

---

---

# 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*

A thesis submitted in partial fulfilment of the requirements for the degree of

*Doctor of Philosophy*  
(in Computer Science)

publicly defended on March 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



# Contents

<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>I Introduction to uncertainty quantification and wind energy</b>	<b>7</b>
<b>1 Uncertainty quantification in computer experiments</b>	<b>9</b>
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 statement . . . . .	27
1.5.2 Rare event estimation methods . . . . .	29
1.5.3 Summary and discussion . . . . .	36
1.6 Global sensitivity analysis . . . . .	36
1.6.1 Screening methods . . . . .	36
1.6.2 Variance-based importance measures . . . . .	38
1.6.3 Moment-independent importance measures . . . . .	41
1.6.4 Summary and discussion . . . . .	42
1.7 Surrogate modeling . . . . .	42
1.7.1 Common framework . . . . .	42

1.7.2	General purposes surrogate model . . . . .	44
1.7.3	Goal-oriented active surrogate model . . . . .	46
1.7.4	Summary and discussion . . . . .	48
1.8	Conclusion . . . . .	48
<b>2</b>	<b>Introduction to wind turbine modeling and design</b>	<b>49</b>
2.1	Introduction . . . . .	50
2.2	Metocean conditions simulation . . . . .	50
2.2.1	Turbulent wind generation . . . . .	51
2.2.2	Wake modeling . . . . .	54
2.2.3	Irregular wave generation . . . . .	55
2.3	Wind turbine multi-physics modeling . . . . .	56
2.3.1	Aerodynamics of horizontal axis wind turbines . . . . .	56
2.3.2	Hydrodynamics . . . . .	57
2.3.3	Command and control . . . . .	57
2.3.4	Structural dynamics . . . . .	57
2.3.5	Fatigue damage . . . . .	57
2.4	Design and operation practices . . . . .	57
2.4.1	Wind turbine design and operation . . . . .	57
2.4.2	Further design considerations . . . . .	57
2.4.3	Maintenance and end-of-life management . . . . .	58
2.5	Uncertain inputs . . . . .	58
2.5.1	Environmental inputs . . . . .	58
2.5.2	System inputs . . . . .	58
2.5.3	Probabilistic fatigue assessment . . . . .	58
2.6	Conclusion . . . . .	58
<b>Bibliography</b>		<b>59</b>
<b>Appendix A</b>	<b>Univariate distribution fitting</b>	<b>67</b>
A.1	Main parametric methods . . . . .	67
A.2	Main nonparametric methods . . . . .	68
<b>Appendix B</b>	<b>Nonparametric copula estimation</b>	<b>71</b>
B.1	Empirical copula . . . . .	71
B.2	Empirical Bernstein & Beta copula . . . . .	71
B.3	Goodness-of-fit . . . . .	72
<b>Appendix C</b>	<b>Copulogram</b>	<b>75</b>

<b>Appendix D Dissimilarity measures between probability distributions</b>	<b>77</b>
D.1 Csizár $f$ -divergences . . . . .	77
D.2 Integral probability metrics . . . . .	77
<b>Appendix E Advanced rare event estimation algorithms</b>	<b>79</b>
E.1 Subset simulation (SS) . . . . .	79
E.2 Nonparametric adaptive importance sampling (NAIS) . . . . .	79
E.3 Parametric adaptive importance sampling using cross-entropy optimization (AIS-CE) . . . . .	79
<b>Appendix F Uncertainty quantification practice with OpenTURNS</b>	<b>81</b>
<b>Appendix G Résumé étendu de la thèse</b>	<b>85</b>
G.1 Introduction . . . . .	85
G.2 Résumés des chapitres relatifs à l'état de l'art des méthodes et outils mis en œuvre dans la thèse . . . . .	89
G.3 Résumés des chapitres relatifs aux contributions méthodologiques et apports vis-à-vis des applications . . . . .	90
G.4 Conclusion . . . . .	94
G.4.1 Communications et publications dans revues à comité de lecture . . .	95
G.4.2 Développements informatiques open source . . . . .	96



# List of Figures

1	General uncertainty quantification framework (adapted from <a href="#">Ajenjo (2023)</a> ) . . . . .	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 ( $n = 256$ ) . . . . .	22
1.6	Latin hypercube designs with poor and optimized space-filling properties ( $n = 8$ )	25
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 . . . . .	35
1.11	Illustration of a $k$ -fold cross-validation (with $k = 4$ ) . . . . .	44
1.12	Illustration of an ordinary kriging model fitted on a limited set of observations ( $n = 7$ ) . . . . .	45
1.13	Illustration of the expected improvement learning criterion . . . . .	47
1.14	Illustration of the deviation number learning criterion . . . . .	48
2.1	Wind spectrum from Brookhaven, USA (source: <a href="#">Burton et al. (2021)</a> ) . . . . .	51
2.2	[Representation to be confirmed and mentioned in the text] . . . . .	52
2.3	Example of a turbulent wind field generated by TurbSim (source: <a href="#">Jonkman (2009)</a> )	53
2.4	Illustration of the wake created downstream a wind farm (source: <a href="#">Veers et al. (2019)</a> ) . . . . .	54
2.5	Peirson-Moskowitz and JONSWAP spectra at significant wave height $H_s = 3$ m and peak period $T_p = 7$ s (source: <a href="#">Milano (2021)</a> ) . . . . .	55
2.6	Hierarchy of structural (a) and aerodynamic (b) wind energy systems models (source: <a href="#">Veers et al. (2019)</a> ) . . . . .	56

A.1	Adequation of two different Weibull models using their likelihood with a sample of observations (black crosses). . . . .	68
A.2	Fit of a bimodal density by KDE using different tuning parameters. . . . .	69
A.3	QQ-plot between the data from Example 2 and a KDE model. . . . .	70
B.1	Evolution of $m_{IMSE}$ for different dimensions and sample sizes. . . . .	73
G.1	Schéma générique de la quantification des incertitudes (de Rocquigny et al. (2008), adapté par Ajenjo (2023)) . . . . .	88

# List of Tables



# 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.

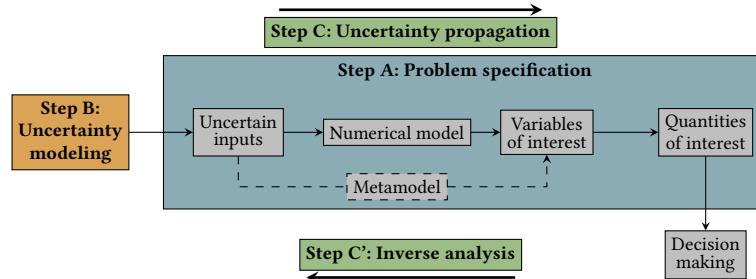


Figure 1 General uncertainty quantification framework (adapted from [Ajenjo \(2023\)](#))

## 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 industrial application presents various specificities which raise some 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`<sup>1</sup>
- This Python package, developed in collaboration with J.Muré, is available on the platform Pypi and fully documented.

---

- This Python package proposes an implementation of the “Bernstein Adaptive Nonparametric Conditional Sampling” method for rare event estimation.  
`bancs`<sup>2</sup>
- This Python package is available on the PyPI platform and is illustrated with examples and analytical benchmarks.

---

- This Python package presents a standardized process to benchmark different sampling methods for central tendency estimation.  
`ctbenchmark`<sup>3</sup>
- This Python package is available on a GitHub repository with analytical benchmarks.

---

- This Python package proposes an implementation of a synthetic visualization tool for multivariate distributions.  
`copulogram`<sup>4</sup>
- This Python package, developed in collaboration with V.Chabridon, is available on the Pypi platform.

---

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

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

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

4. Repository: <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). “Given-data probabilistic fatigue assessment for offshore wind turbines 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 &amp; 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” <sup>5</sup> . 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)

5. This 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 statement . . . . .	27
1.5.2	Rare event estimation methods . . . . .	29
1.5.3	Summary and discussion . . . . .	36
1.6	Global sensitivity analysis . . . . .	36
1.6.1	Screening methods . . . . .	36
1.6.2	Variance-based importance measures . . . . .	38
1.6.3	Moment-independent importance measures . . . . .	41
1.6.4	Summary and discussion . . . . .	42
1.7	Surrogate modeling . . . . .	42
1.7.1	Common framework . . . . .	42
1.7.2	General purposes surrogate model . . . . .	44
1.7.3	Goal-oriented active surrogate model . . . . .	46
1.7.4	Summary and discussion . . . . .	48
1.8	Conclusion . . . . .	48

---

## 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 1.2 describes the context of the model specification step; Section 1.3 presents a classification of the inputs uncertainties and the probabilistic framework to model them; Section 1.4 and 1.5 introduce various methods to propagate the input uncertainties through the numerical model for different purposes; Section 1.6 presents the main inverse methods to perform sensitivity analysis in our framework; Finally, 1.7 introduces the concept of surrogate models to emulate a model by realizing a statistical learning on a limited dataset.

**OpenTURNS**<sup>1</sup>. 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 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 open source 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 **Persalys**<sup>2</sup>. 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

---

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

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

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.

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:

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

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. 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  $\Omega$ , which contains a set of outcomes  $\omega \in \Omega$ . An *event* is defined as a set of outcomes in the sample space. Then, a  $\sigma$ -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  $\mu$  (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  $\mu$  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  $\mathbf{x}$  tends to  $-\infty$  and to 1 as  $\mathbf{x}$  tends to  $+\infty$ . In the continuous case, one can also define a corresponding *probability density function* (PDF)  $f_{\mathbf{X}} : \mathcal{D}_{\mathbf{x}} \rightarrow \mathbb{R}_+$  with  $f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^d F_{\mathbf{X}}(\mathbf{x})}{\partial x_1 \dots \partial x_d}$ .

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

$$\mathbb{E}[\mathbf{X}] = \int_{\Omega} \mathbf{X}(\omega) d\mu(\omega) = \int_{\mathcal{D}_{\mathbf{x}}} \mathbf{x} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = (\mathbb{E}[X_1], \dots, \mathbb{E}[X_d])^T. \quad (1.5)$$

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  $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in \mathcal{D}_{\mathbf{x}}^n$ . The corresponding ranked sample is defined as:  $\mathbf{R}_n = \{\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}\}$ , where <sup>3</sup>  $r_j^{(l)} = \sum_{i=1}^n \mathbb{1}_{\{x_j^{(i)} \leq x_j^{(l)}\}}$ ,  $\forall j \in \{1, \dots, d\}$ . Ranking a multivariate dataset allows us to isolate the dependence structure

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

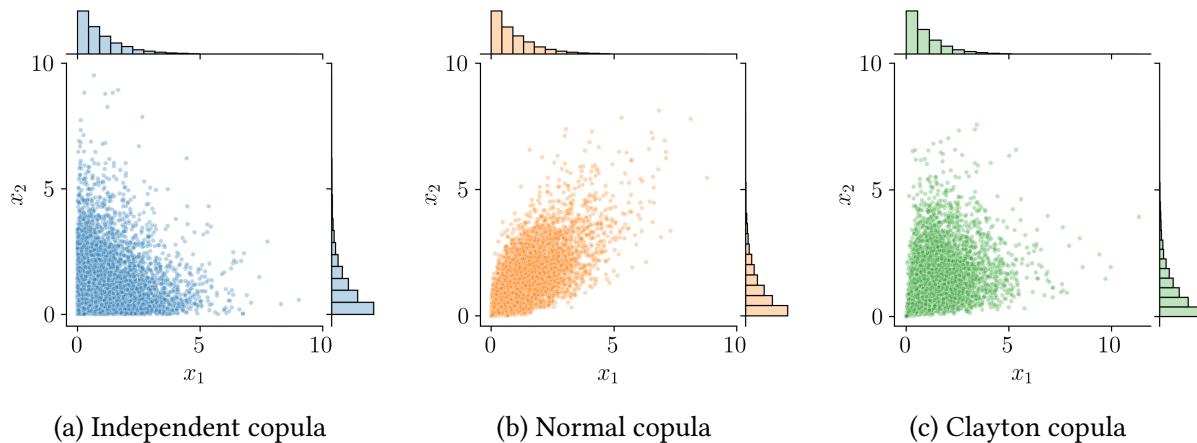


Figure 1.1 Samples of three joint distributions with identical marginals and different dependence structures

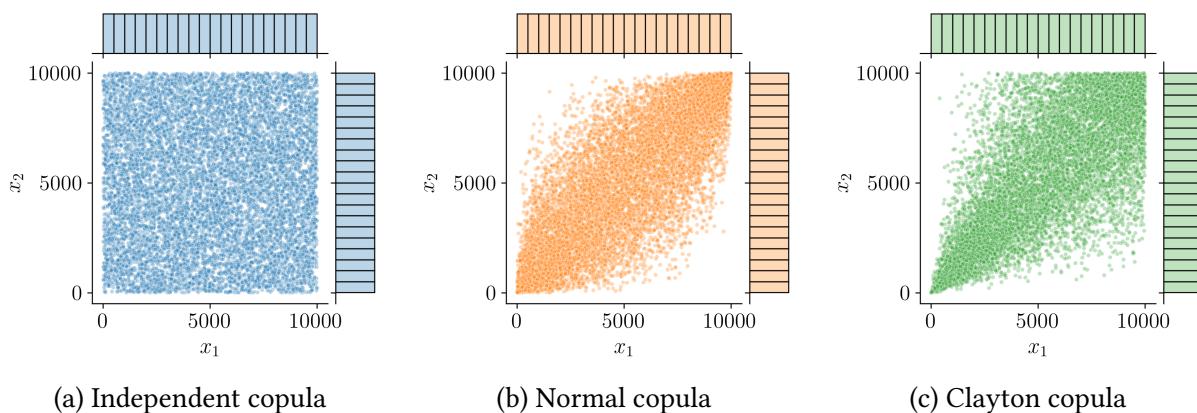


Figure 1.2 Ranked samples represented in the Fig. 1.1

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.

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 [Joe \(2014\)](#) 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:

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

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. In practice, the following points can be checked at this early stage:

- Is the distribution unimodal? If not, mixture methods or nonparametric models might be required;
- Is the validity domain restrictive? If so, specific families of parametric distributions can be chosen or truncation can be applied;
- Is there a dependence structure? Does it concern all the variables together or only some groups of variables?
- Is the dependence structure complex? Transforming the dataset in the ranked space gives an empirical description of the dependence.

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  $\mu$ . 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.

### Classical multivariate deterministic quadrature

Historically, quadrature methods have been developed for univariate integrals. The Gaussian rule and the Fejér-Clebsch-Curtis rule are two univariate deterministic quadratures that will be briefly introduced (see [Sullivan \(2015\)](#) for further elements).

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 Clebsch 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*.

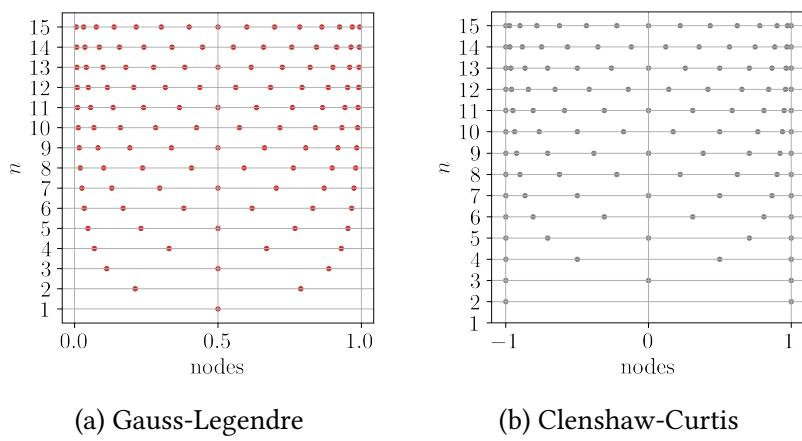


Figure 1.3 Univariate quadratures nodes ( $1 \leq n \leq 15$ )

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. [\[Introduce the recurrent Smolyak formula?\]](#)

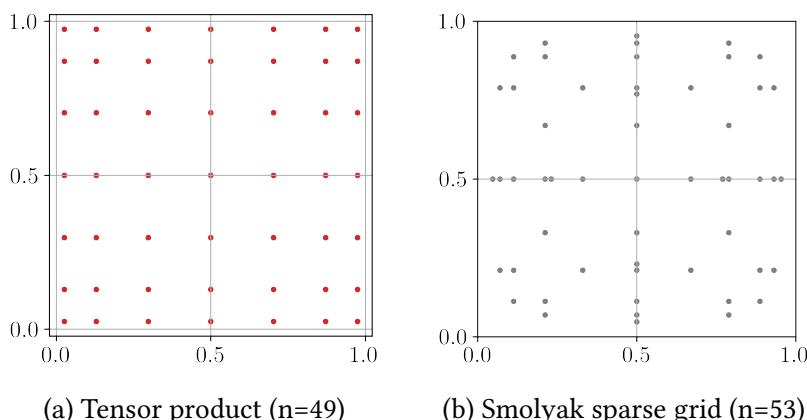


Figure 1.4 Two univariate Gauss-Legendre quadratures combined as a tensor product and a Smolyak sparse grid

## 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  $\mu$ , 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  $4.3 \times 10^{6001}$  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  $\mu$ . 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  $q_\alpha$  is the  $\alpha$ -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 intractable 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 [Leobacher and Pillichshammer \(2014\)](#) Section 3.4 for further mathematical proof.

