UNIVERSITÉ CÔTE D'AZUR | ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Uncertainty quantification in multi-physics model for wind turbine asset management

Elias FEKHARI

EDF R&D, [Université Nice Côte d'Azur]

Présentée en vue de l'obtention du grade de docteur en Automatique, traitement du signal et des images d'Université Côte d'Azur, dirigée par Bertrand Iooss, soutenue le 12 mars 2024.

Devant le jury composé de :

Directeur	Bertrand IOOSS	Ingénieur-chercheur senior	EDF R&D, Chatou
Co-encadrants	Vincent CHABRIDON	Ingénieur-chercheur	EDF R&D, Chatou
	Joseph MURÉ	Ingénieur-chercheur	EDF R&D, Chatou
Rapporteurs	Franck SCHOEFS	Professeur des universités	Nantes Université, Nantes
	Daniel STRAUB	Professeur des universités	TUM, Munich
Examinateurs	Mireille BOSSY	Directrice de recherche	INRIA, Sophia-Antipolis
	Sébastien DA VEIGA	[Maître de conférence]	ENSAI, Rennes
	Bruno SUDRET	Professeur des universités	ETH, Zürich
Invitée	Anaïs LOVERA	Ingénieure-chercheur	EDF R&D, Saclay

EDF

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 fullfilment of the requirements for the degree of

Doctor of Philosophy
(in Computer Science)

publicly defended on March 12, 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

List of Figures

List of Tables

Introduction

Industrial context and motivation

The current challenge of energy transition involves, among other things, reducing the share of fossil fuels in the global electricity mix. In this context, offshore wind energy offers several advantages (**?**). Offshore energy benefits from more consistent winds than onshore, mainly due to the absence of terrain roughness, it also makes possible the installation of larger and more powerful wind turbines. Since the construction of the first offshore wind farm in Vindeby, Denmark, in 1991, the industry has experienced rapid growth, with a total capacity of 56 GW in operation 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 the electricity produced is directly sold on the wholesale market (**?**).

However, despite the progress of this sector, scaling limitations and numerous scientific challenges emerge. 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 several scientific challenges that often involve coupling data with numerical simulations of physical systems and their surrounding environment. The wind energy community is focused on different objectives, including enhancing the design of floating offshore wind turbines, refining wind resource estimation techniques, and optimizing maintenance operations. In general, several decisions are made throughout the lifespan of a wind turbine by its designer, installer, and operator, all while having only partial knowledge of certain physical phenomena. Therefore, modeling and controlling the various sources of uncertainties associated with offshore wind energy proved to be a key success factor in this highly competitive industry.

Overall, the offshore wind industry needs methods for uncertainty management regarding safety margins and industrial asset management (at the component, wind turbine, and overall wind farm levels) (**?**). For wind project developers, the primary focus is on improving the wind potential assessment of candidate sites by combining various sources of information and modeling the multivariate distribution of environmental conditions. In the case of floating wind projects, the goal is to incorporate a probabilistic aspect from the design phase (e.g., of the floaters) to define safer, more robust, and more cost-effective solutions. For wind farm

owners, end-of-life management is another significant concern. An owner of a wind farm at the end of its life has three options: extend the operational life of assets, replace current wind turbines with newer models, or decommission and sell the wind farm. The first two options require evaluating the structural reliability and residual lifespan, with quantitative assessments reviewed by certification bodies and insurers to issue operating permits. To provide rigorous risk assessments, the generic methodology of *uncertainty quantification methodology* is a widely accepted approach in industrial sectors facing these types of issues (**?**).

Generic methodology for uncertainty quantification

Computer experiment is a discipline that emerged with the advent of informatics. This practice produces numerical models that allow the simulation of complex system behavior based on initial conditions defined by the analyst. Numerical models quickly became essential for the analysis, design, and certification of complex systems in cases where experiments or physical measurements are too costly or even unfeasible. However, such numerical models are mostly deterministic: the reproducible result of a simulation is associated with a fixed input set of parameters. The issue of managing uncertainties associated with these inputs arises when performing analysis with numerical models.

Uncertainty quantification aims at modeling and controlling uncertainties around a numerical model. To do so, a generic methodology has been proposed to quantify and analyze uncertainties between input and output variables of a numerical model (**?**). An overview of the mathematical tools used in this field is provided by **?**. This approach improves the understanding of a system, ultimately contributing to more robust decision-making. Figure 1 illustrates the main steps of the generic uncertainty quantification method, which are briefly summarized hereafter:

- **Step A – Problem specification**: This step involves identifying the system under study and constructing a numerical model capable of precisely simulating its behavior. Specifying the problem also involves the definition of a set of parameters inherent to the numerical model. These parameters include both the input variables and the output variables generated by the simulation. In this document, the numerical model is considered a black box, in contrast to approaches that are integrated within the numerical solution schemes for the system's behavioral equations (referred to as intrusive approaches (**?**)). Generally, these numerical models are first calibrated against measured data and pass a process of validation and verification to reduce modeling errors (**?**).

- **Step B – Uncertainty modeling**: The objective of the second step is to identify and model all the sources of uncertainty related to the input variables. Most of the time the uncertainty modeling is done in the probabilistic framework.

- **Step C – Uncertainty propagation**: This step consists of propagating the uncertain inputs through the computer model. Consequently, the output of the numerical model

(commonly scalar) also becomes uncertain. The goal is to estimate a quantity of interest, which is a statistic related to the studied random output variable. The uncertainty propagation method may differ depending on the quantity of interest targeted (e.g., central tendency, a quantile, a rare event probability, etc.).

- **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**: Considering the high computational cost associated with some simulations, statistical approaches emulate these expensive simulators with a limited number of simulations. Uncertainty quantification can then be carried out using a "surrogate model" (or metamodel) for a reduced computational cost. This optional step of statistical learning is not strictly a part of uncertainty quantification, but it often proves to be essential for enabling its practical implementation.

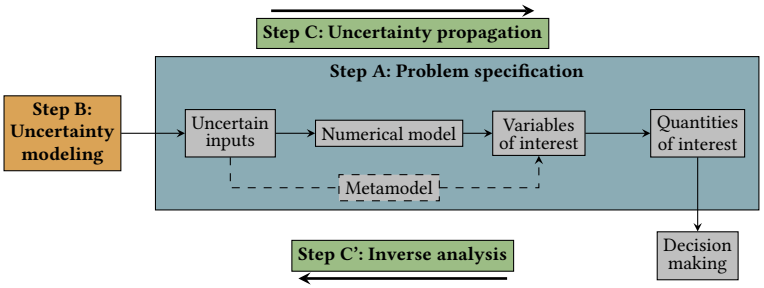


Figure 1 General uncertainty quantification framework (**?**, adapted by Ajenjo (2023))

Problem statement and outline of the thesis

Risk and uncertainty management in the field of wind energy is a significant concern for the electric utility Électricité de France (EDF). This thesis aims to adapt and apply the generic uncertainty quantification methodology to industrial offshore wind energy studies. As such, this use case raises scientific challenges related to its specific characteristics, described in the following:

- The numerical model exploited in the present work consists of a series of numerical models executed sequentially. This chain is divided into three parts: first, a temporal and stochastic generation of wind and wave velocity fields, followed by the simulation of the coupled hydro-aero-servo-elastic behavior of the wind turbine, and finally a post-processing phase to obtain scalar quantities of interest, aggregated over the temporal outputs.
- The complexity of this simulator, along with the high unit computational cost (about 40 minutes per simulation), requires the use of efficient sampling methods and high-performance computing systems. In addition to the complexity associated with the

numerical model, modeling the input uncertainties also represents a challenge. Indeed, the joint distribution associated with environmental conditions presents a complex dependence structure. The quality of the inference step is critical as it directly impacts the conclusions of uncertainty propagation.

