hat{\sigma}_Y^2$. The variance of the Monte Carlo estimator results from a manipulation of the central tendency theorem:

$$\text{Var}\left(\bar{y}_n^{\text{MC}}\right) = \frac{1}{\sqrt{n}} \text{Var}(g(\mathbf{X})). \quad (1.16)$$

This estimator also comes with theoretical confidence intervals at $\alpha\%$, regardless of the output distribution:

$$I_\mu(g) \in \left[\bar{y}_n^{\text{MC}} - q_\alpha \frac{\text{Var}(g(\mathbf{X}))}{\sqrt{n}}, \bar{y}_n^{\text{MC}} + q_\alpha \frac{\text{Var}(g(\mathbf{X}))}{\sqrt{n}} \right], \quad (1.17)$$

where q_α is the α -quantile of the standard normal distribution. Monte Carlo presents the advantage of being a universal method, with no bias and strong convergence guarantees. Moreover, it is worth noting that its convergence properties do not depend on the dimension of the input domain. Unlike the previous multivariate deterministic quadrature, Monte Carlo doesn't suffer from the curse of dimensionality. The main limit of crude Monte Carlo is its convergence speed, making it untractable in most practical cases. More recent methods aim at keeping the interesting properties of this technique while making it more efficient. Among the *variance reduction* family of methods, let us mention importance sampling, stratified sampling (e.g., latin hypercube sampling), control variates and multi-level Monte Carlo (see Chapters 8, 9 and 10 from Owen (2013) and (Giles, 2008)).

Quasi-Monte Carlo and Koksma-Hlawka inequality

Among the methods presented so far, classical deterministic quadratures are subject to the curse of dim while Monte Carlo methods deliver contrasted performances. Quasi-Monte Carlo is a deterministic family of numerical integration schemes over $[0, 1]^d$ with respect to the uniform measure on $[0, 1]$. It offers powerful performances with strong guarantees by choosing nodes respecting *low discrepancy* sequences.

The discrepancy of a set of nodes (or a design) can be seen as a metric of its uniformity. The lowest the discrepancy of a design is, the “closest” it is to uniformity.

The Koksma-Hlawka theorem (Leobacher and Pillichshammer, 2014; Morokoff and Caflisch, 1995) is a fundamental result for understanding the role of the discrepancy in numerical integration.

Theorem 2 (Koksma-Hlawka). *If $g : [0, 1]^d \rightarrow \mathbb{R}$ has a bounded variation (i.e., its total variation is finite), then for any design $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in [0, 1]^d$:*

$$\left| \int_{[0,1]^d} g(\mathbf{x}) d\mathbf{x} - \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}^{(i)}) \right| \leq V(g) D^*(\mathbf{X}_n). \quad (1.18)$$

Where $D^*(\mathbf{X}_n)$ is the star discrepancy of the design \mathbf{X}_n , while $V(g)$ quantifies the complexity of the integrand, which is related to its total variation. The reader might refer to [cite proof] for further mathematical proof.

Where the function variation $V(g)$ in the Eq. (4.8) can formally be defined as the Hardy-Klause variation:

$$V(g) = \sum_{u \subseteq \{1, \dots, p\}} \int_{[0,1]^u} \left| \frac{\partial^u g}{\partial \mathbf{x}^u}(\mathbf{x}_u, 1) \right| d\mathbf{x}_u. \quad (1.19)$$

Where the L_p star discrepancy of a design \mathbf{X}_n defined as the L_p -norm of the difference between the empirical CDF of the design $\widehat{F}_{\mathbf{X}_n}$ and the CDF of the uniform distribution F_U :

$$D_p^*(\mathbf{X}_n) = \|\widehat{F}_{\mathbf{X}_n} - F_U\|_p = \left(\int_{[0,1]^d} |\widehat{F}_{\mathbf{X}_n}(\mathbf{x}) - F_U(\mathbf{x})|^p d\mathbf{x} \right)^{1/p}. \quad (1.20)$$

Additionally, the L_∞ star discrepancy can be defined from a geometric point of view. Let us consider the number of a design \mathbf{X}_n , falling in a subdomain $[\mathbf{0}, \mathbf{x})$ as $\#(\mathbf{X}_n \cap [\mathbf{0}, \mathbf{x}))$. Then, this empirical quantification is compared with the volume of the rectangle $[\mathbf{0}, \mathbf{x})$, noted $\text{vol}([\mathbf{0}, \mathbf{x}))$. Finally, this star discrepancy is written:

$$D^*(\mathbf{X}_n) = \sup_{\mathbf{x} \in [0,1]^d} \left| \frac{\#(\mathbf{X}_n \cap [\mathbf{0}, \mathbf{x}))}{n} - \text{vol}([\mathbf{0}, \mathbf{x})) \right| \quad (1.21)$$

[Figure xx empirically represents discrepancy concept] Let us point out that this star discrepancy is equivalent to the Kolmogorov-Smirnov test verifying whether the design follows a uniform distribution.

One can notice how the Koksma-Hlawka inequality dissociates the quadrature performance into a contribution from the function complexity and one from the repartition of the quadrature nodes. Knowing that the complexity of the studied integrand is fixed, this property explains the motivation to generate low-discrepancy quadratures in numerical integration.

Note that the design can also be considered as a discrete distribution (uniform sum of Dirac distributions). The discrepancy can then be expressed as a probabilistic distance between

this discrete distribution and the uniform distribution. A generalized discrepancy between distributions will be presented in the [Part 2].

Some famous low-discrepancy sequences (e.g., van der Corput, Halton, Sobol', Faure, etc.) can offer a bounded star discrepancy $D^*(\mathbf{X}_n) \leq \frac{C \log(n)^d}{n}$, with C a constant depending on the sequence. Therefore, using these sequences as a quadrature rule with uniform weights provides the following absolute error upper bound:

$$\left| \int_{[0,1]^d} g(\mathbf{x}) d\mathbf{x} - \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}^{(i)}) \right| \leq \frac{V(g) \log(n)^d}{n} \quad (1.22)$$

The generation of these sequences doesn't necessarily require more effort than pseudo-random sampling. Chapter 15 in [Owen \(2013\)](#) offers an extended presentation of the ways to generate different low-discrepancy sequences. For example, the van der Corput and Halton sequences rely on congruential generators. To overcome the limits of Halton sequences, digital nets such as the famous Sobol' or Faure sequences have been developed. Sobol' sequences are in base two and have the advantage of being extensible in dimension. Note that by construction, these sequences offer significantly lower discrepancies for specific values. Typically designs with sizes equal to powers of two or power of prime numbers will be favorable. To illustrate the different patterns and properties of different methods, Fig. 1.5 represents the three designs of 256 points. Each is split into the first 128 points (in red) and the following 128 points (in black) to show the nested properties of the QMC sequences.

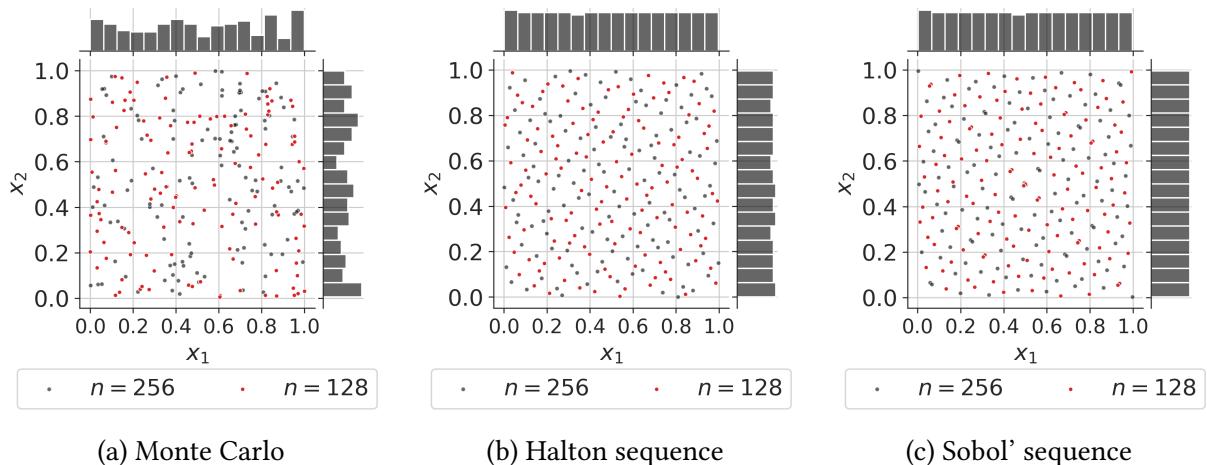


Fig. 1.5 Nested Monte Carlo and quasi-Monte Carlo designs ($n = 256$)

A quantity estimated by crude MC comes with some associated confidence. This complementary information is essential to deliver an end-to-end uncertainty quantification and misses in QMC methods. *Randomized quasi-Monte Carlo* (RQMC) is a method adding some randomness in QMC in order to compute confidence intervals while benefiting from a low variance. A specific review of the randomized (also called “scrambled”) QMC is proposed by [L'Ecuyer \(2018\)](#). Various authors recommend the use of RQMC by default instead of QMC as a good practice.

Recent works aim at exploring the use of these methods to estimate different quantities of interest (such as an expected value (Gobet et al., 2022) or a quantile (Kaplan et al., 2019))

Quasi-Monte Carlo methods easily generate powerful integration schemes. The KH inequality associates an upper bound and a convergence rate to most integrals. A randomization overlay fades the deterministic property of these designs to allow confidence interval computation. In the following, sampling techniques are presented from the numerical *design of experiments* point of view. Even if the finality might look different from the previous numerical integration, it shares many methods and concepts.

OpenTURNS 2 (Numerical integration). The following Python code proposes a minimalist OpenTURNS implementation to build a quadrature rule.

1.4.2 Numerical design of experiments

The numerical design of experiments aims at exploring uniformly the input domain, e.g., to build the learning set of a regression model, or to initialize a multi-start optimization strategy. A design of experience (also simply called design) is then qualified as *space-filling* when it properly covers a domain. As well as in integration, a design of experiments allows propagating uncertainties through a numerical model (or an actual experiment from a laboratory test bench). However, a difference comes from the fact that this community often works with designs of very limited sizes. Users of designs of experiments might also need to build designs with various properties.

- Some might be interested in the sequentiality of a sampling method, to eventually add new points as they get a computational budget extension.
- Some might request a sampling method conserving its properties in any sub-domains. This second property can be useful to reduce the problem's dimension by dropping a few unimportant marginals.

Different metrics are commonly used to quantify how space-filling a design of experiments is. The previously introduced different types of discrepancies are space-filling metrics. Other types of space-filling metrics rely on purely geometrical considerations.

This section will first define some space-filling metrics. Secondly, the *latin hypercube sampling* (LHS) will be introduced as a variance-reduction that became popular in this community. Finally, a general discussion on uncertainty propagation with respect to non-uniform measures will be presented.

Space-filling metrics and properties

Space-filling criteria are key to evaluating designs and are often used in their construction to optimize their performances. In the previous section, the star discrepancy was introduced as a

distance of a finite design to uniformity. However, the L_∞ star discrepancy is hard to estimate, fortunately, [Warnock \(1972\)](#) elaborated an explicit expression specific to the L_2 star discrepancy:

$$\left[D_2^*(\mathbf{X}_n)\right]^2 = \frac{1}{9} - \frac{2}{n} \sum_{i=1}^n \prod_{l=1}^d \frac{(1-x_l^{(i)})}{2} + \frac{1}{n^2} \sum_{i,j=1}^n \prod_{l=1}^d \left[1 - \max(x_l^{(i)}, x_l^{(j)})\right]. \quad (1.23)$$

One can notice that this expression is similar to the Cramér-von Mises test statistic. Even if this expression is tractable, [Fang et al. \(2018\)](#) detailed its limits. First, the star L_2 discrepancy generates designs that are not robust to projections in sub-spaces. Then, this metric is not invariant in rotation and reflection. Finally, by construction, L_p discrepancies give a special role to the point $\mathbf{0}$ by anchoring the box $[\mathbf{0}, \mathbf{x}]$.

Two improved criteria were proposed by [Hickernell \(1998\)](#) with the *centered L_2 discrepancy* and the *wrap-around L_2 discrepancy*. Those are widely used in practice since they solve the previous limits while satisfying the Koksma-Hlawka inequality with a modification of the total variation. Let us introduce the formula of the centered L_2 discrepancy:

$$\begin{aligned} CD_2^*(\mathbf{X}_n) = & \left(\frac{13}{12}\right)^d - \frac{2}{n} \sum_{i=1}^n \prod_{l=1}^d \left(1 + \frac{1}{2}|x_l^{(i)} - 0.5| - \frac{1}{2}|x_l^{(i)} - 0.5|^2\right) \\ & + \frac{1}{n^2} \sum_{i,j=1}^n \prod_{l=1}^d \left(1 + \frac{1}{2}|x_l^{(i)} - 0.5| + \frac{1}{2}|x_l^{(j)} - 0.5| - \frac{1}{2}|x_l^{(i)} - x_l^{(j)}|\right). \end{aligned} \quad (1.24)$$

As an alternative to discrepancies, many geometrical criteria exist to assess a space-filling design. The most common way to do so is to maximize the minimal distance among the pairs of Euclidian distances between the points of a design. The criterion to maximize is then simply called the *minimal distance* of a design (denoted ϕ_{min}). For numerical reasons, the ϕ_p criterion is often used instead of the minimal distance. The following ϕ_p criterion converges towards the minimum distance as $p \geq 1$ tends to infinity:

$$\phi_{min}(\mathbf{X}_n) = \min_{i \neq j} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2, \quad \phi_p(\mathbf{X}_n) = \sum_{i=1}^j \sum_{j=1}^n \left(|x^{(i)} - x^{(j)}|^{-p}\right)^{\frac{1}{p}}. \quad (1.25)$$

More space-filling criteria are reviewed in [Abtini \(2018\)](#). [\[Add reviews from the appendix A in Bertrand's book\]](#) Further relations between some mathematical objects related to space-filling are developed in [Pronzato and Müller \(2012\)](#). These space-filling metrics are widely used to optimize a different sampling technique.

Latin hypercube sampling

The LHS is a method introduced in 1979 ([McKay et al., 1979](#)), initially for numerical integration. This stratified sampling technique forces the distribution of each sub-projection of a bounded domain to be as uniform as possible To do so, for a n -sized design, each marginal's domain

is divided into n identical segments. This creates a regular grid of n^d squared cells over the domain.

An LHS design does not allow more than one point within a segment. That way, a new LHS can be built as a permutation of the marginals of an existing LHS. Inside each selected cell from the grid, the point can be placed in the center or randomly.

Various contributions provided first a variance, then a central limit theorem to the LHS ([Koehler and Owen, 1996](#)). Identically to the Monte Carlo variance [[add pointer](#)], LHS variance can be expressed as:

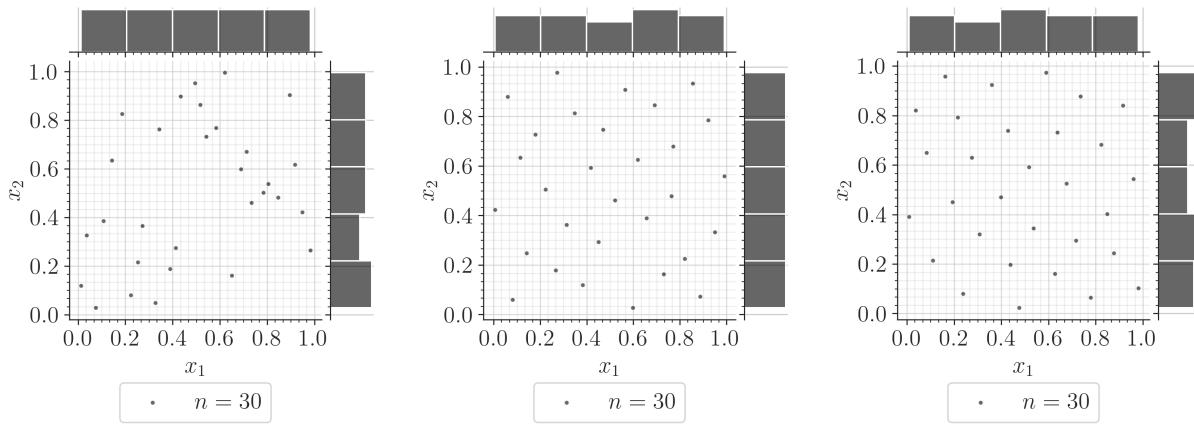
$$\text{Var} \left(\bar{y}_n^{\text{LHS}} \right) = \frac{1}{\sqrt{n}} \text{Var} (g(\mathbf{X})) - \frac{C}{n} + o \left(\frac{1}{n} \right). \quad (1.26)$$

Where C is a positive constant, showing that the LHS usually reduces the variance for numerical integration. Because of its stratified structure, LHS can generate poor designs from a space-filling point of view (see e.g., Figure [[add diagonal design](#)]). The following section presents various methods aiming at optimizing these designs.

Optimized Latin hypercube sampling

To improve the space-filling property of LHD, it is common to add an optimization step. The goal of this optimization is to improve a space-filling criterion by generating LHD from permutations of an initial LHD. [Damblin et al. \(2013\)](#) reviews LHS optimization using different discrepancy criteria and subprojection properties. This optimization can be performed by different algorithms, such as the stochastic evolutionary algorithm or simulated annealing. The results from this work show that LHD optimized by L_2 centered or wrap-around discrepancies offer strong robustness to two-dimensional projections. It also shows that these designs keep this property for dimensions larger than 10, while scrambled Sobol' sequences lose it.

More recent work developed different ways to get optimized LHD. Let us mention the maximum projection designs from [Joseph et al. \(2015\)](#) and the uniform projection designs from [Sun et al. \(2019\)](#). [[Add a sentence to introduce the papers](#)]



(a) Poorly space-filling LHS (b) L_2 centered optimized LHS (c) ϕ_p optimized LHS

Fig. 1.6 Latin hypercube designs with poor and optimized space-filling properties ($n = 8$)

OpenTURNS 3 (Design of experiments). The following Python code proposes a minimalist OpenTURNS implementation to build a LHS and a LHS optimized w.r.t. to a space-filling metric (here the L2-centered discrepancy) using the simulated annealing algorithm.

1.4.3 Summary and discussion

A wide panel of sampling techniques exists for numerical integration or design of experiments purposes. In both cases, the studied domain was bounded and the targeted measure was uniform. However, uncertainty propagation is often performed on complex input distributions, with possibly unbounded domains. In uncertainty quantification, this step might be referred to as the estimation of the output random variable's central tendency (i.e., its mean and variance). Central tendency estimation is a numerical integration with respect to any input distribution, also named *probabilistic integration* by [Briol et al. \(2019\)](#).

To generate i.i.d samples following any distribution (i.e., non-uniform), one may use *inverse transform* sampling. This method first generates a sample in the unit hypercube, then, the inverse CDF function (i.e., quantile function) is applied on marginals. Finally, possible dependence effects can be added using the Sklar theorem Eq. (1).

One may wonder if the properties from the uniform design are conserved after this nonlinear transformation. [Li et al. \(2020\)](#) explores this question from a discrepancy point of view. The authors find correspondences between discrepancies with respect to uniformity and discrepancy with respect to the target distribution. However this result show practical limits, sometimes making the interpretation of the last discrepancy easier. This question will be further discussed using a more general framework in the [Chapter xx].

Let us also remark that, depending on the distribution, defining the inverse CDF is not always possible. For example, samples following truncated distributions or mixture distributions might sometime be generated with a different technique. The *acceptance-rejection* method offers a versatile generation only based on the PDF f_x . Assuming that a well-known proposal PDF f_x^* exists such that $f_x \leq c \times f_x^*, c \in [1, +\infty]$. Then, one may generate a sample according to $c \times f_x^*$ and only retain from this sample the points under the PDF f_x . [\[add illustration with truncated normal and triangular.\]](#) Note that QMC sampling is not well suited with acceptance-rejection since its structure gets perturbed.

- [\[Markov Chain Monte Carlo: when it is hard to sample from the distribution.\]](#)
- [\[Add the summary table with the properties of each methods?\]](#)

In this section, many methods were presented to propagate input uncertainties against a deterministic function. The propagation with the three following goals and contexts were introduced:

- building a quadrature rule for numerical integration against a uniform distribution,

- creating a space-filling design of experiments to uniformly explore the space, often in a small data context (e.g., to build the learning set of a surrogate model),
- generating a design for central-tendency estimation, which is simply a numerical integration against a non uniform density.

These three objectives have been explored in different communities but actually mostly share similar methods. They all have in common the general analysis (i.e., global behavior) of the output random variable. However, some studies require to shift the focus on specific areas of the output random variables. When using uncertainty propagation to perform a risk analysis, the events studied are often contained in the tails of the output distribution. In this case, dedicated uncertainty propagation methods will significantly improve the estimation of the associated statistical quantities.

1.5 Reliability-oriented uncertainty propagation

This section aims at presenting another type of uncertainty propagation. In the context of a risk analysis applied to the engineering field, the reliability of a system needs to be assessed. Most often, a risk measure associated with a failure mode of the studied system is estimated.

Since most systems studied in risk analysis should be highly reliable, the occurrence of such event is qualified as rare. Only an unlikely small amount of extreme input conditions or an unlikely unfavorable combinations of inputs lead to the failure of the system. Hence the usage of the equivalent terms *reliability analysis* and *rare event estimation*. The notion of risk associated with an event is often decomposed as a product of likelihood and impact. The failure of a system might be very rare but its consequences can be severe (e.g., civil engineering structures, nuclear infrastructure, telecommunication networks, electrical grid, railway signalling, etc.).

Different risk measures (i.e., quantities of interest) can be studied depending on the type of risk analysis. Quantiles are a first conservative measure, widely used for risk analysis. The α -quantile q_α of the output random variable Y is defined as:

$$q_\alpha = \inf_{y \in \mathbb{R}} \{F_Y(y) \geq \alpha\}, \quad \alpha \in [0, 1]. \quad (1.27)$$

[define superquantiles in text?] As an alternative, one can define a scalar safety threshold y_{th} that should not be exceeded to keep the system safe. Then, a second risk measure is probability of exceeding this safety threshold, also called *failure probability*:

$$p_f = \mathbb{P}(Y \geq y_{\text{th}}), \quad y_{\text{th}} \in \mathbb{R}. \quad (1.28)$$

To illustrate this quantity, Fig. 1.7 shows the one-dimensional propagation of a normal distribution (represented by the PDF on the left), through a function $g(\cdot)$. The probability of exceeding a given threshold y_{th} is represented by the area in red under the output PDF on top.

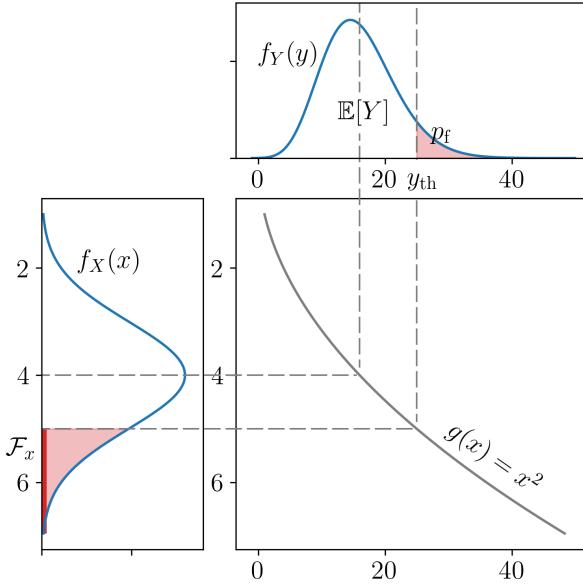


Fig. 1.7 One-dimensional reliability analysis example

An interesting reflection on the use and the interpretation of risk measures⁴ is presented in Rockafellar and Royset (2015).

In the following, the formalism for reliability analysis problems will be first presented, then the main methods solving this specific problem will be introduced. Note that the present work will not address the problems of time-dependent reliability analysis tackled in Hawchar et al. (2017).

1.5.1 Problem formalization

Following to the UQ methodology, the behavior of the system is modeled by $\mathcal{M}(\cdot)$. Considering the problem of exceeding a safety threshold in [add pointer], the system's performance is commonly defined as the difference between the model's output and a safety threshold $y_{\text{th}} \in \mathbb{R}$. Formally, the *limit-state function* (LSF) is a deterministic function $g : \mathbb{R} \rightarrow \mathbb{R}$ quantifying this performance:

$$g(\mathbf{x}) = y_{\text{th}} - \mathcal{M}(\mathbf{x}). \quad (1.29)$$

Depending on the sign of its images, this function splits the inputs space into two disjoint and complementary domains called the *failure domain* $\mathcal{F}_{\mathbf{x}}$, and the *safe domain* $\mathcal{S}_{\mathbf{x}}$ which are defined as:

$$\mathcal{F}_{\mathbf{x}} = \{\mathbf{x} \in \mathcal{D}_{\mathbf{x}} \mid g(\mathbf{x}) \leq y_{\text{th}}\}, \quad \mathcal{S}_{\mathbf{x}} := \{\mathbf{x} \in \mathcal{D}_{\mathbf{x}} \mid g(\mathbf{x}) > y_{\text{th}}\}. \quad (1.30)$$

The border between these two domains is a hypersurface called *limit-state surface* (LLS), defined by $\mathcal{F}_{\mathbf{x}}^0 := \{\mathbf{x} \in \mathcal{D}_{\mathbf{x}} \mid g(\mathbf{x}) = 0\}$. Similarly to any UQ study around a numerical model, this problem can require to be resolved using a limited numbers of calls to a black-box simulator. The difficulties of a reliability problem might come from the properties of the LSF: nonlinear, costly to evaluate

⁴Including measures from the finance domain such as the *conditional value-at-risk* (also called superquantile).

or with a multimodal failure domain. Additionally, note that the reliability problem can be the composition of multiple reliability problems, often modeled as system of problems in series and parallel.

A rare event estimation results from a particular uncertainty propagation through the LSF. Considering the resulting output variable of interest $g(\mathbf{X})$, its probability of being negative (i.e., in the failure domain) is a common risk measure [add pointer]. The commonly named *failure probability*, denoted p_f , will be our quantity of interest in reliability analysis. This quantity is formally written⁵:

$$\begin{aligned} p_f &:= \mathbb{P}(Y \geq y_{\text{th}}) = \mathbb{P}(g(\mathbf{X}) \leq 0) \\ &= \int_{\mathcal{F}_x} f_x(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{D}_x} \mathbb{1}_{\mathcal{F}_x}(\mathbf{x}) f_x(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (1.31)$$

were the indicator function applied to the failure domain returns $\mathbb{1}_{\{\mathcal{F}_x\}}(x) = 1$ if $x \in \mathcal{F}_x$ and $\mathbb{1}_{\{\mathcal{F}_x\}}(x) = 0$ otherwise. Rare event estimation implies both contour finding (i.e., characterizing the LSF) and an estimation strategy targeting the failure domain (often with a limited number of simulations). Note that failure event are qualified as rare when its failure probability has an order of magnitude between $10^{-2} \leq p_f \leq 10^{-9}$ (see e.g., Lemaire (2013)).

Instead of directly performing a reliability analysis in the physical space (i.e., \mathbf{x} -space), these problems are usually solved in the *standard normal space* (i.e., \mathbf{u} -space). Working in the standard space reduces numerical issues potentially caused by unscaled or asymmetric marginals. Moreover, a larger panel of methods can be applied in the standard space since the random inputs are independent. The bijective mapping between these two spaces is called an “iso-probabilistic transformation”, denoted $T : \mathcal{D}_x \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^d, \mathbf{x} \mapsto T(\mathbf{X}) = \mathbf{u} = (u_1, \dots, u_d)^\top$. When considering any random vector $\mathbf{X} = (X_1, \dots, X_d)^\top$ and the independent standard Gaussian vector $\mathbf{U} = (U_1, \dots, U_d)^\top$, the following equalities hold:

$$\mathbf{U} = T(\mathbf{X}) \Leftrightarrow \mathbf{X} = T^{-1}(\mathbf{U}). \quad (1.32)$$

A reliability problem can be expressed in the standard normal space. Let us first consider the transformed limit-state function \check{g} defined as:

$$\check{g} : \begin{cases} \mathbb{R}^d &\longrightarrow \mathbb{R} \\ \mathbf{u} &\longmapsto \check{g}(\mathbf{u}) = (g \circ T^{-1})(\mathbf{u}). \end{cases} \quad (1.33)$$

⁵Note that this probabilistic integration is usually written using the PDF $f_x(\cdot)$ but it could identically be expressed in terms of probability measure by taking $f_x(\mathbf{x}) d\mathbf{x} = d\mu(\mathbf{x}), \forall \mathbf{x} \in \mathcal{D}_x$.

Since this transformation is a diffeomorphism⁶, one can apply the change of variable $\mathbf{x} = T(\mathbf{u})$ to express the reliability problem from [add pointer] in the standard space:

$$p_f = \mathbb{P}(\check{g}(\mathbf{U}) \leq 0) = \int_{\mathcal{F}_u} \varphi_d(\mathbf{u}) d\mathbf{u} = \int_{\mathbb{R}^d} \mathbb{1}_{\mathcal{F}_u}(\mathbf{u}) \varphi_d(\mathbf{u}) d\mathbf{u}, \quad (1.34)$$

with the transformed failure domain noted $\mathcal{F}_u = \{\mathbf{u} \in \mathbb{R}^d \mid \check{g}(\mathbf{u}) \leq 0\}$, and the d -dimensional standard Gaussian PDF $\varphi_d(\mathbf{u}) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{\|\mathbf{u}\|_2^2}{2}\right)$. The fact that the failure probability is invariant by this transformation allows the analyst to estimate this quantity in both spaces.

Different types of transformations exist, such as the Rosenblatt or the generalized Nataf transformation (Lebrun, 2013). In practice, the transformation choice depends on the properties of the input distribution studied. [When should we use what? (see Bourinet (2021), Chapt. 1). In the case of non-elliptical distribution, the default method is the Rosenblatt transform on OpenTURNS.]

1.5.2 Rare event estimation methods

The main risk measure chosen for rare event estimation in this work is the previously introduced failure probability. Therefore, let us recall that the goal is to build an efficient estimation (or approximation) of the following d -dimensional integral:

$$p_f = \int_{\mathcal{D}_x} \mathbb{1}_{\mathcal{F}_x}(\mathbf{x}) f_x(\mathbf{x}) d\mathbf{x} \quad (1.35)$$

In the context of rare event estimation using costly to evaluate numerical models, the simulation budget is often limited to n runs with $p_f \ll \frac{1}{n}$. Which explains the need for specific methods offering approximations or simulations targeting the unknown failure domain. Two types of rare event estimation methods are classically presented: first, using approximation approaches, second, using sampling techniques. This section introduced the commonly used rare event methods, see Morio and Balesdent (2015) for a more exhaustive review.