Where the function variation  $V(g)$  in the Eq. (1.18) 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  $L_\infty$  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:

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

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 called *maximum mean discrepancy* is introduced in the Appendix D and used for different purposes in the [Part 2] of this manuscript.

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.

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 computing confidence intervals. In

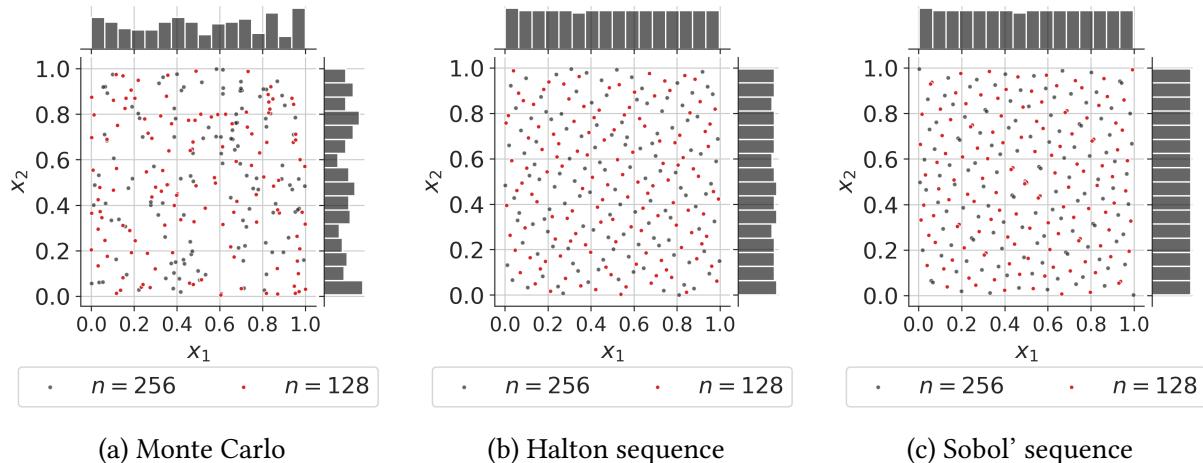


Figure 1.5 Nested Monte Carlo and quasi-Monte Carlo designs ( $n = 256$ )

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 subdomains. 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  $L_\infty$  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  $\phi_{min}$ ). For numerical reasons, the  $\phi_p$  criterion is often used instead of the minimal distance. The following  $\phi_p$  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 \(2018\)](#) and in the Appendix A from [Da Veiga et al. \(2021\)](#). 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 in Eq. (1.16), LHS variance can be expressed as:

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

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., Fig. 1.6a). 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 first mention the maximum projection designs from Joseph et al. (2015) which relies on the optimization of a geometrical criterion and delivers interesting performances. In the same vein, the uniform projection designs from Sun et al. (2019) are also a method to optimize LHS, this time based on a criterion averaging two-dimensional discrepancies.

**OpenTURNS 3** (Design of experiments). The following Python code proposes a minimalist OpenTURNS implementation to build an LHS and an LHS optimized w.r.t. to a space-filling metric (here the  $L_2$ -centered discrepancy) using the simulated annealing algorithm.

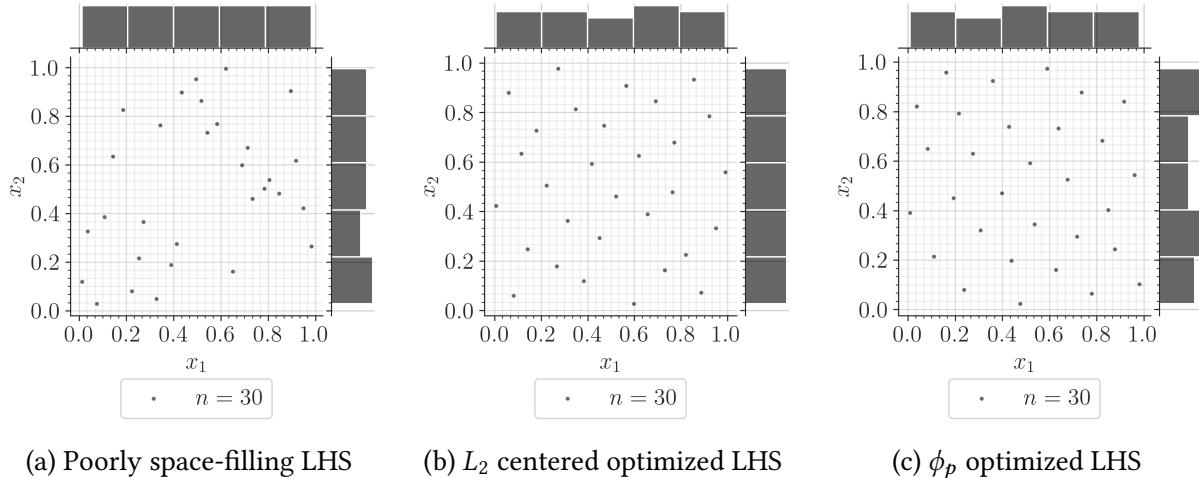


Figure 1.6 Latin hypercube designs with poor and optimized space-filling properties ( $n = 8$ )

### 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 4].

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  $f_x$ . Note that QMC sampling is not well suited with acceptance-rejection since its structure gets perturbed.

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 nonuniform 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 related to the tail of the distributions) can be studied depending on the type of risk analysis. Quantiles are a first conservative measure, widely used for risk analysis. The  $\alpha$ -quantile  $q_\alpha$  of the output random variable  $Y$  is defined as:

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

As an alternative, one can define a scalar safety threshold  $y_{\text{th}}$  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*:

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

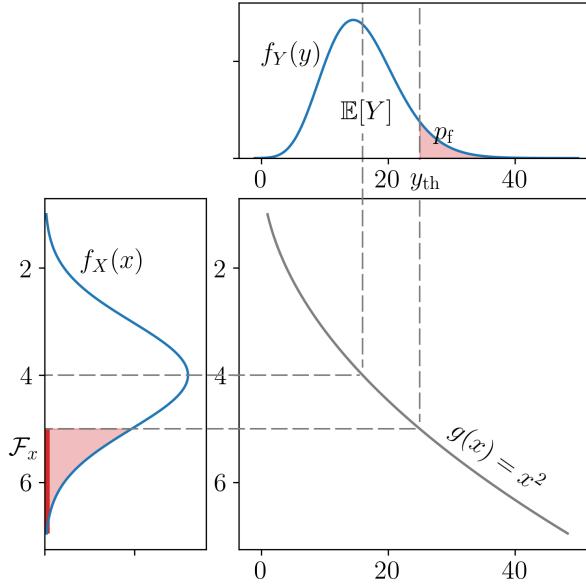


Figure 1.7 One-dimensional reliability analysis example

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  $y_{\text{th}}$  is represented by the area in red under the output PDF on top. An interesting reflection on the use and the interpretation of risk measures including measures from the finance domain such as the *conditional value-at-risk* (also called superquantile) 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 statement

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 Eq. (1.28), 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  $\mathcal{F}_x^0 := \{\mathbf{x} \in \mathcal{D}_x | g(\mathbf{x}) = 0\}$ . Similarly to any UQ study around a numerical model, this problem can require to be resolved using a limited number 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 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. The commonly named *failure probability*, denoted  $p_f$ , will be our quantity of interest in reliability analysis. This quantity is formally written<sup>4</sup>:

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

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 events 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:

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

---

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

Since this transformation is a diffeomorphism<sup>5</sup>, one can apply the change of variable  $\mathbf{x} = T(\mathbf{u})$  to express the reliability problem from Eq. (1.31) 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 introduced by [Lebrun \(2013\)](#). In practice, the transformation choice depends on the properties of the input distribution studied. For example in OpenTURNS, depending on the three following cases different types of transformations are applied:

- for elliptical distributions, a linear Nataf transformation is applied;
- for distributions with an elliptical copula, the generalized Nataf transformation is used;
- otherwise, the Rosenblatt transformation is used.

### 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 ([Lemaire, 2013](#)). They extrapolate a local approximation of the LSF built in the vicinity of a *most-probable-failure-point* (MPFP), also called *design point*.

---

5. Considering 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  $T^{-1} : B \rightarrow A$ .

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:

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

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., Abdo-Rackwitz algorithm ([Abdo and Rackwitz, 1991](#))) but can also use gradient-free techniques (e.g., Cobyla algorithm ([Powell, 1994](#))). 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 (see the illustration in Fig. 1.8).

This distance between the origin and  $P^*$  is a different risk measure, defined as the *Hasofer-Lind reliability index* ([Lemaire, 2013](#)),  $\beta \in \mathbb{R}$  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).

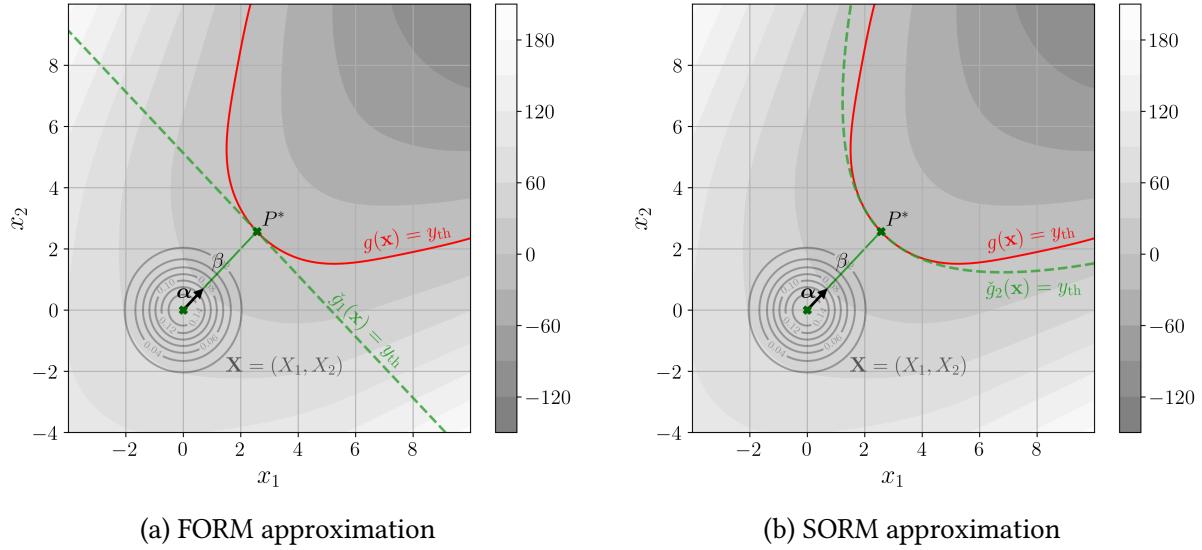


Figure 1.8 FORM and SORM approximation on a two-dimensional example

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 (see [Der Kiureghian and Dakessian \(1998\)](#)) prevents this situation by applying successive FORM. Once the first MPFP  $P^{*(1)}$  found, the LSS is modified by removing a nudge to find to following MPFP  $P^{*(2)}$ .

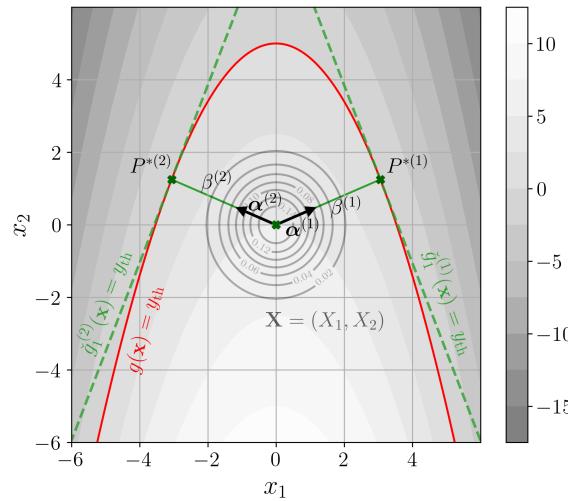


Figure 1.9 Multi-FORM approximation on an example with two MPFPs

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 the Chapter 1 from Chabridon (2018)). Additionally, their main drawback is the lack of complementary information concerning the confidence of the results. The example illustrated in the Fig. 1.9 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 unbiased 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 central limit theorem (similarly to the ones from Eq. (1.17)). Because of the small scale of the quantities manipulated in rare event estimation, the estimator's coefficient of variation is also widely used:

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

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. Third, it is qualified as embarrassingly parallel method since each of the numerical simulations are independent. Finally, it offers strong convergence guaranties and complementary information on the estimation confidence. These properties often make Monte Carlo the reference method in rare event estimation benchmarks.

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 drawing 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  $\{\mathbf{x}^{(i)}\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} h_X$ :

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

Similarly to Monte Carlo, this estimator is unbiased, however, 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 Eq. (1.47):

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

The optimal expression above is unfortunately not usable in practice since it includes the targeted quantity  $p_f$ . Considering this framework, various techniques intend to define instrumental distributions as close as possible to this theoretical result.

The most immediate solution is to combine the information provided by the results of FORM with importance sampling, which is called FORM-IS. In practice, the instrumental distribution is simply the initial distribution centered on the design point resulting from FORM. This strategy is simple to implement, but it inherits the main drawbacks of FORM, such as the problems with multiple failure areas (see the example illustrated in Fig. 1.9). Other importance sampling schemes integrate adaptive mechanisms, progressively leading the sampling towards the failure domain.

**[Adaptive importance sampling]**

The *adaptive importance sampling by cross-entropy* (AIS-CE) is a first adaptive strategy, optimizing the variance reduction by searching the best instrumental distribution throughout a parametric family.

**[Nonparametric adaptive importance sampling]**

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

## 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  $\mathbf{X}_{[k],N}$  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  $f_x(\mathbf{x} | \mathcal{F}_{[k-1]})$ , using as seeds initialization points the  $n = Np_0$  samples given by  $\mathbf{A}_{[k],n} = \{\mathbf{X}_{[k-1]}^{(j)} \subset \mathbf{X}_{[k-1],N} | g(\mathbf{X}_{[k-1]}^{(j)}) > \hat{q}_{[k-1]}^{\alpha}\}_{j=1}^n$ . 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:

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

In practice, the subset sample size should be large enough to properly estimate intermediary quantiles, which leads [Au and Beck \(2001\)](#) to recommend setting  $p_0 = 0.1$ . The efficiency of the SS method depends on the 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 C presenting the algorithms of SS]

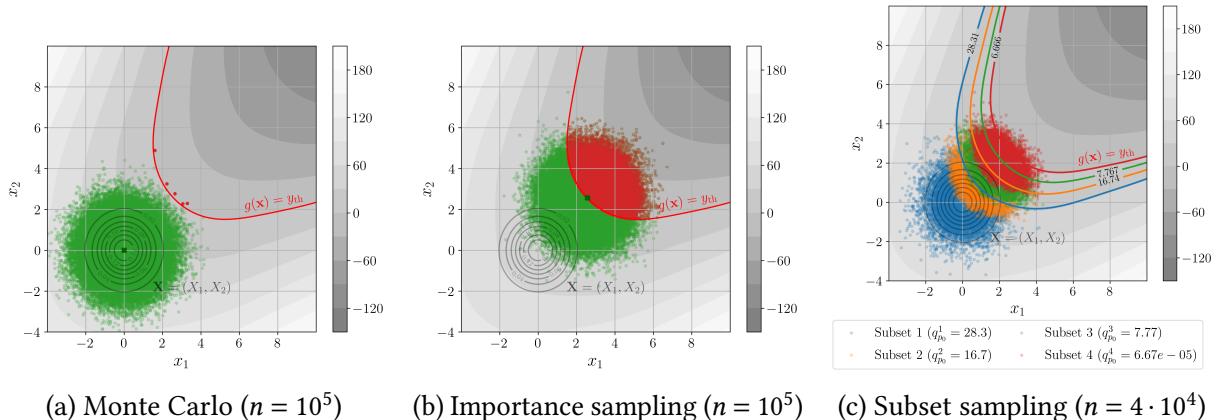


Figure 1.10 Illustration of a rare event estimation

### 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 Global sensitivity analysis

The aim of sensitivity analysis (SA) is to determine the impact of a single (or a group of random inputs) on a random output(s). As described earlier, this step is qualified as an inverse analysis in the general UQ framework (illustrated in Fig. 1), in opposition with the forward uncertainty propagation step. In fact, the analyst studies the effect of the inputs at different scales, hence the distinction between “local” and “global” SA. Local SA focuses on the impact of small perturbations around nominal values of the inputs (i.e., derivative-based approaches), while global sensitivity analysis (GSA), typically studies the general variability (e.g., the variance) of the output. Two types of GSA methods exist in the literature, either proposing qualitative and quantitative approaches:

- *screening methods*: determines the non-influential variables in a UQ study (qualitative);
- *importance measures*: assess the contribution of input(s) in the global variability of the output (quantitative).