In order to apply the generic methodology for uncertainty quantification to the offshore wind turbine case, this thesis aims to answer the following questions:

Q1. *How to accurately model the dependence structure associated with the joint environmental distribution?* (➪ Step B)

Q2. *How to perform uncertainty propagation through a computationally expensive numerical chain uniquely based on an empirical description (measured data) of input uncertainties?* (➪ Step C)

Q3. *How to estimate rare event probabilities related to the fatigue failure of offshore wind turbine structures?* (➪ Step C)

Q4. *How to assess and analyze the sensitivity of uncertain inputs regarding quantities of interest resulting from structural reliability (i.e., reliability-oriented sensitivity analysis)?* (➪ Step C')

To propose an answer to these questions, this manuscript is divided into three parts. The first part offers an introduction to uncertainty quantification methods and offshore wind turbine numerical modeling. The second part presents the contributions of this thesis to uncertainty quantification and propagation while the third part describes the contributions to rare event estimation. This manuscript is divided into seven chapters, which are summarized hereafter:

Chapter 1 – Introduction to uncertainty quantification. This chapter gives a brief overview of various topics in uncertainty quantification (**?**). After a reminder of some mathematical concepts, the model specification step is described, considering a black box and its input and output variables. The different types and sources of uncertainties are then presented, along with their modeling within a probabilistic framework. Uncertainty propagation depends on the estimated quantities of interest, therefore, one section addresses propagation methods for central tendency studies, and another focuses on rare event probability estimation (a statistic related to the tails of output distributions). The section dedicated to central tendency presents numerical integration, sampling, and design of experiment methods (Fang et al., 2018). The one about rare event probabilities introduces usual methods from the field of structural reliability (**??**).

This chapter also covers the main methods for global sensitivity analysis (**?**). This field divides its methods into two major classes: screening methods and importance measures. Screening techniques typically applied in high-dimensional problems, aim to identify variables with low impact on the variability of the output of interest. Importance measures, on the other

hand, quantitatively allocate, for each input variable, a share of the output variability, enabling the ranking of variables based on their influence.

Finally, this chapter presents an overview of the families of metamodels commonly used in uncertainty quantification (**?**). Special attention is given to the Gaussian process regression, which involves conditioning a Gaussian process on a set of observations from the numerical model. Once conditioned, the Gaussian process provides richer information than other types of metamodels. This method simultaneously offers a surrogate model (mean of the Gaussian process, also called predictor) and an error function (variance of the process). Some iterative methods (called "active") use this additional information to progressively enrich the metamodel and improve its predictability. These techniques were quite successful in the 1990s for solving optimization problems with expensive functions (**?**). Since then, their use has expanded to solve problems in structural reliability **?**.

Chapter 2 – Introduction to wind turbine modeling and design. The simulation of an offshore wind turbine involves modeling multiple physical aspects interacting with random environmental conditions. This chapter first introduces spectral methods used to generate wind and wave velocity fields by applying inverse Fourier transforms (e.g., as implemented in the TurbSim tool (**?**)]). These simulated wind velocity fields then become the inputs of a multi-physics wind turbine numerical model. Such simulation includes simplified modeling of the interactions between fluids and structures (using the blade element momentum theory), dynamic modeling of the structure using flexible multibody methods, and modeling of wind turbine control systems [source]. The numerical code studied generates a time series of several physical quantities describing the system's behavior.

This thesis particularly focuses on the probabilistic evaluation of fatigue damage in wind turbine structures. Fatigue damage is a phenomenon that deteriorates the mechanical properties of a material as a result of exposure to many cyclic low-amplitude stresses. Currently, standards recommend the use of deterministic safety factors to address this failure mode (**??**). A probabilistic approach enhances the analysis and can sometimes reveal conservative safety margins. Several recent studies have addressed this topic from different methodological perspectives (**?**Lataniotis, 2019; **?**; **?**; **?**).

In this context, this chapter enumerates the input parameters of the calculation chain that are considered uncertain. These random variables are grouped into two groups: the random vector related to the environment (e.g., average wind speed, wind speed standard deviation, wind direction, significant wave height, wave period, and wave direction), and the random vector related to the system (e.g., controller wind misalignment error, soil stiffness, fatigue calculation curve parameters).

Chapter 3 – Kernel-based uncertainty quantification. This chapter examines perturbations in environmental conditions within an offshore wind farm induced by wake effects **?**. A theoretical offshore wind farm off the southern coast of Brittany is considered as a use case,

and a simplified numerical model of wake in this wind farm is used. This model provides an analytical prediction of the wind speed deficit and turbulence created by the wake, taking into account the influence of the floaters' positions due to rigid body dynamics.

In the second phase, uncertainty propagation is carried out through the wake model, considering the joint distribution of ambient environmental conditions as inputs. In the end, an environmental distribution perturbed by the wake is simulated for each wind turbine. A dissimilarity measure between distributions, based on kernels and named the *maximum mean discrepancy* (MMD), is used to compare the distributions perceived by each wind turbine. This measure allows the clustering of wind turbines exposed to similar environmental conditions, resulting in identical structural responses. Given the high computational cost of aero-servo-hydro-elastic simulations for offshore wind turbines, this preliminary study enables reliability analysis at the wind farm scale without repeating the analysis for each turbine. Ultimately, only four classes are selected to represent a wind farm of 25 turbines.

Chapter 4 – Kernel-based central tendency estimation. Chapter four presents the use of the kernel-based dissimilarity measure (MMD) in the context of probability distribution sampling, a method known as "kernel herding" introduced by **?**. This quadrature technique belongs to the family of "Bayesian quadratures" Briol et al. (2019), which can be viewed as a generalization of quasi-Monte Carlo methods Li et al. (2020).

The properties of this method are highlighted through an industrial application dedicated to estimating the mean fatigue damage of a wind turbine structure. Although this quantity is crucial in the design and certification of wind turbines, the methods used to estimate it are known to be suboptimal (i.e., regular grids). The study is conducted on a model of a fixed offshore wind turbine belonging to a farm in the North Sea. Uncertainties in input environmental conditions are inferred from in-situ measured data.

Finally, a numerical comparison with Monte Carlo and quasi-Monte Carlo sampling reveals the performance and practical advantages of kernel herding. This method allows for direct subsampling from a large environmental database without the need for inference (step B).

Chapter 5 – Kernel-based metamodel validation. This chapter proposes the use of kernel-based sampling methods in the context of model validation for machine learning (or surrogate models). Estimating the predictivity of supervised learning models requires an evaluation of the learned surrogate model on a set of test points that were not used during training. The quality of the validation naturally depends on the properties of the test set and the metric used to summarize the prediction error. This contribution first suggests using space-filling sampling methods to "optimally" select a test set, then, it introduces a new predictivity coefficient that weights the observed errors to improve the global error estimation. A numerical comparison between several sampling methods based on geometric approaches (**?**) or kernel methods **??** is carried out. Our results show that weighted versions of kernel methods offer superior performance. An application to simulated mechanical loads in an offshore wind turbine model

is also presented. This experiment illustrates the practical relevance of this technique as an effective alternative to costly cross-validation techniques.

Chapter 6 – Nonparametric rare event estimation. Estimating rare event probabilities is a common issue in industrial risk management, especially in the field of structural reliability (**?**). To address this, several techniques have been proposed to overcome the known limitations of the Monte Carlo method. Among them, "subset sampling" (**?**) is a technique based on the split of a rare probability into a product of less rare (and thus easier to estimate) conditional probabilities associated with nested failure events. However, this technique relies on conditional simulation using Markov chain Monte Carlo (MCMC) methods. These algorithms, while converging, often produce samples that are not independent and identically distributed (i.i.d.) due to the correlation between the Markov chains. In this chapter, another conditional sampling method is proposed, with the advantage of preserving the i.i.d. property. Independent sampling is particularly relevant for reusing these samples in a posterior reliability-oriented sensitivity analysis. The algorithm introduced is based on the non-parametric inference of the conditional joint distribution using kernel density estimation of marginals combined with dependence inference using the empirical Bernstein copula (Sancetta and Satchell, 2004). The so-called "Bernstein adaptive nonparametric conditional sampling" (BANCS), is compared to the subset sampling method for several structural reliability problems. The initial results are promising, but further investigation is needed to control the estimator's bias.