First and second order reliability methods (FORM/SORM)

The so-called First and second order reliability methods (FORM and SORM) both rely on a geometric approximation to estimate a failure probability. They extrapolate a local approximation of the LSF built in the vicinity of a *most-probable-failure-point* (MPFP), also called *design point*. [add some refs]

Working in the standard space, the methods first look for this MPFP, denoted P^* , with coordinates \mathbf{u}^* . To find it, one can solve the following quadratic optimization problem:

$$\mathbf{u}^* = \arg \max_{\mathbf{u} \in \mathbb{R}^d} (\mathbb{1}_{\mathcal{F}_u}(\mathbf{u}) \varphi_d(\mathbf{u})). \quad (1.36)$$

⁶Considering two manifolds A and B , a transformation $T : A \rightarrow B$ is called a diffeomorphism if it is a differentiable bijection with a differentiable inverse $T^{-1} : B \rightarrow A$.

Using the properties of the standard space allows us to rewrite it as:

$$\mathbf{u}^* = \arg \max_{\mathbf{u} \in \mathbb{R}^d} \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{\mathbf{u}^\top \mathbf{u}}{2}\right) \quad \text{s.t. } \mathbf{u} \in \mathcal{F}_u \quad (1.37)$$

$$= \arg \min_{\mathbf{u} \in \mathbb{R}^d} \mathbf{u}^\top \mathbf{u} \quad \text{s.t. } \check{g}(\mathbf{u}) \leq 0. \quad (1.38)$$

This problem becomes a quadratic optimization under nonlinear constraint. It is classically solved by gradient decent algorithms (e.g., AbdoRackwitz algorithm [add ref]) but can also use gradient-free techniques (e.g., Cobyla algorithm [add ref]). This point (assuming that it is unique) defines the smallest Euclidian distance between the LSS and the origin of the standard space. To understand its role in the reliability problem, let us recall that the density of the standard normal present an exponential decay in its radial and tangential direction. Then, P^* is the point with the biggest contribution to the failure probability [pointer].

This distance between the origin and P^* is a different risk measure, introduced as the *Hasofer-Lind reliability index* [add ref], $\beta \in \mathbb{R}$ such that:

$$\beta = \|\mathbf{u}^*\|_2 = \boldsymbol{\alpha}^\top \mathbf{u}^*, \quad \text{s.t. } \boldsymbol{\alpha} = \frac{\nabla_{\mathbf{u}} \check{g}(\mathbf{u})}{\|\nabla_{\mathbf{u}} \check{g}(\mathbf{u})\|_2}. \quad (1.39)$$

The vector $\boldsymbol{\alpha}$ is the unit vector pointing at P^* from the origin point.

Then, FORM aims at approximating the limit-state function $\check{g}(\cdot)$ by its first-order Taylor expansion around the MPFP, denoted $\check{g}_1(\mathbf{u}^*)$:

$$\begin{aligned} \check{g}(\mathbf{u}) &= \check{g}_1(\mathbf{u}^*) + o(\|\mathbf{u} - \mathbf{u}^*\|_2^2) \\ &= \check{g}(\mathbf{u}^*) + \nabla_{\mathbf{u}} \check{g}(\mathbf{u}^*)^\top (\mathbf{u} - \mathbf{u}^*) + o(\|\mathbf{u} - \mathbf{u}^*\|_2^2) \\ &= \|\nabla_{\mathbf{u}} \check{g}(\mathbf{u})\|_2 (\boldsymbol{\alpha}^\top \mathbf{u}^* - \boldsymbol{\alpha}^\top \mathbf{u}) + o(\|\mathbf{u} - \mathbf{u}^*\|_2^2) \end{aligned} \quad (1.40)$$

Using $\check{g}_1(\cdot)$ as approximation of the LSF, the failure probability can be approximated as:

$$p_f \approx p_f^{\text{FORM}} = \mathbb{P}(-\boldsymbol{\alpha}^\top \mathbf{u} \leq -\beta) = \Phi(-\beta), \quad (1.41)$$

with $\Phi(\cdot)$ the CDF of the standard Gaussian. Depending on the properties of the LFS, this approximation will be more or less accurate. Note that for a linear LFS, $p_f = p_f^{\text{FORM}}$. When the function is nonlinear, adding a quadratic term to the Taylor expansion can help the approximation. The approximation method is then called SORM for *second order reliability method*. However, this added complexity implies the computation of Hessian matrices, which can be complicated (see Chapter 1 from Bourinet (2018) for their estimation).

When the MPFP is not unique, the application of these methods might lead to important errors. From a geometrical point of view, having more than one MPFP means that more than one failure zones are at the same euclidean distance of the origin. Applying a FORM or SORM resolution in this particular case leads to the estimation of only one of the failure zones. The *multi-FORM* algorithm developed by [add ref] prevents this situation by applying successive

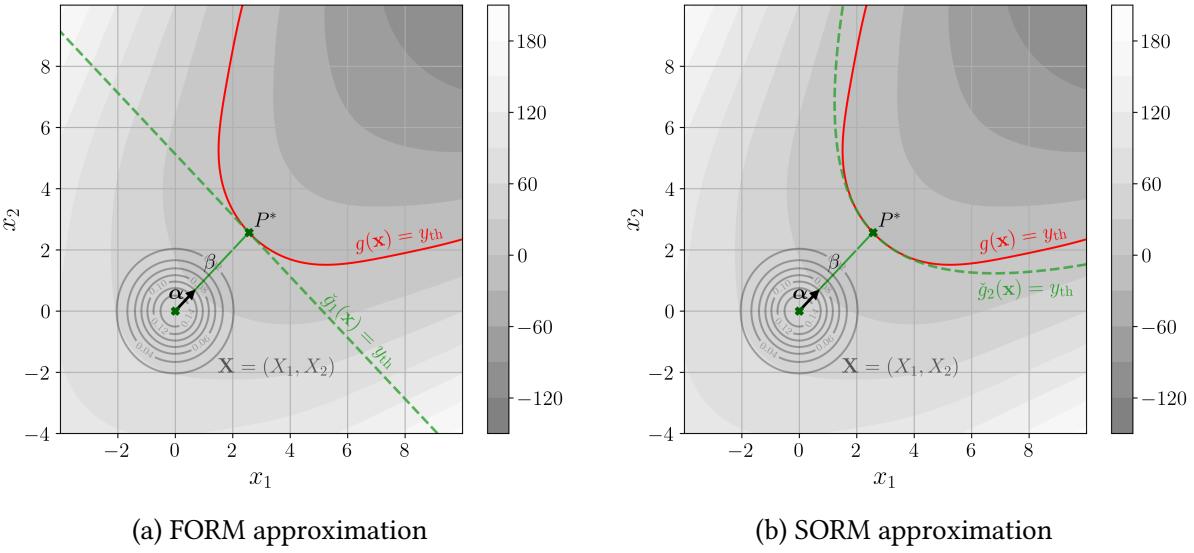


Fig. 1.8 FORM and SORM approximation on a two-dimensional example

FORM. Once the first MPFP $P^{*(1)}$ found, the LSS is modified by removing a nudge to find to following MPFP $P^{*(2)}$.

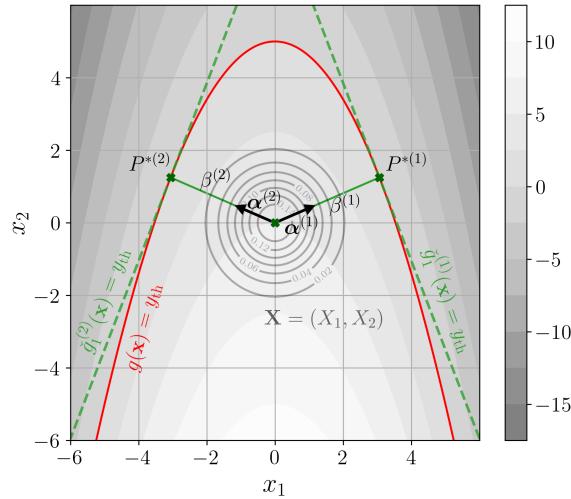


Fig. 1.9 Multi-FORM approximation on an example with two MPFPs

Overall, FORM and SORM methods deliver a very efficient approximation of small probabilities for relatively simple problems (in terms of linearity and dimension). For this reason, they have been widely used in the practical context of limited simulation budget. However, these methods present serious limits as the dimension increases (see the discussion in [Chabri-don] Chapt. 1). Additionally, their main drawback is the lack of complementary information concerning the confidence of the results. The example from the [Fig xx] has shown that the method might miss some important areas of the failure domain, leading to poor estimations. As an alternative to approximation methods, simulation-based methods often provide to the analyst an assessment of the estimation's confidence.

Monte Carlo

Crude Monte Carlo sampling is a universal and empirical method for uncertainty propagation. As introduced earlier, it relies on the pseudo-random generation of a i.i.d. sample $\{\mathbf{x}^{(i)}\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} f_{\mathbf{x}}$. Only the estimator is written using the indicator function with the LSF:

$$p_f \approx \hat{p}_f^{\text{MC}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}^{(i)}) \quad (1.42)$$

Provided that the failure probability is bounded, this estimator converges towards it almost surely according to the LLN. Once again, Monte Carlo offers a nonbiased estimator, regardless of the problem's dimension or regularity of the function $g(\cdot)$. Additionally, the variance of this estimator is fully known:

$$\text{Var}(\hat{p}_f^{\text{MC}}) = \frac{1}{n} p_f (1 - p_f) \quad (1.43)$$

The variance of this estimator can be used to build its confidence interval according to the TCL [pointer to the previous IC]. Because of the small scale of the quantities manipulated in rare event estimation, the estimator's coefficient of variation is also widely used:

$$\delta_{\hat{p}_f^{\text{MC}}} = \frac{\sqrt{\text{Var}(\hat{p}_f^{\text{MC}})}}{\mathbb{E}[\hat{p}_f^{\text{MC}}]} = \sqrt{\frac{1 - p_f}{np_f}}. \quad (1.44)$$

On paper, Monte Carlo estimator presents multiple advantages for rare event estimation. First, this method can be applied directly in the physical space, without transformation (which is practical for complex input distributions). Second, it does not suffer from the curse of dimensionality. Thrid, it is qualified as embarrassingly parallel method since each numerical simulations are independent. Finally, it offers strong convergence guaranties and complementary information on the estimation confidence. [These properties make it the reference method.]

However, these advantages of this estimator are shadowed by its slow convergence. To estimate a target failure probability $p_f = 10^{-\alpha}$, a Monte Carlo estimation with a convergence level $\delta_{\hat{p}_f^{\text{MC}}} = 0.1$ famously requires $n = 10^{\alpha+2}$ simulations.

In the context of rare event estimation, Monte Carlo needs a number of simulation that is often prohibitive in practice. This excessive simulation budget comes from the fact that the vast majority of the samples drawn from the input distribution are not in the failure domain.

Importance sampling

Importance sampling (IS) is a variance reduction method, aiming at improving the performances of crude Monte Carlo sampling. In the context of rare event estimation, the main idea is to deliberately introduce a bias in the sampled density, shifting it towards the failure domain. If this shift actually goes towards the failure domain, it allows to draw more points in it, leading to a better estimate of our quantity.

The challenge in importance sampling is to pick a relevant *instrumental* distribution h_X (also called *auxiliary* distribution) to replace the distribution f_X . Then, by introducing the fully known likelihood ratio $w_X(x) = \frac{f_X(x)}{h_X(x)}$, one can rewrite $f_X(x) = w_X(x)h_X(x)$ and inject it in the failure probability expression:

$$p_f = \int_{\mathcal{D}_x} \mathbb{1}_{\mathcal{F}_x}(x) f_X(x) dx = \int_{\mathcal{D}_x} \mathbb{1}_{\mathcal{F}_x}(x) w_X(x) h_X(x) dx \quad (1.45)$$

This simple writing trick allows us to integrate against the auxiliary distribution. With a Monte Carlo method, this task should be easier than integrating directly against the initial distribution.

The importance sampling estimator of the failure probability is defined for a sample drawn on the auxiliary distribution $\{\mathbf{x}^{(i)}\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} h_X$:

$$\hat{p}_f^{\text{IS}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\mathcal{F}_x}(\mathbf{x}^{(i)}) w_X(\mathbf{x}^{(i)}). \quad (1.46)$$

This estimator is unbiased and its variance is defined as:

$$\text{Var}(\hat{p}_f^{\text{IS}}) = \frac{1}{n} \left(\mathbb{E}_{h_X} \left[(\mathbb{1}_{\mathcal{F}_x}(\mathbf{X}) w_X(\mathbf{X}))^2 \right] - p_f^2 \right). \quad (1.47)$$

The quality of the variance reduction in this method fully depends on the choice of the instrumental distribution. An optimal instrumental distribution h_X^* theoretically gives the smallest variance by setting it equal to zero in [pointer]:

$$h_X^*(\mathbf{x}) = \frac{\mathbb{1}_{\mathcal{F}_x}(\mathbf{x}) f_X(\mathbf{x})}{p_f}. \quad (1.48)$$

This optimal results presented in [add ref] is unfortunately not usable in practice since it uses the targeted quantity p_f . Knowing this framework, various techniques try to define instrumental distributions as close as possible to this theoretical result. Three main importance sampling techniques are used in this work.

[First, FORM-iS]

[Second AIS-CE]

[Finally, NAIS]

Appendix C develops an algorithmic presentation of the two last techniques: NAIS and AIS-CE.

[Discussion on commun advantages and drawback of the IS methods.]

Subset sampling

Subset sampling splits the failure event \mathcal{F}_x into an intersection of $k_\#$ intermediary events $\mathcal{F}_x = \cap_{k=1}^{k_\#} \mathcal{F}_{[k]}$. Each are nested such that $\mathcal{F}_{[1]} \supset \dots \supset \mathcal{F}_{[k_\#]} = \mathcal{F}_x$. The failure probability is then

expressed as a product of conditional probabilities:

$$p_f = \mathbb{P}(\mathcal{F}_x) = \mathbb{P}(\cap_{k=1}^{k_\#} \mathcal{F}_{[k]}) = \prod_{k=1}^{k_\#} \mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}). \quad (1.49)$$

From a practical point of view, the analyst tunes the algorithm by setting the intermediary probabilities $\mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}) = p_0, \forall k \in \{1, \dots, k_\#\}$. Then, the corresponding quantiles $q_{[1]}^{p_0} > \dots > q_{[k_\#]}^{p_0}$ are estimated for each conditional subset samples $X_{[k],N}$ of size N . Note that the initial quantile is estimated by crude Monte Carlo sampling on the input PDF f_x . Following conditional subset samples are generated by *Monte Carlo Markov Chain* (MCMC) sampling of $f_x(x | \mathcal{F}_{[k-1]})$, using as seeds initialisation points the $n = Np_0$ samples given by $A_{[k],n} = \{X_{[k-1]}^{(j)} \subset X_{[k-1],N} | g(X_{[k-1]}^{(j)}) > \hat{q}_{[k-1]}^\alpha\}_{j=1}^n$. This process is repeated until an intermediary quantile exceeds the threshold: $\hat{q}_{[k_\#]}^{p_0} < y_{th}$. Finally, the failure probability is estimated by:

$$p_f \approx \hat{p}_f^{SS} = p_0^{k_\#-1} \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{\{g(x) \leq y_{th}\}}(X_{[k_\#],N}^{(j)}). \quad (1.50)$$

In practice, the subset sample size should be large enough to properly estimate intermediary quantiles, which leads [Au and Beck \(2001\)](#) to recommend setting $p_0 = 0.1$. SS efficiency depends on the proper choice and tuning of the MCMC algorithm ([Papaioannou et al., 2015](#)). [introduce the upper bound of the coefficient of variation and its limits.] [MCMC implies multiple tuning and loses the independent property of the generated samples.]

[Mention the appendix [Appendix C](#) presenting the algorithms of SS]

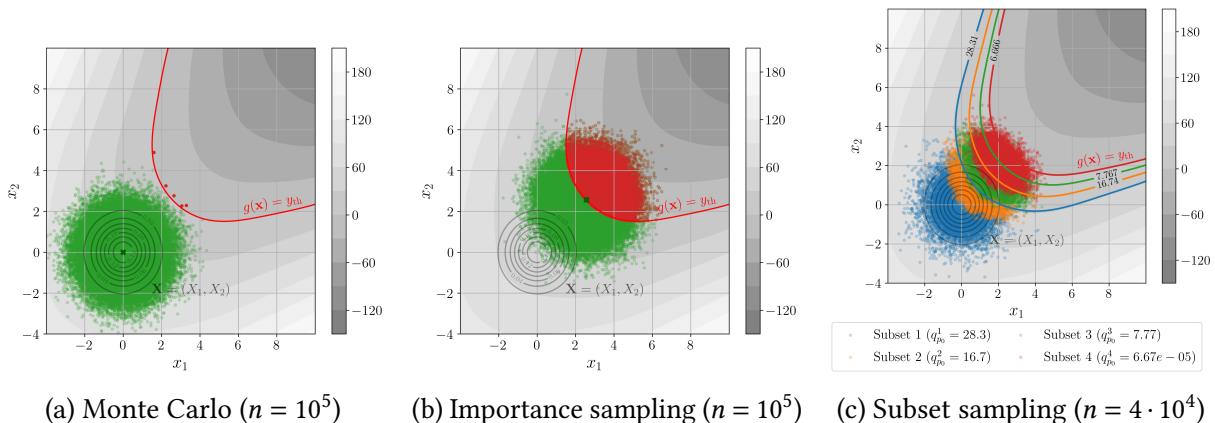


Fig. 1.10 Illustration of a rare event estimation

1.5.3 Summary and discussion

[Discussion: Which strategy to solve these problems?] [The design point does not have the same meaning in high dimension] [Nonlinearities of the LSF are the actual challenge in reliability analysis]

OpenTURNS 4 (Rare event estimation). The following Python code proposes a minimalist OpenTURNS implementation of rare event estimation algorithms.

1.6 Sensitivity analysis

[An inversion problem]

1.6.1 Global sensitivity analysis

[Screening]

- Morris method
- Derivative-based global sensitivity measures
- HSIC

[Importance measures]

- Sobol'
- Shapley

1.6.2 Reliability-oriented sensitivity analysis

- Importance factors from FORM
- Sobol on the indicator function (see Chabridon 2018 chapt 4 and 7, see Papaioannou 2023)

1.7 Surrogate modeling

1.7.1 Common framework

The aim of *surrogate modeling* (or metamodeling) is to build a cheap-to-call statistical model, denoted $\hat{g}_n(\cdot)$, replacing a costly numerical model $g(\cdot)$ over the input domain \mathcal{D}_X . To do so, a statistical learning is performed on a finite number of observations of the costly function g . When manipulating computationally expensive simulations, its size can be limited (i.e., small-data context). This n -sized set is usually called *learning set* denoted:

$$\{\mathbf{X}_n, \mathbf{y}_n\} = \left\{ \mathbf{x}^{(i)}, y^{(i)} \right\}_{i=1}^n = \left\{ \mathbf{x}^{(i)}, g(\mathbf{x}^{(i)}) \right\}_{i=1}^n. \quad (1.51)$$

A very large catalogue of regression methods exist, here is a list of the most encountered ones in the field of UQ: generalized linear regression, polynomial chaos expansion (PCE) (Blatman and Sudret, 2011; Soize and Ghanem, 2004), support vector machine (SVM) (Cortes and Vapnik, 1995), Gaussian processes (GP) (Rasmussen and Williams, 2006), low-rank tensors, and artificial neural network (ANN) (Hastie et al., 2009). The following section will provide a short focus on Gaussian process regression.

Validating the accuracy and precision of a surrogate model is an important step to guarantee its fidelity with regard to the numerical model. When a m -sized input-output set is dedicated to validating the surrogate model, independently from the learning set, it is called *test set* and denoted $\{\mathbf{X}_m, \mathbf{y}_m\} = \left\{ \mathbf{x}^{(i)}, g(\mathbf{x}^{(i)}) \right\}_{i=1}^m$. Note that the analyst may work in two different frameworks, affecting the regression and validation method's choice:

- Given-data context: only using a fixed input-output dataset to build and validate the surrogate model.
- Computer experiment context: allowing to generate simulated data points (often at an important cost).

Validating surrogate models in small-data context appears to be an important challenge. Different validation criteria and techniques exist. The *coefficient of validation*, denoted R^2 , is a first validation metric that can be directly computed on the learning set:

$$R^2(\hat{g}_n) = 1 - \frac{\sum_{i=1}^n (y(\mathbf{x}^{(i)}) - \hat{g}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^n (y(\mathbf{x}^{(i)}) - \bar{y}_n)^2}, \quad (1.52)$$

where $\bar{y}_n = (1/n) \sum_{i=1}^n y^{(i)}$ denotes the empirical mean of the observations in the test sample. However, these metrics are not relevant for every regression method (typically, interpolant method have a $R^2 = 1$). The *predictivity coefficient* is an alternative defined as a normalized *integrated square error* (ISE):

$$Q^2(\hat{g}_n) = 1 - \frac{\text{ISE}(\hat{g}_n)}{\text{Var}(\hat{g}_n)}, \quad (1.53)$$

where

$$\text{ISE}(\widehat{g}_n) = \int_{\mathcal{D}_X} (g(\mathbf{x}) - \widehat{g}(\mathbf{x}))^2 d\mathbf{x}, \quad \text{Var}(\widehat{g}_n) = \int_{\mathcal{D}_X} (g(\mathbf{x}) - \widehat{g}(\mathbf{x}))^2 d\mathbf{x}. \quad (1.54)$$

This quantity can be estimated on a test set $\{\mathbf{X}_m, \mathbf{y}_m\}$:

$$\widehat{Q}^2(\widehat{g}_n) = 1 - \frac{\sum_{i=1}^m (y(\mathbf{x}^{(i)}) - \widehat{g}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^m (\bar{y}_m - y(\mathbf{x}^{(i)}))^2}. \quad (1.55)$$

Note about the interpretation of the two criteria, the higher the value, the better the quality of the fit.

Validating a surrogate model with an independent test-set is sometimes called *Holdout* validation. In a small-data context, dedicating an independent test set to validation might be impossible. Then, *cross-validation* is a generic estimation strategy allowing to use only one sample for learning and testing. The most common cross-validation method is the *k-fold* validation, illustrated in Fig. 1.11. The idea is first to split the n -sized dataset in several equal parts, called folds. A first surrogate can be fitted on all the dataset but the first fold, on which a validation is estimated. The operation is repeated for each fold, providing a virtual validation on the entire dataset. Leave-One-Out validation (LOO) is an extreme case of *k*-fold cross-validation, for which $k = n - 1$. Note that multiple variations of these methods exist, adding for example a permutation or shuffling step.

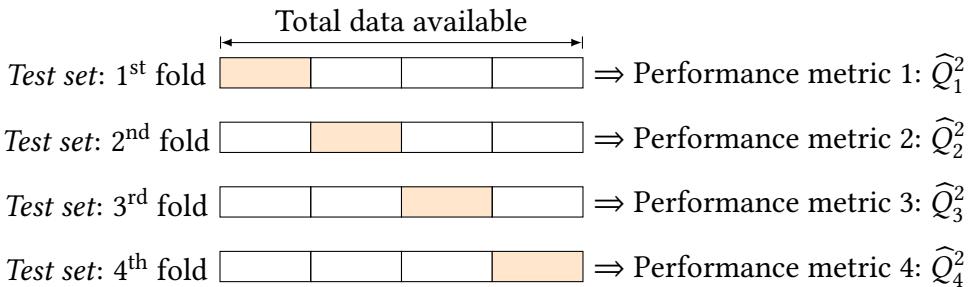


Fig. 1.11 Illustration of a k -fold cross-validation (with $k = 4$)

1.7.2 General purposes surrogate model

In this section, a particular focus is put of Gaussian process (GP) regression (also named kriging after the geostatistician D.G. Krige). Gaussian processes are a widely used regression method in UQ for their performance, flexibility and their associated confidence model. In a small-data context, the way of placing the few points forming the surrogate's learning set is critical. Intuitively, to build a versatile surrogate model, it should get information covering the input space as uniformly as possible. Which is why space-filling designs of experiments are commonly used to build learning sets. In practice, QMC and optimized LHS design introduced in [add pointer] are widely used.

Gaussian process regression

Considering a learning set \mathbf{X}_n , Gaussian process regression aims at approximating the function $g(\cdot)$ by a scalar Gaussian process, conditionned on a set of obervations $\mathbf{y}_n = \{g(\mathbf{x}^{(i)})\}$. Let us first define a Gaussian process prior structure ξ on the function approximated $g(\cdot)$ with a mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$:

$$\xi \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \quad (1.56)$$

with a:

- *trend model*: $m(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta}$, composed of a functional basis $\mathbf{f} = (f_1, \dots, f_d)^\top$ and a vector of coefficients $\boldsymbol{\beta} = (\beta_1, \dots, \beta_d)^\top$,
- *covariance model*: $k(\mathbf{x}, \mathbf{x}')$, usually taken stationary, such that $k(\mathbf{x}, \mathbf{x}') = \sigma^2 k_s(\mathbf{x} - \mathbf{x}', \boldsymbol{\theta})$ with $\sigma^2 > 0$ and $\boldsymbol{\theta} \in \mathcal{D}_b X$.

The trend model of a GP defines its general tendency, while the covariance model influences its regularity. The method takes different names depending on the knowledge of the trend model. It is called “simple kriging” when the trend is fully known, “ordinary kriging”, when the trend is unknown but supposed constant and “universal kriging” otherwise. Note that [Schobi et al. \(2015\)](#) introduced a hybrid method named PC-Kriging setting a PCE as the trend of a kriging model.

To ease the presentation, let us first consider the hyperparameters $\sigma, \boldsymbol{\theta}$ fully known and a zero trend $\boldsymbol{\beta} = \mathbf{0}$. At a given point $\mathbf{x} \in \mathcal{D}_b X$ the realization of the GP is a Gaussian random variable $\xi(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$. Working with Gaussian variables allows to easily write conditioning formulas between $\xi(\mathbf{x})$ and the observations \mathbf{y}_n . This Gaussian variable $\xi(\mathbf{x})$ conditionned on the observations \mathbf{y}_n is sometimes called conditional posterior $\xi_n(\mathbf{x}) := (\xi(\mathbf{x}) | \mathbf{y}_n) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$. The well-known “Kriging equations” (see e.g., [Rasmussen and Williams \(2006\)](#)) offer its explicit expression:

$$\begin{cases} \eta_n(\mathbf{x}) &:= \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{y}_n \\ s_n^2(\mathbf{x}) &:= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \end{cases} \quad (1.57)$$

where $\mathbf{k}(\mathbf{x})$ is the column vector of the covariance kernel evaluations $[k(\mathbf{x}, \mathbf{x}^{(1)}), \dots, k(\mathbf{x}, \mathbf{x}^{(n)})]$ and \mathbf{K} is the $(n \times n)$ variance-covariance matrix such that the (i, j) -element is $\{\mathbf{K}\}_{i,j} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

In practice the surrogate model is defined by the *predictor* function $\eta_n(\cdot)$. This regression model provides an important complementary information with the *kriging variance* $s_n^2(\mathbf{x})$, reaching zero at the learing points. Let us remark that the kriging variance fully depends on the covariance model (defined by its parametric structure and hyperparameters). In practice, the hyperparameters are unknown, therefore, their estimation is a key step in the construction of a kriging model. This estimation can be done using different approaches, most commonly using the maximum likelihood method or a cross-validation.

[Comment the following figure]

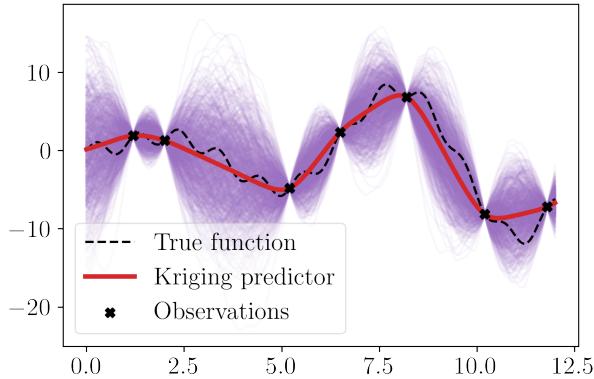


Fig. 1.12 Illustration of an ordinary kriging model fitted on a limited set of observations ($n = 7$)

Associated with kriging models, another validation criterion is relevant to evaluate the kriging variance $s_n^2(\mathbf{x})$. The predictive variance adequation (PVA) has been introduced by Bachoc (2013) to confirm that the kriging variance is reliable. For a validation performed by holdout, using an independant m -sized test set, the PVA is defined as:

$$\text{PVA} = \left| \log \left(\frac{1}{m} \sum_{i=1}^n \frac{(y(\mathbf{x}^{(i)}) - \hat{g}(\mathbf{x}^{(i)}))^2}{s_n^2(\mathbf{x}^{(i)})} \right) \right|. \quad (1.58)$$

The smaller this quantity get, the better the quality of the kriging variance.

Gaussian process regression is an elegant solution, offering a lot of flexibility and an associated error model (i.e., the kriging variance). However, well known numerical issues appear during the estimation of the hyperparameters, especially as the learning size increases. More specifically, the computation and memory allocation for the variance-covariance matrix a recurrent issue. Multiple techniques resolve this issue by applying compression schemes on this matrix, e.g., based on sparse approximations (e.g., Hierarchical Matrices)

This section introduced a general purpose surrogate model, uniformly approximating a function on a given domain. Most surrogates are designed for a well-defined purpose (e.g., optimization with or without constraint, contour finding for reliability analysis, etc.) In these cases, a more efficient approach might be to circumscribe the learning to a given sub-domain, rather than uniformly on the domain.

OpenTURNS 5 (Ordinary kriging). The following Python code proposes a minimalistic OpenTURNS implementation of an ordinary kriging model fitting.

1.7.3 Goal-oriented active surrogate model

A goal-oriented surrogate model fitted for solving a specific problem, related to an area of interest. [Only in the computer experiment framework] For example, to fit a surrogate model for contour finding in reliability analysis, one should concentrate the learning set around the limit-state function. Similarly, to build a surrogate for a global optimization problem, one should

concentrate the learning set around the optimum(s). However, this type of strategies require enough information to define an area of interest. *Active learning* is a general concept, aiming at iteratively increasing the learning set w.r.t. a *learning criterion* and updating the surrogate.

For optimization

[introduce EGO? Jones (1998)]

Fuhg. et al. (2021) reviews adaptive sampling methods for Kriging. [Cite Petit, Spagnol?]