Screening methods are typically used in a statistical learning process, to drop the irrelevant variables to the learning. In this context, *feature selection* serves the same purpose with a slight difference. Screening methods usually assume the inputs to be independent while feature selection does not. Moreover, feature selection not only looks for the irrelevant features to the learning but also the redundant features. The reader may refer to [\[add feature selection review\]](#) for further insights on this topic.

The global sensitivity of an output can be explained by different elements: the single variability of the inputs, their dependence, and their interactions. Two variables present interactions when their simultaneous effect on an output is not additive. Note that SA on dependent inputs is an active field of research and the inputs will mostly be considered as independent in the following.

### 1.6.1 Screening methods

Many UQ methods suffer from the curse of dimensionality, thankfully, high-dimensional problems often only depend on a few variables. This observation was formalized with the concept of *effective dimension* introduced by [Owen \(2003\)](#). Screening methods allow to discriminate the

non-influential variables, which can be considered afterwards as determinist to simplify the problem.

### Morris method

The Morris method ([Morris, 1991](#)) is a screening method historically commonly used in engineering applications. It starts by mapping the input domain  $\mathcal{D}_X$  into a unit hypercube  $[0, 1]^d$ , which is discretized as a regular grid with step  $\Delta \in \mathbb{R}$ . The algorithm computes local elementary sensitivity by building “one at a time” (OAT) local trajectories over the regular grid. Each OAT designs start at a random node  $\mathbf{x}^{(t)} = (x_1^{(t)}, \dots, x_j^{(t)}, \dots, x_d^{(t)})$  of the grid and move only in one direction by an increment equal to the elementary step such that:  $\mathbf{x}^{(t)} + \Delta_j = (x_1^{(t)}, \dots, x_j^{(t)} + \Delta, \dots, x_d^{(t)})$ . The elementary effect in the direction of the variable  $i$  from an OAT design (also called trajectory)  $t$  is expressed as a finite difference:

$$\text{EE}_j^{(t)} = \frac{g(\mathbf{x}^{(t)}) - g(\mathbf{x}^{(t)} + \Delta_j)}{\Delta}. \quad (1.51)$$

The Morris method generates  $T \in \mathbb{N}$  OAT designs and computes theirs respective elementary effects in each direction  $i$ . To assess the global sensitivity of the function, the mean  $\bar{\text{EE}}_j$  and variance  $\widehat{\text{Var}}(\text{EE}_j)$  of the elementary effects are computed:

$$\bar{\text{EE}}_j = \frac{1}{n} \sum_{t=1}^T |\text{EE}_j^{(t)}|, \quad \widehat{\text{Var}}(\text{EE}_j) = \frac{1}{n-1} \sum_{t=1}^T (\text{EE}_j^{(t)} - \bar{\text{EE}}_j)^2. \quad (1.52)$$

It allows to divide the variables into three categories, regardless of any regularity hypothesis on the function: (i) negligible effects; (ii) linear effects without interaction; and (iii) nonlinear effects with possible interactions. This method is very intuitive but quickly shows its limits as the dimension increases since it relies on a discretization of the space by a regular grid. Another disadvantage of this method is that it does not distinguish interactions and nonlinear effects of inputs.

### Derivative-based global sensitivity measures

The Derivative-based global sensitivity measures (DGSM) are a GSA method introduced in [Sobol and Gresham \(1995\)](#) and further studied in [Kucherenko et al. \(2009\)](#). As the Morris method, they study the mean value of local derivatives of the model output with regard to the inputs:

$$v_j = \int_{\mathcal{D}_X} \left( \frac{\partial g(\mathbf{x})}{\partial x_j} \right)^2 f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \mathbb{E} \left[ \left( \frac{\partial g(\mathbf{X})}{\partial X_j} \right)^2 \right]. \quad (1.53)$$

This continuous formulation does not require using OAT designs, which was proven to be more efficient when exploiting designs such as quasi-Monte Carlo. The efficiency of the DGSMs for screening purposes was outlined in many papers, such as [Kucherenko and Iooss \(2017\)](#). Since their value depends on the probability distribution of the input, a normalized version

was developed. The connections between DGSM and variance-based GSA measures (i.e., Sobol' indices introduced hereafter), revealed bounding properties between DGSMs and Sobol' total indices (Lamboni et al., 2013).

### 1.6.2 Variance-based importance measures

Screening methods determine the non-influential variables in a UQ problem. Beyond this information, importance measures quantify the influence of inputs, allowing to rank the inputs according to their contribution to the output variability.

#### Functional variance decomposition and Sobol' indices

Sobol' indices are the most popular importance measure in GSA. Their universality comes from the functional decomposition of the output's variance, attributing variance share to the inputs. Considering a squared-integrable and measurable function  $g(\cdot)$  and the independent random vector  $\mathbf{X}$ . The output random variable  $Y = g(\mathbf{X})$  can be decomposed, according to Hoeffding (1948), as:

$$Y = g(\mathbf{X}) = g_0 + \sum_{j=1}^d g_j(X_j) + \sum_{j < l} g_{jl}(X_j, X_l) + \dots + g_{1\dots d}(\mathbf{X}), \quad (1.54)$$

with the previous terms defined according this recurrence:

$$g_0 = \mathbb{E}[g(\mathbf{X})] \quad (1.55)$$

$$g_j(X_j) = \mathbb{E}[g(\mathbf{X})|X_j] - g_0 \quad (1.56)$$

$$g_{jl}(X_j, X_l) = \mathbb{E}[g(\mathbf{X})|X_j, X_l] - g_j(X_j) - g_l(X_l) - g_0 \quad (1.57)$$

$$\dots \quad (1.58)$$

Sobol in Sobol' (1993) proved that this decomposition is unique by exploiting the orthogonality of the terms of the decomposition. Therefore, this decomposition can be transposed in terms of functional decomposition of variance (also called functional analysis of variance or F-ANOVA):

$$\text{Var}(Y) = \sum_{j=1}^d V_j(Y) + \sum_{j < l} V_{jl}(Y) + \dots + V_{1\dots d}(Y), \quad (1.59)$$

with the previous terms defined in a recurrent way as  $V_j(Y) = \text{Var}(\mathbb{E}[Y|X_j])$ ,  $V_{jl}(Y) = \text{Var}(\mathbb{E}[Y|X_j, X_l]) - V_j(Y) - V_l(Y)$ , and so on for higher order interaction terms. The Sobol' indices of different order are defined as normalized shares of variance. The first-order Sobol' index  $S_j$  quantifies the share of variance of the output only explained by the marginal  $X_j$  (also called main effect). Second order  $S_{jl}$  (or higher order) Sobol' indices quantify the effect of the interactions between a group

of marginals.

$$S_j = \frac{V_j(Y)}{\text{Var}(Y)} = \frac{\text{Var}(\mathbb{E}[Y|X_j])}{\text{Var}(Y)} \quad (1.60)$$

$$S_{jl} = \frac{V_{jl}(Y)}{\text{Var}(Y)} = \frac{\text{Var}(\mathbb{E}[Y|X_j, X_l]) - V_j(Y) - V_l(Y)}{\text{Var}(Y)} \quad (1.61)$$

$$\dots \quad (1.62)$$

The generic definition of the Sobol' sensitivity indices associated to a subset of inputs  $A \in \mathcal{P}_d$ , with  $\mathcal{P}_d$  the set of all possible subsets of  $\{1, \dots, d\}$ , is given by:

$$S_A = \frac{V_A(Y)}{\text{Var}(Y)} = \frac{\sum_{B \subset A} (-1)^{|A|-|B|} \text{Var}(\mathbb{E}[Y|X_B])}{\text{Var}(Y)}. \quad (1.63)$$

By using the functional decomposition of variance in Eq. (1.59), one can show that the Sobol' indices add up to one:

$$\sum_{A \in \mathcal{P}_d} S_A = 1. \quad (1.64)$$

Assessing Sobol' indices for every order becomes complex in medium to high dimension. The total Sobol' index  $S_j^T$  associated with the variable  $j$ , see Saltelli et al. (2008), quantifies the share of output variance which is explained by all the interactions of the variable  $X_j$ :

$$S_j^T = 1 - \frac{\text{Var}(\mathbb{E}[Y|\mathbf{X}_{-j}])}{\text{Var}(Y)} = \frac{\mathbb{E}[\text{Var}(Y|\mathbf{X}_{-j})]}{\text{Var}(Y)}, \quad (1.65)$$

where  $\mathbf{X}_{-j}$  represents all the marginals from  $\mathbf{X}$  but  $X_j$ . This definition can also be generalized for a subset of inputs  $A \in \mathcal{P}_d$ , such that:

$$S_A^T = 1 - \sum_{A' \subset A} S_{A'} = 1 - \frac{\text{Var}(\mathbb{E}[Y|\mathbf{X}_A])}{\text{Var}(Y)}, \quad (1.66)$$

where the second term of this expression is called *closed Sobol' index*,  $S_A^{\text{clos}} = 1 - S_A^T = \frac{\text{Var}(\mathbb{E}[Y|\mathbf{X}_A])}{\text{Var}(Y)}$ .

Jointly analyzing the first and total Sobol' indices offers a quantified decomposition between the marginal and interaction effects. Note that the total indexes are only equal to the first indexes when the model does not present interactions (i.e., purely additive model).

Estimating Sobol' indices can be achieved in various ways. The pick-freeze scheme is the historical estimation technique based on two samples, however, it requires an often prohibitive number of calls to the function. Many estimators using the pick-freeze generic scheme were developed to estimate Sobol' indices (e.g., Saltelli's, Jansen's, Martinez's etc.), see further references are collected in the Chapter 3 of Da Veiga et al. (2021). Alternatively, the surrogate models were exploited to estimate such sensitivity measure. Using a unique input-output dataset, the analyst may build a *polynomial chaos expansion* (PCE) surrogate model, which gives

an explicit expression of the Sobol' indices ([Sudret, 2008](#)). Authors such as [Marrel et al. \(2009\)](#) also studied the use of Gaussian processes for this purpose.

In the case of independent inputs, the first and total Sobol' indices is a complete tool for GSA. The main advantage of this approach is the quantitative nature of its results, allowing to objectively compare the effect of inputs variables. When the inputs present a dependence structure, it becomes complicated to distinguish its effects from possible interactions. However, many authors tried to adapt the Sobol's indices to this context. Chapter 5 of [Da Veiga et al. \(2021\)](#) reviews four of these approaches. For example, [Mara and Tarantola \(2012\)](#) proposed two extra Sobol' indices, called "full indices", detecting the contributions associated with the inputs' dependence. Note that the interpretation and estimation of this solution becomes complicated. Moreover, unlike the independent case, the four Sobol' indices do not divide the output variance between the inputs. Beyond Sobol' indices, another important GSA method was adapted from the theory of Shapley values by [Owen \(2014\)](#), allowing to work with dependent inputs.

## Shapley effects

Shapley effects are an adaptation to GSA by [Owen \(2014\)](#) of the Shapley values from the cooperative games' theory ([Shapley et al., 1953](#)). This method is an alternative to the Sobol' indices in the case of dependent inputs, for which the natural interpretation of single interaction effects no longer holds. In the theory of Shapley values, the Shapley values act as a rule to share the value created by a team between its members (players). The Shapley values allocated to the player  $X_j$  is given considering the indices  $\{-j\} = \{1, \dots, d\} \setminus \{j\}$ :

$$\varsigma_j = \sum_{A \subset -\{j\}} \binom{d-1}{|A|}^{-1} (\text{val}(A \cup \{j\}) - \text{val}(A)), \quad (1.67)$$

where the value (or cost) function is denoted  $\text{val}(A)$ , and  $A$  is a subset of  $\{1, \dots, d\}$  with cardinality  $|A|$ . The Shapley effects adapted this concept to perform a GSA by considering the variables as players and the closed Sobol' indices for the value function:

$$Sh_j = \sum_{A \subset -\{j\}} \binom{d-1}{|A|}^{-1} \left( S_{A \cup \{j\}}^{\text{clos}} - S_A^{\text{clos}} \right). \quad (1.68)$$

Conceptually, this expression compares a performance defined by a cost function with or without the variable  $X_j$ , and averages it over all the possible combinations of inputs. This importance measure offers the following decomposition:

$$\sum_{j=1}^d Sh_j = 1. \quad (1.69)$$

In the case of independent inputs, the Shapley effects present properties related to the Sobol' indices. The following equation (see proof in [Owen \(2014\)](#)) reveal that the Shapley effects

equally divide the interaction effects between the implicated variable:

$$S_j \leq Sh_j \leq S_j^T, \quad Sh_j = \sum_{A \in \mathcal{P}_d, j \in A} \frac{S_A}{|A|}. \quad (1.70)$$

Unlike the Sobol' indices, Shapley effects are a nonnegative allocation of output variance with equitable division of the interaction effects. This method presents an interesting alternative in the dependent case, however, estimating Shapley effects creates computational difficulties. The reader may refer to the permutation-based algorithm from [Song et al. \(2016\)](#) and the introduction of surrogate model in [Benoumechiara and Elie-Dit-Cosaque \(2019\)](#) for further elements on this topic.

Shapley effects are a promising importance measure based on variance allocation. However, in some cases the variance of the output distribution does not represent well its variability (e.g., multimodal distribution). The following section introduces another family of GSA methods based on distances between distributions.

### 1.6.3 Moment-independent importance measures

Beyond variance-based GSA, many types of distances between distributions have been used to evaluate the dependence between the input and the output distributions. Comparing the entire distributions instead of their moments might be more robust in some cases (e.g., cases when the variance is a poor indicator of the variability). The tools used to do so are generally called *dissimilarity measures* between distributions. Appendix D briefly introduces two families of *dissimilarity measures*: the class of  $f$ -Csiszár divergences (e.g., the Kullback-Leibler divergence, total variation distance) and the class of integral probability metrics (IPM) (e.g., Wasserstein distance, total variation distance, maximum mean discrepancy). This appendix focuses on the *maximum mean discrepancy* (MMD), a kernel-based dissimilarity measure between distributions with interesting estimation properties.

Considering the probability measures  $\mathbb{P}_{X_j}$  and  $\mathbb{P}_Y$  (associated with the random variables  $X_j$  and  $Y$ ) and a dissimilarity measure  $\Delta(\cdot, cdot)$ , one can define two formulations to perform a GSA:

- by directly using a dissimilarity measures to assess  $\Delta(\mathbb{P}_Y, \mathbb{P}_{Y|X_j})$ ;
- by building a *dependence measures* evaluating  $\Delta(\mathbb{P}_{(X_j, Y)}, \mathbb{P}_{X_j} \otimes \mathbb{P}_Y)$ .

The first approach was studied in association with  $f$ -divergences in [Da Veiga \(2015\); Rahman \(2016\)](#). However, these dissimilarity measures introduce estimation issues, which are simplified with kernel-based metrics such as the MMD. The following section presents the *Hilbert-Schmidt Independence Criterion* (HSIC) was initially introduced by [Gretton et al. \(2006\)](#) for dependence testing, and adapted as a dependence measure in GSA by [Da Veiga \(2015\)](#).

### Hilbert-Schmidt independence criterion

Let us first recall the definition of the maximum mean discrepancy, further introduced in Appendix D. This distance between two probability distributions  $\mu$  and  $\zeta$  is defined as the worst-case error for any function within a unit ball of a function space  $\mathcal{H}$ :

$$\text{MMD}(\mu, \zeta) := \sup_{\|g\|_{\mathcal{H}(k)} \leq 1} \left| \int_{\mathcal{D}_X} g(\mathbf{x}) d\mu(\mathbf{x}) - \int_{\mathcal{D}_X} g(\mathbf{x}) d\zeta(\mathbf{x}) \right| = \|P_\mu(\mathbf{x}) - P_\zeta(\mathbf{x})\|_{\mathcal{H}(k)}. \quad (1.71)$$

This powerful distance is used in GSA to measure the dependence of inputs and outputs. Considering the pair of random variables  $(X_j, Y)$ , with probability distributions  $\mathbb{P}_{X_j}$  and  $\mathbb{P}_Y$  let us define the RKHS  $\mathcal{H}$  induced by the product kernel  $k((x_j, y), (x'_j, y')) = k_{X_j}(x_j, x'_j)k_Y(y_j, y'_j)$ . The HSIC between the variable  $X_j$  and  $Y$  is defined as the following dependence measure:

$$\text{HSIC}(X_j, Y) = \text{MMD}(\mathbb{P}_{(X_j, Y)}, \mathbb{P}_{X_j} \otimes \mathbb{P}_Y)^2. \quad (1.72)$$

[Add analysis about the FANOVA-like decomposition.]

[The simplicity of the estimation.]