Chapter 7 – Sequential reliability oriented sensitivity analysis. This chapter deals with sensitivity analysis for risk measures (e.g., rare event probabilities). Global sensitivity analysis (**?**) assigns a portion of the global output variability to each variable (or group of variables), often using a functional decomposition of the output variance. However, when studying risk measures (often located in the distributions' tails), the global sensitivity might be very different from the sensitivity to the risk measure. "Reliability-oriented sensitivity analysis" (ROSA), studies the impact of the inputs in regard to a risk measure such as a rare event probability (see e.g., **?**). Using the nested subsets obtained with the BANCS algorithm (presented in Chapter 6), the idea of this chapter is to study the ROSA evolution as the subsets get closer to the failure domain. For each subset, a ROSA is carried out with a kernel-based importance measure called the "Hilbert-Schmidt Independence Criterion" adapted to this context (**?**).

Numerical developments

Several implementations developed in this thesis are available on different platforms, allowing the reader to reproduce some numerical results in an open-data approach:

`otkerneldesign`[1]	• 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. • This Python package, developed in collaboration with J.Muré, is available on the platform Pypi and fully documented.
`bancs`[2]	• This Python package proposes an implementation of the "Bernstein Adaptive Non-parametric Conditional Sampling" method for rare event estimation. • This Python package is available on the PyPI platform and is illustrated with examples and analytical benchmarks.
`ctbenchmark`[3]	• This Python package presents a standardized process to benchmark different sampling methods for central tendency estimation. • This Python package is available on a GitHub repository with analytical benchmarks.
`copulogram`[4]	• This Python package proposes an implementation of a synthetic visualization tool for multivariate distributions. • This Python package, developed in collaboration with V.Chabridon, is available on the Pypi platform.

[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.	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.
Jour. Pap.	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*.
Int. Conf.	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Numerical Studies of Bayesian Quadrature Applied to Offshore Wind Turbine Load Estimation". In: *SIAM Conference on Uncertainty Quantification (SIAM UQ22)*, Atlanta, USA. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *22nd Annual Conference of the European Network for Business and Industrial Statistics (ENBIS 2022)*, Trondheim, Norway. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Efficient techniques for fast uncertainty propagation in an offshore wind turbine multi-physics simulation tool". In: *Proceedings of the 5th International Conference on Renewable Energies Offshore (RENEW 2022)*, Lisbon, Portugal. (Paper & Talk) E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). "Bernstein adaptive nonparametric conditional sampling: a new method for rare event probability estimation"[5]. In: *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)*, Dublin, Ireland. (Paper & Talk) E. Vanem, E. Fekhari, N. Dimitrov, M. Kelly, A. Cousin and M. Guiton (2023). "A joint probability distribution model for multivariate wind and wave conditions". In: *Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2023)*, Melbourne, Australia. (Paper) 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 (Paper)
Nat. Conf.	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Kernel-based quadrature applied to offshore wind turbine damage estimation". In: *Proceedings of the Mascot-Num 2022 Annual Conference (MASCOT NUM 2022)*, Clermont-Ferrand, France (Poster) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2023). "Rare event estimation using nonparametric Bernstein adaptive sampling". In: *Proceedings of the Mascot-Num 2023 Annual Conference (MASCOT-NUM 2023)*, Le Croisic, France (Talk)
Invited Lec.	Le Printemps de la Recherche 2022, Nantes, France. "Traitement des incertitudes pour la gestion d'actifs éoliens". (Talk) Journées Scientifiques de l'Eolien 2024, Saint-Malo, France. "Evaluation probabiliste de la fiabilité en fatigue des structures éoliennes en mer". (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

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 an uncertainty quantification (UQ) analysis with these numerical models is a key element in improving the understanding of the phenomena studied. A wide panel of UQ methods has been developed over the years to pursue these studies for a reasonable computational cost.

This chapter presents the essential tools and methods from the generic UQ framework, including elements partially inspired from **?** and **?**. It is structured as follows: Section 1.2 describes the context of the model specification step; Section 1.3 presents a classification of the input 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, Section 1.7 introduces the concept of surrogate models to emulate a model by realizing statistical learning on a limited dataset.

`OpenTURNS`[1]**.** Is a high-performance Python library dedicated to UQ (**?**). `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 guarantee software quality, the development follows robust processes such as exhaustive unit testing and multiplatform continuous integration. A dedicated forum hosts an active community, which helps new users and discusses future developments. Finally, no-code users can benefit from `OpenTURNS`'s free-download Graphical User Interface software, named `Persalys`[2]. In this chapter, the methodological concepts introduced are linked to minimal `OpenTURNS` implementations examples, available in Appendix D.

1.2 Black-box model specification

In our computer experiments context, uncertainty quantification is 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

[2] `OpenTURNS` installation guide and documentation are available at https://openturns.github.io/www/
[2] `Persalys`, a free-download graphical user interface available at https://www.persalys.fr/obtenir.php

resolution of the equations of the physics (see e.g., **?**). In practice, numerical models might be a sequence of codes executed in series to obtain a variable of interest.

While simulation models are in most cases deterministic, they may also be qualified as intrinsically stochastic (i.e., two runs of the same model taking the same inputs return different outputs). Additionally, numerical simulation always 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 $\mathbf{x} = (x_1, \ldots, x_d)^\top$ considered as uncertain (e.g., wind speed, wave period, etc.). The outputs studied are also defined at this stage, which will only be of scalar type in the present work. UQ methods suited to other types of outputs exist (see e.g., for time series outputs Lataniotis 2019, for functional outputs **??**). Let us then define the following numerical model:

$$\mathcal{M} : \left| \begin{array}{ccc} \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d & \longrightarrow & \mathcal{D}_{\mathrm{y}} \subseteq \mathbb{R} \\ \mathbf{x} & \longmapsto & y. \end{array} \right. \tag{1.1}$$

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 a 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, surrogate models (or metamodels) are statistical approximations of the costly numerical model, that can be used to perform tractable UQ. Surrogate models are built and validated on a limited number of simulations (in a *supervised learning* framework). In practice, note that the model specification step is often associated with the development of a *wrapper* of the code. It is an overlay of code allowing its execution in a parametric way, which is often associated with *high-perfomance computer* (HPC) deployment. Once the model is specified, a critical step in uncertainty quantification is enumerating the input uncertainties and building their associated mathematical model.

1.3 Enumerating and modeling the uncertain inputs

1.3.1 Sources of the input uncertainties

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

- **aleatory uncertainty** regroups the uncertainties arising 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 (e.g., material properties). Contrarily to the aleatory ones, epistemic uncertainties might be reduced by investigating their origin (often at a certain cost).

Der Kiureghian and Ditlevsen (2009) discuss the relevance of this classification. They affirm that this split is practical for decision-makers to identify possible ways to reduce their uncertainties. However, it 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* is a measure space with total measure summing to one, also called probability triple and denoted $(\Omega, \mathcal{A}, \mathbb{P})$. This mathematical concept first includes a sample space Ω, which contains a set of outcomes $\omega \in \Omega$. Note that an *event* is defined as a set of outcomes in the sample space. Then, a σ-algebra $\mathcal{A}$, also called event space, is a set of events. Finally, a probability function $\mathbb{P} : \mathcal{A} \to [0,1]$, 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:

$$\mathbf{X} : \left| \begin{array}{ccc} \Omega & \longrightarrow & \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d \\ \omega & \longmapsto & \mathbf{X}(\omega) = \mathbf{x}. \end{array} \right. \tag{1.2}$$

In the following, the random vector $\mathbf{X}$ will be considered to be a squared-integrable function against the measure $\mathbb{P}$ (i.e., $\int_\Omega |\mathbf{X}(\omega)|^2 \, \mathrm{d}\mathbb{P}(\omega) < \infty$). Moreover, the present thesis deals with continuous random variables.