For reliability analysis

Rare event estimation often requires large amounts of calls to the limit-state function (which is possibly a costly numerical model). Emulating this function by a surrogate model can drastically limit the number of calls to LSF. This surrogate approximates the contour (i.e., border) of the failure domain. However, in most cases, the failure domain represents a very restricted area of the input domain. To build a learning set concentrated around this border, active methods exist.

[introduce AK methods]

1.7.4 Summary and discussion

[The overfitting problem]

[What if the data is noisy or the numerical model stochastic?]

[Note that the calibration error is often larger than the metamodeling error.]

1.8 Conclusion

Chapter **2**

Introduction to wind turbine modeling and design

2.1	Introduction	44
2.2	Wind turbine modeling	44
2.2.1	Synthetic wind generation [TurbSim, Kaimal spectrum]	44
2.2.2	Synthetic wave generation	44
2.2.3	Aerodynamic interactions	44
2.2.4	Servo-Hydro-Aero-Elastic wind turbine simulation [DIEGO]	44
2.2.5	Soil modeling	44
2.2.6	Wake modeling [FarmShadow]	44
2.3	Recommended design practices	44
2.3.1	Design load cases	44
2.3.2	Dynamic response design	44
2.3.3	Fatigue response design	44
2.4	Uncertain inputs	44
2.4.1	Environmental inputs	44
2.4.2	System inputs	44
2.4.3	Probabilistic fatigue assessment	44
2.5	Conclusion	44

2.1 Introduction

2.2 Wind turbine modeling

2.2.1 Synthetic wind generation [**TurbSim, Kaimal spectrum**]

2.2.2 Synthetic wave generation

2.2.3 Aerodynamic interactions

2.2.4 Servo-Hydro-Aero-Elastic wind turbine simulation [**DIEGO**]

2.2.5 Soil modeling

2.2.6 Wake modeling [**FarmShadow**]

2.3 Recommended design practices

2.3.1 Design load cases

2.3.2 Dynamic response design

2.3.3 Fatigue response design

2.4 Uncertain inputs

2.4.1 Environmental inputs

2.4.2 System inputs

2.4.3 Probabilistic fatigue assessment

2.5 Conclusion

PART II:

CONTRIBUTIONS TO UNCERTAINTY QUANTIFICATION AND PROPAGATION

*Le doute est un état mental désagréable,
mais la certitude est ridicule.*

VOLTAIRE

Chapter **3**

Kernel-based uncertainty quantification

3.1	Introduction	48
3.2	Nonparametric fit of the environmental inputs	48
3.3	Quantifying and clustering the wake-induced perturbations within a wind farm	48
3.3.1	Uncertainty propagation in a wake model	49
3.3.2	Statistical metric of wake-induced perturbations	50
3.3.3	Conclusion	53
3.4	Conclusion	54

3.1 Introduction

3.2 Nonparametric fit of the environmental inputs

3.3 Quantifying and clustering the wake-induced perturbations within a wind farm

In the offshore wind industry, the wake effect is considered crucial for electricity production and structural fatigue of turbine components. For instance, the standards developed by the International Electrotechnical Commission (see appendix E in ?) review some analytical ?, and numerical models (e.g., the dynamic wake meandering approach by ?) to simulate the wind speed deficit and the wake added turbulence. Since the pioneering work of ?, several wake models were developed and compared in a benchmark by ?. This work takes advantage of the low computational cost of steady-state wake models to propagate the uncertainty from ambient to wake-induced wind conditions seen on a farm. It is worth noting that the wake creates a heterogeneous wind field in a wind farm, resulting in different loading conditions which should be considered at the stage of reliability based design (RBD). Such heterogeneity is not taken into account by the design load cases of international standards (see e.g., ??) where wind and wave conditions are derived from scenarios occurring over long-term periods. Wake models allow us to simulate the wind conditions' distribution at each turbine, however, the RBD step is too costly to be performed for each turbine. For further details regarding the RBD, one may refer to the work of [Huchet \(2019\)](#); [Slot et al. \(2020\)](#); [Stieng and Muskulus \(2020\)](#); [Wilkie and Galasso \(2021\)](#), proposing several approaches to reduce the computational cost of this step. In order to speed up the RBD at the scale of a wind farm, the present work aims at building clusters of WT similarly affected by the wake. Then, the RBD over a wind farm can be computed only on a few WT, each representing a cluster of turbines enduring similar wake-modified wind loading. This clustering is done on two wind parameters, following the conclusions of the global sensitivity analysis of ?. In order to discriminate the wake-perturbed distributions of wind parameters, the maximum mean discrepancy (MMD) is used as a statistical metric between multivariate distributions (as reviewed by ?). To illustrate this novel approach, a theoretical wind farm for the ongoing tenders off the coast of South Brittany in France is studied, with a modified version of the floating offshore wind turbine (FOWT) IEA-15MW (initially proposed in ??). Figure 3.1 illustrates the layout of the 25 FOWT considered in the following. This layout is regular with an inter-turbine distance of seven times the rotor diameter in the dominant wind direction and five times the rotor diameter in the orthogonal (crosswind) direction. More details regarding the FOWT modifications and theoretical wind farm can be found in ??.

Based on the results of previous numerical studies, with either dynamic wake meandering ? or LES ?, we retain the time-averaged floater position (translation and rotation) as the main effect of the floater motion on the far wake. It was shown that this effect is small, both on the

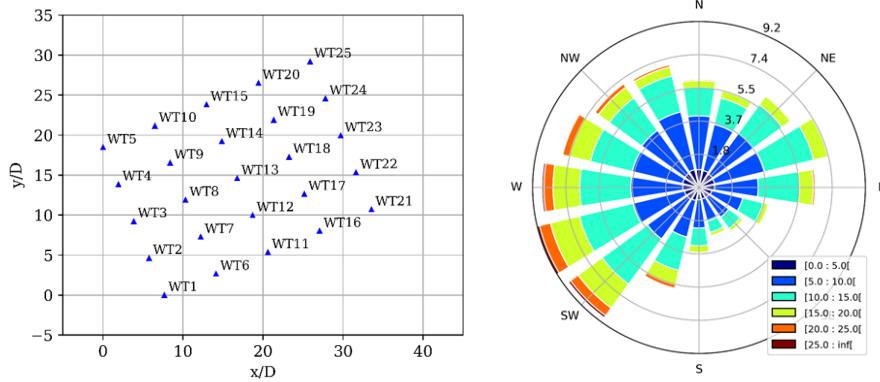


Fig. 3.1 South Brittany wind farm layout (left). South Brittany wind rose from the ANEMOC data (right, source: ?).

wake-added turbulence and on the wind speed deficit. However, noticeable uplift of the wake may influence the fatigue design.

The key idea in this paper to reduce the number of cases for RBD is to employ clustering techniques on the wake-induced wind parameters, to constitute groups of WT that are exposed to similar design load conditions. To do so, the present work is divided into four sections. First, the wake model used in this work is presented in section ???. Then, section 3.3.1 describes the uncertainty propagation through the wake model, from the probabilistic distribution of ambient wind parameters to the wake-modified wind parameters distribution within the farm. In order to regroup the similar wake-modified distributions, an adapted statistical metric on multivariate distributions is introduced in section 3.3.2. Finally, the application of several clustering methods to the South Brittany case study is compared in section ???. The main conclusion suggests four clusters among the 25 FOWT which can then be used for RBD analysis of the farm.

3.3.1 Uncertainty propagation in a wake model

The wake model described in section ?? takes as input a set of variables describing the ambient wind conditions $\mathbf{x} \in \mathbb{R}^3$ and computes the perturbed wind conditions at each WT represented by the vector $\mathbf{x}'_l, l \in (1, \dots, n_{WT})$, where n_{WT} is the total number of turbines in the farm:

$$g : \mathbb{R}^3 \rightarrow \mathbb{R}^{3n_{WT}} \quad (3.1)$$

$$\mathbf{x} \longmapsto g(\mathbf{x}) = (\mathbf{x}'_1, \dots, \mathbf{x}'_{n_{WT}}) \quad (3.2)$$

The uncertainties associated with the ambient wind conditions are represented by a random vector \mathbf{X} following the distribution p_0 . Note that the index 0 is a reference to the fact that these conditions are not perturbed by the wake. A parametric model has been fitted in Vanem et al. (2023) using conditional probability density functions to capture the dependence structure, with an approach similar to the one presented in Kelly and Vanem (2022). The random vector \mathbf{X} is described by the following input random variables:

- Mean wind speed (u) is the 10-min average horizontal wind speed at hub height.
- Wind turbulence intensity (TI) is the 10-min wind speed turbulence intensity at hub height.
- Wind direction (θ) is the 10-min average wind direction.

In the following, we assume the wind orientation variable θ to be unaffected by the wake. When perturbed by the wake of the wind farm, the WT l sees a wind flow represented by the random vector $\mathbf{X}'_l, l \in (1, \dots, n_{WT})$, following the distribution p'_l . Then, the two remaining parameters are u_{rotor} and TI_{rotor} to represent this modified wind flow on a vertical plane located at each WT. These two quantities of interest are averaged over the rotor while the input parameters are given at hub height. For the sake of simplicity, we will neglect this difference in what follows in order to consider the transformation $\mathbf{X}' = g(\mathbf{X})$ as a simple perturbation of \mathbf{X} . We will abusively denote u and TI both the input and output quantities. The output of the uncertainty propagation is a set of perturbed environmental distributions $p'_l, l \in (1, \dots, n_{WT})$. A Monte-Carlo sample $\mathbf{X}_n = \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$ of the three random input variables is generated. Then, considering the farm layout illustrated in Figure 3.1 and a constant wind shear exponent of 0.1 like in section ??, a wake simulation is run for every wind condition of the Monte Carlo design of experiments. The code output consists in a multivariate joint distribution of modified u and TI for each WT. As the Monte Carlo procedure is known to converge slowly but surely, it was chosen to perform this uncertainty propagation with a number of simulations of size $n = 6\,000$ because of its simplicity and the low computational cost of the simulations.

We can plot a preview of the wake perturbations on the joint distribution for given WT in the two dimensions u and TI . Figure 3.2 illustrates this perturbation for three WT differently affected by the wake depending on their position in the farm (cf. Figure 3.1). One can notice that the WT 25 distribution (in orange) is very close to the ambient distribution (in black), as expected, this WT being located on the edge of the farm and facing the dominant wind direction. Meanwhile, the WT 13 distribution (in red) seems more affected by the wake, by getting a higher wind turbulence with lower wind speed. This analysis can be completed with the two marginals in Figure 3.3 and Figure 3.4, both describing the ambient marginal distributions (in black) and wake-disturbed distributions. In general, a small wind speed deficit is noticed as indicated by the small shifts of the probability density functions to the left on Figure 3.3. Also, a small added turbulence is indicated by the small shifts of the curves to the right on Figure 3.4. Ideally, a tool should allow us to quantify the perturbation between the ambient and perturbed distributions.

3.3.2 Statistical metric of wake-induced perturbations

Maximum mean discrepancy: a distance between distributions In the literature, the maximum mean discrepancy was introduced by [Gretton et al. \(2006\)](#) as a statistical test to discriminate two distinct distributions. After further work on this tool, authors such as [Sriperumbudur et al. \(2010\)](#) showed that the MMD is a distance between two distributions embedded

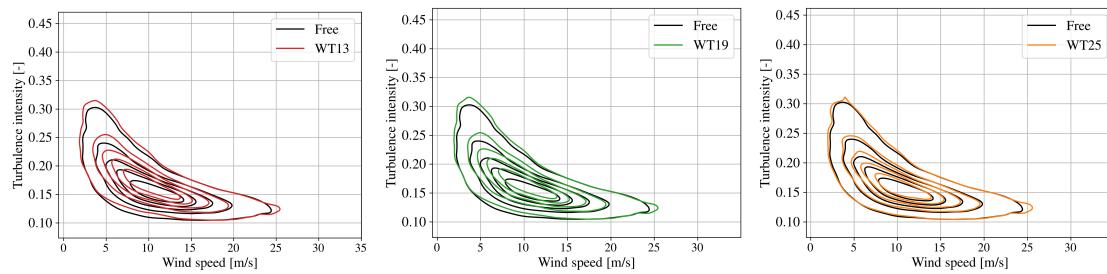


Fig. 3.2 Joint perturbation at WT 13, 19, and 25

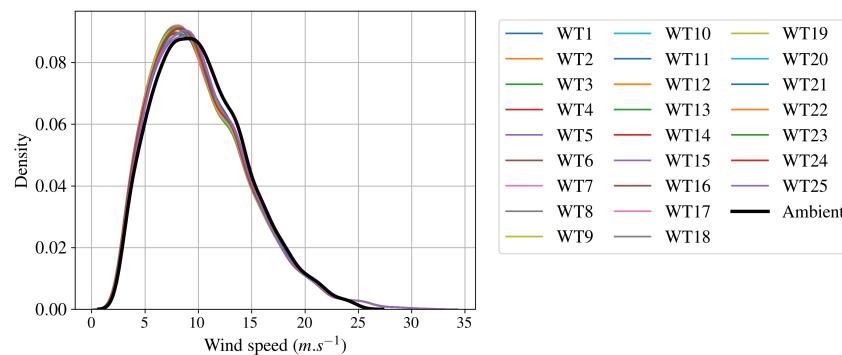


Fig. 3.3 Ambient and wake-disturbed distributions of the wind speed

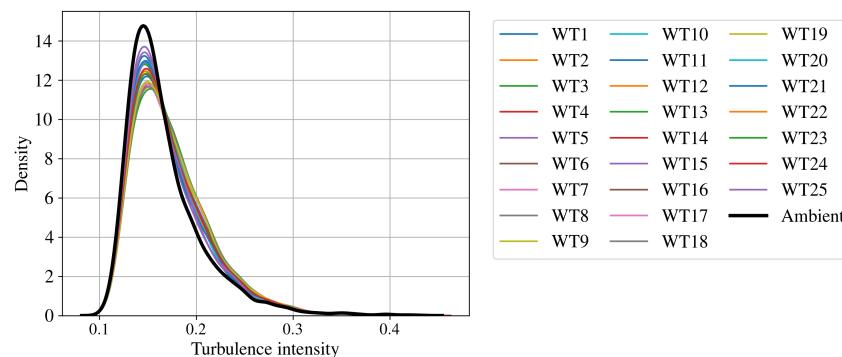


Fig. 3.4 Ambient and wake-disturbed distributions of the turbulence intensity

in a specific function space. Therefore, this concept relies on the embedding of distributions in a convenient Hilbert space. Considering a positive-definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}, d \in \mathbb{N}$ generating a unique Hilbert space \mathcal{H}_k of functions equipped with inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ and norms $\|\cdot\|_{\mathcal{H}_k}$ (also called Reproducing Kernel Hilbert Space (RKHS) when the function $k(\mathbf{x}, \cdot)$ satisfies the reproducing property). Then, let us define the kernel mean embedding of the distribution P in the function space \mathcal{H}_k :

$$\mu_P(\mathbf{x}) := \int_{\mathbb{R}^d} k(\mathbf{x}, \mathbf{y}) dP(\mathbf{y}) \approx \frac{1}{n} \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^{(i)}), \mathbf{x}^{(i)} \in \mathbf{X}_n. \quad (3.3)$$

The kernel mean embedding is approximated on sample $\mathbf{X}_n = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$ following the distribution P . Figure 3.5 illustrates the kernel mean embedding of two distributions in the function space \mathcal{H}_k defined previously. Notice that this procedure allows us to embed continuous distributions (such as P) as well as discrete distributions (such as Q).

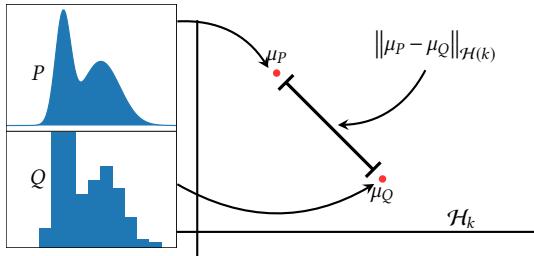


Fig. 3.5 Kernel mean embedding of two probability distributions P and Q mapped in the RKHS H_k

The distance between the two kernel mean embeddings μ_P and μ_Q is called the maximum mean discrepancy (MMD). This distance between two distributions P and Q is initially defined by the worst-case error for any function within a unit ball of a Hilbert space \mathcal{H}_k , generated by the kernel k :

$$\text{MMD}_k(P, Q) := \sup_{\|g\|_{\mathcal{H}_k} \leq 1} \left| \int_{\mathbb{R}^d} g(\mathbf{x}) dP(\mathbf{x}) - \int_{\mathbb{R}^d} g(\mathbf{x}) dQ(\mathbf{x}) \right| = \|\mu_P(\mathbf{x}) - \mu_Q(\mathbf{x})\|_{\mathcal{H}_k}. \quad (3.4)$$

The MMD fully relies on the difference of kernel mean embeddings. Moreover, according to Sriperumbudur et al. (2010), a kernel is called “characteristic kernel” when the following equivalence is true, $\text{MMD}_k(P, Q) = 0 \iff P = Q$, making the MMD a metric on \mathbb{R}^d . For its good convergence behavior, the squared MMD has been used for multiple other purposes than numerical integration: statistical testing Gretton et al. (2006), sensitivity analysis Da Veiga (2015). When elevated to the square, it can be estimated using one n -sized representative sample of P denoted $\{\mathbf{x}^{(i)}\}_{i \in (1, \dots, n)}$ (and respectively one m -sized sample of Q denoted $\{\mathbf{y}^{(i)}\}_{i \in (1, \dots, m)}$:

$$\widehat{\text{MMD}}_k(P, Q)^2 = \frac{1}{n^2} \sum_{i,j=1}^n k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) + \frac{1}{m^2} \sum_{i,j=1}^m k(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}). \quad (3.5)$$

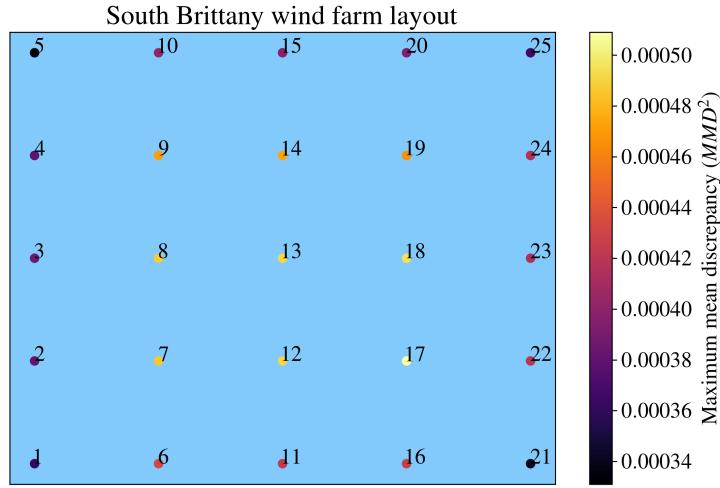


Fig. 3.6 South Brittany layout and wake effects measured by the squared MMD on wind conditions. Note that the vertical direction on this plot does not represent the north direction.

In the following, the idea is to compare the ambient wind distribution p_0 to the wake-perturbed wind conditions p'_l at the WT l using the previously defined squared MMD.

Application to the South Brittany wind farm project Once the joint perturbed distributions of each WT are estimated by a large Monte Carlo sample (cf. section 3.3.1), the MMD with the ambient wind conditions can be computed. Figure 3.6 illustrates for each WT the squared MMD value computed to measure the wake-induced perturbation. Let us remind that MMD is a distance between the joint perturbed distribution at a WT computed for all wind orientations with the ambient wind distribution. Despite the wake obviously depend on the wind direction, our final goal is to define a small number of WT for RBD thus independently of the wind orientation. The lower this metric gets, the closer to the ambient wind distribution. Quite logically, the WT in the center of the farm are more affected by the wake since they are subject to the wake regardless of the wind direction.

The values of squared MMD given in the previous figure are estimated between two samples:

- the Monte Carlo sample of the free environmental distribution: \mathbf{X}_n ,
- the wake-perturbed Monte Carlo sample at the WT l : $\mathbf{X}'_{n,l}$ (output of the steady-state wake numerical model).

To ensure that the Monte-Carlo estimation converged, Figure 3.7 plots the squared MMD between the sample \mathbf{X}_n and the increasing samples $\mathbf{X}'_{i,l}$, $i \in 400, \dots, 6000$. After a few thousands of simulations, the MMD of each WT tends to converge towards a stable value, as expected. The design of experiment with $n = 6\,000$ is thus considered as sufficient.

3.3.3 Conclusion

A steady-state engineering wake model is coupled with a hydrostatic solver to take into account the effect of the floaters position in the wake computation of a floating offshore wind farm. The

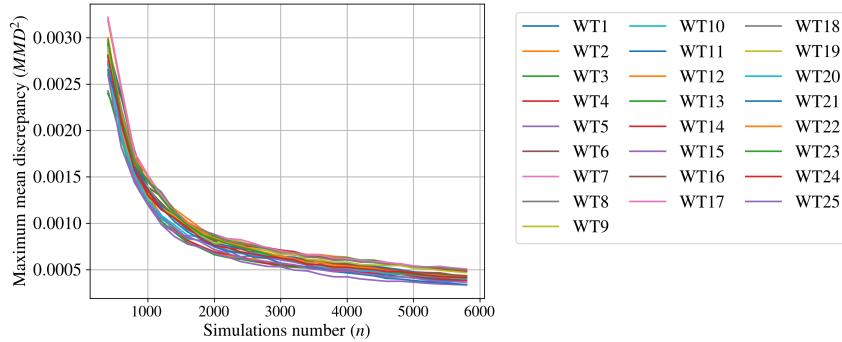


Fig. 3.7 Convergence of the squared MMD estimation

main impact of the floater's position is the increased rotor tilt, which leads to a larger vertical deflection of the wake. In the South-Brittany farm, the elevation of wake center is significant and could modify the fatigue loads on the downstream WT. This low-fidelity, or engineering model is very fast (about 3 minutes on a regular computer), while higher fidelity models can simulate the wake in a wind farm more accurately, but with much higher computational cost (several days for LES Meso-NH solver, with intensive parallelisation). In this paper, we consider that the modelling error made by the engineering wake model as reasonable. Further investigations should be done on the integration of the different fidelity levels in the uncertainty propagation. In this work, the uncertainty propagation is performed for two random inputs: the wind speed and the turbulence intensity, leading to a wake perturbed distribution per WT. A metric was then defined to measure the distance between distributions in order to build clusters of WT seeing similar wind conditions. After applying the metric to the South Brittany case, a clustering approach is used to determine a limited number of WTs representing the farm for RBD. Several clustering methods are compared and provide similar results for the current case study. In this case, a solution with 4 clusters is a good compromise between low relative error with clusters and low number of clusters. Getting a low number of clusters allows us to reduce the number of representative WT on which a RBD study can be done to assess the RBD over the whole farm. The differences among a cluster and between clusters have to be studied when looking directly to the output of interest for ultimate or fatigue reliability. However the difference on load output quantities may be reduced when compared to that on wake output quantities thanks to the damping of the WT. Depending on the definition of a wind farm failure (series system: one WT fails or parallel system: all WT fail or intermediate system), the probability of failure can be estimated from the probability of failure of the representative WT. These considerations need to be further explored to improve RBD at the farm scale.

3.4 Conclusion

Chapter **4**

Kernel-based central tendency estimation

4.1	Introduction	56
4.2	Treatment of uncertainties on the Teesside wind farm	57
4.2.1	Numerical simulation model	58
4.2.2	Measured environmental data	59
4.2.3	Non parametric fit with empirical Bernstein copula	60
4.2.4	Fatigue assessment	62
4.3	Numerical integration procedures for mean damage estimation	63
4.3.1	Quadrature rules and quasi-Monte Carlo methods	63
4.3.2	Kernel discrepancy	65
4.3.3	Kernel herding sampling	68
4.3.4	Bayesian quadrature	71
4.4	Numerical experiments	74
4.4.1	Illustration through analytical toy-cases	74
4.4.2	Application to the Teesside wind turbine fatigue estimation	75
4.5	Conclusion	78

4.1 Introduction

As a sustainable and renewable energy source, offshore wind turbines (OWT) are likely to take a growing share of the global electric mix. However, to be more cost-effective, wind farm projects tend to move further from the coast, exploiting stronger and more regular wind resources. Going further offshore, wind turbines are subject to more severe and uncertain environmental conditions (i.e., wind and waves). In such conditions, their structural integrity should be certified. To do so, numerical simulation and probabilistic tools have to be used. In fact, according to [Graf et al. \(2016\)](#), for new environmental conditions or new turbine models, international standards such as [IEC \(2019\)](#) from the International Electrotechnical Commission and [DNV-GL \(2016b\)](#) from Det Norske Veritas recommend performing over 200,000 simulations distributed over a grid. Numerical simulations are computed by a costly hydro-servo-aero-elastic wind turbine model, making the design process time-consuming. In the following, the simulated output cyclic loads studied are aggregated over the simulation period to assess the mechanical fatigue damage at hot spots of the structure. To compute the risks associated with wind turbines throughout their lifespan, one can follow the steps of the universal framework for the treatment of uncertainties ([de Rocquigny et al., 2008](#)) presented in Fig. 4.1. After defining the problem (Step A), one can quantify the uncertainties related to site-specific environmental conditions denoted by the random variable X (Step B). Then, one can propagate them through an OWT simulation model $g(\cdot)$ (Step C), and estimate a relevant quantity of interest $\psi(Y) = \psi(g(X))$ (e.g., mean, quantile, failure probability). A proper estimation of the quantity relies on a good uncertainty model and an efficient sampling method to estimate the quantity of interest.

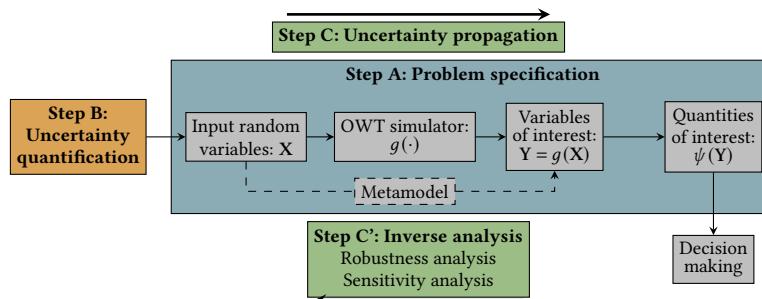


Fig. 4.1 General uncertainty quantification and propagation framework (adapted from [Ajenjo \(2023\)](#))

The uncertainties related to the OWT environment follow a joint distribution with a complex dependence structure. This challenging distribution has been fitted with different parametric approaches in the literature (step B in Fig. 4.1), mainly using conditional distributions ([Vanem et al., 2023](#)), but also vine copulas ([Li and Zhang, 2020](#)). When one has access to data, another way is to directly use the data as empirical representation of input uncertainties.

These uncertainties have then been propagated to fatigue damage (the output), making it random, and to the associated quantities of interest (step C in Fig. 4.1). When uncertainty propagation aims at central tendency studies, the methods employed are split into two groups. Methods of the first group rely on numerical integration by Monte Carlo sampling ([Graf et al.,](#)

2016), quasi-Monte Carlo sampling (Müller and Cheng, 2018), or deterministic quadrature rules (Van den Bos, 2020). All these methods estimate the quantity directly on the numerical simulator's outputs. Methods of the second group use metamodels (or surrogate models) to emulate the costly numerical model by a statistical model such as polynomial chaos expansion (Dimitrov et al., 2018; Murcia et al., 2018), Gaussian processes (Huchet, 2019; Slot et al., 2020; Wilkie and Galasso, 2021), or artificial neural networks (Bai et al., 2023).

When uncertainty propagation aims at studying the tail of the output distribution (e.g., reliability analysis), one can estimate a quantile or a failure probability. Failure probabilities were studied, in static reliability analysis (Slot et al., 2020; Wilkie and Galasso, 2021; Zwick and Muskulus, 2015) or time-dependent reliability analysis (Abdallah et al., 2019; Lataniotis, 2019). To get a better understanding of the OWT numerical models behavior, authors have used sensitivity analysis methods (Da Veiga et al., 2021), which determine the most influential inputs on the damage (step C' in Fig. 4.1). Among others, one can cite the application of Spearman's rank correlation coefficients and Morris method's by Petrovska (2022); Velarde et al. (2019), the direct calculation of Sobol' indices after fitting a polynomial chaos model by Murcia et al. (2018) and the use of Kullback-Leibler divergence by Teixeira et al. (2017). Each of those methods brings something different to the analysis.

This paper will focus on central tendency estimation (i.e., $\psi(\mathbf{X}) = \mathbb{E}(\mathbf{X})$) by: (1) direct sampling on the numerical model, (2) sampling on a static regression model, or (3) sampling on an active regression model (i.e., observations of the numerical model are progressively added to enhance a goal-oriented metamodel). In the specific context of wind turbines, this paper explores how to study the central tendency study of the fatigue damage output, by carrying out the uncertainty propagation of a complex input joint distribution through a costly wind turbine simulator. This work proposes a new approach for given data, fast, and fully-distributable uncertainty propagation for OWT models. Overall, this paper reviews the methods of Bayesian quadrature and presents its application on the industrial wind turbines case. In this paper, Section 2 will detail the industrial use-case related to a wind farm operating in Teesside, UK. Then, Section 3 will introduce different methods for central tendency estimation. Section 4 will analyze the results of numerical experiments on analytical and industrial cases. Then, the last section will present discussions and conclusions.

4.2 Treatment of uncertainties on the Teesside wind farm

An OWT is a complex system interacting with its environment. To simulate the response of this system against a set of environmental solicitations, multi-physics numerical models are developed. In our case, it is a chain of three numerical codes executed sequentially. As illustrated in Fig. 4.2, a simulation over a time period is the sequence of (1) turbulent wind speed field generation, (2) wind turbine simulation (computing various outputs including mechanical stress), and (3) post-processing to assess the fatigue damage of the structure.