[Links with Sobol indices for a simple kernel.]

#### 1.6.4 Summary and discussion

## 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  $\widehat{g}_n(\cdot)$ , 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:

$$\{\mathbf{X}_n, \mathbf{y}_n\} = \left\{ \mathbf{x}^{(i)}, y^{(i)} \right\}_{i=1}^n = \left\{ \mathbf{x}^{(i)}, g(\mathbf{x}^{(i)}) \right\}_{i=1}^n. \quad (1.73)$$

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 tensor approximations (Grasedyck et al., 2013), 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 guaranty its fidelity with regard to the numerical model. When a  $m$ -sized input-output set is dedicated to validating the surrogate model, independently of 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:

$$R^2(\hat{g}_n) = 1 - \frac{\sum_{i=1}^n (y(\mathbf{x}^{(i)}) - \hat{g}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^n (y(\mathbf{x}^{(i)}) - \bar{y}_n)^2}, \quad (1.74)$$

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):

$$Q^2(\hat{g}_n) = 1 - \frac{\text{ISE}(\hat{g}_n)}{\text{Var}(\hat{g}_n)}, \quad (1.75)$$

where

$$\text{ISE}(\hat{g}_n) = \int_{\mathcal{D}_X} (g(\mathbf{x}) - \hat{g}(\mathbf{x}))^2 d\mathbf{x}, \quad \text{Var}(\hat{g}_n) = \int_{\mathcal{D}_X} (g(\mathbf{x}) - \hat{g}(\mathbf{x}))^2 d\mathbf{x}. \quad (1.76)$$

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

$$\widehat{Q}^2(\hat{g}_n) = 1 - \frac{\sum_{i=1}^m (y(\mathbf{x}^{(i)}) - \hat{g}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^m (\bar{y}_m - y(\mathbf{x}^{(i)}))^2}. \quad (1.77)$$

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.

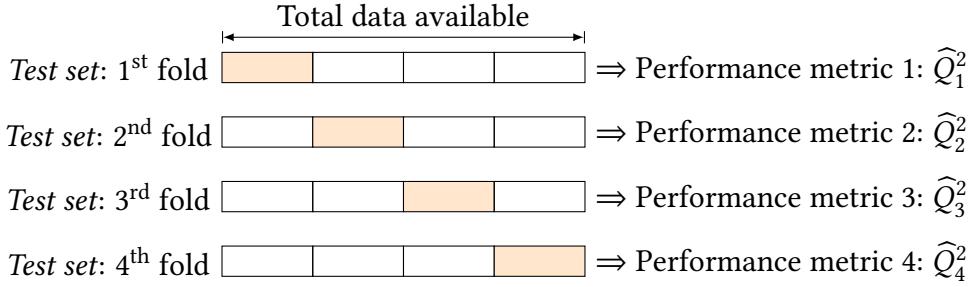


Figure 1.11 Illustration of a  $k$ -fold cross-validation (with  $k = 4$ )

### 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 Section 1.4 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, conditioned on a set of observations  $\mathbf{y}_n = \left\{ g\left(\mathbf{x}^{(i)}\right) \right\}$ . Let us first define a Gaussian process prior structure  $\xi$  on the function approximated  $g(\cdot)$  with a mean function  $m(\cdot)$  and covariance function  $k(\cdot, \cdot)$ :

$$\xi \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \quad (1.78)$$

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  $y_n$ . This Gaussian variable  $\xi(\mathbf{x})$  conditioned on the observations  $y_n$  is sometimes called conditional posterior  $\xi_n(\mathbf{x}) := (\xi(\mathbf{x})|y_n) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$ . The well-known “Kriging equations” (see e.g., [Rasmussen and Williams \(2006\)](#)) offer its explicit expression:

$$\begin{cases} \eta_n(\mathbf{x}) &:= \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{y}_n \\ s_n^2(\mathbf{x}) &:= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \end{cases} \quad (1.79)$$

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 learning 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]

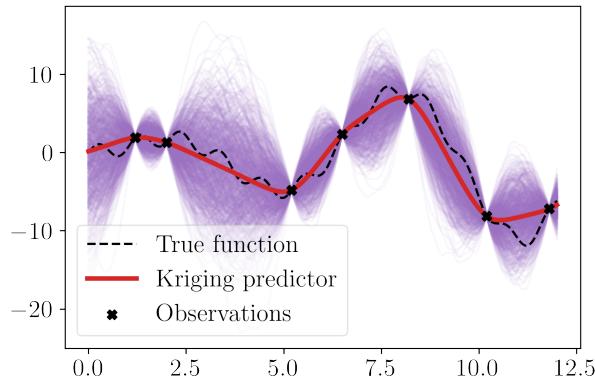


Figure 1.12 Illustration of an ordinary kriging model fitted on a limited set of observations ( $n = 7$ )

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 independent  $m$ -sized test set, the PVA is defined as:

$$\text{PVA} = \left| \log \left( \frac{1}{m} \sum_{i=1}^m \frac{(y(\mathbf{x}^{(i)}) - \hat{g}(\mathbf{x}^{(i)}))^2}{s_n^2(\mathbf{x}^{(i)})} \right) \right|. \quad (1.80)$$

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 domain, however surrogates are often used for specific purposes (e.g., contour finding for reliability analysis).

**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

Surrogates are often fitted for a specific purpose, requiring an accurate approximation over a limited subdomain only. In these cases, a more efficient approach might be to circumscribe the learning to this subdomain (i.e., *goal-oriented learning*), rather than uniformly over the entire domain. 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 focus the learning set around the optimum(s). Unfortunately, the area(s) of interest is usually unknown before evaluating the true function. *Active learning* is a general concept, aiming at iteratively increasing the learning set w.r.t. a *learning criterion* (also called acquisition function) depending on the surrogate's goal to enhance the surrogate in area(s) of interest. An exploration/exploitation trade-off arises in active learning, mostly resolved by the learning criterion.

*Remark 1.* This section introduces active learning methods in the computer experiment context, where the true function can be evaluated anywhere for a high computational cost. However, active learning is also used to handle big data frameworks in the machine learning community ([Qiu et al., 2016](#)). When datasets become so large that surrogate methods cannot be applied in practice, the analyst needs to select a subset on which the learning is performed.

#### Active kriging for optimization

In the field of black-box optimization, many methods rely on approximating the function by a surrogate. The use of Gaussian processes as probabilistic surrogates for optimization was popularized by the *efficient global optimization* (EGO) algorithm ([Jones et al., 1998](#)). Ever since, many related methods were developed under the generic name of *Bayesian optimization*. The main idea is to exploit the uncertainty model from the GP to direct the point selection. Concretely, the learning criterion depends on the gaussian process variance model. Numerous reviews of this field were proposed by [Gramacy \(2020\)](#); [Shahriari et al. \(2015\)](#) and numerical benchmarks presented in [Le Riche and Picheny \(2021\)](#).

The generic black-box optimization problem tackled is defined as:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{D}_X} g(\mathbf{x}) \approx \arg \min_{\mathbf{x} \in \mathcal{D}_X} \widehat{g}(\mathbf{x}) \quad (1.81)$$

To illustrate Bayesian optimization, let us present the EGO algorithm, defined by its specific learning criterion: the “expected improvement”. Considering an initial learning set  $\{\mathbf{X}_n, \mathbf{y}_n\}$  built on a space-filling input design  $\mathbf{X}_n$  to explore the domain. A first surrogate  $\xi_n(\mathbf{x}) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$  is fitted using Eq. (1.79). The expected improvement is then written as:

$$\mathcal{A}^{EI}(\mathbf{x}; \mathbf{y}_n) = \mathbb{E}[\max(g_{\min} - \xi_n(\mathbf{x}))] \quad (1.82)$$

$$= (g_{\min} - \eta_n(\mathbf{x})) \Phi\left(\frac{g_{\min} - \eta_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x}) \phi\left(\frac{g_{\min} - \eta_n(\mathbf{x})}{s_n(\mathbf{x})}\right), \quad (1.83)$$

where  $g_{\min} = \min(\mathbf{y}_n)$ ,  $\phi$  and  $\Phi$  respectively stand for the PDF and the CDF of the standard Gaussian distribution. This learning criterion is relatively inexpensive and allows to progressively enhance the Gaussian process to solve the optimization problem with a limited number of calls to the true function. Bayesian optimization is an active research field, with different open problems such as constrained Bayesian optimization (Petit, 2022), or Bayesian optimization on stochastic functions [cite Gramacy, M.Binois?].

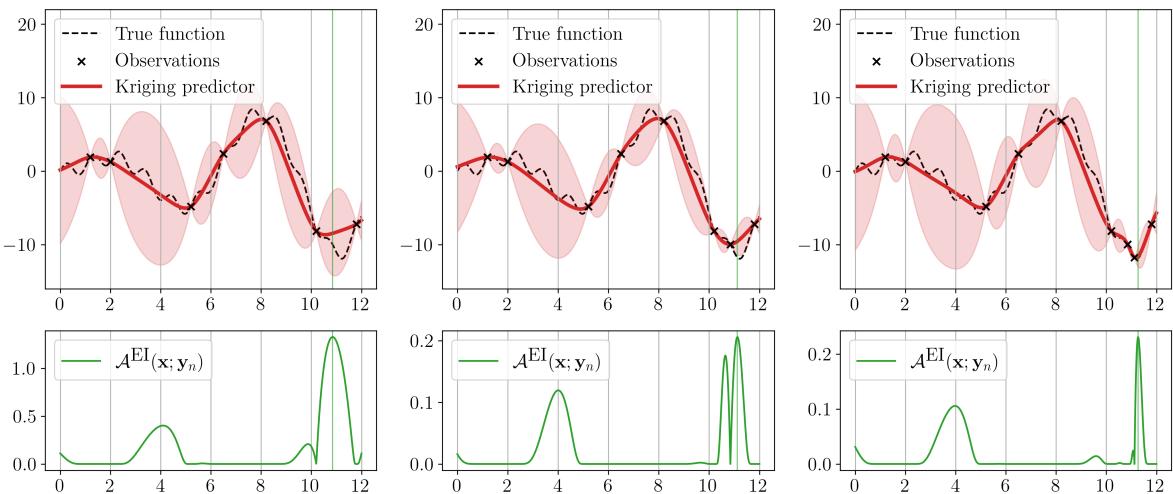


Figure 1.13 Illustration of the expected improvement learning criterion

### Active kriging for reliability analysis

Rare event estimation often requires large amounts of calls to the limit-state function (becoming intractable for 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. Active learning were proposed to iteratively concentrate the learning set around this border.

For rare event estimation, the surrogate only needs to be accurate near the limit state function. In other words, it should accurately discriminate the points leading to the safe domain from those leading to the failure domain. In fact, this problem can be seen as a two-class classification problem. Note that active learning procedure using SVM classifiers were adapted to this specific goal (Bourinet, 2018).

The following paragraph introduces the most popular kriging-based learning criterion: the deviation number  $U$  (Echard et al., 2011). Morio and Balesdent (2015) introduce further active learning techniques dedicated to rare event estimation. Teixeira et al. (2021) and Moustapha et al. (2022) review this topic with the presentation of wide numerical benchmarks.

Considering an initial learning set  $\{\mathbf{X}_n, \mathbf{y}_n\}$  built on a space-filling input design  $\mathbf{X}_n$  to explore the domain. A first Gaussian process  $\xi_n(\mathbf{x}) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$  is fitted using Eq. (1.79). Let us introduce the deviation number  $U$ , looking for points close to the limit-state function and presenting a high kriging variance:

$$\mathcal{A}^U(\mathbf{x}; \mathbf{y}_n) = \frac{|y_{th} - \eta_n(\mathbf{x})|}{s_n^2(\mathbf{x})}, \quad (1.84)$$

where  $y_{th} \in R$  is a threshold defining the failure domain.

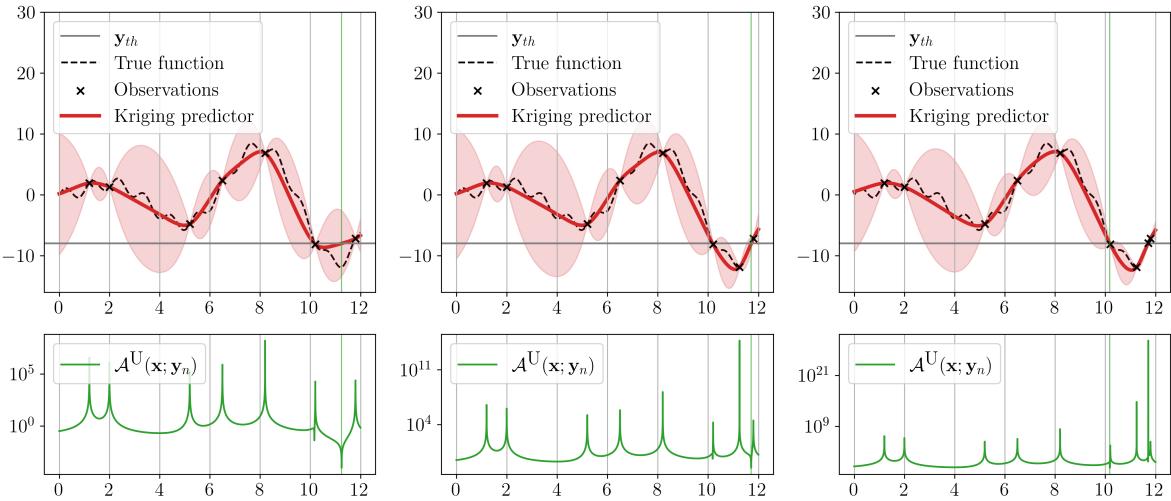


Figure 1.14 Illustration of the deviation number learning criterion

#### 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 surrogate model error.]

## 1.8 Conclusion

Chapter **2**

# Introduction to wind turbine modeling and design

---

2.1	Introduction . . . . .	50
2.2	Metocean conditions simulation . . . . .	50
2.2.1	Turbulent wind generation . . . . .	51
2.2.2	Wake modeling . . . . .	54
2.2.3	Irregular wave generation . . . . .	55
2.3	Wind turbine multi-physics modeling . . . . .	56
2.3.1	Aerodynamics of horizontal axis wind turbines . . . . .	56
2.3.2	Hydrodynamics . . . . .	57
2.3.3	Command and control . . . . .	57
2.3.4	Structural dynamics . . . . .	57
2.3.5	Fatigue damage . . . . .	57
2.4	Design and operation practices . . . . .	57
2.4.1	Wind turbine design and operation . . . . .	57
2.4.2	Further design considerations . . . . .	57
2.4.3	Maintenance and end-of-life management . . . . .	58
2.5	Uncertain inputs . . . . .	58
2.5.1	Environmental inputs . . . . .	58
2.5.2	System inputs . . . . .	58
2.5.3	Probabilistic fatigue assessment . . . . .	58
2.6	Conclusion . . . . .	58

---

## 2.1 Introduction

Wind energy is a highly competitive industry with increasing regulations regarding its impact on ecosystems, land-use conflicts, landscapes, or air traffic management (Beauregard et al., 2022). During the long process to win calls for tenders, obtain construction permits, or throughout the wind farm exploitation, an advanced technical understanding of such systems might offer a competitive advantage.

The operation of offshore wind turbines are driven by multiple physics coupled. This behavior results from different external solicitations which are highly turbulent and uncertain. Among them, the *metocean* (abbreviation of meteorology and oceanography) environmental conditions play an important role. Note that many other types of solicitations also affect the exploitation of offshore wind turbines (e.g., corrosion of the structure, global scour, marine growth, stress concentration factor induced by the manufacturing quality, etc.).

In this context, numerical models have been developed to certify the structural integrity of OWTs with respect to their solicitations. A wind farm project planned at given location should pass different validation procedures established by international standards such as the International Electrotechnical Commission (IEC-61400-1, 2019). As wind turbine structures face a large amount of stress cycles in their lifetime (up to  $10^8$  for 25 years of operation), this chapter will particularly focus on fatigue damage assessment.

This chapter briefly introduces wind turbine modeling and design, in the following layout: Section 2.2 presents the methods used for wind and wave generation and wake simulation at a farm scale; Section 2.3 recalls elements of theory associated with wind turbine modeling; Section 2.4 introduces recommended practices regarding design and operation; finally, Section 2.5 gives a description of the various sources of uncertainties considered in this thesis. Considering the standard uncertainty quantification diagram presented in Fig. 1, the material of this chapter is related to the step A (problem specification) and the step B (uncertainty quantification).

## 2.2 Metocean conditions simulation

In the atmosphere, the wind is the air movements caused by the heterogeneous solar heating of Earth's surface. Winds usually move from high-pressure to low-pressure regions. Earth's rotation also impacts large-scale climate patterns, including winds, according to the well-known Coriolis effect. The wind is a highly variable resource, making its exploitation for energy production uncertain. This variability is both expressed in space and time with different behaviors depending on the scales studied.