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

$$\mathbb{P}_{\mathbf{X}}(B) = \mathbb{P}(\mathbf{X} \in B) = \mathbb{P}(\omega \in \Omega : \mathbf{X}(\omega) \in B), \quad \forall B \in \mathcal{A}. \tag{1.3}$$

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

$$F_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}(\mathbf{X} \leq \mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \ldots, X_d \leq x_d) = \mathbb{P}_{\mathbf{X}}\left( ]-\infty, x_1] \times \cdots \times]-\infty, x_d] \right). \tag{1.4}$$

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) $f_{\mathbf{X}} : \mathcal{D}_{\mathbf{x}} \to \mathbb{R}_+$ 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:

$$\mu_{\mathbf{X}} = \mathbb{E}[\mathbf{X}] = \int_{\Omega} \mathbf{X}(\omega)\, \mathrm{d}\mathbb{P}(\omega) = \int_{\mathcal{D}_{\mathbf{x}}} \mathbf{x}\, f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = (\mathbb{E}[X_1], \dots, \mathbb{E}[X_d])^\top . \tag{1.5}$$

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

$$\mathrm{Var}(X_i) = \mathbb{E}\left[X_i - \mathbb{E}[X_i]\right], \tag{1.6}$$

and a covariance describing their joint variability:

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}\left[\left(X_i - \mathbb{E}[X_i]\right)\left(X_j - \mathbb{E}[X_j]\right)\right]. \tag{1.7}$$

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

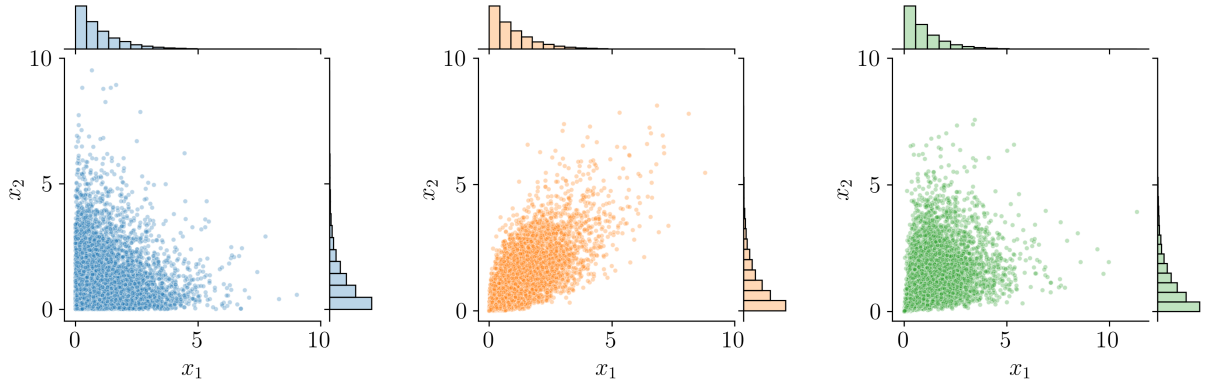
1.3.3 Joint input probability distribution

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

In general, the single effects of multivariate distributions tend to be well modeled. However, modeling the dependence structure underlying 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. Judging from this example, one can assume that the joint distribution results from the composition of the single effects, also called marginals, and an application governing the dependence between them.

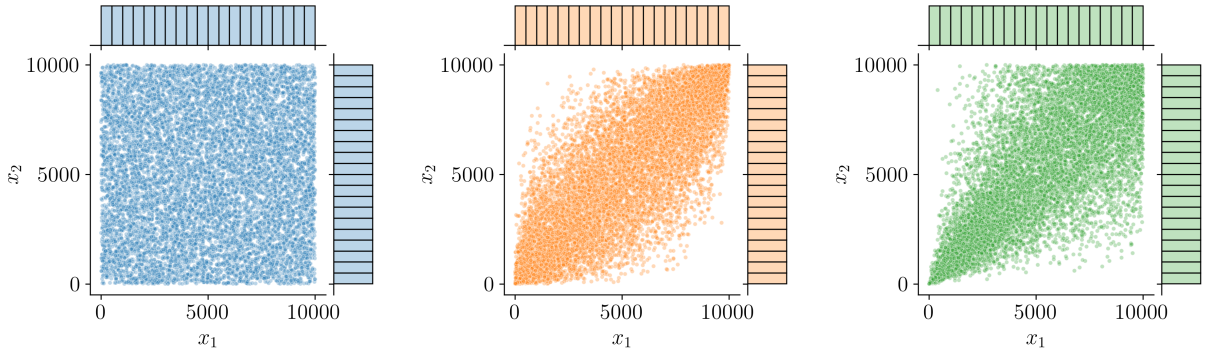
An empirical way of isolating the dependence structures from this example is to transform the samples in the ranked space. Let us consider an n-sized sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\right\} \in \mathcal{D}_{\mathbf{x}}^n$. The corresponding ranked sample is defined as: $\mathbf{R}_n = \left\{\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(n)}\right\}$, where[3] $r_j^{(i)} = \sum_{l=1}^{n} \mathbb{1}_{\left\{x_j^{(l)} \leq x_j^{(i)}\right\}}$, $\forall j \in \{1, \dots d\}, i \in \{1, \dots n\}$. Ranking a multivariate dataset allows us to isolate the dependence structure witnessed empirically (**?**). Fig. 1.2 shows the same three samples from Fig. 1.1 in the ranked space. One can first notice that the marginals are uniform since each rank is uniformly distributed. Then, the scatter plot from the distribution with independent copula (left plot) is uniform while the two others present different patterns.

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



(a) Independent copula (b) Normal copula (c) Clayton copula

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



(a) Independent copula (b) Normal copula (c) Clayton copula

Figure 1.2 Ranked samples represented in the Fig. 1.1

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

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

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

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:*

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

where c *is the density of the copula, sometimes also called copula by misuse of language. The reader might refer to* **?** *for three different mathematical proofs.*

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:

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

This equation allows us to extract a copula from a joint distribution by knowing its marginals. Additionally, copulas are invariant under creasing 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 Π implies that the distribution is defined as the product of its marginals $\Pi = \prod_{j=1}^{d} u_j$. In an inference context, this theorem divides the fitting problem into two independent problems: fitting the marginals and fitting the copula. Provided a dataset, this framework allows the potential 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 asked 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.

To ease the reading, a few techniques for estimating marginal distributions are available in Appendix A. In addition, two nonparametric methods are introduced in Chapter 3 to infer a copula: the "empirical Bernstein copula" and the "Beta copula". The adequation between a fitted probabilistic model and a dataset should be validated, therefore, Appendices A recall visual and quantitative tools for univariate goodness-if-fit evaluation.

`OpenTURNS` 1 (Bivariate distribution)**.** The Python code available in Appendix D proposes a minimalistic `OpenTURNS` example of a probabilistic uncertainty modeling. Figures illustrating the present section may be reproduced, using the `OpenTURNS` scripts available on GitHub[4].

[4]https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/copulas.ipynb

1.4 Uncertainty propagation for central tendency study

The previous section aimed at building a probabilistic model of the uncertainties considering the knowledge available. This one introduces diverse methods for forward propagation of the input uncertainties through a numerical model. In the present section, uncertainty propagation is dedicated to the "central tendency" as its goal is to study the mean and variance of the output distribution. This approach contrasts with the uncertainty propagation committed to rare event probability estimation, which will be introduced in Section 1.5 (e.g., used to assess reliability).

The difficulties related to any uncertainty propagation mostly arise from the practical properties of the numerical model. Its potential high dimension, irregularity and nonlinearity each represent a challenge. Such studies rely on a finite number of observations of the numerical model, depending on the computational budget affordable. Uncertainty propagation is at the end of the generic UQ approach (step C), however, it is affected by the "garbage in, garbage out" concept. Meaning that its conclusions depend on the accuracy of the inputs' uncertainty modeling.