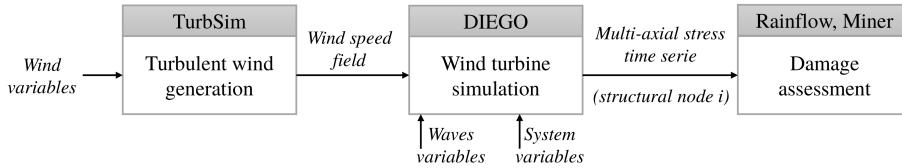


Fig. 4.2 Diagram of the chained OWT simulation model

Table 4.1 Teesside Offshore Wind turbine datasheet

Siemens SWT-2.3–93	
Rated power	2.3 MW
Rotor diameter	93 m
Hub height	83 m
Cut-in, cut-out wind speed	4 m/s, 25 m/s

4.2.1 Numerical simulation model

This section describes more precisely the modeling hypotheses considered in this case. Starting from the turbulent wind field simulator TurbSim (developed by [Jonkman \(2009\)](#) from the National Renewable Energy Laboratory, USA) that uses a Kaimal spectrum ([Kaimal et al., 1972](#)). To extrapolate the wind speed vertically, the shear is modeled by a power law. Since the wind field generation shows inherent stochasticity, each 10-minute long simulation is repeated with different pseudo-random seeds and the average damage over these repetitions is studied. This question was widely studied by some authors, (e.g., [Slot et al. \(2020\)](#)), who concluded that the six repetitions recommended by the [IEC \(2019\)](#) are insufficient to properly average this stochasticity. In the following, the simulations are repeated eleven times (allowing direct access to the median value). This number of repetitions was chosen as a compromise between the general number of simulations and the storage capacity of the generated simulations.

DIEGO (for “Dynamique Intégrée des Éoliennes et Génératrices Offshore¹”) is a code developed by EDF R&D ([Kim et al., 2022](#)) to simulate the aero-hydro-servo-elastic behavior of OWTs. It takes the turbulent wind speed field generated by TurbSim as input and computes the dynamical behavior of the system (including the multiaxial mechanical stress at different nodes of the structure). For our application, the control system is modeled by the open-source DTU controller ([Hansen and Henriksen, 2013](#)), and no misalignment between the wind and the OWT is assumed. As for the waves, they are modeled in DIEGO using a JONSWAP spectrum (named after the 1975 Joint North Sea Wave Project). Our study uses a DIEGO model of a Siemens SWT 2.3MW bottom-fixed turbine on a monopile foundation (see the datasheet in Table 4.1), currently operating in Teesside, UK (see the wind farm layout and wind turbine diagram in Fig. 4.3). Although wind farms are subject to the wake effect, affecting the behavior and performance of some turbines in the farm, this phenomenon is not considered in the following. To avoid numerical perturbations and reach the stability of the dynamical system, our simulation period

¹In english, “Integrated Dynamics of Wind Turbines and Offshore Generators”.

is extended to 1000 seconds and the first 400 seconds are cropped in the post-processing step. This chained OWT numerical simulation model has been deployed on an EDF R&D HPC facility to benefit from parallel computing speed up (a single simulation on one CPU takes around 20 minutes).

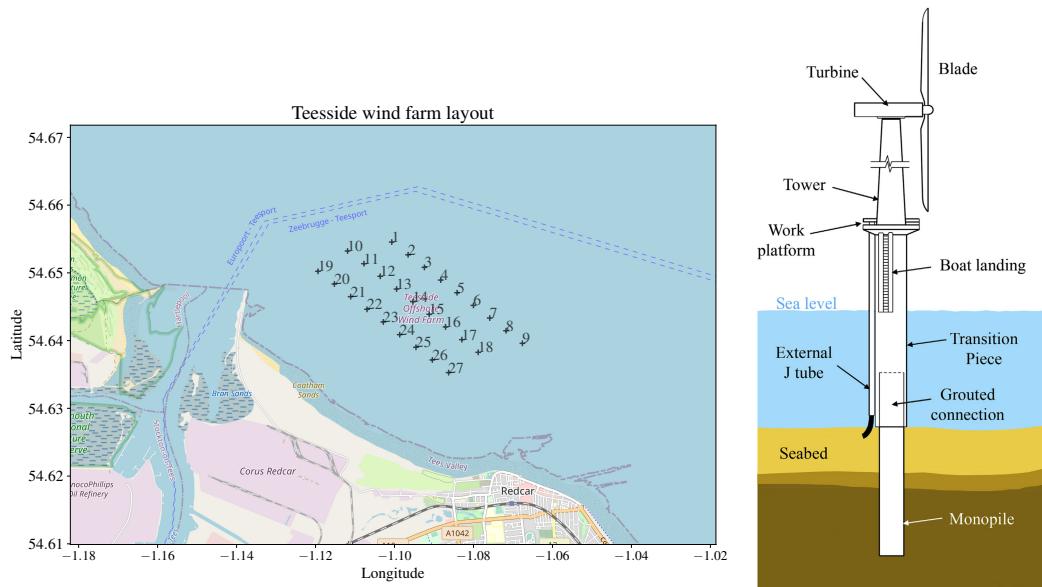


Fig. 4.3 Teesside wind farm layout (left). Monopile OWT diagram (Chen et al., 2018) (right)

4.2.2 Measured environmental data

During the lifespan of a wind farm project, environmental data is collected at different phases. In order to decide on the construction of a wind farm, meteorological masts and wave buoys are usually installed on a potential site for a few years. After its construction, each wind turbine is equipped with monitoring instruments (e.g., cup anemometers). In total, five years of wind data have been collected on the turbines which are not affected by the wake on this site. Their acquisition system (usually called SCADA, for “Supervisory Control And Data Acquisition”) have a sampling period of ten minutes. The wave data arise from a buoy placed in the middle of the farm. This data describes the physical features listed in Table 4.2.

The farm of Teesside is located close to the coast, making the environmental conditions very different depending on the direction (see the wind farm layout in Fig. 4.3). Since measures are also subject to uncertainties, a few checks were realized to ensure that the data were physically consistent. The truncation bounds defined in Table 4.2 were applied since this study is not interested in extreme values but in central tendency estimation (i.e., mean behavior). In addition, a simple trigonometric transform is applied to each directional feature to take into account their cyclic structure. Finally, the remaining features are rescaled (i.e., using a min-max normalization). The matrix plot of the transformed data in Fig. 4.4 is an innovative plot named *copulogram*. A copulogram is an innovative plot as it decomposes the data between the effects of the marginals

and those of the dependence between features. To do so, it represents the marginals with univariate kernel density estimation plots (diagonal), and the dependence structure with scatter plots in the ranked space (upper triangle). On the bottom triangle the scatter plots are set in the physical space, gathering the effects of the marginals and the dependencies. Since the dependence structure is theoretically modeled by an underlying copula, this plot is called *copulogram*, generalizing the well-known “correlogram” to nonlinear dependencies. It gives a synthetic and empirical decomposition of the dataset.

On Fig. 4.4, a large sample \mathcal{S} (with a size $N = 10^4$) is randomly drawn from the entire Teesside data (with size $N_{\text{Teesside}} = 2 \times 10^5$), and plotted in grey. In the same figure, the orange matrix plot is a subsample of the sample \mathcal{S} , selected by kernel herding, a method that will be presented in Section 3. Visually, this orange subsample seems to match the original sample both in terms of marginal distributions and dependence structure. In the following study, the large samples \mathcal{S} will be considered as an empirical representation of the multivariate environmental distribution $\mathbf{X} \in \mathcal{D}_X \subset \mathbb{R}^p$, of density f_X , and called *candidate set*. Contrarily to parametric approaches which can be used to describe the joint environmental uncertainty, this method intends to directly subsample from this large and representative dataset. This technique samples a joint distribution without modeling it. Indeed, a proper parametric model fit would be challenging for complex dependence structures such as the one plotted on Fig. 4.4. Li and Zhang (2020) built a parametric model of a similar multivariate distribution using vine copulas. Alternatively, a nonparametric approach coupling empirical Bernstein copula fitting with kernel density estimation of the marginals is introduced in Section 4.2.3.

Table 4.2 Description of the environmental data.

Variable	Notation	Unit	Description
Mean wind speed	U	m.s^{-1}	10-min. average horizontal wind speed
Wind turbulence	σ_U	m.s^{-1}	10-min. wind speed standard deviation
Wind direction	θ_{wind}	deg.	10-min. average wind direction
Significant wave height	H_s	m	Significant wave height
Peak wave period	T_p	s	Peak 1-hour spectral wave period
Wave direction	θ_{wave}	deg.	10-min. average wave direction

4.2.3 Non parametric fit with empirical Bernstein copula

The Sklar theorem (Joe, 1997) states that the multivariate distribution of any random vector $\mathbf{X} \in \mathbb{R}^p$ can be broken down into two objects:

1. A set of univariate marginal distributions to describe the behavior of the individual variables;
2. A function describing the dependence structure between all variables, called a copula.

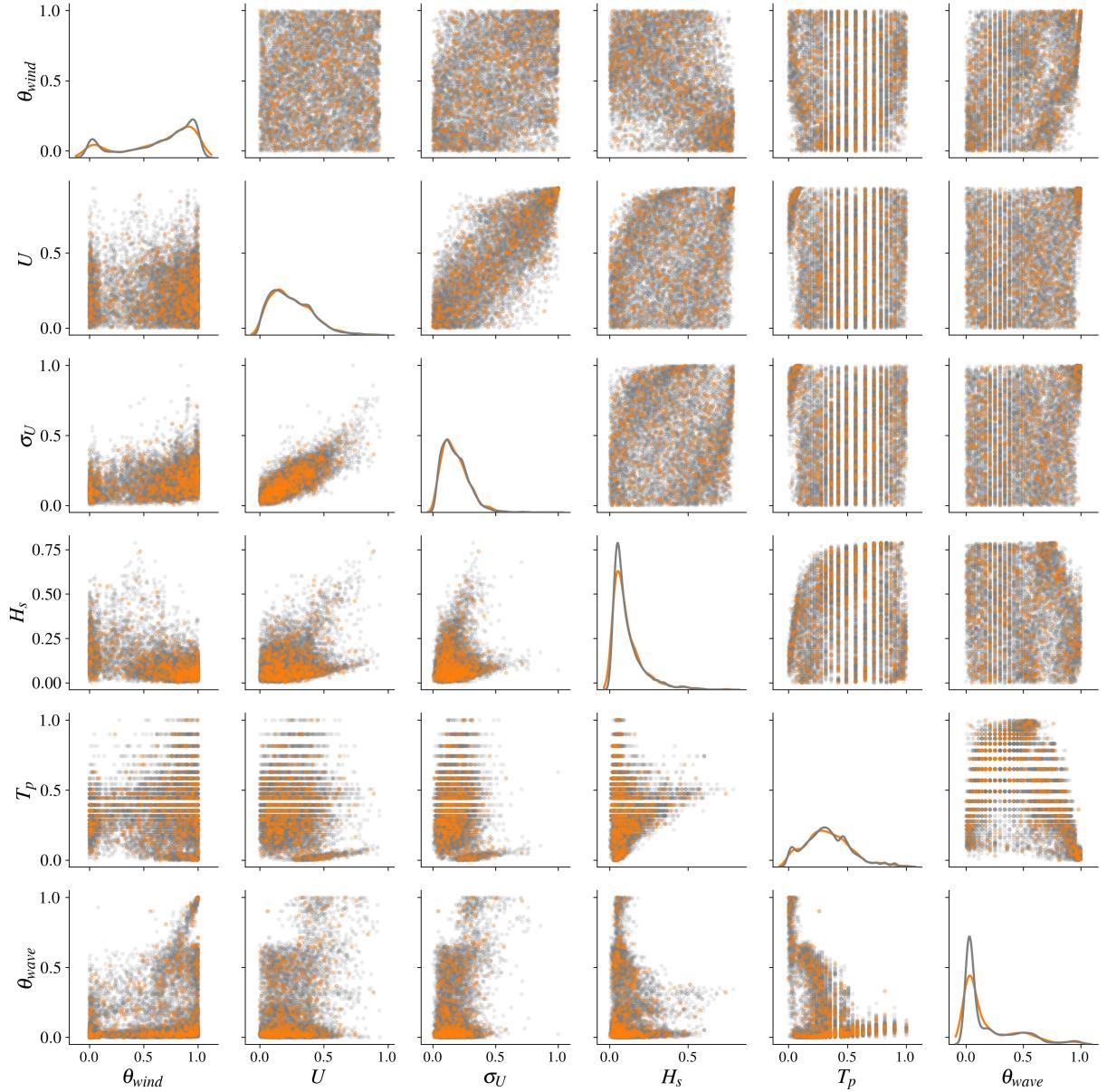


Fig. 4.4 Copulogram of the Teesside measured data ($N = 10^4$ in grey), kernel herding subsample ($n = 500$ in orange). Marginals are represented by univariate kernel density estimation plots (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space

This theorem states that considering a random vector $\mathbf{X} \in \mathbb{R}^p$, with its distribution F and its marginals $\{F_i\}_{i=1}^p$, there exists a copula $C : [0, 1]^p \rightarrow [0, 1]$, such that:

$$F(x_1, \dots, x_p) = C(F_1(x_1), \dots, F_p(x_p)). \quad (4.1)$$

It allows us to divide the problem of fitting a joint distribution into two independent problems: fitting the marginals and fitting the copula. Additionally, when the joint distribution is continuous, this copula is unique. Copulas are continuous and bounded functions defined on a compact set (the unit hypercube). Bernstein polynomials allow to uniformly approximate as

closely as desired any continuous and real-valued function defined on a compact set (Weierstrass approximation theorem). Therefore, they are good candidates to approximate unknown copulas. This concept was introduced as *empirical Bernstein copula* (EBC) by [Sancetta and Satchell \(2004\)](#) for applications in economics and risk management. Later on, [Segers et al. \(2017\)](#) offered further asymptotic studies. Formally, the multivariate Bernstein polynomial for a function $C : [0, 1]^p \rightarrow \mathbb{R}$ on a grid over the unit hypercube $G := \left\{ \frac{0}{h_1}, \dots, \frac{h_1}{h_1} \right\} \times \dots \times \left\{ \frac{0}{h_p}, \dots, \frac{h_p}{h_p} \right\}, \mathbf{h} = (h_1, \dots, h_p) \in \mathbb{N}^p$, is written as follows:

$$B_{\mathbf{h}}(C)(\mathbf{u}) := \sum_{t_1=0}^{h_1} \dots \sum_{t_p=0}^{h_p} C\left(\frac{t_1}{h_1}, \dots, \frac{t_p}{h_p}\right) \prod_{j=1}^p P_{h_j, t_j}(u_j), \quad (4.2)$$

with $\mathbf{u} = (u_1, \dots, u_p) \in [0, 1]^p$, and the Bernstein polynomial $P_{h, t}(u) := \frac{t!}{h!(t-h)!} u^h (1-u)^{t-h}$. When C is a copula, then $B_{\mathbf{m}}(C)$ is called “Bernstein copula”. Therefore, the empirical Bernstein copula is an application of the Bernstein polynomial in Eq. (4.2) to the so-called “empirical copula”. In practice, considering a sample $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{np}$ and the associated ranked sample $\mathbf{R}_n = \{\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}\}$, the corresponding empirical copula is written:

$$C_n(\mathbf{u}) := \frac{1}{n} \sum_{i=0}^n \prod_{j=1}^p \mathbb{1} \left\{ \frac{r_j^{(i)}}{n} \leq u_j \right\}, \mathbf{u} = (u_1, \dots, u_p) \in [0, 1]^p. \quad (4.3)$$

Provided a large enough learning set \mathbf{X}_n , the EBC combined with kernel density estimation for the marginals fit well the environmental joint distribution related to the dataset in Fig. 4.4. Moreover, the densities of the EBC are available in an explicit form, making Monte Carlo or quasi-Monte Carlo generation easy. For a thorough presentation of this method and theoretical results regarding the EBC tuning, see the manuscript of [Lasserre \(2022\)](#). Further discussions and numerical experiments on the estimation of nonparametric copula models are presented in [Nagler et al. \(2017\)](#).

4.2.4 Fatigue assessment

As described in Fig. 4.2, a typical DIEGO simulation returns a 10-minute multiaxial stress time series at each node $i \in \mathbb{N}$ of the 1D meshed structure. Since fatigue laws are established for uniaxial stresses, the first step is to compute one equivalent Von Mises stress time series at each structural node.

However, the foundation and the tower of an OWT are a succession of tubes with various sections connected by bolted or welded joints. Our work studies the welded joints at the mudline level, identified as a critical area for the structure. To compute fatigue in this joint, the external circle of the welding ring is discretized for a few azimuth angles $\theta \in \mathbb{R}_+$ (see the red points in the monopile cross-section on the right in Fig. 4.5). The equivalent Von Mises stress time series is then reported on the external welding ring for an azimuth θ . According to [Li and Zhang \(2020\)](#) and our own experience, the most critical azimuth angles are roughly aligned with the main wind and wave directions (whose distributions are illustrated in Fig. 4.5). According to

in these illustrations, the wind and wave conditions have a very dominant orientation, which is explained by the closeness of the wind farm to the shore. Then, it is assumed that azimuth angles in these directions will be more solicited, leading to higher fatigue damage. To assess fatigue damage, rainflow counting (Dowling, 1972) first identifies the stress cycles and their respective amplitudes (using the implementation of the ASTM E1049-85 rainflow cycle counting algorithm from the Python package `rainflow`²). For each identified stress cycle of amplitude s , the so-called “Stress vs. Number of cycles” curve (also called the “Wöhler curve”) allows one to estimate the number N_c of similar stress cycles necessary to reach fatigue ruin:

$$N_c := W(s) = as^{-m}, a \in \mathbb{R}, m \in \mathbb{R}. \quad (4.4)$$

Finally, a usual approach to compute the damage is to consider the fatigue contribution of each stress cycle identified using Miner’s rule. Damage occurring during a 10-minute operating time is simulated and then scaled up to the OWT lifetime. More details regarding damage assessment are available in Appendix ???. For a realization of environmental conditions $\mathbf{x} \in \mathcal{D}_X$, at a structural node i , an azimuth angle θ ; k stress cycles of respective amplitude $\{s_{i,\theta}^{(j)}(\mathbf{x})\}_{j=1}^k$ are identified. Then, Miner’s rule (Fatemi and Yang, 1998) defines the damage function $g_{i,\theta}(\mathbf{x}) : \mathcal{D}_X \rightarrow \mathbb{R}_+$ by:

$$g_{i,\theta}(\mathbf{x}) = \sum_{j=1}^k \frac{1}{N_c} = \sum_{j=1}^k \frac{1}{W(s_{i,\theta}^{(j)}(\mathbf{x}))}. \quad (4.5)$$

As defined by the DNV standards for OWT fatigue design (DNV-GL, 2016a), the quantity of interest in the present paper is the “mean global damage” $d_{i,\theta}$, computed at the node i , for an azimuth angle θ :

$$d_{i,\theta} = \mathbb{E}[g_{i,\theta}(\mathbf{X})] = \int_{\mathcal{D}_X} g_{i,\theta}(\mathbf{x}) f_X(\mathbf{x}) \mathbf{x}. \quad (4.6)$$

To get a preview of the distribution of this output random variable $g_{i,\theta}(\mathbf{X})$, a histogram of a large Monte Carlo simulation ($N_{\text{ref}} = 2000$) is represented in Fig. 4.6 (with a log scale). The log-damage presents a little asymmetry, so it is unlikely to be normally distributed.

4.3 Numerical integration procedures for mean damage estimation

4.3.1 Quadrature rules and quasi-Monte Carlo methods

The present section explores different methods aiming at estimating the expected value of a function against a probability measure. Considering a measurable space $\mathcal{D}_X \subset \mathbb{R}^p, p \in \mathbb{N}_+$, associated with a known Lebesgue measure μ , let us study the approximation of integrals

²<https://github.com/iamlikeme/rainflow>

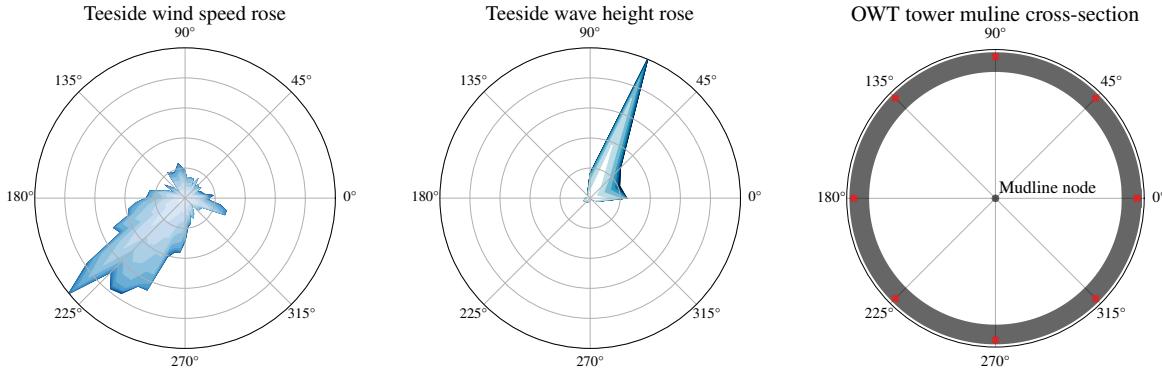


Fig. 4.5 Angular distribution of the wind and waves with a horizontal cross-section of the OWT structure and the mudline

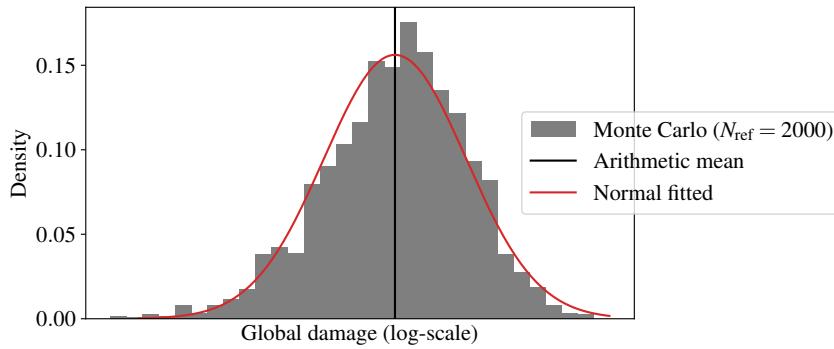


Fig. 4.6 Histogram of the log-damage, at mudline, azimuth 45 deg. (Monte Carlo reference sample)

of the form $\int_{\mathcal{D}_X} g(\mathbf{x})d\mu(\mathbf{x})$, with g the map $g(\mathbf{x}) : \mathcal{D}_X \rightarrow \mathbb{R}$. This problem is equivalent to the central tendency estimation of $\mathbf{Y} = g(\mathbf{X})$, the image of the environmental random variable \mathbf{X} by the damage function g (see Eq. (4.6)). Some authors also named this generic problem *probabilistic integration* (Briol et al., 2019). In practice, this quantity of interest is estimated on an n -sized set of damage realizations $\mathbf{y}_n = \{g(\mathbf{x}^{(1)}), \dots, g(\mathbf{x}^{(n)})\}$ of an input sample $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$. Our numerical experiment framework often implies that the function g is costly to evaluate, making the realization number limited. A weighted arithmetic mean of the realizations $\{g(\mathbf{x}^{(1)}), \dots, g(\mathbf{x}^{(n)})\}$ is called a *quadrature rule* with a set of unconstrained weights $\mathbf{w}_n = \{w_1, \dots, w_n\} \in \mathbb{R}^n$:

$$I_\mu(g) := \int_{\mathcal{D}_X} g(\mathbf{x})d\mu(\mathbf{x}) \approx \sum_{i=1}^n w_i g(\mathbf{x}^{(i)}). \quad (4.7)$$

For a given sample size n , our goal is to find a set of tuples $\{\mathbf{x}^{(i)}, w_i\}_{i=1}^n$ (i.e., quadrature rule), giving the best approximation of our quantity. In the literature, a large panel of numerical integration methods has been proposed to tackle this problem. In a recent work, Van den Bos (2020) applies a first family of numerical integration methods based on tensor products of quadrature rules to a similar industrial OWT use case. Unfortunately, the tensor formulation fails when

inputs present a strong dependency structure and will not be studied in this paper. Alternatively, sampling methods rely on generating a set of points \mathbf{X}_n drawn from the input distribution to compute the arithmetic mean of their realizations (i.e., uniform weights $\{w_i = \frac{1}{n}\}_{i=1}^n$). Among them, low-discrepancy sequences, also called “quasi-Monte Carlo” sampling (e.g., Sobol’, Halton, Faure sequences) are known to improve the standard Monte Carlo convergence rate and will be used as a deterministic reference method in the following numerical experiments (Morokoff and Caflisch, 1995).

4.3.2 Kernel discrepancy

Quantization of probability measures and quadrature When dealing with probabilistic integration such as Eq. (4.7), a quadrature rule is a finite representation of a continuous measure μ by a discrete measure $\zeta_n = \sum_{i=1}^n w_i \delta(\mathbf{x}^{(i)})$ (weighted sum of Dirac distributions at the design points \mathbf{X}_n). In the literature, this procedure is also called *quantization* of a continuous measure μ . Overall, numerical integration is a particular case of probabilistic integration against a uniform input measure. For uniform measures, the Koksma-Hlawka inequality (Morokoff and Caflisch, 1995) provides a useful upper bound on the absolute error of a quadrature rule:

$$\left| \int_{[0,1]^p} g(\mathbf{x}) d\mathbf{x} - \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}^{(i)}) \right| \leq V(g) D_n^*(\mathbf{X}_n). \quad (4.8)$$

As explained in Oates (2021), $V(g) = \sum_{u \subseteq \{1, \dots, p\}} \int_{[0,1]^u} \left| \frac{\partial^u g}{\partial \mathbf{x}_u}(\mathbf{x}_u, 1) \right| d\mathbf{x}$, quantifies the complexity of the integrand, while $D_n^*(\mathbf{X}_n)$ evaluates the discrepancy to uniformity of the design \mathbf{X}_n . Therefore, the Koksma-Hlawka inequality shows that the quadrature rule’s accuracy relies on the good quantization of μ by \mathbf{X}_n . For a uniform target measure μ , the star discrepancy is a metric assessing how far from uniformity a sample \mathbf{X}_n is. When generalizing to a non-uniform measure, a good quantization of μ should also lead to a good approximation of the quantity.

Reproducing kernel Hilbert space and kernel mean embedding To generalize the Koksma-Hlawka inequality to any probability measure, let us assume that the integrand g lives in a specific function space $\mathcal{H}(k)$. $\mathcal{H}(k)$ is a *reproducing kernel Hilbert space* (RKHS), which is an inner product space $\mathcal{H}(k)$ of functions $g : \mathcal{D}_X \rightarrow \mathbb{R}$. Considering a symmetric and positive definite function $k : \mathcal{D}_X \times \mathcal{D}_X \rightarrow \mathbb{R}$, later called a “reproducing kernel” or simply a “kernel”, an RKHS verifies the following axioms:

- The “feature map” $\phi : \mathcal{D}_X \rightarrow \mathcal{H}(k); \phi(\mathbf{x}) = k(\cdot, \mathbf{x})$ belongs to the RKHS: $k(\cdot, \mathbf{x}) \in \mathcal{H}(k), \forall \mathbf{x} \in \mathcal{D}_X$.
- The “reproducing property”: $\langle g, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}(k)} = g(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}_X, \forall g \in \mathcal{H}(k)$.

Note that it can be shown that every positive semi-definite kernel defines a unique RKHS (and vice versa) with a feature map ϕ , such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}(k)}$. This framework allows us to embed a continuous or discrete probability measure in an RKHS, as illustrated in Fig. 4.7.

For any measure μ , its *kernel mean embedding* (Sejdinovic et al., 2013), also called “potential” $P_\mu(\mathbf{x})$ in Pronzato and Zhigljavsky (2020), associated with the kernel k is defined as:

$$P_\mu(\mathbf{x}) := \int_{\mathcal{D}_X} k(\mathbf{x}, \mathbf{x}') d\mu(\mathbf{x}'). \quad (4.9)$$

Respectively, the potential $P_{\zeta_n}(\mathbf{x})$ of a discrete distribution $\zeta_n = \sum_{i=1}^n w_i \delta(\mathbf{x}^{(i)})$, $w_i \in \mathbb{R}$ associated with the kernel k can be written as:

$$P_{\zeta_n}(\mathbf{x}) = \int_{\mathcal{D}_X} k(\mathbf{x}, \mathbf{x}') d\zeta_n(\mathbf{x}') = \sum_{i=1}^n w_i k(\mathbf{x}, \mathbf{x}^{(i)}). \quad (4.10)$$

The potential $P_\mu(\mathbf{x})$ of the targeted measure μ will be referred to as “target potential” and the potential $P_{\zeta_n}(\mathbf{x})$ associated with the discrete distribution ζ_n called “current potential” when its support is the design X_n . If $P_{\zeta_n}(\mathbf{x})$ is close to $P_\mu(\mathbf{x})$, it can be interpreted to mean that ζ_n is an adequate quantization or representation of μ by the discrete distribution ζ_n (and therefore lead to a good estimation of a quantity such as $I_\mu(g)$ from Eq. (4.7)). Potentials can be computed in closed forms for specific pairs of distribution and associated kernel. Summary tables of some of these cases are detailed in Briol (2019) (section 3.4), Pronzato and Zhigljavsky (2020) (section 4), and extended in Fekhari et al. (2023). However, in most cases, the target potentials must be estimated on a large and representative sample, typically a large quasi-Monte Carlo sample of μ .