Regarding large timescales, yearly seasonal fluctuations of wind conditions are well-defined using probability distributions (typically Weibull distributions). Predictions at a shorter timescale are usually unreliable beyond a few days ahead. Under a few days, the spectral wind energy distribution per time unit is represented by its power spectral density (PSD). Historically, the spectral study of horizontal wind by Van der Hoven (1957) for timescales between a few seconds

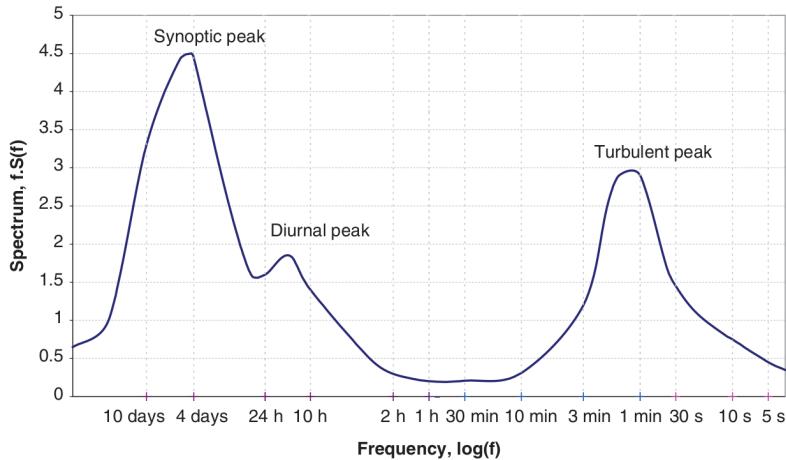


Figure 2.1 Wind spectrum from Brookhaven, USA (source: [Burton et al. \(2021\)](#))

and ten days revealed distinct ranges of behaviors. The PSD, such as the one illustrated in Fig. 2.1, presents three main separated peaks, explaining how the wind energy is split. The two first peaks are named “synoptic” and “diurnal” peak, which respectively correspond to return periods around four days and one day. While these two peaks are relatively close together, the third peak is completely separated. This peak describes the energy related to the wind turbulence, which evolves in the range below ten minutes. Considering this energy distribution, wind behaviors are often referred to as “short-term” (for turbulent wind) and “long-term” (otherwise). In wind turbine simulation, ten minutes simulations became a common practice to fully consider turbulent winds.

Note that the spectrum presented in the research paper of [Van der Hoven \(1957\)](#) was build from wind measures near New York, USA. The same pattern between the three peaks is rather constant between sites, however, the geography (including the surface roughness, the topology, the proximity to the coast, etc.) may affect this distribution. At a larger timescale than one year, assessing trends becomes complicated. Wind resource assessment over decades are made more uncertain by the human-caused climate change [Nagababu et al. \(2023\)](#), which disrupts large weather trends and the occurrence of extreme events.

[Add something general regarding wave modeling?]

### 2.2.1 Turbulent wind generation

The wind turbulence is a complex and aleatory process, often described as chaotic, since a small perturbation of its initial conditions might have an important impact on the response. However, the wind over short-term periods (i.e., ten minutes periods) is usually assumed to be a Gaussian process with constant mean  $\bar{U}$  and standard deviation  $\sigma_U$  ([Burton et al., 2021](#)). Its mean is modeled by the long-term wind conditions (i.e., mean wind speed), often described by a probabilistic model such as a Weibull distribution. Note that this assumption is based on the bimodal wind energy distribution observed in Fig. 2.1, which might vary at some specific locations.

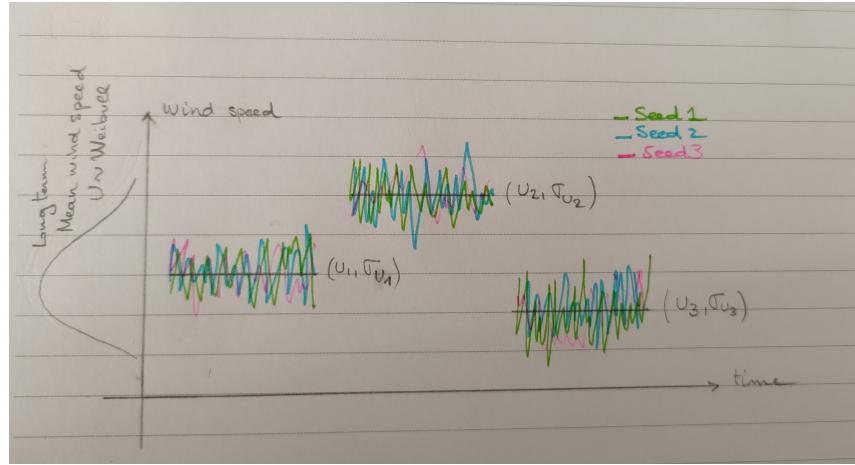


Figure 2.2 [Representation to be confirmed and mentioned in the text]

The *turbulence intensity* is a commonly used normalized statistic of the wind variability:

$$I = \frac{\sigma_U}{\bar{U}}. \quad (2.1)$$

As the wind depends on differences between pressure, humidity, air density, different models exist to represent vertical wind profiles. The vertical change in wind conditions is referred to as *vertical wind shear*. Assuming a constant standard deviation over the altitude, the power law is a widely used approximated shear model ([IEC-61400-1, 2019](#)):

$$\bar{U}(z) = \bar{U}_0 \left( \frac{z}{z_0} \right)^\alpha, \quad (2.2)$$

with  $\bar{U}_0$  a well-defined mean wind speed at the height  $z_0$  (typically corresponding to a measurement height),  $z$  the studied height (e.g., the turbine's hub-height), and  $\alpha$  the vertical shear coefficient (defined according to measures or standards recommendations).

To generate a turbulent wind field on a mesh around the turbine, the general mechanism is to apply inverse Fourier transforms on a turbulent wind spectrum. Two types of parametric spectrums are commonly used in wind energy: the *Kaimal model* ([Kaimal et al., 1972](#)) and the *Mann model* ([Mann, 1998](#)). In this thesis, the Kaimal spectrum as defined in [IEC-61400-1 \(2019\)](#) is used for turbulent wind generation over the Cartesian component  $k \in \{u, v, w\}$ :

$$S_k(f) = \frac{4\sigma_k^2 \frac{L_k}{\bar{U}}}{\left(1 + 6f \frac{L_k}{\bar{U}}\right)^{5/3}}, \quad (2.3)$$

such that  $f$  is the frequency,  $\bar{U}$  is the longitudinal mean speed at hub-height,  $L_k$  are the Kaimal length scales, and  $\sigma_k$  standard deviations (see the complete definition in Annex C of [IEC-61400-1 \(2019\)](#)). Along with the Kaimal wind speed spectrum, a spacial coherence model is usually

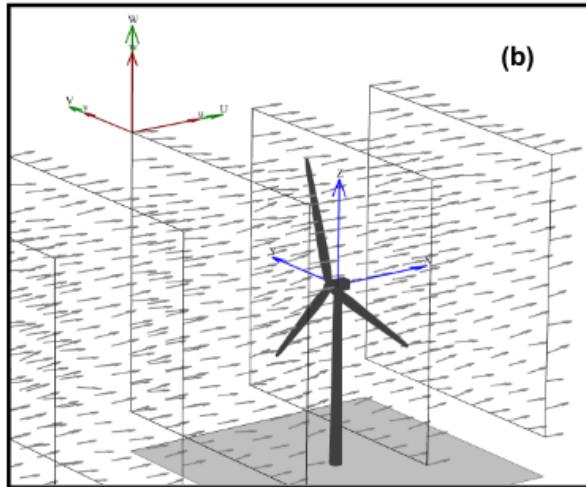


Figure 2.3 Example of a turbulent wind field generated by TurbSim (source: [Jonkman \(2009\)](#))

defined in the frequency domain. Each couple of nodes in the mesh are correlated, for example, using an exponential coherence model (for further detail in Annex C from [IEC-61400-1 \(2019\)](#)).

In this thesis, the full-field turbulent wind fields (i.e., over a regular mesh) are generated using TurbSim, a software developed by the National Renewable Energy Laboratory (NREL) ([Jonkman, 2009](#)). TurbSim generates time realizations by adapting the spectral method proposed in [Veers \(1988\)](#) (relying on the inverse Fourier transforms of each component). Considering a wind spectrum (e.g., Kaimal model) and a vertical shear model (e.g., power law), TurbSim takes as inputs a mean wind speed, a turbulence standard deviation and a mean wind orientation. Fig. 2.3 illustrates the corresponding wind field generated by a ten-minutes TurbSim simulation, considering a set of input long-term conditions.

In their recent review of the challenges in wind energy, ([Veers et al., 2019](#)) list some limits of the two spectral turbulence models recommended by the standards. First, their parameters were fitted using a restricted amount of data ([Dimitrov et al., 2017](#)). Second, the spacial coherence models associated with Kaimal models showed differences with turbulence measured on site ([Saranyasoontorn et al., 2004](#)). Finally, recent studies showed that the choice of spectral model impacts the resulting wind turbines loads ([Doubrawa et al., 2019](#)). These approximations generally tend to overestimate wind flows, leading to conservative designs.

To ensure the most realistic turbulent wind field generation, two research perspectives are actively explored. Authors recently developed hybrid methods, including measurement data to enhance spectral models ([Dimitrov and Natarajan, 2017](#)). Alternatively, higher fidelity models were studied in this domain, see for example the use of vortex methods ([Branlard, 2017](#)) and large eddy simulations (LES) ([Bui and Bakhoday-Paskyabi, 2022; Doubrawa et al., 2019](#)). Such complex models allow the simulation of mesoscale conditions (e.g., at the farm scale), and extreme transient events (e.g., gusts and storms). However, their computational cost is often prohibitive in uncertainty quantification studies. When studying the wind resources at a wind farm scale, modeling wind energy losses induced by the turbines' wake becomes essential.

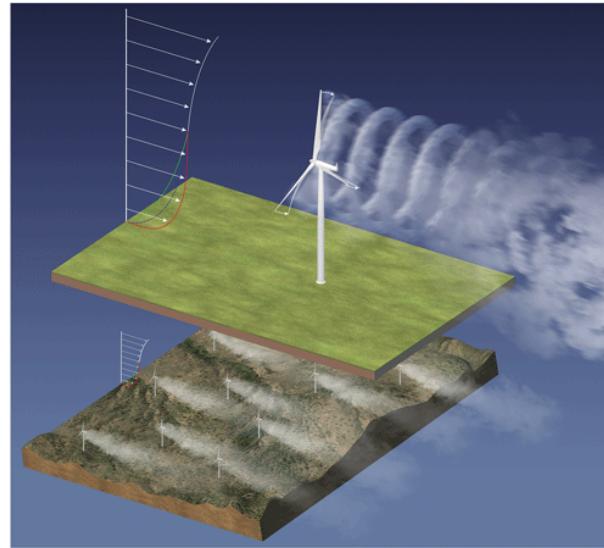


Figure 2.4 Illustration of the wake created downstream a wind farm (source: [Veers et al. \(2019\)](#))

## 2.2.2 Wake modeling

The wake is caused by the wind energy extraction, reducing the wind speed and increasing the turbulence downstream of the turbines (see the illustration in Fig. 2.4). In a wind farm, this effect depends on the spacing between turbines, as well as the ambient wind speed and turbulence intensity. The turbines positioned at the center of the farm are the most impacted by the wake. As a wind farm owner, the consequence of the wake is twofold: a loss of energy production (in the range of 10 to 20 percents depending on the farm), and an increase of fatigue loads (due to the asymmetric loading from the created turbulences).

The initiation of the wake is a complex physical mechanism, however, the wake almost becomes axisymmetric after two turbine diameters downstream. At this stage, the wind speed deficit often presents a Gaussian profile centered on the hub ([Burton et al., 2021](#)). Numerical models of different fidelities aim at simulating the wake. For example, computational fluid dynamics (CFD) models give a detailed description of the wake (including near the turbine) but require high computational efforts. In practice, simple analytical models (often called “engineering models”) are widely used and recommended by standards (see e.g., Annex E in [IEC-61400-1 \(2019\)](#)). These models mostly rely the equivalence between the thrust load on the turbine wind energy deficit. Since the seminal engineering model proposed by [Jensen \(1983\)](#), multiple enhancements were proposed. A wide benchmark of the wake modeling solutions for different fidelities was performed in [Doubrawa et al. \(2020\)](#) and [Ardillon et al. \(2023\)](#). The optimal tuning of these engineering models was studied using measurements from a Doppler wind lidar in [Zhan et al. \(2020\)](#). Different software programs propose wake engineering models, such as: FLORIS (developed by the NREL [Fleming et al. \(2020\)](#)), FarmShadow (developed by IFPEN).

To take into account the wake effect, control strategies increasingly move from the turbine scale to the farm scale. This concept, called “active wake control”, introduces small yaw

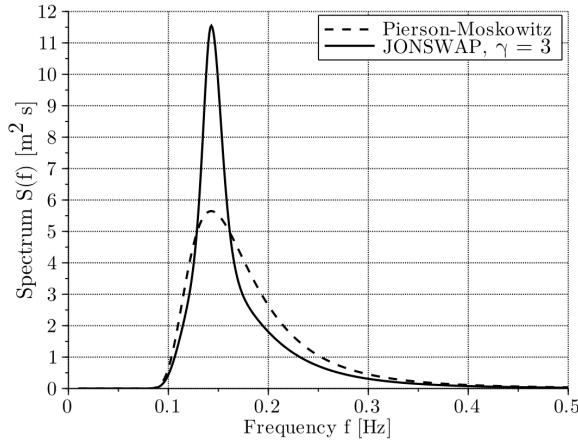


Figure 2.5 Peirson-Moskowitz and JONSWAP spectra at significant wave height  $H_s = 3$  m and peak period  $T_p = 7$  s (source: [Milano \(2021\)](#))

misalignments (making the control of turbines individually suboptimal) to optimize the global wake inside the farm ([Rott et al., 2018](#); [Simley et al., 2020](#)).

### 2.2.3 Irregular wave generation

The propagation of wind generated waves has long been studied in hydrodynamics, leading to various wave theories including Airy's, and Stokes'. Airy wave theory (also referred to as linear wave theory), models sea states under the hypothesis of small waves relatively to the water depth. This spectral approach superposes many regular waves, following the same wave spectrum, to model irregular waves. Standard statistics are used in oceanography to represent sea states and their corresponding wave spectra: the wave period  $T_p$  (with respective frequency  $f_p$ ), and the significant wave height  $H_s$  (average over the highest third of the waves measured).

The most commonly used parametric wave spectrum is called JONSWAP, after the Joint North Sea Wave Project ([Hasselmann et al., 1973](#)):

$$S(f) = \delta \frac{H_s^2}{f} \left( \frac{f_p}{f} \right)^4 \exp \left[ -\frac{5}{4} \left( \frac{f_p}{f} \right)^4 \right] \gamma^\alpha. \quad (2.4)$$

The JONSWAP spectrum is a correction of the Pierson-Moskowitz spectrum, adding the peak enhancement factor  $\gamma^\alpha$ . Further details regarding the numerical values to choose in Eq. (2.4) are given in [Burton et al. \(2021\)](#). An illustration of the two spectra is presented in Fig. 2.5, revealing the artificial enhancement factor proposed in the JONSWAP model to better fit sea states measurements.

[Add elements from E.Petrovska p.13 regarding bimodal spectra]

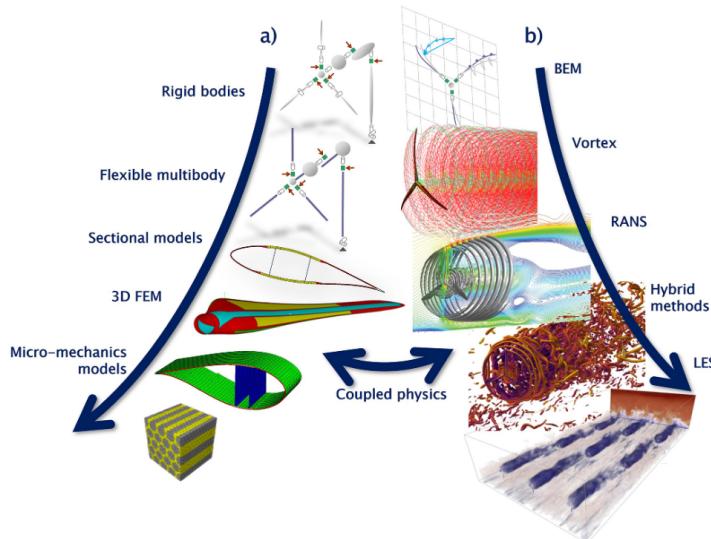


Figure 2.6 Hierarchy of structural (a) and aerodynamic (b) wind energy systems models (source: Veers et al. (2019))

## 2.3 Wind turbine multi-physics modeling

Offshore wind turbine models are coupling multiple physics such as aerodynamics, hydrodynamics, mechanical elasticity, control and mooring dynamics for floating OWT. Similarly to the usual practices from the offshore oil & gas industry, OWT have been first modeled in the frequency domain. At an early design stage, a study in the frequency domain gives a rough idea of the system's feasibility by computing its natural frequencies. An OWT should not have its natural frequencies in the same range as the main frequencies of the wave energy spectra. Otherwise, such systems can be subject to critical dynamic resonance, leading to their failure.