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

1.4.1 Numerical integration

Forward uncertainty propagation aims at integrating a measurable function $g : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}$ with respect to a probability measure $\mathbb{P}_{\mathbf{X}}$. Numerical integration provides algorithmic tools to help the resolution of this probabilistic integration (i.e., Lebesgue integration). Note that the measurable function g, in the context of computer experiments, becomes the numerical model $\mathcal{M}$ introduced in Eq. (1.1).

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

$$I_{\mathbb{P}_{\mathbf{X}}}(g) := \int_{\mathcal{D}_{\mathbf{X}}} g(\mathbf{x}) \, \mathrm{d}\mathbb{P}_{\mathbf{X}}(\mathbf{x}) \approx \sum_{i=1}^{n} w_i \, g\left(\mathbf{x}^{(i)}\right). \tag{1.11}$$

For a given sample size n, the goal is to find a set of tuples $\left\{\mathbf{x}^{(i)}, w_i\right\}_{i=1}^{n}$ (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 $(\Omega, \mathcal{A}, \mathbb{P}_{\mathbf{X}})$, 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 to present the main multivariate integration techniques. These methods encompass different properties: some are deterministic and some are aleatory; some are sequential (i.e., 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-Clenshaw-Curtis rule are two univariate deterministic quadratures that will be briefly introduced (see **?** 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 Gaussian rules exist, the most common 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:

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

Gauss-Legendre quadrature 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), at the expense of a slightly lower accuracy.

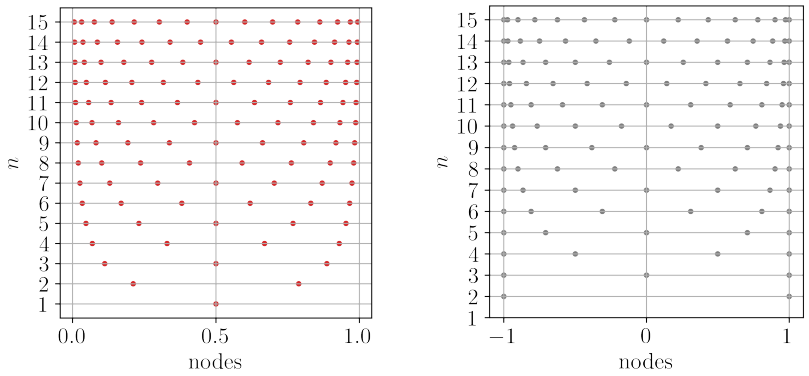
To overcome this practical drawback, Fejér then Clenshaw with Curtis proposed a nested rule with mostly equivalent accuracy as Gaussian quadratures. This method is usually presented to integrate a function with respect to the uniform measure on $[-1,1]$ and starts with a change of variables:

$$\int_{-1}^{1} g(x)\,\mathrm{d}x = \int_{0}^{\pi} g\left(\cos(\theta)\right)\,\sin(\theta)\,\mathrm{d}\theta. \tag{1.13}$$

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

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

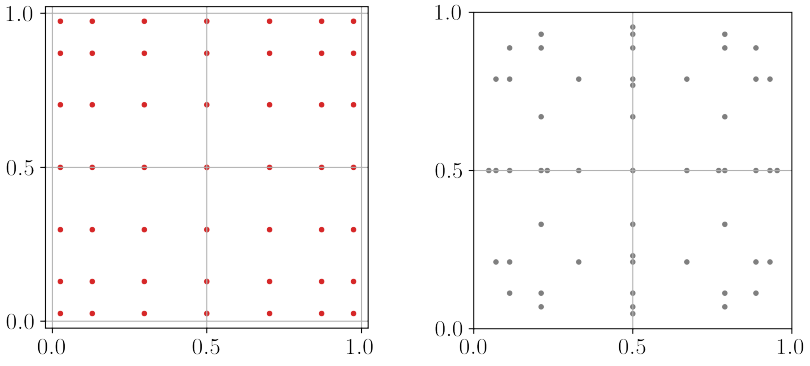
These two univariate integration schemes are both very efficient on a wide panel of functions. Yet, Fejér-Clenshaw-Curtis is sequential and offers easy implementations, benefitting from powerful algorithms such as the *fast Fourier transform*. Fig. 1.3 illustrates the nested properties of Fejér-Clenshaw-Curtis quadrature by representing the nodes of quadrature rules with increasing size.



(a) Gauss-Legendre (b) Clenshaw-Curtis

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

Uncertainty quantification problems are rarely unidimensional, but one can directly 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. In Fig. 1.4, the left plot represents a two-dimensional tensor product of identical Gauss-Legendre quadratures. Alternatively, sparse multivariate quadratures (i.e., Smolyak sparse grid) explore the joint domain more efficiently. Using the Smolyak recurrent formula (see e.g., **?**), two univariate quadratures can be combined as illustrated on the right of Fig. 1.4.



(a) Tensor product (n=49) (b) Smolyak sparse grid (n=53)

Figure 1.4 Two identical univariate Gauss-Legendre quadratures combined as a tensor product (left) and a Smolyak sparse grid (right).

Monte Carlo methods

Monte Carlo methods were initially developed in the 1940s to solve problems in neutronics. Ever since, this frequentist technique has been applied to the resolution of the Lebesgue integral. To integrate a function g against a measure $\mathbb{P}_{\mathbf{X}}$, it randomly generates points following the input measure. The integral is estimated by taking the uniform arithmetic mean of the nodes' images 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 uniformly generate a sequence of random points on $[0,1]$. These sequences mimic actual randomness but are in fact generated by deterministic algorithms, also called pseudorandom number generators. Pseudorandom algorithms generate a sequence of numbers with a very large, but finite length. This sequence can be exactly repeated by fixing the same initial point, also called *pseudorandom seed*. Most programming languages use the Mersenne-Twister pseudorandom generator (Matsumoto and Nishimura, 1998), offering a very long period (around 4.3×10^{6001} iterations).

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

$$I_{\mathbb{P}_{\mathbf{X}}}(g) \approx \overline{y}_n^{\mathrm{MC}} = \frac{1}{n} \sum_{i=1}^{n} g(\mathbf{x}^{(i)}). \tag{1.15}$$

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 $\widehat{\sigma}_Y^2$. The variance of the Monte Carlo estimator results from a manipulation of the central limit theorem:

$$\mathrm{Var}\left(\overline{y}_n^{\mathrm{MC}}\right) = \frac{1}{\sqrt{n}} \mathrm{Var}\left(g(\mathbf{X})\right). \tag{1.16}$$

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

$$I_{\mathbb{P}_{\mathbf{X}}}(g) \in \left[\overline{y}_n^{\mathrm{MC}} - q_\alpha \frac{\mathrm{Var}\left(g(\mathbf{X})\right)}{\sqrt{n}}, \overline{y}_n^{\mathrm{MC}} + q_\alpha \frac{\mathrm{Var}\left(g(\mathbf{X})\right)}{\sqrt{n}}\right], \tag{1.17}$$

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

variates and multi-level Monte Carlo. For further details, the reader may refer to 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 dimension while Monte Carlo methods deliver contrasted performances. Quasi-Monte Carlo is a deterministic family of numerical integration schemes with respect to the uniform measure on $[0,1]$. It offers powerful performances with strong guarantees by choosing nodes according to *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 (Morokoff and Caflisch, 1995; Leobacher and Pillichshammer, 2014) is a fundamental result for understanding the role of the discrepancy in numerical integration.

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

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

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.

The function's variation $V(g)$ in Eq. (1.18) can be formally defined as the Hardy-Klause variation:

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

In which the L_p *star discrepancy* of a design $\mathbf{X}_n$ is 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 $F_{\mathbf{U}}$:

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

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

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

Let us point out that this star discrepancy is equivalent to the Kolmogorov-Smirnov statistic, 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 Appendix D and used for efficient sampling in Chapter 4 of this manuscript.