Definition 1. The *energy* of a measure μ is defined as the integral of the potential P_μ against the measure, which leads to the following scalar quantity:

$$\varepsilon_\mu := \int_{\mathcal{D}_X} P_\mu(\mathbf{x}) d\mu(\mathbf{x}) = \iint_{\mathcal{D}_X^2} k(\mathbf{x}, \mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}'). \quad (4.11)$$

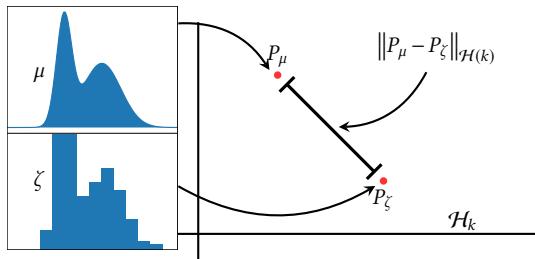


Fig. 4.7 Kernel mean embedding of a continuous and discrete probability distribution

Finally, using the reproducing property and writing the Cauchy-Schwarz inequality on the absolute quadrature error leads to the following inequality, similar to the Koksma-Hlawka inequality Eq. (4.8) (see Briol et al. (2019)):

$$\left| \sum_{i=1}^n w_i g(\mathbf{x}^{(i)}) - \int_{\mathcal{D}_X} g(\mathbf{x}) d\mu(\mathbf{x}) \right| = \left| \langle g, P_{\zeta_n}(\mathbf{x}) \rangle_{\mathcal{H}(k)} - \langle g, P_\mu(\mathbf{x}) \rangle_{\mathcal{H}(k)} \right| \quad (4.12a)$$

$$= \left| \langle g, (P_{\zeta_n}(\mathbf{x}) - P_\mu(\mathbf{x})) \rangle_{\mathcal{H}(k)} \right| \quad (4.12b)$$

$$\leq \|g\|_{\mathcal{H}(k)} \|P_\mu(\mathbf{x}) - P_{\zeta_n}(\mathbf{x})\|_{\mathcal{H}(k)}. \quad (4.12c)$$

Maximum mean discrepancy A metric of discrepancy and quadrature error is offered by the *maximum mean discrepancy* (MMD). This distance between two probability distributions μ and ζ is given by the worst-case error for any function within a unit ball of the Hilbert space $\mathcal{H}(k)$, associated with the kernel k :

$$\text{MMD}_k(\mu, \zeta) := \sup_{\|g\|_{\mathcal{H}(k)} \leq 1} \left| \int_{\mathcal{D}_X} g(\mathbf{x}) d\mu(\mathbf{x}) - \int_{\mathcal{D}_X} g(\mathbf{x}) d\zeta(\mathbf{x}) \right| \quad (4.13)$$

According to the inequality in Eq. (4.12c), $\text{MMD}_k(\mu, \zeta) = \|P_\mu - P_\zeta\|_{\mathcal{H}(k)}$, meaning that the MMD fully relies on the difference of potentials. Moreover, Sriperumbudur et al. (2010) defines a kernel as “characteristic kernel” when the following equivalence is true: $\text{MMD}_k(\mu, \zeta) = 0 \Leftrightarrow \mu = \zeta$. This property makes the MMD a metric on \mathcal{D}_X . The squared MMD has been used for other purposes than numerical integration: e.g., statistical testing (Gretton et al., 2006), and global sensitivity analysis (Da Veiga, 2015). It can be written as follows:

$$\text{MMD}_k(\mu, \zeta)^2 = \|P_\mu(\mathbf{x}) - P_\zeta(\mathbf{x})\|_{\mathcal{H}(k)}^2 \quad (4.14a)$$

$$= \langle (P_\mu(\mathbf{x}) - P_\zeta(\mathbf{x})), (P_\mu(\mathbf{x}) - P_\zeta(\mathbf{x})) \rangle_{\mathcal{H}(k)} \quad (4.14b)$$

$$= \langle P_\mu(\mathbf{x}), P_\mu(\mathbf{x}) \rangle_{\mathcal{H}(k)} - 2 \langle P_\mu(\mathbf{x}), P_\zeta(\mathbf{x}) \rangle_{\mathcal{H}(k)} + \langle P_\zeta(\mathbf{x}), P_\zeta(\mathbf{x}) \rangle_{\mathcal{H}(k)} \quad (4.14c)$$

$$= \iint_{\mathcal{D}_X^2} k(\mathbf{x}, \mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}') - 2 \iint_{\mathcal{D}_X^2} k(\mathbf{x}, \mathbf{x}') d\mu(\mathbf{x}) d\zeta(\mathbf{x}') + \iint_{\mathcal{D}_X^2} k(\mathbf{x}, \mathbf{x}') d\zeta(\mathbf{x}) d\zeta(\mathbf{x}'). \quad (4.14d)$$

Taking a discrete distribution with uniform weights $\zeta_n = \frac{1}{n} \sum_{i=1}^n \delta(\mathbf{x}^{(i)})$, the squared MMD reduces to:

$$\text{MMD}_k(\mu, \zeta_n)^2 = \varepsilon_\mu - \frac{2}{n} \sum_{i=1}^n P_\mu(\mathbf{x}^{(i)}) + \frac{1}{n^2} \sum_{i,j=1}^n k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}). \quad (4.15)$$

4.3.3 Kernel herding sampling

Herein, the MMD is used to quantize the known target measure μ by a design sample \mathbf{X}_n . For practical reasons, design construction is done sequentially. Sequential strategies can also be used to learn and validate regression models for statistical learning (see [Fekhari et al. \(2023\)](#)). Moreover, since each realization is supposed to be obtained at the same unitary cost, we fix the quadrature weights as uniform during the construction of the design \mathbf{X}_n .

Kernel herding (KH), proposed by [Chen et al. \(2010\)](#), is a sampling method that offers a quantization of the measure μ by minimizing a squared MMD when adding points iteratively. With a current design \mathbf{X}_n and its corresponding discrete distribution with uniform weights $\zeta_n = \frac{1}{n} \sum_{i=1}^n \delta(\mathbf{x}^{(i)})$, a KH iteration can be written as an optimization problem involving the following criterion over the point $\mathbf{x}^{(n+1)} \in \mathcal{D}_{\mathbf{X}}$:

$$\mathbf{x}^{(n+1)} \in \arg \min_{\mathbf{x} \in \mathcal{D}_{\mathbf{X}}} \left\{ \text{MMD}_k \left(\mu, \frac{1}{n+1} \left(\delta(\mathbf{x}) + \sum_{i=1}^n \delta(\mathbf{x}^{(i)}) \right) \right)^2 \right\}. \quad (4.16)$$

In the literature, two formulations of this optimization problem can be found. The first one uses the Frank-Wolfe algorithm (or “conditional gradient algorithm”) to compute a linearization of the problem under the convexity hypothesis (see [Lacoste-Julien et al. \(2015\)](#) and [Briol et al. \(2015\)](#) for more details). The second one is a straightforward greedy optimization. Due to the combinatorial complexity, the greedy formulation is tractable for sequential construction. To see this, let us develop the MMD from Eq. (4.15):

$$\text{MMD}_k \left(\mu, \frac{1}{n+1} \left(\delta(\mathbf{x}) + \sum_{i=1}^n \delta(\mathbf{x}^{(i)}) \right) \right)^2 = \varepsilon_{\mu} - \frac{2}{n+1} \sum_{i=1}^{n+1} P_{\mu} \left(\mathbf{x}^{(i)} \right) + \frac{1}{(n+1)^2} \sum_{i,j=1}^{n+1} k \left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right) \quad (4.17a)$$

$$= \varepsilon_{\mu} - \frac{2}{n+1} \left(P_{\mu}(\mathbf{x}) + \sum_{i=1}^n P_{\mu} \left(\mathbf{x}^{(i)} \right) \right) \quad (4.17b)$$

$$+ \frac{1}{(n+1)^2} \left(\sum_{i,j=1}^n k \left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right) + 2 \sum_{i=1}^n k \left(\mathbf{x}^{(i)}, \mathbf{x} \right) - k(\mathbf{x}, \mathbf{x}) \right). \quad (4.17c)$$

In the previously developed expression, only a few terms actually depend on the next optimal point $\mathbf{x}^{(n+1)}$ since the target energy, denoted by ε_{μ} , and $k(\mathbf{x}, \mathbf{x}) = \sigma^2$ are constant (by taking a stationary kernel). Therefore, the greedy minimization of the MMD can be equivalently written as:

$$\mathbf{x}^{(n+1)} \in \arg \min_{\mathbf{x} \in \mathcal{D}_{\mathbf{X}}} \left\{ \frac{1}{n+1} \sum_{i=1}^n k \left(\mathbf{x}^{(i)}, \mathbf{x} \right) - P_{\mu}(\mathbf{x}) \right\} = \arg \min_{\mathbf{x} \in \mathcal{D}_{\mathbf{X}}} \left\{ \frac{n}{n+1} P_{\zeta_n}(\mathbf{x}) - P_{\mu}(\mathbf{x}) \right\}. \quad (4.18)$$

Remark 1. For the sequential and uniformly weighted case, the formulation in Eq. (4.18) is almost similar to the Frank-Wolfe formulation. Our numerical experiments showed that these two versions generate very close designs, especially as n becomes large. [Pronzato and Rendas \(2021\)](#) express the Frank-Wolfe formulation in the sequential and uniformly weighted case as follows:

$$\mathbf{x}^{(n+1)} \in \arg \min_{\mathbf{x} \in \mathcal{D}_X} \{P_{\zeta_n}(\mathbf{x}) - P_\mu(\mathbf{x})\}. \quad (4.19)$$

Remark 2. In practice, the optimization problem is solved by a brute-force approach on a fairly dense finite subset $\mathcal{S} \subseteq \mathcal{D}_X$ of candidate points with size $N \gg n$ that emulates the target distribution, also called the “candidate set”. This sample is also used to estimate the target potential $P_\mu(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^N k(\mathbf{x}^{(i)}, \mathbf{x})$.

As explained previously, choosing the kernel defines the function space on which the worst-case function is found (see Eq. (4.13)). Therefore, this sampling method is sensitive to kernel choice. A kernel is defined, both by the choice of its parametric family (e.g., Matérn, squared exponential) and the choice of its tuning. The so-called “support points” method developed by [Mak and Joseph \(2018\)](#) is a special case of kernel herding that uses the characteristic and parameter-free “energy-distance” kernel (introduced by [Székely and Rizzo \(2013\)](#)). In the following numerical experiments, the energy-distance kernel will be compared with an isotropic tensor product of a Matérn kernel (with regularity parameter $v = 5/2$ and correlation lengths θ_i), or a squared exponential kernel (with correlation lengths θ_i) defined in Table 4.3. Since the Matérn and squared exponential kernels are widely used for Gaussian process regression ([Rasmussen and Williams, 2006](#)), they were naturally picked to challenge the energy-distance kernel. The correlation lengths for the squared exponential and Matérn kernels are set using the heuristic given in [Fekhari et al. \(2023\)](#), $\theta_i = n^{-1/p}$, $i \in \{1, \dots, p\}$.

Table 4.3 Kernels considered in the following numerical experiments.

Energy-distance	$k_E(\mathbf{x}, \mathbf{x}') = \frac{1}{2} (\ \mathbf{x}\ + \ \mathbf{x}'\ - \ \mathbf{x} - \mathbf{x}'\)$	
Squared exponential	$k_G(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p k_{\theta_i}(x_i - x'_i)$	$k_\theta(x - x') = \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$
Matérn ($v = 5/2$)	$k_M(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p k_{5/2, \theta_i}(x_i - x'_i)$	$k_{5/2, \theta}(x - x') = \left(1 + \frac{\sqrt{5}}{\theta} x - x' + \frac{5}{3\theta^2} (x - x')^2\right) \exp\left(-\frac{\sqrt{5}}{\theta} x - x' \right)$

Fig. 4.8 represents the covariance structure of the three kernels. One can notice that the squared exponential and Matérn $v = 5/2$ kernels are closer to one another than they are to the energy-distance. In fact, as v tends to infinity, the Matérn kernel tends toward the squared exponential kernel (which has infinitely differentiable sample paths, see [Rasmussen and Williams \(2006\)](#)). For these two stationary kernels, the correlation length controls how fast the correlation between two points decreases as their distance from one another increases. Meanwhile, the

energy distance is not stationary (but still positive and semi-definite). Therefore, its value does not only depend on the distance between two points but also on the norm of each of the points.

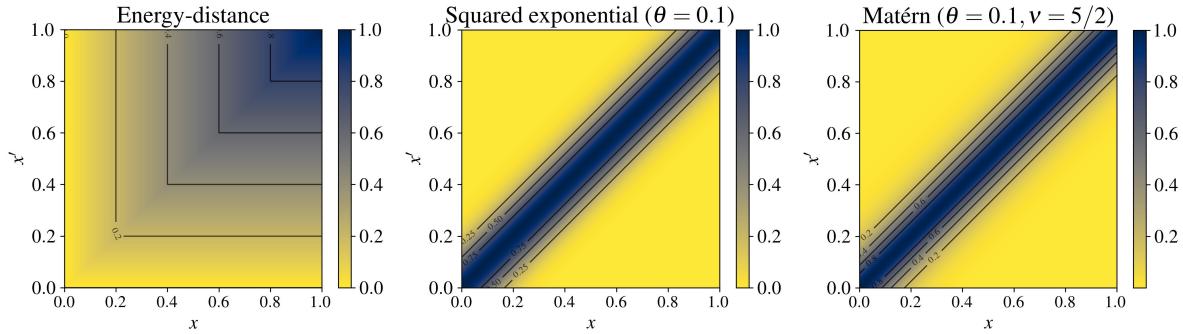


Fig. 4.8 Kernel illustrations (left to right: energy-distance, squared exponential, and Matérn 5/2)

To illustrate the sequential sampling of a complex distribution, Fig. 4.9 shows three nested kernel herding samples (orange crosses for different sizes $n \in \{10, 20, 40\}$) of a mixture of Gaussian distributions with complex nonlinear dependencies (with density represented by the black isoprobability contours). In this example, the method seems to build a parsimonious design between each mode of the distribution. The candidate set (in light grey) was generated by a large quasi-Monte sample of the underlying Gaussian mixture. In this two-dimensional case, this candidate set is sufficient to estimate the target potential P_μ . However, the main bottleneck of kernel herding is the estimation of the potentials, which becomes costly in high dimension.

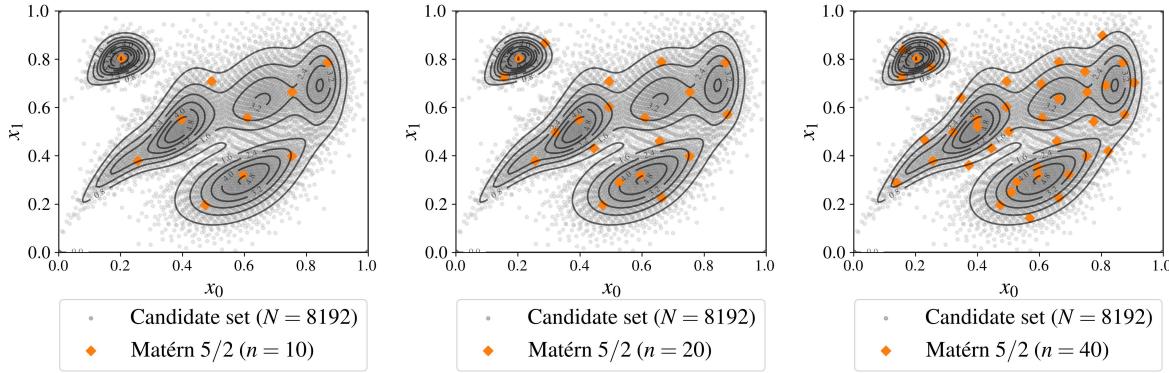


Fig. 4.9 Sequential kernel herding for increasing design sizes ($n \in \{10, 20, 40\}$) built on a candidate set of $N = 8196$ points drawn from a complex Gaussian mixture μ

Other approaches take advantage of the progressive knowledge acquired sequentially on g to select the following points in the design. These methods are sometimes called “active learning” or “adaptive strategies” (Fuhr. et al., 2021). Many of them rely on a sequentially updated Gaussian process (or Kriging) metamodel. To solve a probabilistic integration problem, the concept of Bayesian quadrature is introduced in the following.

4.3.4 Bayesian quadrature

Gaussian processes for Bayesian quadrature Kernel methods and Gaussian processes present a lot of connections and equivalences, thoroughly reviewed by [Kanagawa et al. \(2018\)](#). In numerical integration, Gaussian processes have been used to build quadrature rules in the seminal paper of [O'Hagan \(1991\)](#), introducing the concept of *Bayesian quadrature* (BQ). Let us recall the probabilistic integration problem $I_\mu(g) = \int_{\mathcal{D}_X} g(\mathbf{x}) d\mu(\mathbf{x})$ (introduced in Eq. (4.7)). From a general point of view, this quantity could be generalized by composing g with another function ψ (e.g., other moments, quantiles, exceedance probabilities). The quantity of interest then becomes, $I_\mu(\psi(g))$, for example when ψ is a monomial, it gives a moment the distribution of the output.

Let us assume, adopting a Bayesian point of view, that ξ is a stochastic process describing the uncertainty affecting the knowledge about the true function g . Let ξ be a Gaussian process (GP) prior with a zero trend (denoted by $\mathbf{0}$) to ease the calculation, and a stationary covariance kernel (denoted by $k(\cdot, \cdot)$). The conditional posterior $\xi_n := (\xi | \mathbf{y}_n) \sim \mathcal{GP}(\eta_n, s_n^2)$ has been conditioned on the function observations $\mathbf{y}_n = [g(\mathbf{x}^{(1)}), \dots, g(\mathbf{x}^{(n)})]^\top$ computed from the input design \mathbf{X}_n and is fully defined by the well-known “Kriging equations” (see e.g., [Rasmussen and Williams \(2006\)](#)):

$$\begin{cases} \eta_n(\mathbf{x}) &:= \mathbf{k}_n^\top(\mathbf{x}) \mathbf{K}_n^{-1} \mathbf{y}_n \\ s_n^2(\mathbf{x}) &:= k_n(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n^\top(\mathbf{x}) \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}) \end{cases} \quad (4.20)$$

where $\mathbf{k}_n(\mathbf{x})$ is the column vector of the covariance kernel evaluations $[k_n(\mathbf{x}, \mathbf{x}^{(1)}), \dots, k_n(\mathbf{x}, \mathbf{x}^{(n)})]$ and \mathbf{K}_n is the $(n \times n)$ variance-covariance matrix such that the (i, j) -element is $\{\mathbf{K}_n\}_{i,j} = k_n(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

In BQ, the main object is the random variable $I_\mu(\xi_n)$. According to [Briol et al. \(2019\)](#), its distribution on \mathbb{R} is the pushforward of ξ_n through the integration operator $I_\mu(\cdot)$, sometimes called *posterior distribution*:

$$I_\mu(\xi_n) = \int_{\mathcal{D}_X} (\xi(\mathbf{x}) | \mathbf{y}_n) d\mu(\mathbf{x}) = \int_{\mathcal{D}_X} \xi_n(\mathbf{x}) d\mu(\mathbf{x}). \quad (4.21)$$

Fig. 4.10 provides a one-dimensional illustration of the Bayesian quadrature of an unknown function (dashed black curve) against a given input measure μ (with corresponding grey distribution at the bottom). For an arbitrary design, one can fit a Gaussian process model, interpolating the function observations (black crosses). Then, multiple trajectories of this conditioned Gaussian process ξ_n are drawn (orange curves) whilst its mean function, also called “predictor”, is represented by the red curve. Therefore, the input measure μ is propagated through the conditioned Gaussian process to obtain the random variable $I_\mu(\xi_n)$, with distribution represented on the right plot (brown curve). Again on the right plot, remark how the mean of this posterior distribution (brown line) is closer to the reference output expected value (dashed black line) than the arithmetic mean of the observations (black line). This plot was inspired by the paper of [Huszár and Duvenaud \(2012\)](#).

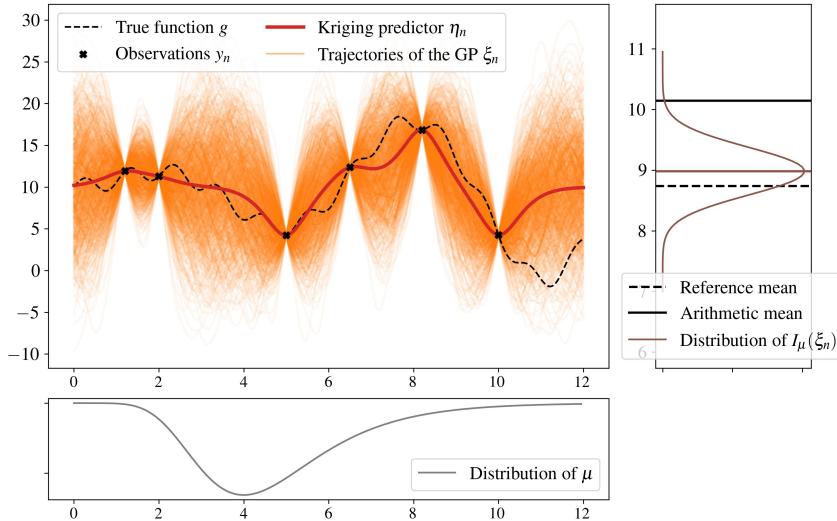


Fig. 4.10 Bayesian quadrature on a one-dimensional case

Optimal weights computed by Bayesian quadrature Taking the random process ξ_n as Gaussian conveniently implies that its posterior distribution $a_\mu(\xi_n)$ is also Gaussian. This comes from the linearity of the infinite sum of realizations of a Gaussian process. The posterior distribution is described in a closed form through its mean and variance by applying Fubini's theorem (see the supplementary materials from Briol et al. (2019) for the proof regarding the variance):

$$\bar{y}_n^{\text{BQ}} := \mathbb{E}[I_\mu(\xi_n)|\mathbf{y}_n] = \int_{\mathcal{D}_X} \eta_n(\mathbf{x}) d\mu(\mathbf{x}) = \left[\int_{\mathcal{D}_X} \mathbf{k}_n^\top(\mathbf{x}) d\mu(\mathbf{x}) \right] \mathbf{K}_n^{-1} \mathbf{y}_n = P_\mu(\mathbf{X}_n) \mathbf{K}_n^{-1} \mathbf{y}_n, \quad (4.22)$$

$$\left(\sigma_n^{\text{BQ}} \right)^2 := \text{Var}(I_\mu(\xi_n)) = \iint_{\mathcal{D}_{X^2}} k_n(\mathbf{x}, \mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}') = \varepsilon_\mu - P_\mu(\mathbf{X}_n) \mathbf{K}_n^{-1} P_\mu(\mathbf{X}_n)^\top. \quad (4.23)$$

Where $P_\mu(\mathbf{X}_n)$ is the row vector of potentials $[\int k_n(\mathbf{x}, \mathbf{x}^{(1)}) d\mu(\mathbf{x}), \dots, \int k_n(\mathbf{x}, \mathbf{x}^{(n)}) d\mu(\mathbf{x})]$, and ε_μ is given in Eq. (1). As in the one-dimensional example presented in Fig. 4.10, the expected value of $I_\mu(\xi_n)$ expressed in Eq. (4.22) is a direct estimator of the quantity of interest Eq. (4.7). The so-called “Bayesian quadrature estimator” appears to be a simple linear combination of the observations by taking the row vector of “optimal weights” as:

$$\mathbf{w}_{\text{BQ}} := P_\mu(\mathbf{X}_n) \mathbf{K}_n^{-1} \quad (4.24)$$

For any given sample, an optimal set of weights can be computed, leading to the mean of the posterior distribution. Remark here that this enhancement depends on the evaluation of the inverse variance-covariance matrix \mathbf{K}_n^{-1} , which can present numerical difficulties, either when design points are too close, making the conditioning bad. Moreover, a prediction interval on the BQ estimator can be computed since the posterior distribution is Gaussian, with a variance

expressed in closed-form in Eq. (4.23). The expressions in Eq. (4.22) and Eq. (4.23) were extended to Gaussian processes in the case of constant and linear trends in Pronzato and Zhigljavsky (2020). In the following numerical experiments, the expression with a hypothesis of constant trend β_n is used, which leads to:

$$\mathbb{E}[I_\mu(\xi_n)] = \beta_n + P_\mu(\mathbf{X}_n)\mathbf{K}_n^{-1}(\mathbf{y}_n - \beta_n\mathbf{1}_n). \quad (4.25)$$

Then, an a posteriori 95% prediction interval around the mean Bayesian estimator is directly given by:

$$\bar{y}_n^{\text{BQ}} \in \left[\bar{y}_n^{\text{BQ}} - 2\sigma_n^{\text{BQ}}, \bar{y}_n^{\text{BQ}} + 2\sigma_n^{\text{BQ}} \right]. \quad (4.26)$$

Variance-based Bayesian quadrature rule The link between the posterior variance and the squared MMD has been first made by Huszár and Duvenaud (2012) in their Proposition 1: the expected variance in the Bayesian quadrature $\text{Var}(I_\mu(\xi_n))$ is the MMD between the target distribution μ and $\zeta_n = \sum_{i=1}^n \mathbf{w}_{\text{BQ}}^{(i)} \delta(\mathbf{x}^{(i)})$. The proof is reproduced below (as well as in Proposition 6.1 from Kanagawa et al. (2018)):

$$\text{Var}(I_\mu(\xi_n)) = \mathbb{E}\left[\left(I_\mu(\xi_n) - I_{\zeta_n}(\xi_n)\right)^2\right] \quad (4.27a)$$

$$= \mathbb{E}\left[\left(\langle \xi_n, P_\mu \rangle_{\mathcal{H}(k)} - \langle \xi_n, P_{\zeta_n} \rangle_{\mathcal{H}(k)}\right)^2\right] \quad (4.27b)$$

$$= \mathbb{E}\left[\langle \xi_n, P_\mu - P_{\zeta_n} \rangle_{\mathcal{H}(k)}^2\right] \quad (4.27c)$$

$$= \|P_\mu - P_{\zeta_n}\|_{\mathcal{H}(k)}^2 \quad (4.27d)$$

$$= \text{MMD}_k(\mu, \zeta_n)^2. \quad (4.27e)$$

Note that the transition from equation (27c) to (27d) relies on the property stating that if ξ is a standard Gaussian process then $\forall g \in \mathcal{H}(k) : \langle \xi, g \rangle_{\mathcal{H}(k)} \sim \mathcal{N}(0, \|g\|_{\mathcal{H}(k)}^2)$. The method that sequentially builds a quadrature rule by minimizing this variance is called by the authors “Sequential Bayesian Quadrature” (SBQ). According to the previous proof, this criterion can be seen as an optimally-weighted version of the kernel herding criterion, as stated in the title of the paper from Huszár and Duvenaud (2012). Later, Briol et al. (2015) proved the weak convergence of $I_\mu(\xi_n)$ towards the target integral. Closer to wind turbines applications, Huchet (2019) and Huchet et al. (2019) introduced the “Adaptive Kriging Damage Assessment” method: a Kriging-based method for mean damage estimation that is very close to SBQ. However, This type of method inherits the limits from both KH and BQ since it searches for optimal design points among a candidate set and computes an inverse variance-covariance matrix. These numerical operations both scale hardly in high dimension.

Remark 3. Every quadrature method introduced in this section has been built without any observation of the possibly costly function g . Therefore, they cannot be categorized as active learning approaches. Contrarily, Kanagawa and Hennig (2019) presents a set of methods for

BQ with transformations (i.e., adding a positivity constraint on the function g), which are truly active learning methods.

4.4 Numerical experiments

This section presents numerical results computed on two different analytical toy-cases, respectively in dimension 2 (toy-case #1) and dimension 10 (toy-case #2), with easy-to-evaluate functions $g(\cdot)$ and associated input distributions μ . Therefore, reference values can easily be computed with great precision. For each toy-case a large reference Monte Carlo sample ($N_{\text{ref}} = 10^8$) is taken. This first benchmark compares the mean estimation of toy-cases given by quasi-Monte Carlo Sobol' sequences (abbreviation by QMC in the next figures), and kernel herdings with the three kernels defined in Table 4.3. Notice that the quasi-Monte Carlo designs are first generated on the unit cube, then transformed using the generalized Nataf transformation to follow the target distribution (Lebrun and Dutfoy, 2009). Additionally, the performances of kernel herding for both uniform and optimally-weighted Eq. (4.25) estimators are compared.

The kernel-based sampling and BQ methods were implemented in a new open-source Python package named `otkerneldesign`³. This development mostly relies on OpenTURNS⁴, an “Open source initiative for the Treatment of Uncertainties, Risks’N Statistics”, see Baudin et al. (2017). The following numerical experiments are available in the Git repository named `ctbenchmark`⁵.

4.4.1 Illustration through analytical toy-cases

The toy-cases were chosen to cover a large panel of complex probabilistic integration problems, completing the ones from ?. To assess the complexity of numerical integration problems, Owen (2003) introduced the concept of the “effective dimension” of an integrand function (number of the variables that actually impact the integral). The author showed that functions built on sums yield a low effective dimension (unlike functions built on products). In the same vein, Kucherenko et al. (2011) build three classes of integrand sorted by difficulty depending on their effective dimension:

- *class A*: problem with a few dominant variables.
- *class B*: problem without unimportant variables, and important low-order interaction terms.
- *class C*: problems without unimportant variables, and important high-order interaction terms.

³<https://efekhari27.github.io/otkerneldesign/master/index.html>

⁴<https://openturns.github.io/www/>

⁵<https://github.com/efekhari27/ctbenchmark>

The 10-dimensional “GSobol function” (toy-case #2) with a set of coefficient $\{a_i = 2\}_{i=1}^{10}$ has an effective dimension equal to 10 and belongs to the hardest class C from Kucherenko et al. (2011). In the case of the two-dimensional Gaussian mixture problem, the complexity is carried by the mixture of Gaussian distributions with highly nonlinear dependencies. Probabilistic integration results are presented in Fig. 4.11 (toy-case #1) and Fig. 4.12 (toy-case #2). Kernel herding samples with kernels defined in Table 4.3 (squared exponential in green, Matérn in orange, and energy-distance in red), are compared with a quasi-Monte Carlo sample (Sobol’ sequences in grey). Convergences of the arithmetic means are plotted on the left and MMDs on the right. The respective BQ estimators of the means are plotted in dashed lines.

Table 4.4 Analytical toy-cases

Toy-case #1	$dim = 2$	$g_1(\mathbf{x}) = x_1 + x_2$	Gaussian mixture from Fig. 4.9
Toy-case #2	$dim = 10$	$g_2(\mathbf{x}) = \prod_{i=1}^{10} \frac{ 4x_i - 2 + a_i}{1 + a_i}, \{a_i = 2\}_{i=1}^{10}$	Gaussian $\mathcal{N}(0.5, \mathbf{I}_{10})$

Remark 4. Different kernels are used in these numerical experiments. First, the generation kernel, used by the kernel herding algorithm to generate designs (with the heuristic tuning defined in Section 4.3.3). Second, the BQ kernel allows computation of the optimal weights (arbitrarily set up as a Matérn 5/2 with the heuristic tuning). Third, the evaluation kernel, which must be common to allow a fair comparison of the computed MMD results (same as the BQ kernel).

About toy-case #1. KH consistently converges faster than quasi-Monte Carlo in this case, especially for small sizes in terms of MMD. BQ weights tend to reduce the fluctuations in the mean convergence, which ensures better performance for any size. Overall, applying the weights enhances to convergence rate.

About toy-case #2. Although quasi-Monte Carlo is known to suffer the “curse of dimensionality”, KH does not outperform it drastically in this example. In fact, KH with uniform weights performs worse than quasi-Monte Carlo while optimally-weighted KH does slightly better. Moreover, the results confirm that $MMD_{BQ} < MMD_{unif}$ for all our experiments. The application of optimal-weights to the quasi-Monte Carlo sample slightly improves the estimation on this case. Note that the prediction interval around the BQ estimator is not plotted for the sake of readability.

In these two toy-cases, the MMD is shown to quantify numerical integration convergence well, which illustrates the validity of the inequality given in Eq. (4.12c), similar to the Koksma-Hlawka inequality, recalled in Eq. (4.8).