Beyond this preliminary check, frequency-domain approaches present limits for OWT modeling. As they rely on linear assumptions, they are unable to model the non-linearities and transient loading phases (Matha et al., 2011). These aspects happen to be essential in the design of OWT (Cordle and Jonkman, 2011). As an alternative, the behavior of OWT systems are also simulated in the time domain.

In the time domain, such systems may be models at different fidelities. The diagram in Fig. 2.6, illustrates the increasing complexities for two physics involved in OWT modeling (aerodynamics and structural dynamics). To perform an uncertainty quantification around a wind turbine model, its fidelity is preferably low. At the wind turbine scale, the numerical model studied in this work is actually a chain of three models executed sequentially. Note that the wake should also be considered at the wind farm scale, as described earlier.

### 2.3.1 Aerodynamics of horizontal axis wind turbines

The structural response...

### 2.3.2 Hydrodynamics

### 2.3.3 Command and control

### 2.3.4 Structural dynamics

### 2.3.5 Fatigue damage

## 2.4 Design and operation practices

Generally speaking, the energy available in the wind is proportional to the cube of the wind speed, given the well-known power production equation:

$$P = \frac{1}{2} C_p \rho A U^3, \quad (2.5)$$

where [continue the description].

### 2.4.1 Wind turbine design and operation

- Different parts of the turbine (monopile, transition piece, fundation, tower), different turbine designs
- What are the key steps for a wind farm project in France
- Pre-installation measures are usually done for a period of 1 to 3 year. This data is often validated against long-term regional datasets [Sempreviva et al. \(2008\)](#). Note that floating lidars present an opportunity [add ref]
- general design from IEC61400 with part 1 and part 3
- Different designs : soft-stiff, stiff-stiff
- Design load cases
- How does it work : different ranges of function depending on the wind speed (cut-in, cut-off)
- 

### 2.4.2 Further design considerations

- Soil modeling
- Marine growth
- Global scour
- Port logistics
- Grid impact
- Environmental impact and social acceptance
- Manufacturing quality inducing stress concentration factor

### 2.4.3 Maintenance and end-of-life management

- Operation and management
- Repowering vs. revamping
- Recycling

## 2.5 Uncertain inputs

[Ref: Floating lidar as an advanced offshore wind speed measurement technique: current technology status and gap analysis in regard to full maturity]

### 2.5.1 Environmental inputs

### 2.5.2 System inputs

### 2.5.3 Probabilistic fatigue assessment

## 2.6 Conclusion

# Bibliography

- Abdo, T. and Rackwitz, R. (1991). A new beta-point algorithm for large time-invariant and time-variant reliability problems. In *Reliability and Optimization of Structural Systems' 90: Proceedings of the 3rd IFIP WG 7.5 Conference*, pages 1–12.
- 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.
- Ardillon, E., Paskyabi, M., Cousin, A., Dimitrov, N., Dupoirion, M., Eldevik, S., Fekhari, E., Ferreira, C., Guiton, M., Jezequel, B., et al. (2023). Turbine loading and wake model uncertainty. Technical report, European Union.
- 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.
- 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.
- Beauregard, E., Bérille, E., Berrabah, N., Berthelot, M., Burrows, J., Capaldo, M., Cornet, S., Costan, V., Duchet, M., Dufossé, E., Dupont, E., Franchet, M., Gouze, E., Grau, A., Joly, A., Kell, N., de Laleu, V., Latraube, F., Lovera, A., de Bazelaire, A., Monnot, E., Nogaro, G., Pagot, J., Pérony, R., Peyrard, C., Piguet, C., Régnier, A., Santibanez, E., Senn, C., Smith, C., Soriano, F., Stephan, P., Terte, N., Veyan, P., Vizireanu, D., and Yeow, L. (2022). *L'éolien en mer : un défi pour la transition énergétique*. Lavoisier.
- Benoumechiara, N. and Elie-Dit-Cosaque, K. (2019). Shapley effects for sensitivity analysis with dependent inputs: bootstrap and kriging-based algorithms. *ESAIM: Proceedings and Surveys*, 65:266–293.
- 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.
- Branlard, E. (2017). *Wind turbine aerodynamics and vorticity-based methods: Fundamentals and recent applications*. Springer.
- Briol, F., Oates, C., Girolami, M., Osborne, M., and Sejdinovic, D. (2019). Probabilistic Integration: A Role in Statistical Computation? *Statistical Science*, 34:1 – 22.
- Bui, H. and Bakhoday-Paskyabi, M. (2022). Mesoscale Simulation of Open Cellular Convection: Roles of Model Resolutions and Physics Parameterizations. In *Journal of Physics: Conference Series*, volume 2362, page 012006.
- Burton, T., Jenkins, N., Bossanyi, E., Sharpe, D., and Graham, M. (2021). *Wind energy handbook*. John Wiley & Sons.
- Chabridon, V. (2018). *Reliability-oriented sensitivity analysis under probabilistic model uncertainty—Application to aerospace systems*. PhD thesis, Université Clermont Auvergne.
- 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, 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.
- Cordle, A. and Jonkman, J. (2011). State of the art in floating wind turbine design tools. In *ISOPE International Ocean and Polar Engineering Conference*.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.
- Cousin, A. (2021). *Optimisation sous contraintes probabilistes d'un système complexe : Application au dimensionnement d'une éolienne offshore flottante*. PhD thesis, Institut Polytechnique de Paris.
- 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 Dakessian, T. (1998). Multiple design points in first and second-order reliability. *Structural Safety*, 20:37–49.
- Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112.
- Dimitrov, N. and Natarajan, A. (2017). Application of simulated lidar scanning patterns to constrained Gaussian turbulence fields for load validation. *Wind Energy*, 20(1):79–95.

- Dimitrov, N., Natarajan, A., and Mann, J. (2017). Effects of normal and extreme turbulence spectral parameters on wind turbine loads. *Renewable Energy*, 101:1180–1193.
- DNV-GL (2016). DNVGL-ST-0437: Loads and site conditions for wind turbines. Technical report, DNVGL.
- Doubrawa, P., Churchfield, M., Godvik, M., and Sirnivas, S. (2019). Load response of a floating wind turbine to turbulent atmospheric flow. *Applied Energy*, 242:1588–1599.
- Doubrawa, P., Quon, E., Martinez-Tossas, L., Shaler, K., Debnath, M., Hamilton, N., Herges, T., Maniaci, D., Kelley, C., Hsieh, A., et al. (2020). Multimodel validation of single wakes in neutral and stratified atmospheric conditions. *Wind Energy*, 23(11):2027–2055.
- Echard, B., Gayton, N., and Lemaire, M. (2011). AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, 33:145–154.
- Fang, K., Liu, M.-Q., Qin, H., and Zhou, Y.-D. (2018). *Theory and application of uniform experimental designs*, volume 221. Springer.
- Fleming, P., King, J., Bay, C., Simley, E., Mudafort, R., Hamilton, N., Farrell, A., and Martinez-Tossas, L. (2020). Overview of FLORIS updates. In *Journal of Physics: Conference Series*, volume 1618, page 022028.
- Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.
- 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.
- Gramacy, R. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. CRC press.
- Grasedyck, L., Kressner, D., and Tobler, C. (2013). A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78.
- 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.
- Hasselmann, K., Barnett, T., Bouws, E., Carlson, H., Cartwright, D., Enke, K., Ewing, J., Gienapp, A., Hasselmann, D., Kruseman, P., et al. (1973). Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). *Ergaenzungsheft zur Deutschen Hydrographischen Zeitschrift, Reihe A*.
- 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.
- Hirvoas, A. (2021). *Development of a data assimilation method for the calibration and continuous update of wind turbines digital twins*. PhD thesis, Université Grenoble Alpes.

- Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Annals of Mathematical Statistics*, 19(3):293–325.
- Huchet, Q. (2019). *Kriging based methods for the structural damage assessment of offshore wind turbines*. PhD thesis, Université Blaise Pascal.
- IEC-61400-1 (2019). IEC 61400-1: Wind energy generation systems - Part 1: Design requirements. Technical report, International Electrotechnical Commission (IEC).
- Jensen, N. (1983). A note on wind generator interaction. Technical report, DTU.
- Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Chapman and Hall.
- Joe, H. (2014). *Dependence modeling with copulas*. CRC press.
- Joe, H. and Kurowicka, D. (2011). *Dependence modeling: vine copula handbook*. World Scientific.
- Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492.
- 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.
- 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.
- 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. and Iooss, B. (2017). *Derivative-Based Global Sensitivity Measures*, pages 1241–1263. Springer International Publishing, Cham.
- Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C., and Shah, N. (2009). Monte Carlo evaluation of derivative-based global sensitivity measures. *Reliability Engineering & System Safety*, 94:1135–1148. Special Issue on Sensitivity Analysis.
- Lamboni, M., Iooss, B., Popelin, A.-L., and Gamboa, F. (2013). Derivative-based global sensitivity measures: General links with Sobol’ indices and numerical tests. *Mathematics and Computers in Simulation*, 87:45–54.
- Larsen, G. C., Madsen, H., Thomsen, K., and Larsen, T. (2008). Wake meandering: a pragmatic approach. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 11(4):377–395.
- 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.
- Le Riche, R. and Picheny, V. (2021). Revisiting Bayesian optimization in the light of the COCO benchmark. *Structural and Multidisciplinary Optimization*, 64(5):3063–3087.
- Lebrun, R. (2013). *Contributions à la modélisation de la dépendance stochastique*. PhD thesis, Université Paris-Diderot – Paris VII. (in English).
- 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, 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.
- Mann, J. (1998). Wind field simulation. *Probabilistic engineering mechanics*, 13(4):269–282.
- Mara, T. A. and Tarantola, S. (2012). Variance-based sensitivity indices for models with dependent inputs. *Reliability Engineering & System Safety*, 107:115–121.
- 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., Laurent, B., and Roustant, O. (2009). Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751.
- Matha, D., Schlipf, M., Pereira, R., and Jonkman, J. (2011). Challenges in simulation of aerodynamics, hydrodynamics, and mooring-line dynamics of floating offshore wind turbines. In *ISOPE International Ocean and Polar Engineering Conference*.
- 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.
- Milano, D. (2021). *Numerical prototype for floating offshore wind turbines*. PhD thesis, The University of Edinburgh.
- 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.
- Morris, M. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33:161–174.

- Moustapha, M., Marelli, S., and Sudret, B. (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety*, 96:102174.
- Nagababu, G., Srinivas, B., Kachhwaha, S., Puppala, H., and Kumar, S. (2023). Can offshore wind energy help to attain carbon neutrality amid climate change? A GIS-MCDM based analysis to unravel the facts using CORDEX-SA. *Renewable Energy*, page 119400.
- Oates, C. J. (2021). Minimum Discrepancy Methods in Uncertainty Quantification. Lecture Notes at École Thématique sur les Incertitudes en Calcul Scientifique (ETICS21), <https://www.gdr-mascotnum.fr/etics.html>.
- Oberkampf, W. and Roy, C. (2010). *Verification and validation in scientific computing*. Cambridge university press.
- Owen, A. (2003). The dimension distribution and quadrature test functions. *Statistica Sinica*, 13:1–17.
- Owen, A. (2013). *Monte Carlo theory, methods and examples*.
- Owen, A. (2014). Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251.
- Papaioannou, I., Betz, W., Zwirglmaier, K., and Straub, D. (2015). MCMC algorithms for Subset Simulation. *Probabilistic Engineering Mechanics*, 41:89–103.
- Petit, S. (2022). *Improved Gaussian process modeling : Application to Bayesian optimization*. PhD thesis, Université Paris-Saclay.
- Petrovska, E. (2022). *Fatigue life reassessment of monopile-supported offshore wind turbine structures*. PhD thesis, University of Edinburgh.
- Powell, M. (1994). *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer.
- Pronzato, L. and Müller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701.
- Pronzato, L. and Zhigljavsky, A. (2020). Bayesian quadrature and energy minimization for space-filling design. *SIAM/ASA Journal on Uncertainty Quantification*, 8:959 – 1011.
- Qiu, J., Wu, Q., Ding, G., Xu, Y., and Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016:1–16.
- Rahman, S. (2016). The f-sensitivity index. *SIAM/ASA journal on uncertainty quantification*, 4(1):130–162.
- Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. Springer.
- Rockafellar, R. and Royset, J. (2015). Engineering Decisions under Risk Averseness. *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.
- Rott, A., Doekemeijer, B., Seifert, J., van Wingerden, J., and Kühn, M. (2018). Robust active wake control in consideration of wind direction variability and uncertainty. *Wind energy science*, 3(2):869–882.

- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.
- 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.
- Saranyasoontorn, K., Manuel, L., and Veers, P. (2004). A Comparison of Standard Coherence Models for Inflow Turbulence With Estimates from Field Measurements. *Journal of Solar Energy Engineering*, 126(4):1069–1082.
- 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.
- Sempreviva, A., Barthelmie, R., and Pryor, S. (2008). Review of methodologies for offshore wind resource assessment in European seas. *Surveys in Geophysics*, 29:471–497.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.
- Shang, B. and Apley, D. (2020). Fully-sequential space-filling design algorithms for computer experiments. *Journal of Quality Technology*, 53:1 – 24.
- Shapley, L. S. et al. (1953). A value for n-person games.
- Simley, E., Fleming, P., and King, J. (2020). Design and analysis of a wake steering controller with wind direction variability. *Wind Energy Science*, 5(2):451–468.
- Sobol̄, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407.
- Sobol, I. and Gresham, A. (1995). On an alternative global sensitivity estimators. *Proceedings of SAMO*, pages 40–42.
- 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.
- Song, E., Nelson, B. L., and Staum, J. (2016). Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083.
- 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.
- Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979.
- 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.
- Teixeira, R., Nogal, M., and O'Connor, A. (2021). Adaptive approaches in metamodel-based reliability analysis: A review. *Structural Safety*, 89:102019.

- Trefethen, L. (2008). Is Gauss quadrature better than Clenshaw–Curtis? *SIAM review*, 50(1):67–87.
- Van der Hoven, I. (1957). Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour. *Journal of Atmospheric Sciences*, 14(2):160–164.
- Van Kuik, G., Peinke, J., Nijssen, R., Lekou, D., Mann, J., Sørensen, J., Ferreira, C., van Wingerden, J., Schlipf, D., Gebraad, P., et al. (2016). Long-term research challenges in wind energy—a research agenda by the European Academy of Wind Energy. *Wind energy science*, 1(1):1–39.
- Veers, P. (1988). Three-dimensional wind simulation. Technical report, Sandia National Labs., Albuquerque, NM (USA).
- Veers, P., Dykes, K., Lantz, E., Barth, S., Bottasso, C., Carlson, O., Clifton, A., Green, J., Green, P., Holttinen, H., et al. (2019). Grand challenges in the science of wind energy. *Science*, 366(6464):eaau2027.
- Warnock, T. (1972). Computational investigations of low-discrepancy point sets. In *Applications of number theory to numerical analysis*, pages 319–343. Elsevier.
- Zhan, L., Letizia, S., and Iungo, G. (2020). Optimal tuning of engineering wake models through lidar measurements. *Wind Energy Science*, 5(4):1601–1622.

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:

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

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  $\theta$  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  $\theta$  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

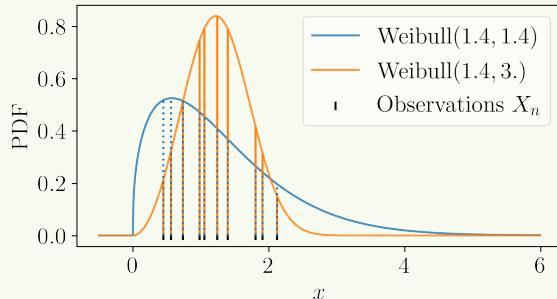


Figure A.1 Adequation of two different Weibull models using their likelihood with a sample of observations (black crosses).

## 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.

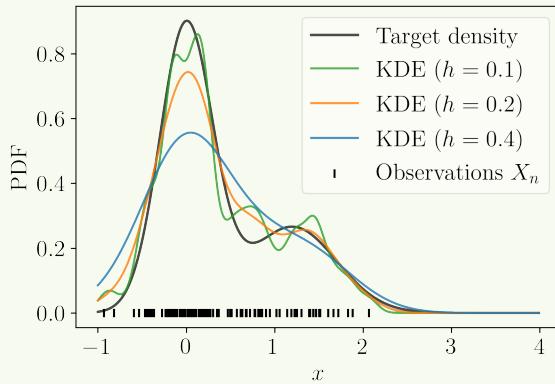


Figure A.2 Fit of a bimodal density by KDE using different tuning parameters.