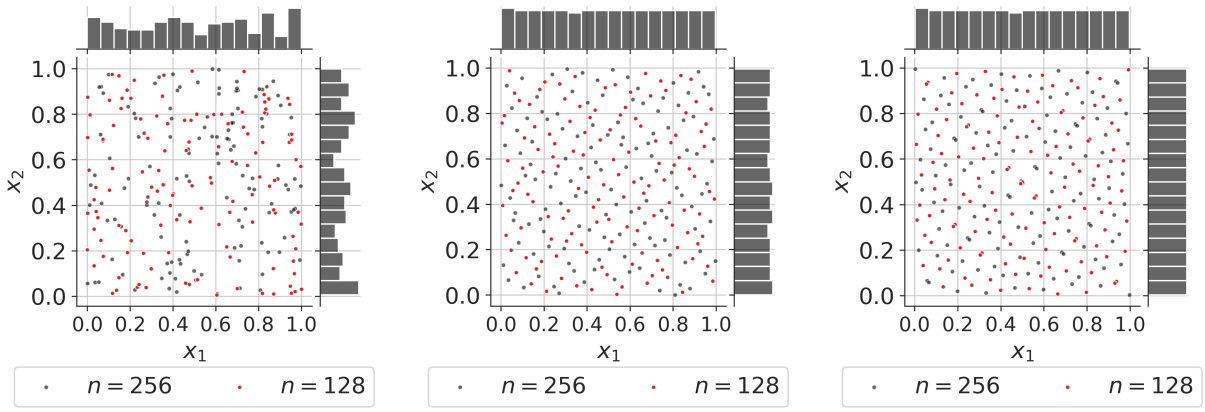
Some famous low-discrepancy sequences (e.g., van der Corput, Halton, Sobol', Faure, etc.) can offer a bounded star discrepancy $D^*(\mathbf{X}_n) \leq \frac{C\log(n)^d}{n}$, with the constant C depending on the sequence. Therefore, using these sequences as a quadrature rule with uniform weights provides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

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

The generation of such sequences does not 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.

Halton sequences in medium dimension, unfortunately, introduce pathological patterns when looking at their subprojections. To overcome these limits, digital nets such as the famous Sobol' or Faure sequences were 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 size values. Typically, designs with sizes equal to powers of two or power of prime numbers will be favorable. To illustrate the different repartition and properties of the methods, Fig. 1.5 represents the three Monte Carlo and quasi-Monte Carlo designs (with size $n = 256$). Each is split into the first 128 points (in red) and the following 128 points (in black) to show to nested properties of the QMC sequences.

Crude Monte Carlo estimators provide guarantees associated with the estimate. This complementary information is essential to deliver an end-to-end uncertainty quantification, which is missed in deterministic QMC methods. *Randomized quasi-Monte Carlo* (RQMC) is a method introducing 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).



(a) Monte Carlo (b) Halton sequence (c) Sobol' sequence

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

Ultimately, quasi-Monte Carlo methods generate powerful integration schemes. The Koksma-Hlawka inequality associates an upper bound and a convergence rate to most integrals. A randomization overlay fades the deterministic property of these designs, allowing us to compute confidence intervals. In the following, sampling techniques are presented from the numerical *design of experiments* point of view. Even if the goal might look different from numerical integration, these two topics share many methods and concepts.

`OpenTURNS` 2 (Numerical integration)**.** The Python code available in Appendix D proposes a minimalistic `OpenTURNS` example to build multivariate quadrature rules. Figures illustrating the present section may be reproduced, using the `OpenTURNS` scripts available on GitHub[5].

1.4.2 Numerical design of experiments

The numerical design of experiments aims at uniformly exploring the input domain, e.g., to build a learning set for a regression model, or to initialize a multi-start optimization strategy. A design of experiment (also simply called design) is qualified as *space-filling* when it uniformly covers a domain. As well as in integration, a design is used to propagate uncertainties through a numerical model (or a physical experiment). However, a difference comes from the fact that this community often works with designs of very limited size. Users of designs of experiments also consider various properties.

- Some might be interested in the sequentiality of a sampling method, to eventually add new points as they get their computational budget extended.
- 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 variables (see the following Section 1.6 on global sensitivity analysis).

[5]https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/integration.ipynb

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

This section will first define a few space-filling metrics. Secondly, the *Latin hypercube sampling* (LHS) will be introduced as a variance-reduction method that became popular in the UQ 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 to optimize their performances. In the previous section, the star discrepancy was introduced as a distance of a finite design to uniformity. However, the L_∞ star discrepancy is hard to estimate, fortunately, Warnock (1972) elaborated an explicit expression specific to the L_2 star discrepancy:

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

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: the star L_2 discrepancy generates designs that are not robust to projections in sub-spaces; it is not an invariant metric by rotation and reflection; and finally, by construction, L_p discrepancies give a disproportionate 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 explicit formula of the centered L_2 discrepancy:

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

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

minimum distance as $p \geq 1$ tends to infinity:

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

More space-filling criteria are reviewed in Abtini (2018) and in Appendix A from **?**. Further relations between some mathematical objects related to space-filling are developed in **?**. These space-filling metrics are widely used to optimize different sampling techniques.

Latin hypercube sampling

Latin hypercube sampling is a method introduced in 1979 (Mckay et al., 1979), initially for numerical integration. In a bounded domain, this stratified sampling technique forces the distribution of each sub-projection to be as uniform as possible. To do so, for an n-sized design, each marginal domain is divided into n identical segments. This creates a regular grid of n^d squared cells over the domain.

Then, a Latin hypercube design (LHD) does not allow more than one point within a segment. That way, new LHDs can be built as a permutation of the marginals of an existing LHD. Inside each selected cell from the grid, the point can be placed at the center or randomly.

Various contributions proposed a variance, and a central limit theorem to LHS (Koehler and Owen, 1996). Similarly to the Monte Carlo variance in Eq. (1.16), LHS variance can be expressed as:

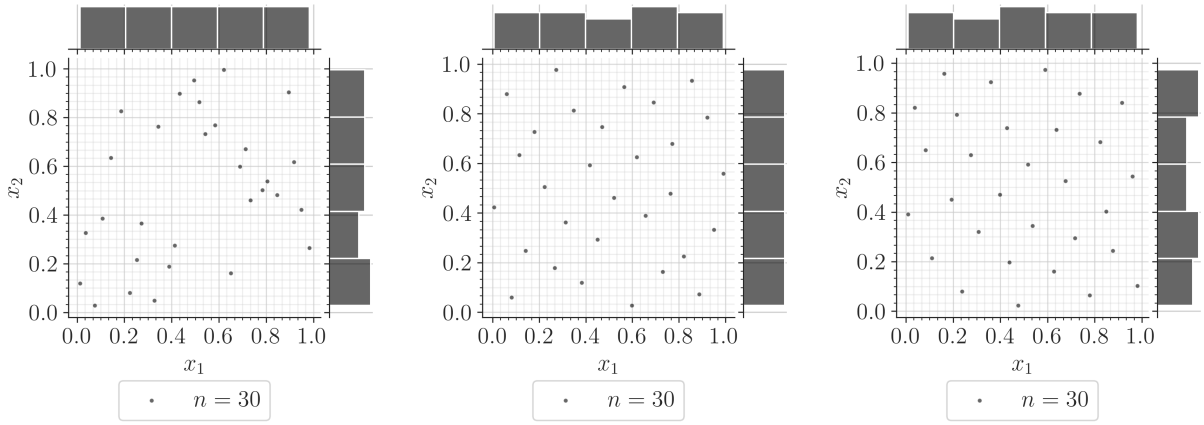
$$\mathrm{Var}\left(\overline{y}_n^{\mathrm{LHS}}\right) = \frac{1}{\sqrt{n}} \mathrm{Var}\left(g(\mathbf{X})\right) - \frac{C}{n} + o\left(\frac{1}{n}\right). \tag{1.26}$$

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., the illustration in Fig. 1.6a). The following section presents various methods aiming at optimizing LHDs.

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. Fig. 1.6 illustrates two LHD, optimized by the L_2 centered discrepancy and the geometrical ϕ_p. The space-filling difference is not obvious in two-dimensional problems, and they both spread uniformly.