4.4.2 Application to the Teesside wind turbine fatigue estimation

Before analyzing the performance of the KH on the industrial application, let us notice that the copulogram Fig. 4.14 seems to be in line with the global sensitivity analysis presented in Murcia et al. (2018) and Li and Zhang (2020). In particular, the fact that the scatter plot of mean

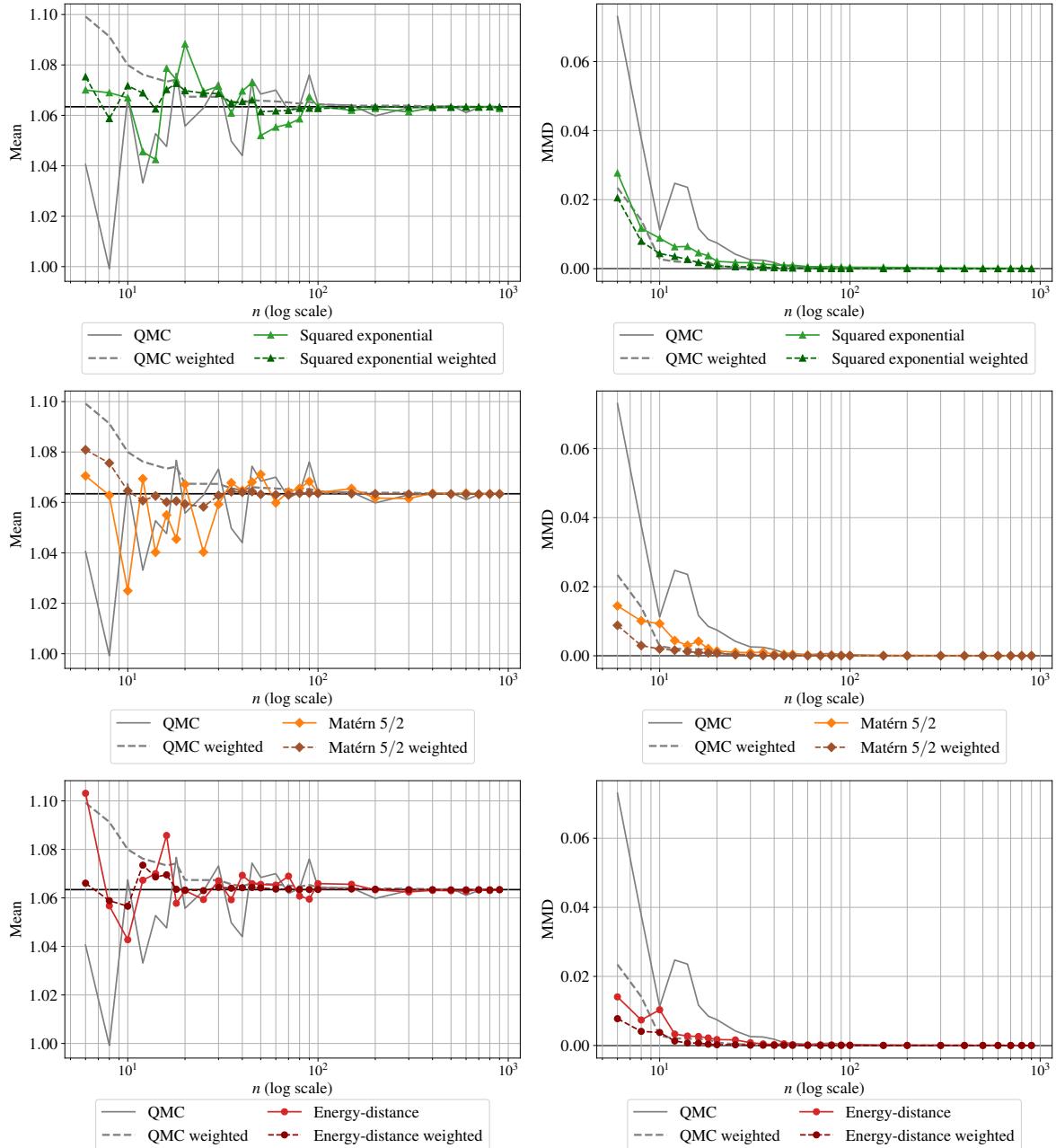


Fig. 4.11 Analytical benchmark results on the toy-case #1

wind speed vs. turbulence wind speed is the main factor explaining the variance of the output $Y = g(\mathbf{X})$. Judging from these references, the numerical model does not seem to have high effective dimension, however, the input dependence structure is challenging and the damage assessment induces strong nonlinearities (see Eq. (4.4)).

Crude Monte Carlo and kernel herding both subsample directly from a large dataset (previously referred to as candidate set). Unlike them, quasi-Monte Carlo generates a uniform sample in the unit hypercube, which can then be transformed according to a target distribution. In our case, this distribution is only known empirically via the candidate set. Since its dependence structure is complex (see Fig. 4.4), a parametric model might fit the distribution poorly (and

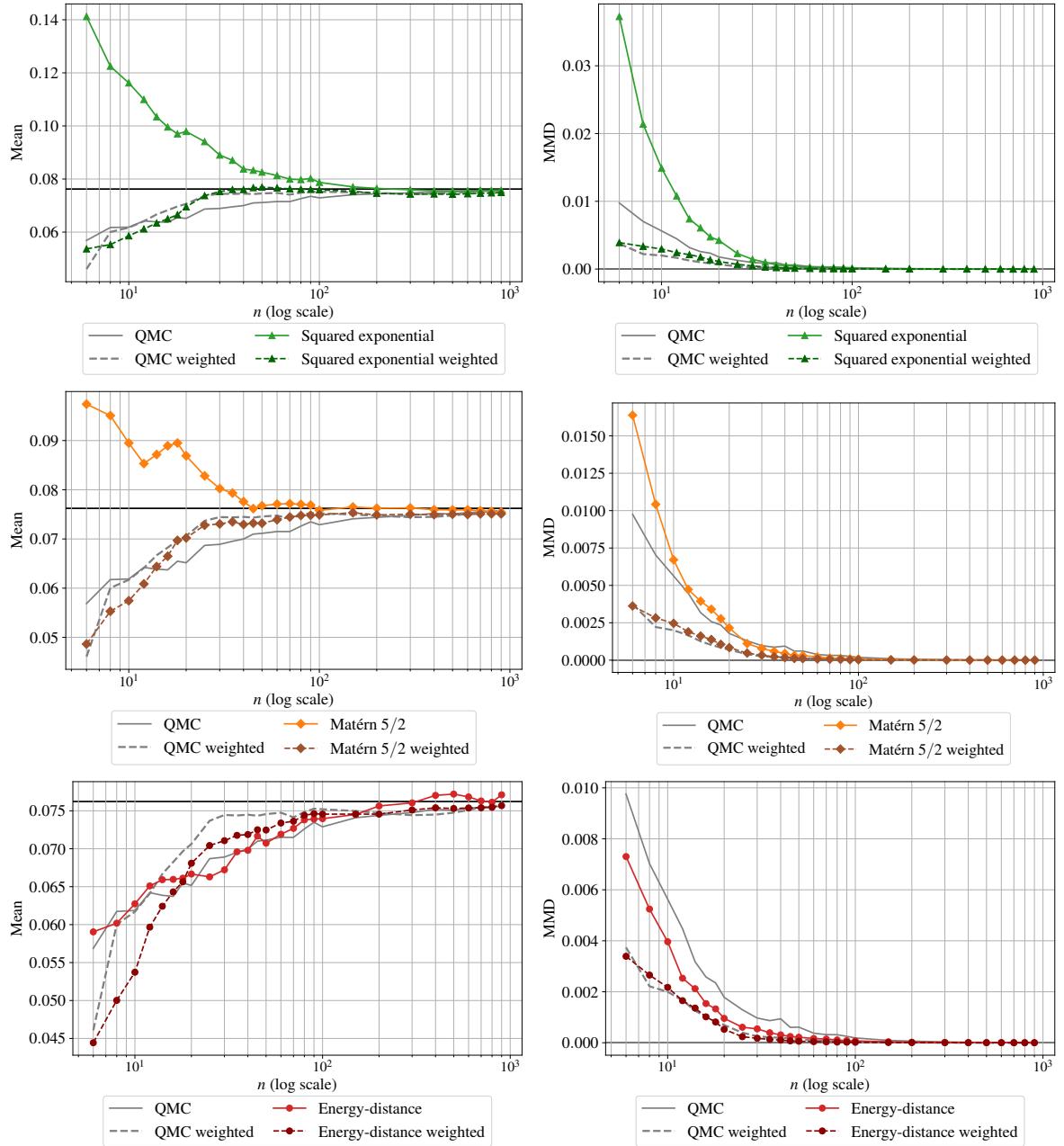


Fig. 4.12 Analytical benchmark results on the toy-case #2

therefore lead to a poor quasi-Monte Carlo estimation of the quantity). Then, a nonparametric fit using the empirical Bernstein copula (introduced in Section 4.2.3) coupled with a kernel density estimation on each marginal is applied to the candidate set (with the EBC parameter $m = 100 > m_{MISE}$ to avoid bias). Subsequently, quasi-Monte Carlo sampling is applied to this nonparametric model. These two approaches are summarized in Fig. 4.13, showing a practical advantage to the subsampling methods.

The results presented are compared in the following to a reference Monte Carlo sample with a confidence interval computed by bootstrap (see Fig. 4.15). The performance of the KH is good: it quickly converges towards the confidence interval of the Monte Carlo obtained with

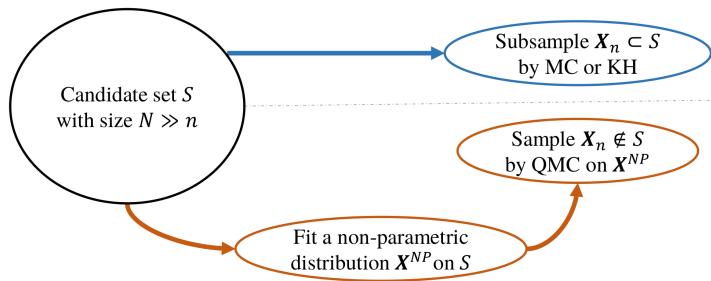


Fig. 4.13 Sampling techniques used for the industrial use case

the reference sample. In addition, the Bayesian quadrature estimator also offers a posteriori prediction interval, which can reassure the user. The BQ prediction intervals are smaller than the ones obtained by bootstrap on the reference Monte Carlo sample. To provide more representative results, note that a set of scale parameters is computed with a kriging procedure to define the kernel used to compute BQ intervals. Since other methods do not generate independent samples, bootstrapping them is not legitimate. Contrarily to the other kernel, we observe that the energy-distance kernel presents a small bias with the MC reference for most of the azimuth angles computed in this experiment. Finally, combining nonparametric fitting with quasi-Monte Carlo sampling also delivers good results as long as the fitting step does not introduce a bias.

4.5 Conclusion

Wind energy assets are subject to highly uncertain environmental conditions. Uncertainty propagation through numerical models is performed to ensure their structural integrity (and energy production). For this case, the method recommended by the standards (regular grid sampling) is intractable. This can lead, in practice, to poor uncertainty propagation under the constraint of a simulation budget. This industrial use case induces two practical constraints. First, active learning methods are hard to set up on such a numerical model, and they restrict the use of high-performance computers. Second, the input distribution of the environmental conditions presents a complex dependence structure, hard to model with parametric approaches.

In this paper, the association of kernel herding sampling with Bayesian quadrature for central tendency has been both explored theoretically and numerically. This method fits with the practical constraints induced by the industrial use case. Kernel herding sampling subsamples the relevant points directly from a given dataset (here from the measured environmental data). Moreover, the method is fully compatible with intensive high-performance computer use. This work provides an MMD-based upper bound on numerical integration absolute error. Kernel herding and Bayesian quadrature both aim at finding the quadrature rule minimizing the MMD, and therefore the absolute integration error. The numerical experiments confirmed that the MMD is an appropriate criterion since it leads to results being better or equivalent to quasi-Monte Carlo sampling. This numerical benchmark relied on the Python package, called `otkerneldesign`, implementing the methods.

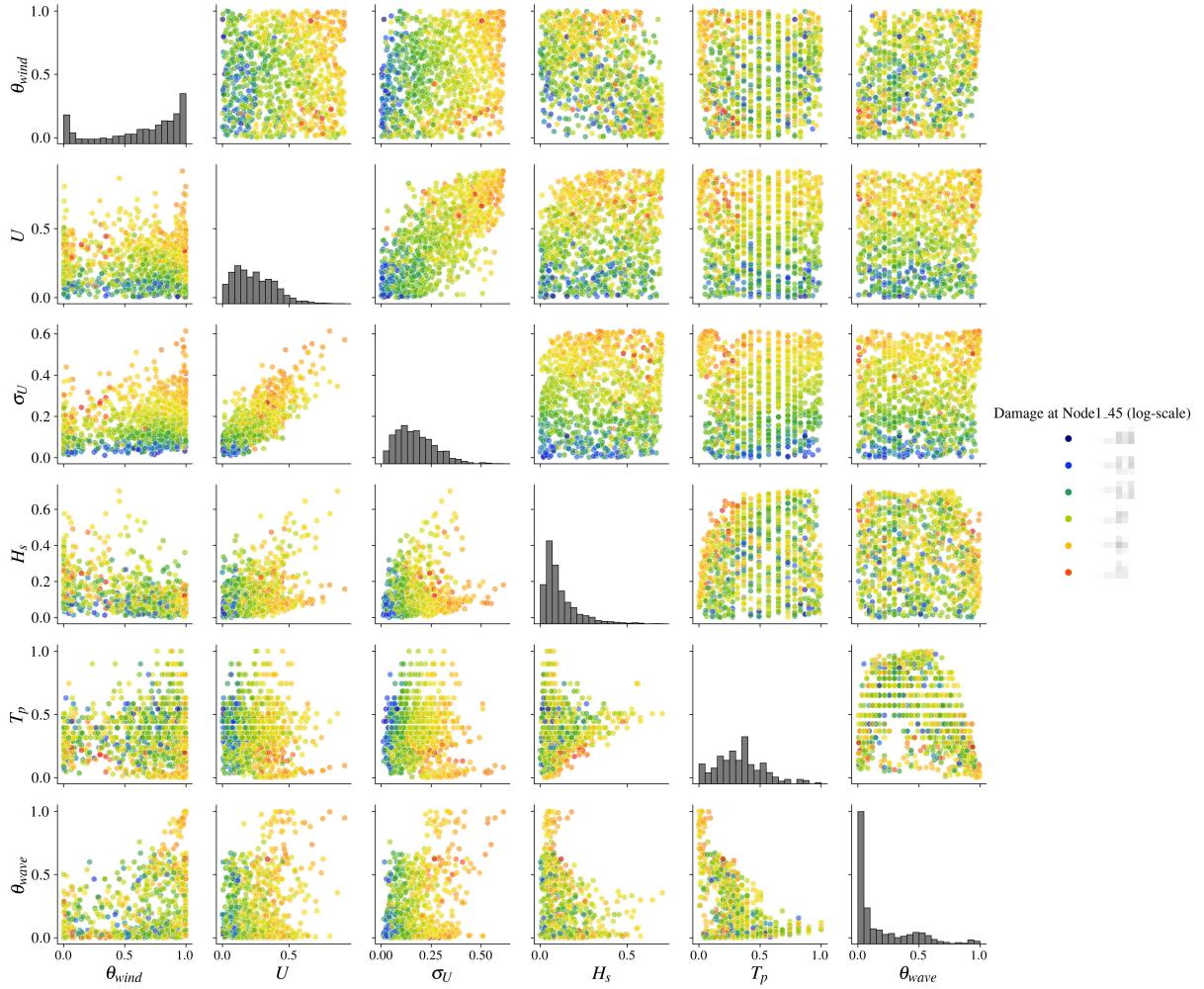


Fig. 4.14 Copulogram of the kernel herding design of experiments with corresponding outputs in color (log-scale) on the Teesside case ($n = 10^3$). The highest values are in red while the lowest values are in blue. Marginals are represented by histograms (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.

The limits of this method are reached when the problem dimension increases considerably. Moreover, it showed to be sensitive to the choice of the kernel and its tuning (although good practices were offered). From a methodological point of view, further interpretation of the impact of the different kernels should be explored. Then, the kernel herding sampling could be used to estimate quantiles, following the work on randomized quasi-Monte Carlo for quantiles of [Kaplan et al. \(2019\)](#). Kernel herding could also be used to quantize conditional distributions, using the conditional kernel mean embedding concept reviewed by [Klebanov et al. \(2020\)](#). Regarding the industrial use case, the next step is to realize a reliability analysis by considering another group of random variables (related to the wind turbine). Among other ideas, our upcoming work could explore a reliability-oriented sensitivity analysis by adapting recent kernel-based sensitivity indices ([Marrel and Chabridon, 2021](#)) to the sensitivity of a failure probability.

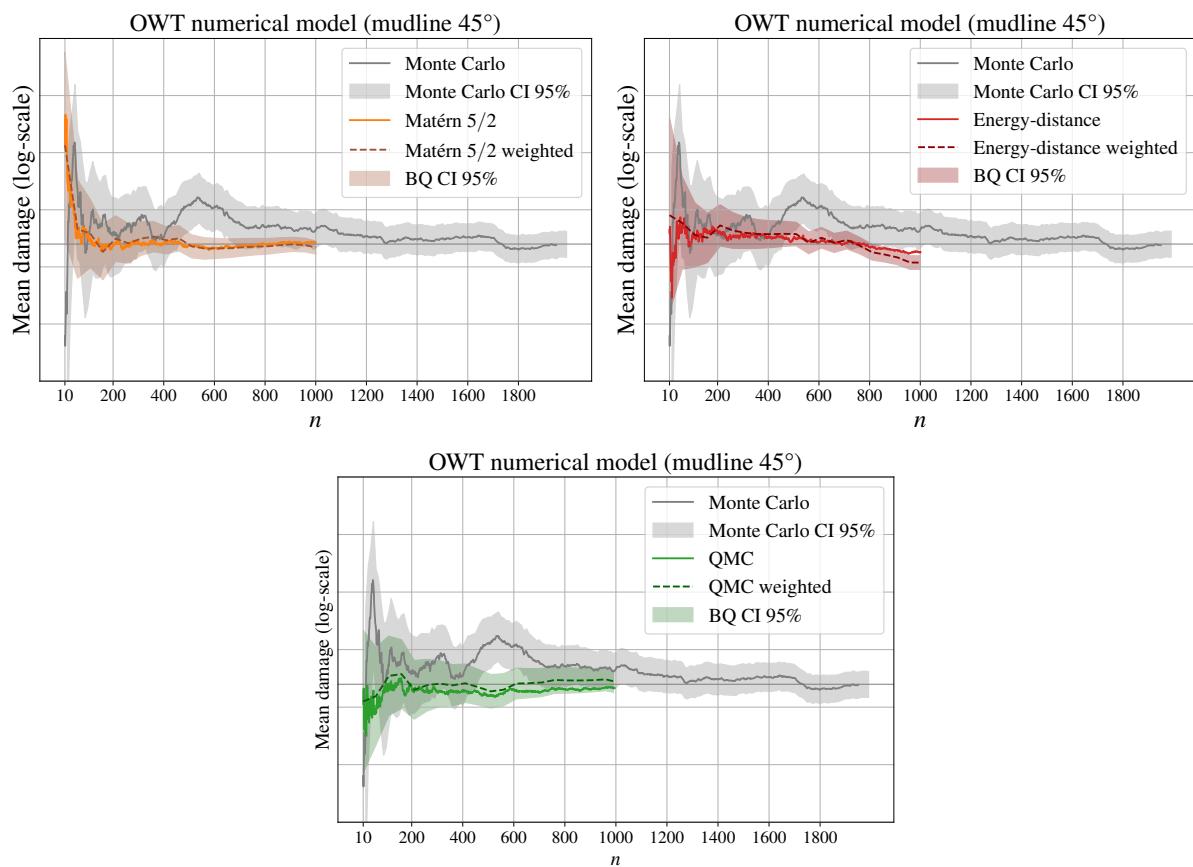


Fig. 4.15 Mean estimation convergence (at the mudline, azimuth $\theta = 45$ deg.) on the Teesside case

Chapter **5**

Kernel-based metamodel validation

5.1	Introduction	82
5.2	Sequential validation design	82
5.3	Computer experiments context	82
5.4	Machine learning given-data context	82
5.5	Application to wind turbine production metamodel	82
5.6	Conclusion	82

5.1 Introduction**5.2 Sequential validation design****5.3 Computer experiments context****5.4 Machine learning given-data context****5.5 Application to wind turbine production metamodel****5.6 Conclusion**

PART III:

CONTRIBUTIONS TO RARE EVENT ESTIMATION

La résignation est un suicide quotidien.

H. BALZAC

Nonparametric rare event estimation

6.1	Introduction	86
6.1.1	Background	87
6.2	Bernstein adaptive nonparametric conditional sampling	88
6.2.1	Empirical Bernstein copula	88
6.2.2	Bernstein adaptive nonparametric conditional sampling algorithm . .	90
6.3	Numerical experiments	91
6.3.1	Results analysis	93
6.4	Application to wind turbine fatigue reliability	94
6.5	Conclusion	94

6.1 Introduction

Reliability analysis of a system is often associated with rare event probability estimation. Considering that the system's performance is modeled by a deterministic scalar function $g : \mathcal{D}_x \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$, called *limit-state function* and a critical threshold on the system's output $y_{\text{th}} \in \mathbb{R}$, one can define the *failure domain* as $\mathcal{F}_x := \{\mathbf{x} \in \mathcal{D}_x | g(\mathbf{x}) \leq y_{\text{th}}\}$. Uncertain inputs are represented by a continuous random vector $\mathbf{X} \in \mathcal{D}_x$ assumed to be distributed according to its joint probability density function (PDF) f_X . In this context, uncertainty propagation consists in composing the random vector \mathbf{X} by the function g to get an output variable of interest $Y = g(\mathbf{X}) \in \mathbb{R}$. A usual risk measure in reliability analysis is the *failure probability*, denoted by p_f , and defined as the probability that the system exceeds the threshold y_{th} : Rare event problems are usually solved in the so-called *standard normal space* after applying an “iso-probabilistic transformation” which can be either the Rosenblatt or the generalized Nataf one (Lebrun, 2013). Additionally, the limit-state function g can be viewed as an input-output “black-box” model which can be costly to evaluate (e.g., a complex numerical model), making the failure probability estimation nontrivial. When the limit-state function is a costly computer model, one can build a surrogate model and use specific active learning methods (see, e.g., Moustapha et al. (2022)). However, using surrogate models is not always possible for practical engineering applications as they might introduce another level of approximation, which can be prohibitive from safety auditing. Moreover, their validation as well as their behavior with respect to large input dimension case make also their use quite complex (see, e.g., (Marrel et al., 2022)).

Going back to the rare event estimation literature, one can consider two major types of techniques for failure probability calculation (Morio and Balesdent, 2015): (i) Geometric approaches, such as the *first-/second-order reliability method* (FORM/SORM) whose aim is to approximate the limit-state function by a first-/second-order Taylor expansion at the most probable failure point; (ii) Simulation-based techniques such as the *crude Monte Carlo* method. Unfortunately, FORM/SORM methods do not provide a lot of statistical information as they are purely geometric approaches. Meanwhile, estimating a rare event probability by crude Monte Carlo becomes rapidly intractable. To overcome this limit, advanced simulation techniques have been developed: among others, one can mention several “variance reduction methods” such as the non-adaptive and adaptive versions of the *Importance Sampling* (Rubinstein and Kroese, 2008) (either parametric, using the Cross-Entropy method Kurtz and Song (2013), or nonparametric Morio (2011)) and splitting techniques (Cérou et al., 2012) such as the *Subset Simulation* (SS) Au and Beck (2001). In these techniques, the idea is to write the rare event p_f as a product of larger conditional probabilities, each one of them being easier to estimate. To generate intermediary conditional samples, this method uses Markov chain Monte Carlo (MCMC) sampling, which presents numerous versions (Papaioannou et al., 2015). However, MCMC algorithms are known to be highly tunable algorithms which produce non-i.i.d. samples, which consequently, cannot be used for direct statistical estimation (e.g., failure probability or sensitivity indices (Da Veiga et al., 2021).

The present work proposes a new rare event estimation method, adopting the same sequential structure as SS while using a strictly different sampling mechanism to generate conditional samples. This method intends to fit the intermediary conditional distributions with a nonparametric tool called the *Empirical Bernstein Copula*. Contrarily to SS, the proposed method named “Bernstein adaptive nonparametric conditional sampling” (BANCS), generates i.i.d. samples of the intermediary conditional distributions. For instance, a practical use of such i.i.d. samples can be to estimate dedicated reliability-oriented sensitivity indices (see, e.g., Chabridon et al. (2021); Marrel and Chabridon (2021)).

In this paper, Section 2 will recall the methodology of subset sampling and probabilistic modeling. Then, Section 3 will introduce the BANCS method for rare event estimation. Section 4 will apply this method to three toy-cases and analyze the results with respect to SS performances. Then, the last section present some conclusions and research perspectives.

6.1.1 Background

Subset sampling

Subset sampling splits the failure event \mathcal{F}_x into an intersection of $k_\#$ intermediary events $\mathcal{F}_x = \cap_{k=1}^{k_\#} \mathcal{F}_{[k]}$. Each are nested such that $\mathcal{F}_{[1]} \supset \dots \supset \mathcal{F}_{[k_\#]} = \mathcal{F}_x$. The failure probability is then expressed as a product of conditional probabilities:

$$p_f = \mathbb{P}(\mathcal{F}_x) = \mathbb{P}(\cap_{k=1}^{k_\#} \mathcal{F}_{[k]}) = \prod_{k=1}^{k_\#} \mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}). \quad (6.1)$$

From a practical point of view, the analyst tunes the algorithm by setting the intermediary probabilities $\mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}) = p_0, \forall k \in \{1, \dots, k_\#\}$. Then, the corresponding quantiles $q_{[1]}^{p_0} > \dots > q_{[k_\#]}^{p_0}$ are estimated for each conditional subset samples $X_{[k],N}$ of size N . Note that the initial quantile is estimated by crude Monte Carlo sampling on the input PDF f_x . Following conditional subset samples are generated by MCMC sampling of $f_x(x | \mathcal{F}_{[k-1]})$, using as seeds initialisation points the $n = Np_0$ samples given by $A_{[k],n} = \{X_{[k-1]}^{(j)} \subset X_{[k-1],N} | g(X_{[k-1]}^{(j)}) > \hat{q}_{[k-1]}^{\alpha}\}_{j=1}^n$. This process is repeated until an intermediary quantile exceeds the threshold: $\hat{q}_{[k_\#]}^{p_0} < y_{\text{th}}$. Finally, the failure probability is estimated by:

$$p_f \approx \hat{p}_f^{\text{SS}} = p_0^{k_\# - 1} \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{\{g(x) \leq y_{\text{th}}\}}(X_{[k_\#],N}^{(j)}). \quad (6.2)$$

In practice, the subset sample size should be large enough to properly estimate intermediary quantiles, which leads Au and Beck (2001) to recommend setting $p_0 = 0.1$. SS efficiency depends on the proper choice and tuning of the MCMC algorithm (Papaioannou et al., 2015). Our work uses the SS implementation from OpenTURNS¹ (Baudin et al., 2017) which integrates a

¹<https://openturns.github.io/www/index.html>

component-wise Metropolis-Hastings algorithm. As an alternative to generating samples on a conditional distribution by MCMC, one could try to fit this conditional distribution.

Multivariate modeling using copulas

The Sklar theorem ([Joe, 1997](#)) affirms that the multivariate distribution of any random vector $\mathbf{X} \in \mathbb{R}^d$ can be broken down into two objects:

1. A set of univariate marginal distributions to describe the behavior of the individual variables;
2. A function describing the dependence structure between all variables, called a copula.

This theorem states that considering a random vector $\mathbf{X} \in \mathbb{R}^d$, with its distribution F and its marginals $\{F_i\}_{i=1}^d$, there exists a copula $C : [0, 1]^d \rightarrow [0, 1]$, such that:

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_p(x_d)). \quad (6.3)$$

It allows us to divide the problem of fitting a joint distribution into two independent problems: fitting the marginals and fitting the copula. Note that when the joint distribution is continuous, this copula is unique. Provided a dataset, this framework allows to combine a parametric (or nonparametric) fit of marginals with a parametric (or nonparametric) fit of the copula. When the distribution's dimension is higher than two, one can perform a parametric fit using vine copulas ([Joe and Kurowicka, 2011](#)), implying the choice of multiple types of parametric copulas. Otherwise, nonparametric fit by multivariate kernel density estimation (KDE) presents a computational burden as soon as the dimension increases ([Chabridon et al., 2021](#)). Since univariate marginals are usually well-fitted with nonparametric tools (e.g., KDE), let us introduce an effective nonparametric method for copula fitting.

6.2 Bernstein adaptive nonparametric conditional sampling

6.2.1 Empirical Bernstein copula

Copulas are continuous and bounded functions defined on a compact set (the unit hypercube). Bernstein polynomials allow to uniformly approximate as closely as desired any continuous and real-valued function defined on a compact set (Weierstrass approximation theorem). Therefore, they are good candidates to approximate unknown copulas. This concept was introduced as *empirical Bernstein copula* (EBC) by [Sancetta and Satchell \(2004\)](#) for applications in economics and risk management. Later on, [Segers et al. \(2017\)](#) offered further asymptotic studies. Formally, the multivariate Bernstein polynomial for a function $C : [0, 1]^d \rightarrow \mathbb{R}$ on a grid over the unit

hypercube $G := \left\{ \frac{0}{m_1}, \dots, \frac{m_1}{m_1} \right\} \times \dots \times \left\{ \frac{0}{m_d}, \dots, \frac{m_d}{m_d} \right\}$, $\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}^d$, writes:

$$B_{\mathbf{m}}(C)(\mathbf{u}) := \sum_{t_1=0}^{m_1} \dots \sum_{t_d=0}^{m_d} C\left(\frac{t_1}{m_1}, \dots, \frac{t_d}{m_d}\right) \prod_{j=1}^d P_{m_j, t_j}(u_j), \quad (6.4)$$

with $\mathbf{u} = (u_1, \dots, u_d) \in [0, 1]^d$, and the Bernstein polynomial $P_{m, t}(u) := \frac{t!}{m!(t-m)!} u^m (1-u)^{t-m}$. Notice how the grid definition implies the polynomial's order. When C is a copula, then $B_{\mathbf{m}}(C)$ is called “Bernstein copula”. Therefore, the empirical Bernstein copula is an application of the Bernstein polynomial in Eq. (B.2) to the so-called “empirical copula”.