## 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.

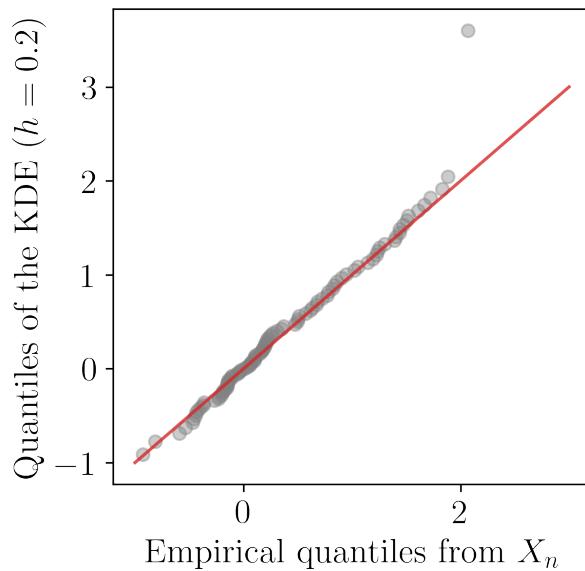


Figure A.3 QQ-plot between the data from Example 2 and a KDE model.

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  $m_{\text{IMSE}}$  for different dimensions and sample sizes (see Fig. B.1). In high dimension, the values of  $m_{\text{IMSE}}$  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]

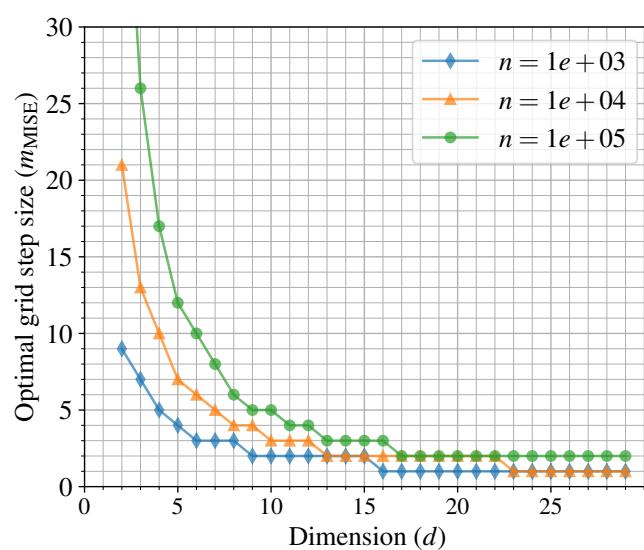


Figure B.1 Evolution of  $m_{\text{IMSE}}$  for different dimensions and sample sizes.



Appendix C

## Copulogram



# Appendix D

## Dissimilarity measures between probability distributions

Beyond the discrepancy measure to the uniform distribution, this section introduces different dissimilarity measures between probability distributions.

### D.1 Csizár $f$ -divergences

[General definition]

[Numerous examples depending on the function chosen: see the book culte]

[Link between KL and mutual information] [Further inputs in the review from Rahman, maybe some in the PhD subject from A.Dutfoy.] [Problems generated in the estimation]

### D.2 Integral probability metrics

[general definition]

[Numerous examples see the book culte]

[No closed form expression unlike the  $f$ -divergence but the use of RKHS goes around this issue.]

### Maximum discrepancy measure

A metric of discrepancy between distributions is introduced as the *maximum mean discrepancy* (MMD). This distance between two probability distributions  $\mu$  and  $\zeta$  is defined as the worst-case error for any function within a unit ball of a function space  $\mathcal{H}$ :

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

To ease the calculation of the quantity, this metric was studied for a particular function space, offering specific properties. A *reproducing kernel Hilbert space* (RKHS), denoted  $\mathcal{H}(k)$ , is an inner product space  $\mathcal{H}(k)$  of functions  $g : \mathcal{D}_X \rightarrow \mathbb{R}$ . It verifies the following axioms, 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”:

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

Every positive semi-definite kernel defines a unique RKHS (and vice versa) with a feature map  $\phi$ , such that  $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}(k)}$ . 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  $\mathcal{D}_X$ .

Then, a probability measure has a representation in the RKHS through its *kernel mean embedding* ([Sejdinovic et al., 2013](#)), also called “potential”  $P_\mu(\mathbf{x})$  in [Pronzato and Zhigljavsky \(2020\)](#), defined as:

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

The reproducing property from the RKHS allows to express the squared MMD as expectations of kernels:

$$\text{MMD}_k(\mu, \zeta)^2 = \int_{\mathcal{D}_X} P_\mu(\mathbf{x}) d\mu(\mathbf{x}) - 2 \int_{\mathcal{D}_X} P_\mu(\mathbf{x}) d\zeta(\mathbf{x}) + \int_{\mathcal{D}_X} P_\zeta(\mathbf{x}) d\zeta(\mathbf{x}). \quad (\text{D.3})$$

[Add a sentence on estimation]