More recent work developed different ways to optimize LHD. Among them, let us mention the maximum projection designs from Joseph et al. (2015) which rely on the optimization of a geometrical criterion and deliver interesting performances. In the same vein, the uniform projection designs from Sun et al. (2019) is also a method to optimize LHS, this time based on a criterion averaging discrepancies between each pair of marginals.



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

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

OpenTURNS 3 (Design of experiments)**.** The Python code available in Appendix D proposes a minimalistic `OpenTURNS` example to build an LHS and an LHS optimized w.r.t. to a space-filling metric (here the L2-centered discrepancy) using the simulated annealing algorithm. Figures illustrating the present section may be reproduced, using the `OpenTURNS` scripts available on GitHub[6].

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 is bounded and the targeted measure is 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, sometimes called *probabilistic integration* (Briol et al., 2019) as part of *probabilistic numerics* (?).

To generate i.i.d samples following any distribution (i.e., non-uniform), one may use *inverse transform* sampling. After generating samples in the unit hypercube, the inverse CDF function (i.e., quantile function) is applied to the marginals. Finally, possible dependence effects may be added using the Sklar theorem Eq. (1).

[6]https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/designofexperiments.ipynb

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 discrepancies with respect to the target distribution. However, this result shows practical limits, sometimes making the interpretation of the last discrepancy easier. This question will be further discussed in Chapter 4, using a more general framework.

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 sometimes be generated with a different technique. The *acceptance-rejection* method offers a versatile generation only based on the PDF $f_{\mathbf{x}}$. Assuming that a well-known proposal PDF $f_{\mathbf{x}}^*$ exists such that $f_{\mathbf{x}} \leq c \times f_{\mathbf{x}}^*, c \in [1, +\infty]$. One may generate a sample according to $c \times f_{\mathbf{x}}^*$ and only retain from this sample the points under the PDF $f_{\mathbf{x}}$. Note that some sampling methods, such as QMC, are not well suited to acceptance-rejection since their 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 was 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 they actually 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 toward specific areas of the output random variables. When using uncertainty propagation to perform 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 Uncertainty propagation for rare event estimation

This section aims to present 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 must be highly reliable, the occurrence of such an event is qualified as rare. Only a small amount of extreme input conditions or an unlikely unfavorable combination of inputs leads 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

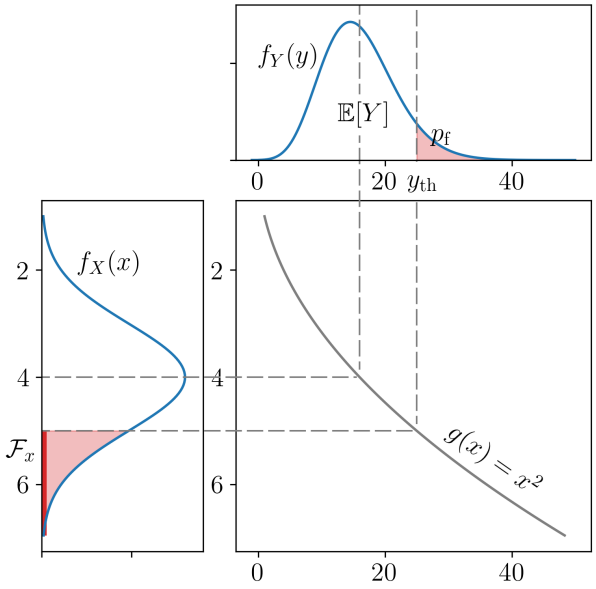


Figure 1.7 One-dimensional reliability analysis example

an event is often decomposed as a product of its likelihood and its consequences. 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 signaling, 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 α-quantile q_α of the output random variable Y is defined as:

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

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

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

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

In the following section, the formalism for reliability analysis problems will be first presented, then the main methods for solving this specific problem will be introduced. Note however that the present work will not address the problems of time-dependent reliability analysis tackled in **?**.

1.5.1 Problem statement

Following 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 the safety threshold $y_{\text{th}} \in \mathbb{R}$. Formally, the *limit-state function* (LSF) is a deterministic function $g : \mathbb{R} \to \mathbb{R}$ quantifying this performance:

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

Depending on the sign of its images, this function splits the input 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:

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

The border between these two domains is a hypersurface called *limit-state surface* (LLS), defined by $\mathcal{F}_{\mathbf{x}}^0 := \{\mathbf{x} \in \mathcal{D}_{\mathbf{x}} | g(\mathbf{x}) = 0\}$. Similarly to any UQ study using a numerical model, this problem may require to be resolved using a limited number of calls to a black-box simulator. The difficulties in a reliability problem 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 a system of problems in series and parallel.

A rare event estimation results from a particular uncertainty propagation through the LSF. Considering the output variable of interest $g(\mathbf{X})$, its probability of being negative (i.e., in the failure domain) is a common risk measure. The so-called *failure probability*, denoted by p_{f}, is the quantity of interest for reliability analysis considered in this work. This quantity is formally written[7]:

$$p_{\text{f}} = \mathbb{P}(Y \geq y_{\text{th}}) = \mathbb{P}(g(\mathbf{X}) \leq 0) = \int_{\mathcal{F}_{\mathbf{x}}} f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_{\mathcal{D}_{\mathbf{x}}} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{1.31}$$

were the indicator function applied to the failure domain returns $\mathbb{1}_{\{\mathcal{F}_{\mathbf{X}}\}}(x) = 1$ if $x \in \mathcal{F}_{\mathbf{X}}$ and $\mathbb{1}_{\{\mathcal{F}_{\mathbf{X}}\}}(x) = 0$ 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 their failure probability has an order of magnitude between $10^{-2} \leq p_{\text{f}} \leq 10^{-9}$ (see e.g., **?**).

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 caused by potentially unscaled or asymmetric marginals. Moreover, a larger panel of methods can be applied in the standard space since the random inputs become independent. The bijective mapping between these two spaces is called an "iso-probabilistic transformation", denoted by $T : \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d \to \mathbb{R}^d, \mathbf{x} \mapsto T(\mathbf{X}) = \mathbf{u} = (u_1, \ldots, u_d)^\top$.

[7]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 $f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathrm{d}\mathbb{P}_{\mathbf{X}}(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{D}_{\mathbf{x}}$.

When considering any random vector $\mathbf{X} = (X_1, \ldots, X_d)^\top$ and the independent standard Gaussian vector $\mathbf{U} = (U_1, \ldots, u_d)^\top$, the following equalities hold:

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

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

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

Since this transformation is a diffeomorphism[8], 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:

$$p_\mathrm{f} = \mathbb{P}(\check{g}(\mathbf{U}) \leq 0) = \int_{\mathcal{F}_\mathbf{u}} \varphi_d(\mathbf{u})\, \mathrm{d}\mathbf{u} = \int_{\mathbb{R}^d} \mathbb{1}_{\mathcal{F}_\mathbf{u}}(\mathbf{u}) \varphi_d(\mathbf{u})\, \mathrm{d}\mathbf{u}, \tag{1.34}$$

with the transformed failure domain denoted by $\mathcal{F}_\mathbf{u} = \{\mathbf{u} \in \mathbb{R}^d \,|\, \check{g}(\mathbf{u}) \leq 0\}$, 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 **?**. 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:

$$p_\mathrm{f} = \int_{\mathcal{D}_\mathbf{x}} \mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{x}) f_\mathbf{X}(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{1.35}$$

In the context of rare event estimation using costly to evaluate numerical models, the simulation budget is often limited to n runs with $p_\mathrm{f} \ll \frac{1}{n}$. This explains the need for specific methods offering approximations or simulations targeting the unknown failure domain. Two

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

types of rare event estimation methods are classically presented: first, using approximation approaches, and second, using sampling techniques. This section introduced the commonly used rare event methods, see **?** for a more exhaustive review.

First and second order reliability methods (FORM/SORM)

The well-known first and second-order reliability methods (FORM and SORM) both rely on a geometric approximation to estimate a failure probability (**?**). They extrapolate a local approximation of the LSF built in the vicinity of a *most-probable-failure-point* (MPFP), also called *design point*.