In practice, considering a sample $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{np}$ and the associated ranked sample $\mathbf{R}_n = \{\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}\}$, the corresponding empirical copula writes:

$$C_n(\mathbf{u}) := \frac{1}{n} \sum_{i=0}^n \prod_{j=1}^p \mathbb{1} \left\{ \frac{r_j^{(i)}}{n} \leq u_j \right\}, \quad (6.5)$$

with $\mathbf{u} = (u_1, \dots, u_d) \in [0, 1]^d$. In the following, the polynomial order is set as equal in each dimension: $\{m_i = m\}_{j=1}^d$. Theoretically, the tuning parameter can be optimized to minimize an “Mean Integrated Squared Error” (MISE), leading to a bias-variance tradeoff. Formally, the MISE of the empirical Bernstein copula $B_{\mathbf{m}}(C_n)$ is defined as follows:

$$\mathbb{E} [\|B_{\mathbf{m}}(C_n) - C\|_2^2] = \mathbb{E} \left[\int_{\mathbb{R}^d} (B_{\mathbf{m}}(C_n)(\mathbf{u}) - C(\mathbf{u}))^2 \right]. \quad (6.6)$$

Then, [Sancetta and Satchell \(2004\)](#) prove in their Theorem 3 that:

- $B_{\mathbf{m}}(C_n)(\mathbf{u}) \rightarrow C(\mathbf{u})$ for any $u_j \in]0, 1[$ if $\frac{m^{d/2}}{n} \rightarrow 0$, when $m, n \rightarrow \infty$.
- The optimal order of the polynomial in terms of MISE is: $m \lesssim m_{\text{IMSE}} = n^{2/(d+4)}$, $\forall u_j \in]0, 1[$. The sign \lesssim means “less than or approximately”.

Let us remark that in the special case $m = n$, also called the “Beta copula” in [Segers et al. \(2017\)](#), the bias is very small while the variance gets large. To illustrate the previous theorem, [Lasserre \(2022\)](#) represents the evolution of the m_{IMSE} for different dimensions and sample sizes (see Fig. B.1). In high dimension, the values of m_{IMSE} tend towards one, which is equivalent to the independent copula. Therefore, high-dimensional problems should be divided into a product of smaller problems on which the EBC is tractable. Provided a large enough learning set \mathbf{X}_n , KDE fitting of marginals combined with EBC fitting of the copula delivers good results even on complex dependence structures. Moreover, EBC provides an explicit expression, making a Monte Carlo generation of i.i.d. samples simple. In the following, this nonparametric tool is used to fit the intermediary conditional distributions present in subset sampling.

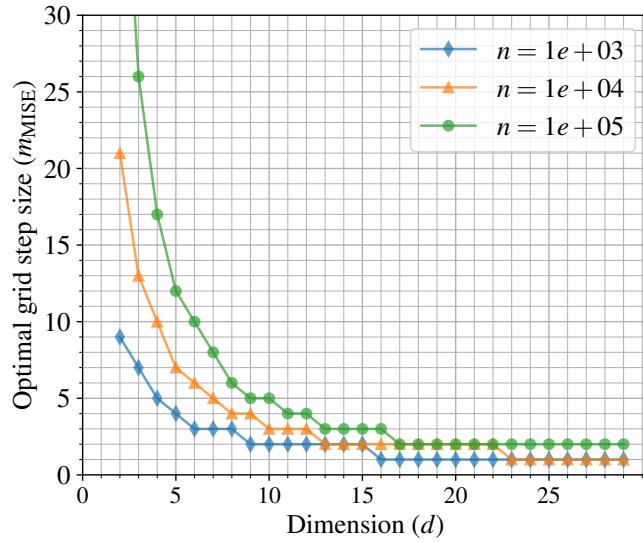


Fig. 6.1 Evolution of m_{IMSE} for different dimensions and sample sizes.

6.2.2 Bernstein adaptive nonparametric conditional sampling algorithm

This new method reuses the main idea from SS while employing a different approach to generate conditional samples. Instead of using MCMC sampling, the conditional distribution is firstly fitted by a nonparametric procedure, before sampling on this nonparametric model. As described in Algorithm 1, conditional sampling is done on a distribution composed by merging marginals $\{\widehat{F}_i\}_{i=1}^d$ fitted by KDE, with a copula $B_m(C_n)$ fitted by EBC. Fig. 6.2 illustrate the nonparametric fit and conditional sampling in BANCS method on a two-dimensional reliability problem (later introduced as “toy-case #1”). At iteration k , after estimating the intermediary quantile $\widehat{q}_{[k]}^{p_0}$, a nonparametric model is fitted on $A_{[k+1],n}$ and used to generate the next N -sized subset sample $X_{[k+1],N}$. Note that the BANCS method does not require iso-probabilistic transform.

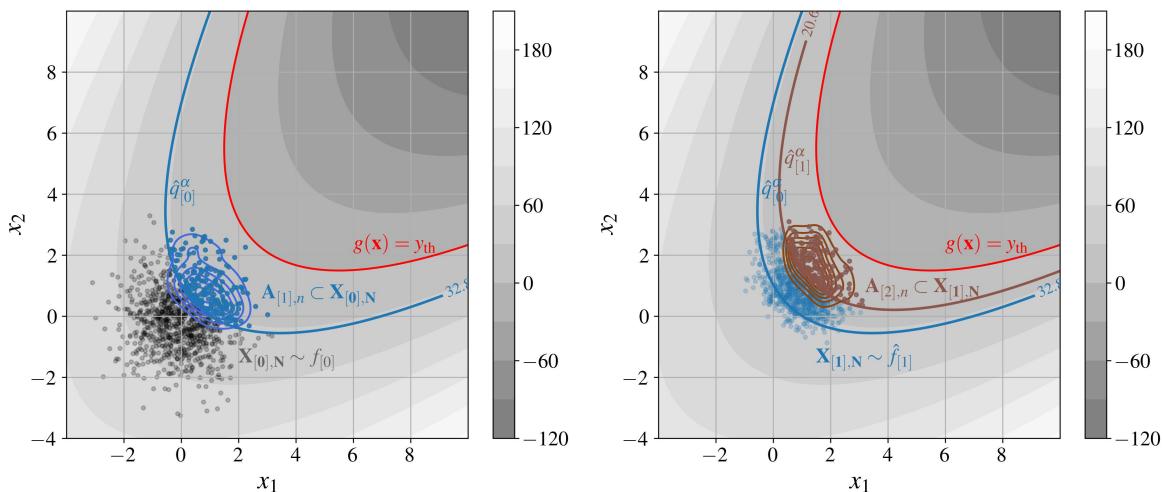


Fig. 6.2 BANCS on toy-case #1: illustration of conditional sampling and nonparametric fit at the first and second iterations.

Algorithm 1 Bernstein adaptive nonparametric conditional sampling (BANCS).

» Inputs:

f_X , joint PDF of the inputs
 $g(\cdot)$, limit-state function
 $y_{\text{th}} \in \mathbb{R}$, threshold defining the failure event
 N , number of samples per iteration
 $m \in \mathbb{N}$, parameter of the EBC fitting
 $p_0 \in]0, 1[$, empirical quantile order (rarity parameter)

» Algorithm:

```

Set  $k = 0$  and  $f_{[0]} = f_X$ 
Sample  $\mathbf{X}_{[0], N} = \{\mathbf{X}_{[0]}^{(j)}\}_{j=1}^N \stackrel{\text{i.i.d.}}{\sim} f_{[0]}$ 
Evaluate  $G_{[0], N} = \{g(\mathbf{X}_{[0]}^{(j)})\}_{j=1}^N$ 
Estimate the empirical  $p_0$ -quantile  $\hat{q}_{[0]}^{p_0}$  of the set  $G_{[0], N}$ 
while  $\hat{q}_{[k]}^{p_0} > y_{\text{th}}$  do
    Subsample  $\mathbf{A}_{[k+1], n} = \{\mathbf{X}_{[k]}^{(j)} \subset \mathbf{X}_{[k], N} | g(\mathbf{X}_{[k]}^{(j)}) > \hat{q}_{[k]}^{p_0}\}_{j=1}^n$ 
    Fit marginals of the subset  $\mathbf{A}_{[k+1], n}$  by KDE  $\{\hat{f}_i\}_{i=1}^d$ 
    Fit the copula of the subset  $\mathbf{A}_{[k+1], n}$  by EBC  $B_m(C_n)$ 
    Build a CDF  $\hat{F}_{[k+1]}(\mathbf{x}) = B_m(C_n)(\hat{f}_1(x_1), \dots, \hat{f}_d(x_d))$ 
    Sample  $\mathbf{X}_{[k+1], N} = \{\mathbf{X}_{[k+1]}^{(j)}\}_{j=1}^N \stackrel{\text{i.i.d.}}{\sim} \hat{f}_{[k+1]}$ 
    Evaluate  $G_{[k+1], N} = \{g(\mathbf{X}_{[k+1]}^{(j)})\}_{j=1}^N$ 
    Estimate the empirical  $p_0$ -quantile  $\hat{q}_{[k+1]}^{p_0}$  of  $G_{[k+1], N}$ 
    Set  $k = k + 1$ 
end while
Set total iteration number  $k_{\#} = k - 1$ 
Estimate  $\hat{p}_f = (1 - p_0)^{k_{\#}} \cdot \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{\{g(\mathbf{X}_{[k_{\#}]}^{(j)}) \geq y_{\text{th}}\}} (\mathbf{X}_{[k_{\#}]}^{(j)})$ 
 $\hat{p}_f$ , estimate of  $p_f$ 
```

As discussed in the previous section, EBC fitting is tuned by the Bernstein polynomial of order m , implying a bias-variance trade off. In Fig. 6.2, conditional distributions fitted by EBC (blue and brown isolines) seem to present a slight bias since they overlay the quantiles. However, reducing this bias implies decreasing the tuning parameter m , until $m = 1$, which is equivalent to an independent copula. Tools to control the goodness of fit of nonparametric conditional distributions are also available. As an example, let us consider the fitted conditional distribution at the first iteration (visible in Fig. 6.2). Its quantile-quantile plot in Fig. 6.3 shows a good fit of the two marginals by KDE. Then, the goodness of fit of copulas can be evaluated by Kendall's plot, represented in Fig. 6.4. This fit is also good, even if a slight bias is again visible.

6.3 Numerical experiments

In the following analytical numerical experiments, the intermediary probabilities were set to $p_0 = 0.1$, allowing a fair comparison with subset sampling. Then, the subset sample size is set to $N = 10^4$, in order to get a reasonable sample size $n = Np_0 = 10^3$ to perform the nonparametric fitting. EBC tuning is setup to minimize the MISE in Eq. (B.3): $m = 1 + n^{\frac{2}{d+4}}$. In order to take into account the variability of the method's results, each experiment is repeated 100 times, allowing

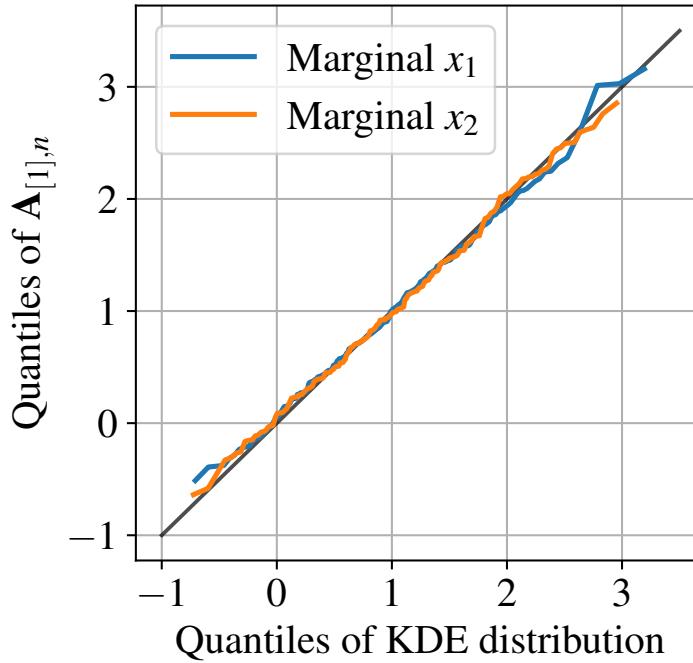


Fig. 6.3 QQ-plot for KDE of marginals of the conditional distribution from Fig. 6.2.

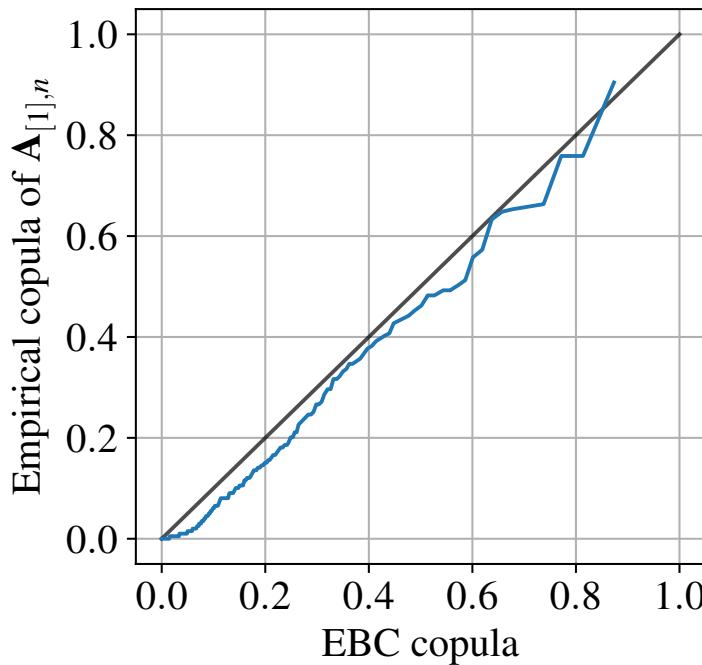


Fig. 6.4 Kendall plot for EBC on the copula of a conditional distribution from Fig. 6.2.

the computation of a coefficient of variation $\widehat{\delta} = \frac{\sigma_{\widehat{P}_f}}{\mu_{\widehat{P}_f}}$. Note that an implementation of the BANCS method and the following numerical experiments are available in a Git repository².

²<https://github.com/efekhari27/icasp14>

Toy-case #1: Parabolic reliability problem Let us define the parabolic reliability problem, considering the function $g_1 : \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$g_1(\mathbf{x}) = (x_1 - x_2)^2 - 8(x_1 + x_2 - 5), \quad (6.7)$$

with the input random vector $\mathbf{X} = (X_1, X_2)$ following a standard 2-dimensional normal distribution. The reliability problem consists in evaluating: $p_{f,1} = \mathbb{P}(g_1(\mathbf{X}) \leq 0) = 1.31 \times 10^{-4}$.

Toy-case #2: Four-branch reliability problem Let us define the four-branch reliability problem (originally proposed by [Waarts \(2000\)](#)), considering the following function $g_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$g_2(\mathbf{x}) = \min \left(\begin{array}{l} 5 + 0.1(x_1 - x_2)^2 - \frac{(x_1 + x_2)}{\sqrt{2}} \\ 5 + 0.1(x_1 - x_2)^2 + \frac{(x_1 + x_2)}{\sqrt{2}} \\ (x_1 - x_2) + \frac{9}{\sqrt{2}} \\ (x_2 - x_1) + \frac{9}{\sqrt{2}} \end{array} \right), \quad (6.8)$$

with the input random vector $\mathbf{X} = (X_1, X_2)$ following a standard 2-dimensional normal distribution. The reliability problem consists in evaluating: $p_{f,2} = \mathbb{P}(g_2(\mathbf{X}) \leq 0) = 2.21 \times 10^{-4}$.

Toy-case #3: high-dimensional reliability problem Let us define the higher-dimensional reliability problem (proposed by [Yun et al. \(2018\)](#)), considering the following function $g_3 : \mathbb{R}^7 \rightarrow \mathbb{R}$:

$$g_3(\mathbf{x}) = 15.59 \times 10^4 - \frac{x_1 x_3^2}{2x_3^2} \frac{x_2^4 - 4x_5 x_6 x_7^2 + x_4(x_6 + 4x_5 + 2x_6 x_7)}{x_4 x_5(x_4 + x_6 + 2x_6 x_7)}, \quad (6.9)$$

with the input random vector $\mathbf{X} = (X_1, \dots, X_7)$, following a product of normal distributions defined in [Yun et al. \(2018\)](#). The reliability problem consists in evaluating: $p_{f,3} = \mathbb{P}(g_3(\mathbf{X}) \leq 0) = 8.10 \times 10^{-3}$.

6.3.1 Results analysis

Results of our numerical experiments are presented graphically (for 2-dimensional problems) in Figures [6.5](#) and [6.6](#), and numerically in Table [6.1](#). In the same fashion as the previous illustrations, the figures represent the intermediary quantiles $\widehat{q}_{[k]}^{p_0}$ estimated over conditional samples of size $N = 10^4$. Moreover, samples $\mathbf{A}_{[k+1],n}$ exceeding these quantiles are also represented in the same color. Notice how the last estimated quantile is set to the problem threshold $y_{th} = 0$. To capture the dispersion of BANCS estimation, 100 repetitions were realized. Let us notice that for each toy-case, BANCS well estimates the failure probabilities' orders of magnitude. Yet the numerical values in Table [6.1](#) consistently present a positive bias, leading to an overestimated failure probability. This bias is partially explained by the EBC tuning choice and could be reduced at the expense of a slightly higher variance.

The variance obtained with the repetitions is quite large. Although, part of it is due to the fact that the algorithm might compute a different total number of subsets (e.g., toy-case #1 is

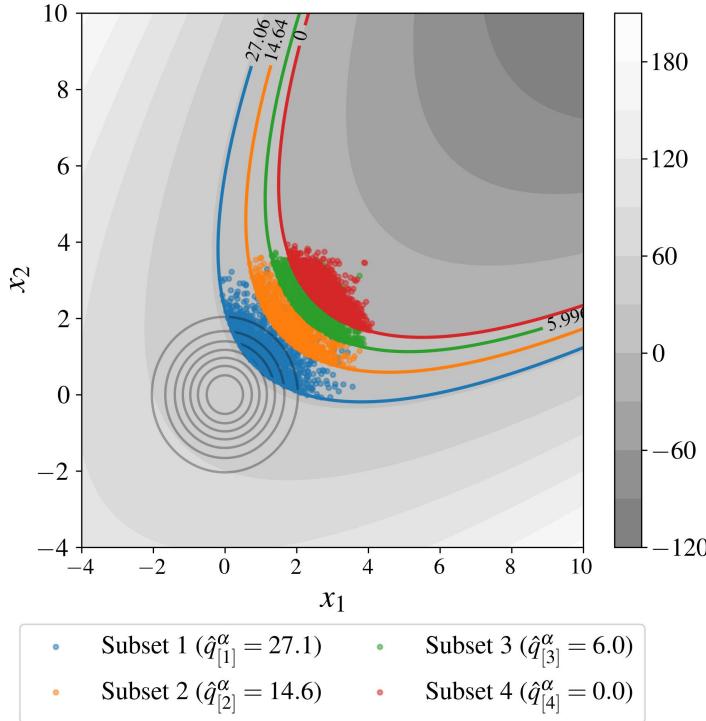


Fig. 6.5 BANCS sampling steps on toy-case #1.

either solved in four or five subsets). Overall, considering the EBC tuning from Eq. (B.3), BANCS performs worst than SS on toy-cases #1 and #2 but performs as well as SS on the toy-case #3. This might be due to the fact that toy-case #3 has a higher input dimension. However, one can note that SS coefficient of variation is computed by an approximation, tending to underestimate the true coefficient of variation (see e.g., Papaioannou et al. (2015)).

Table 6.1 Results of the numerical experiments (subset sample size $N = 10^4$, $p_0 = 0.1$).

	d	p_f^{ref}	\hat{p}_f^{BANCS}	$\hat{\delta}^{\text{BANCS}}$	\hat{p}_f^{SS}	$\hat{\delta}^{\text{SS}}$
Toy-case #1	2	1.31×10^{-4}	2.67×10^{-4}	24%	1.30×10^{-4}	9%
Toy-case #2	2	2.21×10^{-4}	4.23×10^{-4}	7%	2.24×10^{-4}	6%
Toy-case #3	7	8.10×10^{-3}	9.32×10^{-3}	15%	8.92×10^{-3}	6%

6.4 Application to wind turbine fatigue reliability

6.5 Conclusion

Subset Simulation uses MCMC sampling to generate its intermediary conditional samples. However, MCMC algorithms tends to be complex to tune and does not generate i.i.d. conditional samples. In this work, a new method is proposed, replacing MCMC sampling with a simpler procedure. An intermediary conditional distribution is first fitted by a nonparametric approach, mixing kernel density estimation for fitting the marginals and Empirical Bernstein Copula (EBC)

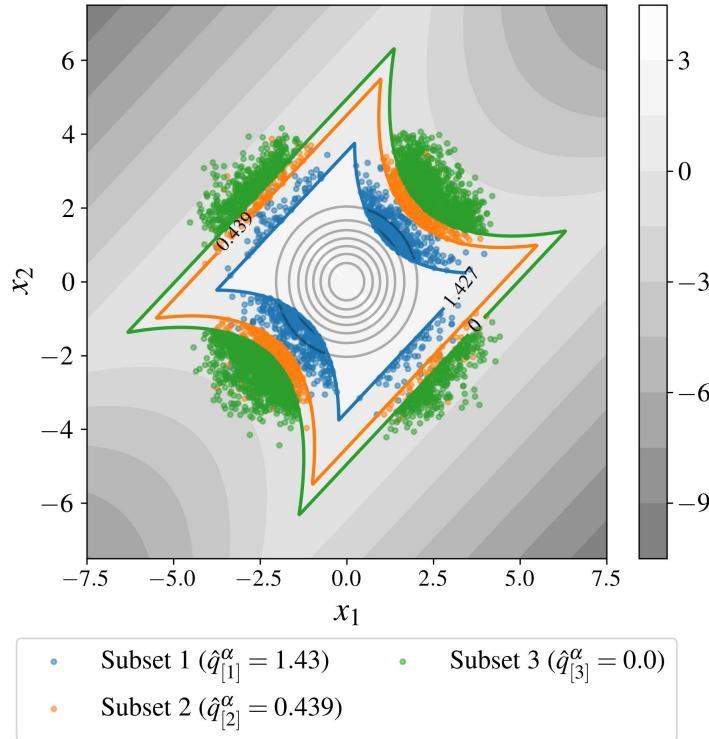


Fig. 6.6 BANCS sampling steps on toy-case #2.

for fitting the copula. Then, the resulting allows to perform direct Monte Carlo sampling. This method is named “Bernstein adaptive nonparametric conditional sampling” (BANCS) and is applied to three toy-cases (two 2-dimensional and one 7-dimensional) and compared with SS.

The method shows promising results, even though a small positive bias consistently appears. This issue results from EBC tuning, creating a bias-variance tradeoff in the copula fit. Theoretical works offer optimal tuning, allowing us to find the optimal compromise. In our numerical experiments, an empirical estimation of BANCS variance is computed over a set of repetitions. BANCS estimated coefficient of variation is higher than SS approximated coefficient of variation. This work can be further explored by building an approximation of BANCS variance and confidence interval. One major advantage remains that the samples generated at each iteration are i.i.d. leading to a possible use of these samples to perform global reliability-oriented sensitivity analysis (Marrel and Chabridon, 2021) in order to detect and analyze the most influential input variables leading to failure.

Chapter **7**

Sequential reliability oriented sensitivity analysis

7.1	Conclusion	98
7.2	HSIC for GSA	98
7.3	HSIC for TSA & CSA	98
7.4	Sequential ROSA	98
7.5	Application to wind turbine fatigue reliability	98
7.6	Conclusion	98

7.1 Conclusion

7.2 HSIC for GSA

7.3 HSIC for TSA & CSA

7.4 Sequential ROSA

7.5 Application to wind turbine fatigue reliability

7.6 Conclusion

Conclusion and perspectives

References

- Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical kriging for multi-fidelity aero-servo-elastic simulators – Application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67 – 77.
- Abtini, M. (2018). *Plans prédictifs à taille fixe et séquentiels pour le krigeage*. PhD thesis, Ecole Centrale Lyon.
- Ajenjo, A. (2023). *Info-gap robustness assessment of reliability evaluations for the safety of critical industrial systems*. PhD thesis, Université Bourgogne Franche-Comté.
- Au, S.-K. and Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277.
- Auder, B., De Crecy, A., Iooss, B., and Marques, M. (2012). Screening and metamodeling of computer experiments with functional outputs. Application to thermal–hydraulic computations. *Reliability Engineering & System Safety*, 107:122–131.
- Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69.
- Bai, H., Shi, L., Aoues, Y., Huang, C., and Lemosse, D. (2023). Estimation of probability distribution of long-term fatigue damage on wind turbine tower using residual neural network. *Mechanical Systems and Signal Processing*, 190:110101.
- Baudin, M., Dutfoy, A., Iooss, B., and Popelin, A. (2017). Open TURNS: An industrial software for uncertainty quantification in simulation. In Ghanem, R., Higdon, D., and Owhadi, H., editors, *Springer Handbook on Uncertainty Quantification*, pages 2001–2038. Springer.
- Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230(6):2345–2367.
- Bourinet, J. (2018). *Reliability analysis and optimal design under uncertainty-Focus on adaptive surrogate-based approaches*. PhD thesis, Université Clermont Auvergne.
- Briol, F., Oates, C., Girolami, M., Osborne, M., and Sejdinovic, D. (2019). Probabilistic Integration: A Role in Statistical Computation? *Statistical Science*, 34:1 – 22.
- Briol, F.-X. (2019). *Statistical computation with kernels*. PhD thesis, University of Warwick.
- Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. (2015). Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees. In *Advances in Neural Information Processing Systems*.

- Cérou, F., Del Moral, P., Furon, T., and Guyader, A. (2012). Sequential monte carlo for rare event estimation. *Statistics and computing*, 22:795–808.
- Chabridon, V., Balesdent, M., Perrin, G., Morio, J., Bourinet, J.-M., and Gayton, N. (2021). Global reliability-oriented sensitivity analysis under distribution parameter uncertainty. *Mechanical Engineering under Uncertainties: From Classical Approaches to Some Recent Developments*, pages 237–277.
- Chen, T., Wang, X., Yuan, G., and Liu, J. (2018). Fatigue bending test on grouted connections for monopile offshore wind turbines. *Marine Structures*, 60:52–71.
- Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 109 – 116. AUAI Press.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.
- Da Veiga, S. (2015). Global sensitivity analysis with dependence measures. *Journal of Statistical Computation and Simulation*, 85:1283 – 1305.
- Da Veiga, S., Gamboa, F., Iooss, B., and Prieur, C. (2021). *Basics and Trends in Sensitivity Analysis: Theory and Practice in R*. Society for Industrial and Applied Mathematics.
- Damblin, G., Couplet, M., and Iooss, B. (2013). Numerical studies of space-filling designs: Optimization of Latin Hypercube Samples and subprojection properties. *Journal of Simulation*, 7.
- de Rocquigny, E., Devictor, N., and Tarantola, S. (2008). *Uncertainty in industrial practice: a guide to quantitative uncertainty management*. John Wiley & Sons.
- Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112.
- Dimitrov, N., Kelly, M., Vignaroli, A., and Berg, J. (2018). From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. *Wind Energy Science*, 3:767 – 790.
- DNV-GL (2016a). DNVGL-RP-C203: Fatigue design of offshore steel structures. Technical report, DNVGL.
- DNV-GL (2016b). DNVGL-ST-0437: Loads and site conditions for wind turbines. Technical report, DNVGL.
- Dowling, N. E. (1972). Fatigue Failure Predictions for Complicated Stress-Strain Histories. *Journal of Materials, JMLSA*, 7:71 – 87.
- Fang, K., Liu, M.-Q., Qin, H., and Zhou, Y.-D. (2018). *Theory and application of uniform experimental designs*, volume 221. Springer.
- Fatemi, A. and Yang, L. (1998). Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials. *International Journal of Fatigue*, 20(1):9–34.
- Fekhari, E., Iooss, B., Muré, J., Pronzato, L., and Rendas, J. (2023). Model predictivity assessment: incremental test-set selection and accuracy evaluation. In Salvati, N., Perna, C., Marchetti, S., and Chambers, R., editors, *Studies in Theoretical and Applied Statistics*, pages 315–347. Springer.

- Fuhg., J., Fau, A., and Nackenhorst, U. (2021). State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging. *Archives of Computational Methods in Engineering*, 28:2689–2747.
- Giles, M. (2008). Multilevel Monte Carlo Path Simulation. *Operations Research*, 56:607–617.
- Gobet, E., Lerasle, M., and Métivier, D. (2022). Mean estimation for randomized quasi Monte Carlo method.
- Graf, P., Stewart, G., Lackner, M., Dykes, K., and Veers, P. (2016). High-throughput computation and the applicability of Monte Carlo integration in fatigue load estimation of floating offshore wind turbines. *Wind Energy*, 19(5):861–872.
- Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. (2006). A Kernel Method for the Two-Sample-Problem. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 513–520.
- Hansen, M. and Henriksen, L. (2013). *Basic DTU Wind Energy controller*. Number 0028 in DTU Wind Energy E. DTU Wind Energy.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Hawchar, L., El Soueid, C.-P., and Schoefs, F. (2017). Principal component analysis and polynomial chaos expansion for time-variant reliability problems. *Reliability Engineering & System Safety*, 167:406–416.
- Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322.
- Huchet, Q. (2019). *Kriging based methods for the structural damage assessment of offshore wind turbines*. PhD thesis, Université Blaise Pascal.
- Huchet, Q., Matstrand, C., Beaurepaire, P., Relun, N., and Gayton, N. (2019). AK-DA: An efficient method for the fatigue assessment of wind turbine structures. *Wind Energy*, 22(5):638–652.
- Huszár, F. and Duvenaud, D. (2012). Optimally-Weighted Herding is Bayesian Quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 377 – 386.
- IEC (2019). IEC 61400-1: Wind energy generation systems - Part 1: Design requirements. Technical report, International Electrotechnical Commission (IEC).
- Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Chapman and Hall.
- Joe, H. and Kurowicka, D. (2011). *Dependence modeling: vine copula handbook*. World Scientific.
- Jonkman, B. (2009). Turbsim User's Guide: Version 1.50. Technical report, NREL.
- Joseph, V., Gul, E., and Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, 102.
- Kaimal, J., Wyngaard, J., Izumi, Y., and Coté, O. (1972). Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 98(417):563–589.
- Kanagawa, M. and Hennig, P. (2019). Convergence Guarantees for Adaptive Bayesian Quadrature Methods. In *Advances in Neural Information Processing Systems*, volume 32.

- Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. (2018). Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. preprint.
- Kaplan, Z., Li, Y., Nakayama, M., and Tuffin, B. (2019). Randomized quasi-Monte Carlo for quantile estimation. In *2019 Winter Simulation Conference (WSC)*, pages 428–439.
- Kelly, M. and Vanem, E. (2022). Environmental joint probability distributions and uncertainties. Technical report, Tech Rep. of HIPERWIND H2020 project, Grant Agreement No. 101006689, <https://www.hiperwind.eu/>.
- Kim, T., Natarajan, A., Lovera, A., Julian, E., Peyrard, E., Capaldo, M., Huwart, G., Bozonnet, P., and Guiton, M. (2022). A comprehensive code-to-code comparison study with the modified IEA15MW-UMaine Floating Wind Turbine for H2020 HIPERWIND project. *Journal of Physics: Conference Series*, 2265(4):042006.
- Klebanov, I., Schuster, I., and Sullivan, T. (2020). A rigorous theory of conditional mean embeddings. *SIAM Journal on Mathematics of Data Science*, 2(3):583–606.
- Koehler, J. and Owen, A. (1996). 9 Computer experiments. In *Design and Analysis of Experiments*, volume 13 of *Handbook of Statistics*, pages 261–308. Elsevier.
- Kucherenko, S., Feil, B., Shah, N., and Mauntz, W. (2011). The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety*, 96:440–449.
- Kurtz, N. and Song, J. (2013). Cross-entropy-based adaptive importance sampling using Gaussian mixture. *Structural Safety*, 42:35–44.
- Lacoste-Julien, S., Lindsten, F., and Bach, F. (2015). Sequential Kernel Herding: Frank-Wolfe Optimization for Particle Filtering. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 544–552.
- Lasserre, M. (2022). *Apprentissages dans les réseaux bayésiens à base de copules non-paramétriques*. PhD thesis, Sorbonne Université.
- Lataniotis, C. (2019). *Data-driven uncertainty quantification for high-dimensional engineering problems*. PhD thesis, ETH Zürich.
- Laurie, D. (1997). Calculation of Gauss-Kronrod quadrature rules. *Mathematics of Computation*, 66(219):1133–1145.
- Le Maître, O. and Knio, O. (2010). *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Science & Business Media.
- Lebrun, R. (2013). *Contributions à la modélisation de la dépendance stochastique*. PhD thesis, Université Paris-Diderot – Paris VII. (in English).
- Lebrun, R. and Dutfoy, A. (2009). A generalization of the Nataf transformation to distributions with elliptical copula. *Probabilistic Engineering Mechanics*, 24(2):172–178.
- L’Ecuyer, P. (2018). Randomized Quasi-Monte Carlo: An Introduction for Practitioners. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 29–52, Cham. Springer International Publishing.
- Lemaire, M. (2013). *Structural reliability*. John Wiley & Sons.
- Leobacher, G. and Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo integration and applications*. Springer.

- Li, X. and Zhang, W. (2020). Long-term fatigue damage assessment for a floating offshore wind turbine under realistic environmental conditions. *Renewable Energy*, 159:570–584.
- Li, Y., Kang, L., and Hickernell, F. (2020). Is a transformed low discrepancy design also low discrepancy? *Contemporary Experimental Design, Multivariate Analysis and Data Mining: Festschrift in Honour of Professor Kai-Tai Fang*, pages 69–92.
- Mak, S. and Joseph, V. (2018). Support points. *The Annals of Statistics*, 46:2562 – 2592.
- Marrel, A. and Chabridon, V. (2021). Statistical developments for target and conditional sensitivity analysis: Application on safety studies for nuclear reactor. *Reliability Engineering & System Safety*, 214:107711.
- Marrel, A., Iooss, B., and Chabridon, V. (2022). The ICSCREAM methodology: Identification of penalizing configurations in computer experiments using screening and metamodel – Applications in thermal-hydraulics. *Nuclear Science and Engineering*, 196:301–321.
- Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30.
- Mckay, M., Beckman, R., and Conover, W. (1979). A Comparison of Three Methods for Selecting Vales of Input Variables in the Analysis of Output From a Computer Code. *Technometrics*, 21:239 – 245.
- Morio, J. (2011). Non-parametric adaptive importance sampling for the probability estimation of a launcher impact position. *Reliability Engineering and System Safety*, 96(1):178–183.
- Morio, J. and Balesdent, M. (2015). *Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems: A Practical Approach*. Woodhead Publishing, Elsevier.
- Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122(2):218–230.
- Moustapha, M., Marelli, S., and Sudret, B. (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety*, 96:102174.
- Murcia, J., Réthoré, P., Dimitrov, N., Natarajan, A., Sørensen, J., Graf, P., and Kim, T. (2018). Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. *Renewable Energy*, 119:910–922.
- Müller, K. and Cheng, P. (2018). Application of a Monte Carlo procedure for probabilistic fatigue design of floating offshore wind turbines. *Wind Energy Science*, 3:149 – 162.
- Nagler, T., Schellhase, C., and Czado, C. (2017). Nonparametric estimation of simplified vine copula models: comparison of methods. *Dependence Modeling*, 5:99–120.
- Oates, C. J. (2021). Minimum Discrepancy Methods in Uncertainty Quantification. Lecture Notes at École Thématische sur les Incertitudes en Calcul Scientifique (ETICS21), <https://www.gdr-mascotnum.fr/etics.html>.
- O'Hagan, A. (1991). Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260.
- Owen, A. (2003). The dimension distribution and quadrature test functions. *Statistica Sinica*, 13:1–17.
- Owen, A. (2013). *Monte Carlo theory, methods and examples*.

- Papaioannou, I., Betz, W., Zwirglmaier, K., and Straub, D. (2015). MCMC algorithms for Subset Simulation. *Probabilistic Engineering Mechanics*, 41:89–103.
- Petrovska, E. (2022). *Fatigue life reassessment of monopile-supported offshore wind turbine structures*. PhD thesis, University of Edinburgh.
- Pronzato, L. and Müller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701.
- Pronzato, L. and Rendas, M. (2021). Validation design I: construction of validation designs via kernel herding. preprint, <https://arxiv.org/abs/2112.05583>.
- Pronzato, L. and Zhigljavsky, A. (2020). Bayesian quadrature and energy minimization for space-filling design. *SIAM/ASA Journal on Uncertainty Quantification*, 8:959 – 1011.
- Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. Springer.
- Rockafellar, R. and Royston, J. (2015). Engineering Decisions under Risk Aversion. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 1.
- Rollón de Pinedo, A., Couplet, M., Iooss, B., Marie, N., Marrel, A., Merle, E., and Sueur, R. (2021). Functional outlier detection by means of h-mode depth and dynamic time warping. *Applied Sciences*, 11(23):11475.
- Rubinstein, R. Y. and Kroese, D. P. (2008). *Simulation and the Monte Carlo Method*. Wiley, Second ed. edition.
- Sancetta, A. and Satchell, S. (2004). The Bernstein copula and its applications to modeling and approximations of multivariate distributions. *Econometric Theory*, 20(3):535–562.
- Schobi, R., Sudret, B., and Wiart, J. (2015). Polynomial-chaos-based Kriging. *International Journal for Uncertainty Quantification*, 5(2).
- Segers, J., Sibuya, M., and Tsukahara, H. (2017). The empirical beta copula. *Journal of Multivariate Analysis*, 155:35–51.
- Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41:2263–2291.
- Slot, R. M., Sørensen, J. D., Sudret, B., Svenningsen, L., and Thøgersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150 – 1162.
- Soize, C. and Ghanem, R. (2004). Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26:395–410.
- Sriperumbudur, B., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. (2010). Hilbert Space Embeddings and Metrics on Probability Measures. *J. Mach. Learn. Res.*, 11:1517–1561.
- Stieng, L. and Muskulus, M. (2020). Reliability-based design optimization of offshore wind turbine support structures using analytical sensitivities and factorized uncertainty modeling. *Wind Energy Science*, 5:171–198.
- Sullivan, T. (2015). *Introduction to uncertainty quantification*, volume 63. Springer.
- Sun, F., Wang, Y., and Xu, H. (2019). Uniform projection designs. *The Annals of Statistics*, 47:641 – 661.

- Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143:1249 – 1272.
- Teixeira, R., O'Connor, A., Nogal, M., Nandakumar, K., and Nichols, J. (2017). Analysis of the design of experiments of offshore wind turbine fatigue reliability design with Kriging surfaces. *Procedia Structural Integrity*, 5:951 – 958.
- Trefethen, L. (2008). Is Gauss quadrature better than Clenshaw–Curtis? *SIAM review*, 50(1):67–87.
- Van den Bos, L. (2020). *Quadrature Methods for Wind Turbine Load Calculations*. PhD thesis, Delft University of Technology.
- Vanem, E., Fekhari, E., Dimitrov, N., Kelly, M., Cousin, A., and Guiton, M. (2023). A joint probability distribution model for multivariate wind and wave conditions. In *Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering*.
- Velarde, J., Kramhøft, C., and Sørensen, J. (2019). Global sensitivity analysis of offshore wind turbine foundation fatigue loads. *Renewable Energy*, 140:177 – 189.
- Waarts, P. (2000). *Structural reliability using finite element methods: an appraisal of directional adaptive response surface sampling (DARS)*. PhD thesis, Technical University of Delft, The Netherlands.
- Warnock, T. (1972). Computational investigations of low-discrepancy point sets. In *Applications of number theory to numerical analysis*, pages 319–343. Elsevier.
- Wilkie, D. and Galasso, C. (2021). Gaussian process regression for fatigue reliability analysis of offshore wind turbines. *Structural Safety*, 88:102020.
- Yun, W., Lu, Z., Zhang, Y., and Jiang, X. (2018). An efficient global reliability sensitivity analysis algorithm based on classification of model output and subset simulation. *Structural Safety*, 74:49–57.
- Zwick, D. and Muskulus, M. (2015). The simulation error caused by input loading variability in offshore wind turbine structural analysis. *Wind Energy*, 18:1421 – 1432.

Appendix **A**

Univariate distribution fitting

This appendix recalls the main methods to infer a univariate distribution considering a n -sized i.i.d sample $X_n = \{x^{(1)}, \dots, x^{(n)}\} \in \mathbb{R}^n$. The goal is to use this finite set of observations of the random variable X to approach its underlying distribution by an estimated distribution. The inference techniques are split into two main groups, the methods assuming that the underlying distribution belongs to a family of parametric distributions are called parametric. Otherwise, the fitting method falls into the nonparametric group. Nonparametric methods often require a larger amount of data but allow more flexibility. In fact, nontrivial distributions (e.g., multimodal) might be easier to model using nonparametric approaches. To assess the quality of this estimation regarding the sample, a panel of goodness-of-fit methods are proposed [add ref], this appendix recalls a few of them. Note that the following tools can be used to estimate the marginals of a multivariate distribution.

A.1 Main parametric methods

Moments method

The moments method aims at looking for a parametric distribution with density $f_X(\theta)$, whose first moments (e.g., $m(\theta)$ and $\sigma^2(\theta)$) match the empirical moments of the sample X_n (e.g., \widehat{m}_{X_n} and $\widehat{\sigma}^2$). After computing the empirical moments:

$$\widehat{m}_n = \frac{1}{n} \sum_{i=1}^n x^{(i)}, \quad \widehat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^n (x^{(i)} - \widehat{m}_{X_n})^2, \quad (\text{A.1})$$

one can solve the system of equations ($m(\theta) = \widehat{m}_n$; $\sigma^2(\theta) = \widehat{\sigma}_n^2$) to determine the optimal set of parameters θ in this situation. Some families of distributions are more suited to this method (i.e., \mathcal{N}) because of the analytical expression of their moments. Moreover, this technique is sensitive to the possible biases in the estimation of the sample moments.

Maximum likelihood estimation

Maximum likelihood estimation (MLE) is a popular alternative to the moments method. Similarly, it aims at maximizing a given correspondence metric between the dataset X_n and a parametric distribution with density $f_X(\theta)$. This metric is the *likelihood* function, defined as:

$$\mathcal{L}(\theta|X_n) = \prod_{i=1}^n f_X(x^{(i)}; \theta), \quad (\text{A.2})$$

with the PDF taking the set of parameters θ written: $f_X(x^{(i)}; \theta)$. For numerical reasons, the optimization is often performed on the natural logarithm of the likelihood function, called *log-likelihood*. The goal is then finding the optimal vector $\hat{\theta}^*$ of parameters minimizing the following expression:

$$\hat{\theta}^* = \arg \min_{\theta \in \mathcal{D}_\theta} \left(- \sum_{i=1}^n \ln(f_X(x^{(i)}; \theta)) \right). \quad (\text{A.3})$$

Remark that the quick analytical results from the moment method can be used as a starting point of the MLE optimization. [Asymptotic behaviors of this method are described in: add ref] [This method can be applied to censored data in the field of survival analysis. Add ref]

Example 1. Considering a small set of observations $X_n = \{1, 2, 3, 4, 6\}$, the following figure xx represents

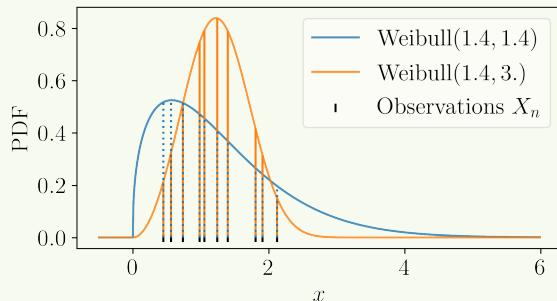


Fig. A.1 Adequation of two different Weibull models using their likelihood with a sample of observations (black crosses).

A.2 Main nonparametric methods

Empirical CDF and histogram

The empirical CDF is a cumulative stair-shaped representation of the sorted sample X_n :

$$\widehat{F}_X(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{x \geq x^{(i)}\}}. \quad (\text{A.4})$$

A histogram consists in sorting and gathering the observations in a sample X_n into a finite number of categories. These categories are called bins and each regroups the same number of observations (identical binwidth). The number of bins is the only tuning parameter of this method. Its definition has a great impact on the visual consistency of the plot, therefore, many rules exist to define it. Note that the empirical CDF can be seen as a cumulative histogram with the number of bins equal to the number of observations.

Kernel density estimation

Kernel density estimation (KDE) is a nonparametric method, it estimates a PDF by weighing a sample of observations X_n with kernels. After setting a kernel $k : \mathbb{R} \rightarrow \mathbb{R}_+$ and a scaling parameter $h > 0$, also called bandwidth, the kernel density estimator is defined as:

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=1}^n k\left(\frac{x-x^{(i)}}{h}\right) \quad (\text{A.5})$$

Different types of kernels are used for KDE, such as the uniform, triangular, squared exponential or Epachnikov. The choice of bandwidth results in a bias-variance trade-off, that has been extensively discussed in the literature [add ref].

Example 2. Considering a small set of observations $X_n = \{1, 2, 3, 4, 6\}$, the following figure xx represents three fits obtained by.

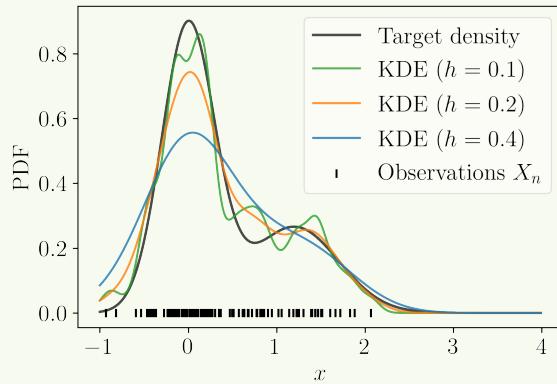


Fig. A.2 Fit of a bimodal density by KDE using different tuning parameters.

Main goodness-of-fit methods

Penalized likelihood criteria

Two quantitative goodness-of-fit criteria are commonly used to assess parametric inference: the *Akaike information criterion* (AIC) and the *Bayesian information criterion* (BIC). The likelihood as a goodness-of-fit criterion should only be applied to the same family of distributions. Otherwise, the comparison would unfairly advantage distributions with a large number of degrees of

freedom. The two following criteria are metrics based on the likelihood with a correction related to the number of degrees of freedom of the distribution. Moreover, let us remind that more flexible models will require more data to provide a robust estimation.

The AIC and BIC are expressed as follows:

$$\text{AIC} = \frac{-2\ln(\mathcal{L}(\theta|X_n))}{n} + \frac{2q}{n}, \quad \text{BIC} = \frac{-2\ln(\mathcal{L}(\theta|X_n))}{n} + \frac{q\ln(n)}{n}, \quad (\text{A.6})$$

with the likelihood $\mathcal{L}(\theta|X_n)$ and the number of distribution's number degrees of freedom denoted q . The second term adds a penalty depending on the number of parameters. The best inference will be given by the model with the smallest AIC or BIC. Note that an additional correction can be applied in a small data context.

Kolmogorov-Smirnov adequacy test

Quantile-quantile plot

The quantile-quantile plot (also called QQ-plot) is a graphical tool providing a qualitative check of the goodness of fit. It compares the CDF of the fitted model with the empirical CDF of the sample X_n . To do so, it represents a scatterplot of the empirical quantiles (i.e., the ranked observations), against the quantiles of the fitted model at the levels $\{\alpha^{(i)}\}_{i=1}^n = \{\widehat{F}_X(x^{(i)})\}_{i=1}^n$. The following [figure xx] is a QQ-plot of the model fitted in [Example xx]. The closer the scatter plot gets to the first bisector line the best the fit is.

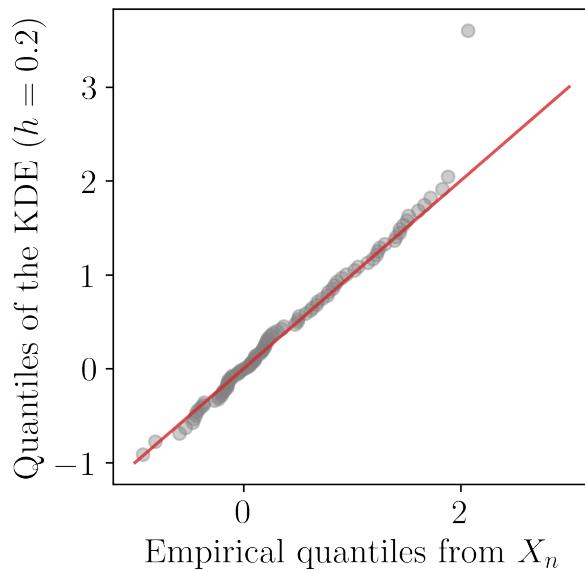


Fig. A.3 QQ-plot between the data from Example 2 and a KDE model.

Appendix B

Nonparametric copula estimation

[update CDF notations]

[Change EBC notations using h for the polynomial orders]

When the distribution's dimension is higher than two, one can perform a parametric fit using vine copulas (Joe and Kurowicka, 2011), implying the choice of multiple types of parametric copulas. Otherwise, nonparametric fit by multivariate kernel density estimation (KDE) presents a computational burden as soon as the dimension increases (Chabridon et al., 2021). Since univariate marginals are usually well-fitted with nonparametric tools (e.g., KDE), let us introduce an effective nonparametric method for copula fitting.

B.1 Empirical copula

In practice, considering a sample $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{np}$ and the associated ranked sample $\mathbf{R}_n = \{\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}\}$, the corresponding empirical copula writes:

$$C_n(\mathbf{u}) := \frac{1}{n} \sum_{i=0}^n \prod_{j=1}^p \mathbb{1} \left\{ \frac{r_j^{(i)}}{n} \leq u_j \right\}, \quad (\text{B.1})$$

with $\mathbf{u} = (u_1, \dots, u_d) \in [0, 1]^d$. In the following, the polynomial order is set as equal in each dimension: $\{m_i = m\}_{j=1}^d$.

B.2 Empirical Bernstein & Beta copula

Copulas are continuous and bounded functions defined on a compact set (the unit hypercube). Bernstein polynomials allow us to uniformly approximate as closely as desired any continuous and real-valued function defined on a compact set (Weierstrass approximation theorem). Therefore, they are good candidates to approximate unknown copulas. This concept was introduced as *empirical Bernstein copula* (EBC) by Sancetta and Satchell (2004) for applications in economics and risk management. Later on, Segers et al. (2017) offered further asymptotic studies. Formally,

the multivariate Bernstein polynomial for a function $C : [0, 1]^d \rightarrow \mathbb{R}$ on a grid over the unit hypercube $G := \left\{ \frac{0}{m_1}, \dots, \frac{m_1}{m_1} \right\} \times \dots \times \left\{ \frac{0}{m_d}, \dots, \frac{m_d}{m_d} \right\}$, $\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}^d$, writes:

$$B_{\mathbf{m}}(C)(\mathbf{u}) := \sum_{t_1=0}^{m_1} \dots \sum_{t_d=0}^{m_d} C\left(\frac{t_1}{m_1}, \dots, \frac{t_d}{m_d}\right) \prod_{j=1}^d P_{m_j, t_j}(u_j), \quad (\text{B.2})$$

with $\mathbf{u} = (u_1, \dots, u_d) \in [0, 1]^d$, and the Bernstein polynomial $P_{m,t}(u) := \frac{t!}{m!(t-m)!} u^m (1-u)^{t-m}$. Notice how the grid definition implies the polynomial's order. When C is a copula, then $B_{\mathbf{m}}(C)$ is called “Bernstein copula”. Therefore, the empirical Bernstein copula is an application of the Bernstein polynomial in Eq. (B.2) to the so-called “empirical copula”. [add a sentence to mean to refer to the previous subsection]

Theoretically, the tuning parameter can be optimized to minimize an “Mean Integrated Squared Error” (MISE), leading to a bias-variance tradeoff. Formally, the MISE of the empirical Bernstein copula $B_{\mathbf{m}}(C_n)$ is defined as follows:

$$\mathbb{E}\left[\|B_{\mathbf{m}}(C_n) - C\|_2^2\right] = \mathbb{E}\left[\int_{\mathbb{R}^d} (B_{\mathbf{m}}(C_n)(\mathbf{u}) - C(\mathbf{u})) d\mathbf{u}\right]^2. \quad (\text{B.3})$$

Then, [Sancetta and Satchell \(2004\)](#) prove in their Theorem 3 that:

- $B_{\mathbf{m}}(C_n)(\mathbf{u}) \rightarrow C(\mathbf{u})$ for any $u_j \in]0, 1[$ if $\frac{m^{d/2}}{n} \rightarrow 0$, when $m, n \rightarrow \infty$.
- The optimal order of the polynomial in terms of MISE is: $m \lesssim m_{\text{IMSE}} = n^{2/(d+4)}$, $\forall u_j \in]0, 1[$. The sign \lesssim means “less than or approximately”.

Let us remark that in the special case $m = n$, also called the “Beta copula” in [Segers et al. \(2017\)](#), the bias is very small while the variance gets large. To illustrate the previous theorem, [Lasserre \(2022\)](#) represents the evolution of the m_{IMSE} for different dimensions and sample sizes (see Fig. B.1). In high dimension, the values of m_{IMSE} tend towards one, which is equivalent to the independent copula. Therefore, high-dimensional problems should be divided into a product of smaller problems on which the EBC is tractable. Provided a large enough learning set \mathbf{X}_n , KDE fitting of marginals combined with EBC fitting of the copula delivers good results even on complex dependence structures. Moreover, EBC provides an explicit expression, making a Monte Carlo generation of i.i.d. samples simple.

B.3 Goodness-of-fit

[Mention the vine copulas and how we want to only use nonparametric methods here.]

[Tails correlation / Kendall plot]

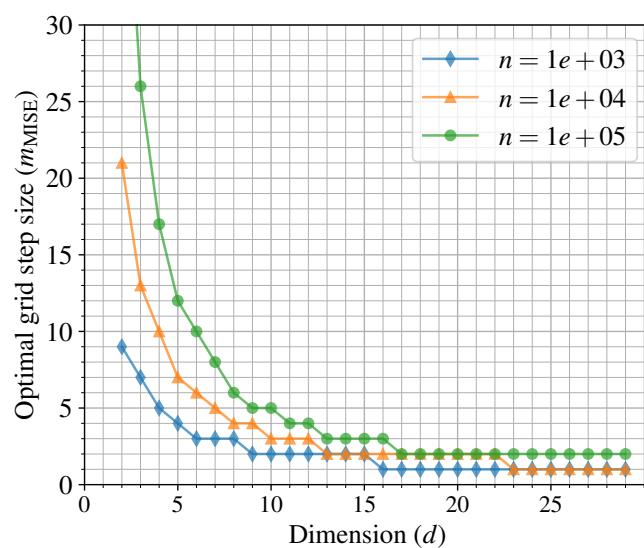


Fig. B.1 Evolution of m_{IMSE} for different dimensions and sample sizes.

Appendix

C

Advanced rare event estimation algorithms

C.1 Subset simulation (SS)

C.2 Nonparametric adaptive importance sampling (NAIS)

C.3 Parametric adaptive importance sampling using cross-entropy optimization (AIS-CE)

Appendix **D**

Résumé étendu de la thèse

Appendix E

Uncertainty quantification implemented with OpenTURNS

[Add short introduction to the motivation]

OpenTURNS 6 (Bivariate distribution). The following Python code proposes a minimalist OpenTURNS implementation of a probabilistic uncertainty modeling.

```
1 import openturns as ot
2 # Build multivariate distribution from marginals and copula
3 copula=ot.GumbelCopula(2.0)
4 marginals=[ot.Uniform(1.0, 2.0), ot.Normal(2.0, 3.0)]
5 distribution=ot.ComposedDistribution(marginals, copula)
6 # Compute first moments
7 mean_vector=distribution.getMean()
8 covariance_matrix=distribution.getCovariance()
9 # Compute CDF (respectively PDF)
10 x_cdf=distribution.computeCDF([1.5, 2.5]) # x=[1.5, 2.5]
11 a_quantile=distribution.computeQuantile([0.9]) # alpha=0.9
```

OpenTURNS 7 (Numerical integration). The following Python code proposes a minimalist OpenTURNS implementation to build a quadrature rule.

```

1 import openturns as ot
2 marginals=[ot.Exponential(1.0), ot.Uniform(-1.0, 1.0)]
3 distribution=ot.ComposedDistribution(marginals)
4 # Build a 2D Gaussian quadrature
5 n_marginal=[4, 4] # Number of nodes per marginal
6 g_quad=ot.GaussProductExperiment(distribution, n_marginal)
7 g_nodes, weights=g_quad.generateWithWeights()
8 # Build a Monte Carlo design
9 n=16
10 mc_nodes=distribution.getSample(n)
11 # Build a quasi-Monte Carlo design
12 sequence=ot.HaltonSequence(2) # d=2
13 qmc_experiment=ot.LowDiscrepancyExperiment(sequence,
14 distribution, n)
14 qmc_nodes=qmc_experiment.generate()
```

OpenTURNS 8 (Design of experiments). The following Python code proposes a minimalist OpenTURNS implementation to build a LHS and a LHS optimized w.r.t. to a space-filling metric (here the L2-centered discrepancy) using the simulated annealing algorithm.

```

1 import openturns as ot
2 marginals=[ot.Uniform(0.0, 1.0), ot.Uniform(0.0, 1.0)]
3 distribution=ot.ComposedDistribution(marginals)
4 # Build a LHS
5 n=10
6 LHS_exp=ot.LHSExperiment(distribution, n)
7 LHS_design=LHS_exp.generate()
8 # Build an optimized LHS using L2-centered discrepancy
9 LHS_exp=ot.LHSExperiment(distribution, n)
10 SF_metric=ot.SpaceFillingC2()
11 SA_profile=ot.GeometricProfile(10., 0.95, 20000)
12 LHS_opt=ot.SimulatedAnnealingLHS(LHS_exp, SF_metric,
13 SA_profile)
13 LHS_opt.generate()
14 LHS_design=LHS_opt.getResult().getOptimalDesign()
```

OpenTURNS 9 (Rare event estimation). The following Python code proposes a minimalist OpenTURNS implementation of rare event estimation algorithms.

```

1 import openturns as ot
2 marginals=[ot.Normal(0.0, 1.0), ot.Exponential(1.0)]
3 distribution=ot.ComposedDistribution(marginals)
4 # Build a limit-state function and failure event
5 g=ot.SymbolicFunction(["x1", "x2"], ["(x1 - x2) ^ 2"])
6 X=ot.RandomVector(distribution)
7 Y=ot.CompositeRandomVector(g, X)
8 th=0.0
9 failure_event=ot.ThresholdEvent(Y, ot.LessOrEqual(), th)
10 # Estimate pf using FORM
11 starting_p=distribution.getMean()
12 FORM_algo=ot.FORM(ot.Cobyla(), failure_event, starting_p)
13 FORM_algo.run()
14 FORM_results=FORM_algo.getResult()
15 design_point=FORM_results.getStandardSpaceDesignPoint()
16 FORM_pf=FORM_results.getEventProbability()
17 # Estimate pf using Monte Carlo
18 MC_exp=ot.MonteCarloExperiment()
19 MC_algo=ot.ProbabilitySimulationAlgorithm(failure_event,
20                                              MC_exp)
21 MC_algo.run()
22 MC_results=MC_algo.getResult()
23 MC_pf=MC_results.getProbabilityEstimate()
24 MC_pf_confidence=MC_results.getConfidenceLength(0.95)
25 # Estimate pf using importance sampling
26 aux_distribution=ot.Normal(design_point, [1.0, 1.0])
27 standard_event=ot.StandardEvent(failure_event)
28 IS_exp=ot.ImportanceSamplingExperiment(aux_distribution)
29 IS_algo=ot.ProbabilitySimulationAlgorithm(standard_event,
30                                             IS_exp)
31 IS_algo.run()
32 IS_results=IS_algo.getResult()
33 IS_pf=IS_results.getProbabilityEstimate()
34 IS_pf_confidence=IS_results.getConfidenceLength(0.95)
35 # Estimate pf using subset sampling
36 SS_algo=ot.SubsetSampling(failure_event)
37 SS_algo.run()
38 SS_results=SS_algo.getResult()
39 SS_pf=SS_results.getProbabilityEstimate()
40 SS_pf_confidence=SS_results.getConfidenceLength(0.95)

```

OpenTURNS 10 (Ordinary kriging). The following Python code proposes a minimalistic OpenTURNS implementation of an ordinary kriging model fitting.

```
1 g=ot.SymbolicFunction(['x'], ['x * sin(x) + sin(6 * x)'])
2 x_train=ot.Uniform(0., 12.).getSample(7) # n=7
3 y_train=g(x_train)
4 basis=ot.ConstantBasisFactory(1).build() # d=1
5 cov_model=ot.MaternModel([1.], 1.5)
6 algo=ot.KrigingAlgorithm(x_train, y_train, cov_model, basis)
7 algo.run()
8 kriging_results=algo.getResult()
9 kriging_predictor=kriging_results.getMetaModel()
```