# Appendix E

## Advanced rare event estimation algorithms

**E.1 Subset simulation (SS)**

**E.2 Nonparametric adaptive importance sampling (NAIS)**

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



# Appendix F

## Uncertainty quantification practice 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 minimalistic 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 minimalistic 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()
```

# Appendix G

## Résumé étendu de la thèse

### G.1 Introduction

#### Contexte industriel

L'enjeu actuel de la transition énergétique implique, entre autres, de réduire la part des énergies fossiles au sein du mix électrique mondial. Dans ce contexte, l'énergie éolienne en mer présente plusieurs avantages [Beauregard et al. \(2022\)](#). L'éolien en mer bénéficie notamment de vents plus constants que l'éolien terrestre, notamment dû à l'absence de relief, et offre la possibilité d'installer des éoliennes plus grandes donc plus puissantes. Depuis l'installation de la première ferme éolienne en mer à Vindeby, au Danemark, en 1991, l'industrie a connu une croissance rapide, avec une capacité totale de 56 GW exploitée dans le monde en 2021. Au fil du temps, la technologie éolienne en mer s'est améliorée, aboutissant à des succès importants tels que la signature de projets non subventionnés en Europe (en anglais *zero-subsidy bids*), pour lesquels l'électricité produite est directement vendue sur le marché de gros [Beauregard et al. \(2022\)](#).

Cependant, malgré les progrès techniques indéniables, des limites industrielles émergent vis-à-vis de ces parcs éoliens en mer, posant ainsi de nombreux défis scientifiques. Pour atteindre les ambitieux objectifs de développement au niveau national et régional, la filière de l'éolien en mer fait face à plusieurs problèmes liés à l'augmentation de la taille des turbines. Ce changement d'échelle crée notamment des tensions liées à la logistique portuaire, aux besoins en ressources primaires et à la gestion durable du démantèlement futur. Ce secteur présente plusieurs défis techniques et scientifiques, qui requièrent l'utilisation conjointe de données mesurées et de simulations numériques d'éoliennes dans leur environnement. La recherche appliquée à l'éolien en mer fait intervenir plusieurs disciplines qui étudient notamment des sujets tels que la conception d'éoliennes flottantes, l'amélioration de l'estimation des ressources éoliennes, l'optimisation des opérations de maintenance et l'augmentation de la durée de vie utile des parcs. De manière générale, plusieurs décisions sont prises durant la vie d'une éolienne par son concepteur, installateur et exploitant, tout en ayant une connaissance partielle de certains phénomènes physiques. Par conséquent, modéliser et maîtriser les diverses sources

d'incertitudes associées à l'éolien en mer s'avère être un élément déterminant dans une industrie hautement concurrentielle.

Dans l'ensemble, l'industrie de l'éolien en mer a besoin de méthodes de traitement des incertitudes pour maîtriser les marges de sûreté et la gestion des actifs industriels (à la maille des composants, de l'éolienne et du parc dans son ensemble) [Van Kuik et al. \(2016\)](#). Pour un développeur de projets éoliens, l'attention est d'abord portée sur l'amélioration du potentiel éolien des sites candidats en combinant différentes sources d'information et en modélisant la distribution multivariée des conditions environnementales au sein d'un parc éolien. Dans le cas de projets en éolien flottant, l'objectif est d'intégrer un aspect probabiliste dès la phase de conception (par exemple, du flotteur) afin de définir des solutions plus sûres, plus robustes et plus rentables. Pour un propriétaire d'un parc éolien, la gestion de la fin de vie est une autre problématique importante. Un propriétaire de parc éolien en fin de vie a le choix entre trois options : prolonger la durée de vie des actifs en exploitation, remplacer les éoliennes actuelles par des modèles plus récents, ou démanteler et vendre le parc éolien. Les deux premières solutions nécessitent d'évaluer la fiabilité de la structure et sa durée de vie résiduelle. Ces évaluations quantitatives sont examinées par des organismes de certification et des assureurs pour délivrer des permis d'exploitation. Pour fournir des évaluations rigoureuses des risques, la méthodologie générique de *traitement des incertitudes* est une démarche qui fait consensus dans les secteurs industriels confrontés à ce genre de problématique [de Rocquigny et al. \(2008\)](#).

## Méthodologie générique de traitement des incertitudes dans les outils de calcul scientifiques

La simulation numérique est une discipline qui a émergé avec l'avènement de l'informatique. Cette pratique produit des outils de calcul scientifique (OCS) qui permettent de simuler le comportement de système complexes compte tenu de conditions initiales définies par l'analyste. Les OCS sont vite devenus indispensables pour l'analyse, la conception, et la certification de systèmes complexes dans les cas où des expériences ou des mesures physiques sont coûteuses à obtenir, voire impossibles à réaliser. Cependant, ces modèles numériques s'intègrent dans une démarche déterministe : le résultat d'une simulation est associé à un vecteur de paramètres fixé en entrée. La question de la gestion des incertitudes associées aux entrées se pose rapidement lors de l'utilisation des OCS.

Le traitement des incertitudes vise à modéliser et à traiter les incertitudes autour d'un modèle numérique. Pour ce faire, une méthodologie générique a été proposée pour quantifier et analyser les incertitudes entre les variables d'entrée et de sortie d'un OCS [de Rocquigny et al. \(2008\)](#). Une présentation des outils mathématiques utilisés dans ce domaine est proposée par [Sullivan \(2015\)](#). Cette approche apporte une meilleure compréhension d'un système, ce qui contribue à une prise de décision plus robuste.

La Figure 1 illustre les étapes génériques de la méthodologie de quantification des incertitudes, qui sont brièvement décrites ci-après :

- **Étape A – Spécification du problème.** Cette étape consiste à déterminer le système étudié et construire un modèle numérique capable de simuler (précisément) son comportement. La spécification du problème implique également de définir l'ensemble des paramètres inhérents au modèle numérique. Ces paramètres comprennent aussi bien les variables d'entrée que les variables de sortie générées par la simulation. Dans ce document, le modèle numérique est considéré comme une boîte-noire, par opposition à des approches qui s'intègrent à l'intérieur des schémas de résolution numérique des équations de comportement du système (approches dites intrusives [Le Maître and Knio \(2010\)](#)). En général, ces modèles numériques sont au préalable calibrés par rapport à des données mesurées et suivent un processus de validation et de vérification pour réduire les erreurs de modélisation [Oberkampf and Roy \(2010\)](#).
- **Étape B – Modélisation et quantification des incertitudes.** L'objectif de la deuxième étape est d'identifier et modéliser toutes les sources d'incertitude associées aux variables d'entrée. Dans la plupart des cas, cette modélisation est effectuée dans un cadre probabiliste.
- **Étape C – Propagation des incertitudes.** Lors de cette étape, les entrées incertaines sont propagées au travers du modèle de simulation numérique. Dès lors, la sortie du modèle numérique (habituellement de type scalaire) devient également incertaine. L'objectif est alors d'estimer une quantité d'intérêt, c'est-à-dire une statistique sur la variable aléatoire de sortie étudiée. La méthode de propagation de l'incertitude peut différer en fonction de la quantité d'intérêt visée (par exemple, la tendance centrale, un quantile, une probabilité d'événement rare, etc.).
- **Étape C' – Analyse de sensibilité.** En complément de la propagation d'incertitudes, une analyse de sensibilité peut être réalisée afin d'étudier le rôle attribué à chaque entrée incertaine dans la variabilité de la sortie d'intérêt.
- **Métamodélisation.** Compte tenu du coût de calcul élevé que représentent certaines simulations, des approches statistiques visent à émuler ces simulateurs coûteux partir d'un nombre limité de simulations. La quantification de l'incertitude peut alors être réalisée avec le modèle statique de substitution (ou métamodèle) pour un moindre coût de calcul. Cette étape optionnelle d'apprentissage statistique ne fait pas à proprement dit partie du traitement des incertitudes mais elle s'avère souvent essentielle pour permettre sa mise en oeuvre pratique.

## Verrous scientifiques et objectifs de la thèse

La maîtrise des risques et des incertitudes dans l'éolien est un enjeu majeur pour le groupe EDF en tant qu'exploitant. Cette thèse vise à adapter et appliquer, sur un cas d'usage issu de l'éolien en mer, une démarche globale de traitement des incertitudes. Ainsi, ce cas d'usage soulève des verrous scientifiques associés à ses particularités qui peuvent être décrites comme suit :

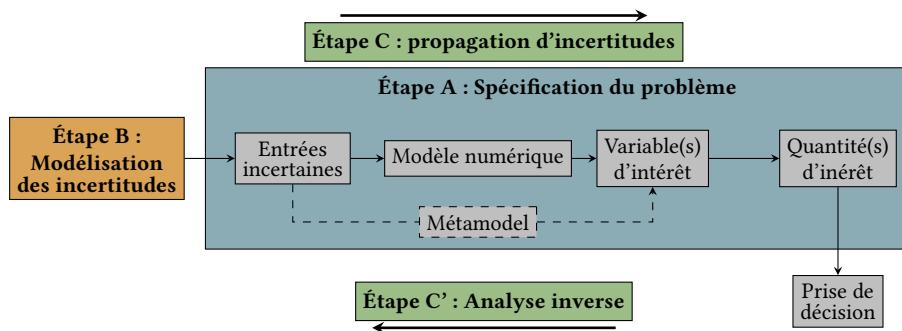


FIGURE G.1 Schéma générique de la quantification des incertitudes (de Rocquigny et al. (2008), adapté par Ajenjo (2023))

- Le code de simulation numérique autour duquel les travaux sont réalisés est constitué d'une chaîne de codes de calcul, exécutés en série. Cette chaîne s'articule en trois étapes : d'abord une génération temporelle et stochastique d'un champ de vitesse de vent et de houle, puis la simulation du comportement hydro-aéro-servo-élastique de l'éolienne et enfin une phase d'agrégation des résultats temporels pour obtenir des quantités d'intérêt scalaires ;
- La complexité de cet outil de calcul scientifique ainsi que le coût de calcul unitaire élevé (de l'ordre de 20 minutes par simulation) nécessite l'utilisation de méthodes d'échantillonnage performantes, ainsi que des systèmes de calcul haute performance. En plus de la complexité liée au modèle numérique, la modélisation des incertitudes en entrée présente elle aussi des difficultés. En effet, la loi conjointe des conditions environnementales liées à un site comporte une structure de dépendance complexe à capturer et à modéliser. L'étape d'inférence vis-à-vis des grandes quantités de données mesurées est d'autant plus importante que sa qualité impacte directement les conclusions de la propagation d'incertitudes.

Afin d'appliquer le schéma global de traitement des incertitudes au cas éolien, cette thèse vise à répondre aux problématiques suivantes :

- Q1.** *Comment précisément modéliser la structure de dépendance complexe associée aux lois conjointes de conditions environnementales ?* (⇒ Étape B)
- Q2.** *Comment réaliser une propagation d'incertitudes au travers d'une chaîne de simulation numérique coûteuse, uniquement basée sur une description empirique (données mesurées) des incertitudes en entrée ?* (⇒ Étape C)
- Q3.** *Comment estimer des probabilités d'événements rares associées à la ruine de structures éoliennes en mer ?* (⇒ Étape C)
- Q4.** *Comment évaluer et interpréter la sensibilité des entrées incertaines vis-à-vis des quantités d'intérêt liées à la fiabilité des structures (analyse de sensibilité fiabiliste) ?* (⇒ Étape C')

Les sections suivantes résument les travaux de thèse, tout en respectant la structure du manuscrit.

## **G.2 Résumés des chapitres relatifs à l'état de l'art des méthodes et outils mis en œuvre dans la thèse**

Les deux premiers chapitres relaterons l'état de l'art dans le domaine du traitement des incertitudes et de la modélisation numérique des systèmes éoliens.

### **Chapitre 1 – Traitement des incertitudes en simulation numérique**

Ce chapitre vise à présenter un état de l'art concis des différentes thématiques en quantification des incertitudes [Sullivan \(2015\)](#). Après un rappel de quelques prérequis mathématiques, l'étape de spécification du modèle numérique (considéré comme étant une boîte-noire), ainsi que les variables d'entrée et de sortie est détaillée. Les différents types et sources d'incertitudes sont ensuite présentés, ainsi que leur modélisation dans un cadre probabiliste. La propagation des incertitudes dépend de la nature des quantités d'intérêt estimées, ainsi, une section aborde les méthodes de propagation pour l'étude en tendance centrale et une autre s'intéresse aux problèmes d'estimation de probabilités d'événements rares (statistiques liées aux queues de distributions). La section dédiée à la tendance centrale présente des méthodes d'intégration numérique, d'échantillonage et de planification d'expériences [Fang et al. \(2018\)](#). Celle consacrée aux probabilités d'événements rares présente des méthodes classiques issues du domaine de la fiabilité des structures [Lemaire \(2013\)](#); [Morio and Balesdent \(2015\)](#).

Ce chapitre aborde également les principales méthodes d'analyse de sensibilité globale [Da Veiga et al. \(2021\)](#). Ce domaine divise ses méthodes en deux grandes classes : les méthodes de criblage et les mesures d'importance. D'une part, les techniques de criblage, généralement mises en œuvre dans les problèmes de grande dimension, visent à identifier les variables n'ayant qu'un faible impact sur la variabilité de la sortie d'intérêt. D'autre part, les mesures d'importances visent, quant à elles, à attribuer de manière quantitative, pour chaque variable d'entrée, une part de variabilité de la sortie, permettant de proposer un classement des variables en fonction de leur influence.

Finalement, ce chapitre présente un panorama des familles de métamodèles communément utilisés en quantification des incertitudes [Forrester et al. \(2008\)](#). Une attention particulière est apportée à la régression par processus gaussiens qui revient à conditionner un processus gaussien par un ensemble d'observations du code de simulation numérique. Une fois conditionné, le processus gaussien apporte une information plus riche que d'autres types de métamodèles. En effet, cette méthode propose conjointement un métamodèle (un prédicteur, ou moyenne du processus), et une fonction d'erreur (variance du processus). Certaines méthodes itératives (dites « actives ») exploitent cette information complémentaire pour enrichir progressivement le métamodèle et améliorer sa prédictivité. Ces techniques ont connu un franc succès dans les années 90 pour résoudre des problèmes d'optimisation de fonctions coûteuses [Jones et al. \(1998\)](#). Depuis, leur utilisation s'est étendue à la résolution de problèmes de fiabilité des structures [Echard et al. \(2011\)](#).

## Chapitre 2 – Introduction à la modélisation et la conception de systèmes éoliens

La simulation d'une éolienne en mer implique la modélisation de plusieurs physiques en interaction avec des conditions environnementales de nature aléatoire. Ce chapitre introduit premièrement les méthodes spectrales utilisées pour générer des champs de vitesse de vent et de houle en appliquant des transformées de Fourier inverses (par exemple implémentées dans l'outil TurbSim [Jonkman \(2009\)](#)). Ces champs de vitesses de vent simulés alimentent par la suite un outil de simulation multi-physique des éoliennes. Cette simulation intègre une modélisation simplifiée des interactions entre fluides et structures (méthode "BEMT" pour *blade element momentum theory*), une modélisation dynamique de la structure par des éléments finis de type poutre, et une modélisation du contrôle-commande de l'éolienne [Milano \(2021\)](#). Ce code numérique produit en sortie des séries temporelles de plusieurs grandeurs physiques décrivant le comportement du système.

Cette thèse s'intéresse particulièrement à l'évaluation probabiliste du dommage en fatigue des structures éoliennes. Le dommage en fatigue est un phénomène qui détériore les propriétés mécaniques d'un matériau suite à sa sollicitation via un grand nombre de contraintes cycliques de faible amplitude. A l'heure actuelle, les standards [DNV-GL \(2016\)](#); [IEC-61400-1 \(2019\)](#) recommandent l'utilisation de coefficients de sécurité déterministes pour faire face à ce mode de défaillance. Une approche probabiliste permet d'enrichir l'analyse et parfois de mettre en évidence le conservatisme des marges de sûreté. Plusieurs travaux récents se sont intéressés à cette thématique en abordant des angles méthodologiques différents [Cousin \(2021\)](#); [Hirvoas \(2021\)](#); [Huchet \(2019\)](#); [Lataniotis \(2019\)](#); [Petrovska \(2022\)](#).

Dans ce contexte, ce chapitre liste les paramètres d'entrée de la chaîne de calcul considérés comme incertains par la suite. Ces variables aléatoires sont regroupées en deux groupes : le vecteur aléatoire lié à l'environnement (par exemple : la vitesse moyenne du vent, l'écart-type de la vitesse du vent, la direction du vent, la hauteur de houle, la période de houle, et la direction de houle), et le vecteur aléatoire lié au système (par exemple : l'erreur de d'alignement au vent du contrôleur, la rigidité du sol, les paramètres des courbes de calcul de fatigue).

### G.3 Résumés des chapitres relatifs aux contributions méthodologiques et apports vis-à-vis des applications

Après avoir dressé l'état de l'art sur ce sujet, les prochains chapitres du manuscrit présentent les nouvelles contributions de la thèse. D'un point de vue méthodologique, un objet mathématique servira de fil conducteur au cours de ces travaux. La *maximum mean discrepancy* (MMD) [Oates \(2021\)](#) est une mesure de dissimilarité entre des lois de probabilité basée sur des noyaux qui est utilisée dans des contextes différents (tests statistiques [Gretton et al. \(2006\)](#), analyse de sensibilité [Da Veiga \(2015\)](#), échantillonage [Pronzato and Zhigljavsky \(2020\)](#), etc.).

## **Chapitre 3 – Quantification des perturbations induites par les effets de sillage au sein d'un parc éolien**

Ce chapitre étudie les perturbations sur les conditions environnementales à l'intérieur d'une ferme éolienne en mer induites par les effets de sillage (*wake effect* en anglais) [Larsen et al. \(2008\)](#). Un parc éolien en mer théorique au large de la côte sud de la Bretagne est considéré comme cas d'usage, et un modèle numérique simulant le sillage de ce parc est exploité. Ce modèle donne une prédition analytique du déficit en vitesse de vent et de la turbulence créés par le sillage, en tenant compte de l'influence de la position des flotteurs en raison des forces moyennes du vent. Une propagation de l'incertitude sur le modèle de sillage est réalisée, en considérant la loi conjointe des conditions environnementales ambiantes en entrée. Au final une distribution environnementale perturbée par le sillage est simulée pour chaque éolienne. Une mesure de dissimilarité (la MMD) est utilisée pour comparer les distributions perçues par chaque éolienne. Cette quantité permet de regrouper les éoliennes (phase de *clustering*) exposées à des conditions environnementales similaires, entraînant une réponse structurelle identiques. Compte tenu du coût de calcul élevé des simulations aéro-servo-hydro-élastiques des éoliennes en mer, cette étude préalable permet de réaliser une analyse de fiabilité à l'échelle d'une ferme éolienne sans répéter l'analyse pour chaque turbine. En fin de compte, seules quatre classes sont retenues pour représenter une ferme de 25 éoliennes. Ce travail a mené à la publication suivante :

✉ A. Lovera, [E. Fekhari](#), B. Jézéquel, M. Dupoiron, M. Guiton and E. Ardillon (2023). "Quantifying and clustering the wake-induced perturbations within a wind farm for load analysis". In : *Journal of Physics : Conference Series (WAKE 2023)*, Visby, Sweden.

## **Chapitre 4 – Méthodes à noyaux pour l'estimation de la tendance centrale**

Ce chapitre présente une utilisation d'une mesure de dissimilarité basée sur des noyaux (la MMD) pour échantillonner suivant une loi de probabilité, méthode du "*kernel herding*" introduite par [Chen et al. \(2010\)](#). Cette technique de quadrature appartient à la famille dite des « quadratures Bayésiennes » [Briol et al. \(2019\)](#) qui s'interprètent comme une généralisation des méthodes de quasi-Monte Carlo [Li et al. \(2020\)](#). Le *kernel herding* est présenté en détails et plusieurs expériences numériques sur des fonctions analytiques illustrent son intérêt.

Les propriétés de cette méthode sont mises en valeur via une application industrielle dédiée à l'estimation de la moyenne du dommage en fatigue d'une structure éolienne. Cette quantité est déterminante dans le dimensionnement et la certification des éoliennes. Toutefois, son estimation par le biais de simulations numériques s'avère coûteuse. L'étude est réalisée sur un modèle d'une éolienne posée appartenant à une ferme installée en mer du Nord. Les incertitudes des conditions environnementales en entrée sont inférées sur des données mesurées in-situ.

Dans ce cadre, une comparaison numérique avec un échantillonnage Monte Carlo et quasi-Monte Carlo révèle la performance et les avantages pratiques du *kernel herding*. Cette méthode

permet notamment sous-échantillonner directement depuis une base de données environnementales importante, sans effectuer d'inférence (étape B). Ce travail a mené à la publication et au développement informatique suivant :

- ☞ E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). “Given-data probabilistic fatigue assessment for offshore wind turbines using Bayesian quadrature”. In : *Data-Centric Engineering*, In press.
- ☞ Le module Python `ctbenchmark` standardise les expériences numériques liées à la quadrature Bayésienne et est disponible sur la plateforme GitHub.
- ☞ Le module Python `copulogram` propose une nouvelle représentation graphique de jeux de données multivariés et est disponible sur la plateforme de téléchargement Pypi.

## **Chapitre 5 – Méthodes à noyaux pour la validation de métamodèles**

Ce chapitre propose une utilisation des méthodes d'échantillonage à base de noyaux dans le cadre de la validation de modèles d'apprentissage (ou métamodèles). L'estimation de la prédictivité des modèles d'apprentissage supervisé nécessite une évaluation de la fonction apprise sur un ensemble de points de test (non utilisés par lors de l'apprentissage). La qualité de l'évaluation dépend naturellement des propriétés de l'ensemble de test et de la statistique d'erreur utilisée pour estimer l'erreur de prédiction. Cette contribution propose d'une part d'utiliser des méthodes d'échantillonage pour sélectionner de manière “optimale” un ensemble de test et d'autre part présente un nouveau critère de prédictivité qui pondère les erreurs observées pour obtenir une estimation globale de l'erreur. Une comparaison numérique entre plusieurs méthodes d'échantillonage basées sur des approches géométriques [Shang and Apley \(2020\)](#) ou sur des méthodes à noyaux [Chen et al. \(2010\); Mak and Joseph \(2018\)](#) est effectuée. Nos résultats montrent que les versions pondérées des méthodes à noyau offrent des performances supérieures. Une application aux efforts mécaniques simulées par un modèle éolien en mer est également présentée. Cette expérience illustre la pertinence pratique de cette technique comme alternative efficace aux techniques coûteuses de validation croisée. Ce travail a mené à la publication et au développement informatique suivant :

- ☞ E. Fekhari, B. Iooss, J. Muré, L. Pronzato and M.J. Rendas (2023). “Model predictivity assessment : incremental test-set selection and accuracy evaluation”. In : *Studies in Theoretical and Applied Statistics*, pages 315–347. Springer.
- ☞ Le module Python `otkerneldesign` est développé en collaboration avec J.Muré. Ce module dédié à la quadrature Bayésienne est documenté et disponible sur la plateforme de téléchargement Pypi.

## Chapitre 6 – Estimation non-paramétrique de probabilités d'événements rares

L'estimation de probabilités d'événements rares est un problème courant dans la gestion des risques industriels, notamment dans le domaine de la fiabilité des structures Chabridon (2018). Pour ce faire, plusieurs techniques ont été proposées pour surmonter les limites connues de la méthode de Monte Carlo. Parmi elles, la méthode de “*subset sampling*” Au and Beck (2001) est une technique qui repose sur la décomposition de la probabilité de l'événement rare en un produit de probabilités conditionnelles moins rares (donc plus simples à estimer) associées à des événements de défaillance imbriqués. Cependant, cette technique repose sur la simulation conditionnelle à base de méthodes de Monte Carlo par chaînes de Markov (MCMC). Ces algorithmes permettent, à la convergence, de simuler selon la densité cible. Cependant, en pratique, ils produisent souvent des échantillons non indépendants et identiquement distribués (i.i.d.) en raison de la corrélation entre les chaînes de Markov. Ce chapitre propose une autre méthode pour échantillonner conditionnellement aux événements de défaillance imbriqués afin d'obtenir des échantillons dont la propriété d'être i.i.d. est préservée. La propriété d'indépendance des échantillons est particulièrement pertinente pour exploiter ces mêmes échantillons pour une analyse de sensibilité fiabiliste. L'algorithme proposé repose sur l'inférence non-paramétrique de la distribution conjointe conditionnelle en utilisant une estimation par noyau des marginales combinée à une inférence de la dépendance à l'aide de la copule empirique de Bernstein Sancetta and Satchell (2004). L'algorithme appelé “*Bernstein adaptive nonparametric conditional sampling*” (BANCS) est comparée à la méthode du *subset sampling* pour plusieurs problèmes de fiabilité des structures. Les premiers résultats sont encourageants, mais le contrôle du biais de l'estimateur doit être plus amplement investigué. Ce travail a mené à la publication et au développement informatique suivant :

☞ E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). “Bernstein adaptive nonparametric conditional sampling : a new method for rare event probability estimation”. In : *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)*, Dublin, Ireland.

☞ Le module Python `bancs` propose une implémentation de la méthode BANCS et est disponible sur la plateforme GitHub.

## Chapitre 7 – Analyse de sensibilité fiabiliste adaptative

Ce chapitre traite d'analyse de sensibilité pour des mesures de risque (par exemple, un probabilité d'événement rare). L'analyse de sensibilité globale Da Veiga et al. (2021) attribue à chaque variable (ou groupe de variable) une part de variabilité globale de la sortie (le plus souvent à l'aide d'une décomposition fonctionnelle de la variance de la sortie). Cependant, les variables ayant un impact sur des quantités liées à une queue de distribution peuvent être

très différentes que celles ayant un impact sur la variabilité globale (pondérée par le poids associé au centre de la distribution). L'analyse de sensibilité fiabiliste (en anglais “*reliability-oriented sensitivity analysis*”, Chabridon (2018)) permet d'expliquer le rôle des entrées vis-à-vis de probabilités d'événements rares. L'idée de ce chapitre est d'étudier l'évolution de la sensibilité au fur et à mesure que l'échantillonnage se rapproche de l'événement rare. Cette analyse permet ainsi d'exploiter les paquets successifs d'échantillons conditionnels générés par l'algorithme BANCS (présenté dans le Chapitre 6). En post-traitement de l'estimation de la probabilité d'un événement rare, cette approche utilise une mesure d'importance à base de noyaux, nommée *Hilbert-Schmidt Independence Criterion*, pour évaluer la dynamique de la sensibilité fiabiliste Marrel and Chabridon (2021).

## G.4 Conclusion

En résumé, cette thèse aborde plusieurs aspects du traitement des incertitudes à l'aide d'outils mathématiques à base de noyaux et présente un débouché industriel lié à l'enjeu de la maîtrise des risques des actifs éoliens en mer. Les contributions de cette thèse ont été principalement réalisées dans le cadre du projet européen HIPERWIND (*Highly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost-efficient offshore wind.*), et de l'ANR INDEX (INcremental Design of EXperiments). Le sous-sections ci-après résument les communications, les publications dans revue à comité de lecture et les développements informatiques.

### G.4.1 Communications et publications dans revues à comité de lecture

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). “Given-data probabilistic fatigue assessment for offshore wind turbines using Bayesian quadrature”. In : <i>Data-Centric Engineering</i> , In press.
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” <sup>1</sup> . In : <i>Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)</i> , Dublin, Ireland. (Paper & Talk)
	E. Vanem, <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)
	A. Lovera, <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)

1. Cette contribution a été récompensée par le “CERRA Student Recognition Award”

### G.4.2 Développements informatiques open source

`otkerneldesign`<sup>2</sup>

- Ce module Python génère des échantillons (aussi appelés plans d’expérience) en utilisant des méthodes à base de noyaux comme le *kernel herding* et les *support points*. Une implementation tensorisée qui améliore grandement les performances est également proposée. En complément, une méthode de pondération “optimale” à l’aide de quadrature Bayésienne est proposée.
- Ce module est développé en collaboration avec J. Muré, est documenté et disponible sur la plateforme de téléchargement Pypi.

`bancs`<sup>3</sup>

- Ce module Python offre une implémentation de la méthode “*Bernstein Adaptive Nonparametric Conditional Sampling*” mentionnée en Section G.3.
- Ce module est disponible sur la plateforme de GitHub et son utilisation est illustrée par des exemples analytiques.

`ctbenchmark`<sup>4</sup>

- Ce module Python standardise les comparaisons numériques réalisés pour étudier les méthodes de quadrature Bayésiennes.
- Le module et les expériences numériques sont disponibles sur un dépôt GitHub.

`copulogram`<sup>5</sup>

- Ce module Python propose une nouvelle représentation graphique de jeux de données multivariés appelée *copulogram*.
- Ce module, développé en collaboration avec V. Chabridon, est disponible sur la plateforme de téléchargement Pypi.

---

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

3. Dépôt: <https://github.com/efekhari27/bancs>

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

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