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:

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^d}{\arg\max} \left(\mathbb{1}_{\mathcal{F}_\mathbf{u}}(\mathbf{u})\varphi_d(\mathbf{u})\right). \tag{1.36}$$

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

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

$$= \underset{\mathbf{u}\in\mathbb{R}^d}{\arg\min}\, \mathbf{u}^\top\mathbf{u} \qquad \text{s.t.} \quad \breve{g}(\mathbf{u}) \leq 0. \tag{1.38}$$

This problem then becomes a quadratic optimization under nonlinear constraint. It is classically solved by gradient descent algorithms (e.g., Abdo-Rackwitz algorithm **?**) but can also use gradient-free techniques (e.g., Cobyla algorithm **?**). This point 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 presents an exponential decay in its radial and tangential directions. 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 another risk measure, defined as the *Hasofer-Lind reliability index* (**?**), $\beta\in\mathbb{R}$ such that:

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

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

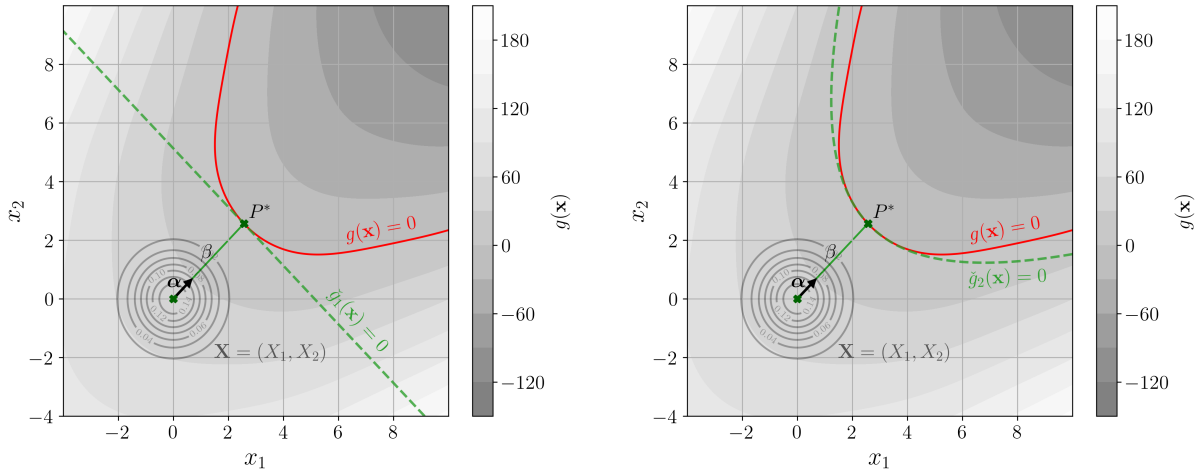
Then, FORM aims at approximating the limit-state function $\breve{g}(\cdot)$ by its first-order Taylor expansion around the MPFP, denoted $\breve{g}_1(\mathbf{u}^*)$:

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

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

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

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 purely 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 **?** for their estimation).



(a) FORM approximation

(b) SORM approximation

Figure 1.8 FORM and SORM approximation on a two-dimensional reliability problem

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 zone is 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 areas. The *muti-FORM* algorithm (see **?**) prevents this situation by applying successive FORM. Once the first MPFP $P^{*(1)}$ is found, the LSS is modified by removing a nudge to find to following MPFP $P^{*(2)}$, positioned at a similar distance but in a different direction.

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 budgets. **?** illustrates the efficiency of FORM approaches on industrial cases such as probabilistic fatigue damage. However, these methods present serious limits as the dimension increases (see the discussion in Chapter 1 from **?**). Additionally, their main drawback is the lack of complementary information concerning the confidence of the results. The textbook example illustrated in Fig. 1.9 shows 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 the analyst with an assessment of the estimation's confidence.

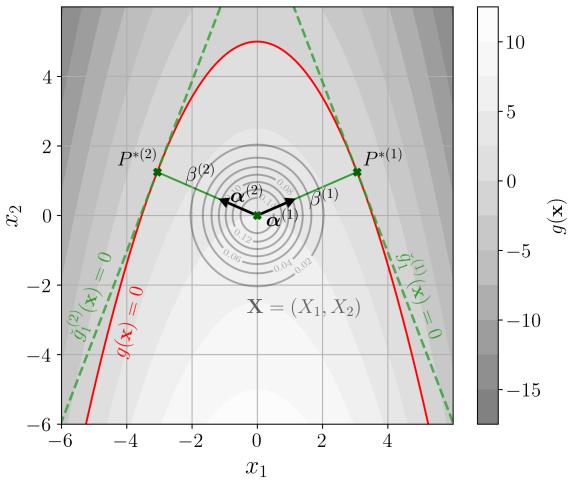


Figure 1.9 Multi-FORM approximation on an example with two MPFPs

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 i.i.d. samples $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} f_{\mathbf{X}}$. Only the estimator is now written using the indicator function applied to the LSF:

$$p_{\text{f}} \approx \widehat{p}_{\text{f}}^{\,\text{MC}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}^{(i)}). \tag{1.42}$$

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

$$\text{Var}\left(\widehat{p}_{\text{f}}^{\,\text{MC}}\right) = \frac{1}{n} p_{\text{f}}(1 - pf). \tag{1.43}$$

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:

$$\delta_{\widehat{p}_{\text{f}}^{\,\text{MC}}} = \frac{\sqrt{\text{Var}\left(\widehat{p}_{\text{f}}^{\,\text{MC}}\right)}}{\mathbb{E}[\widehat{p}_{\text{f}}^{\,\text{MC}}]} = \sqrt{\frac{1 - p_{\text{f}}}{n p_{\text{f}}}}. \tag{1.44}$$

In theory, Monte Carlo estimation 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 an embarrassingly parallel method since each of the numerical simulations is independent. Finally, it offers strong convergence guarantees and

complementary information on the estimation confidence. These properties often make Monte Carlo the reference method in rare event estimation benchmarks.

However, the advantages of this estimator are shadowed by its slow convergence. To estimate a target failure probability $p_\mathrm{f} = 10^{-\alpha}$, a Monte Carlo estimation with a convergence level $\delta_{\widehat{p}_\mathrm{f}^{\,\mathrm{MC}}} = 0.1$ 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 $h_\mathbf{X}$ (also called *auxiliary* distribution) to replace the distribution $f_\mathbf{X}$. Then, by introducing the fully known *likelihood ratio* $w_\mathbf{X}(\mathbf{x}) = \frac{f_\mathbf{X}(\mathbf{x})}{h_\mathbf{X}(\mathbf{x})}$, one can rewrite $f_\mathbf{X}(\mathbf{x}) = w_\mathbf{X}(\mathbf{x})\, h_\mathbf{X}(\mathbf{x})$ and inject it in the failure probability expression:

$$p_\mathrm{f} = \int_{\mathcal{D}_\mathbf{x}} \mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{x})\, f_\mathbf{X}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_{\mathcal{D}_\mathbf{x}} \mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{x})\, w_\mathbf{X}(\mathbf{x})\, h_\mathbf{X}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{1.45}$$

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 $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} h_\mathbf{X}$:

$$\widehat{p}_\mathrm{f}^{\,\mathrm{IS}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{x}^{(i)})\, w_\mathbf{X}(\mathbf{x}^{(i)}). \tag{1.46}$$

Similarly to Monte Carlo, this estimator is unbiased, however, its variance is defined as:

$$\mathrm{Var}\left(\widehat{p}_\mathrm{f}^{\,\mathrm{IS}}\right) = \frac{1}{n}\left(\mathbb{E}_h\left[\left(\mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{X})\, \frac{f_\mathbf{X}(\mathbf{X})}{h_\mathbf{X}(\mathbf{X})}\right)^2\right] - p_\mathrm{f}^2\right). \tag{1.47}$$

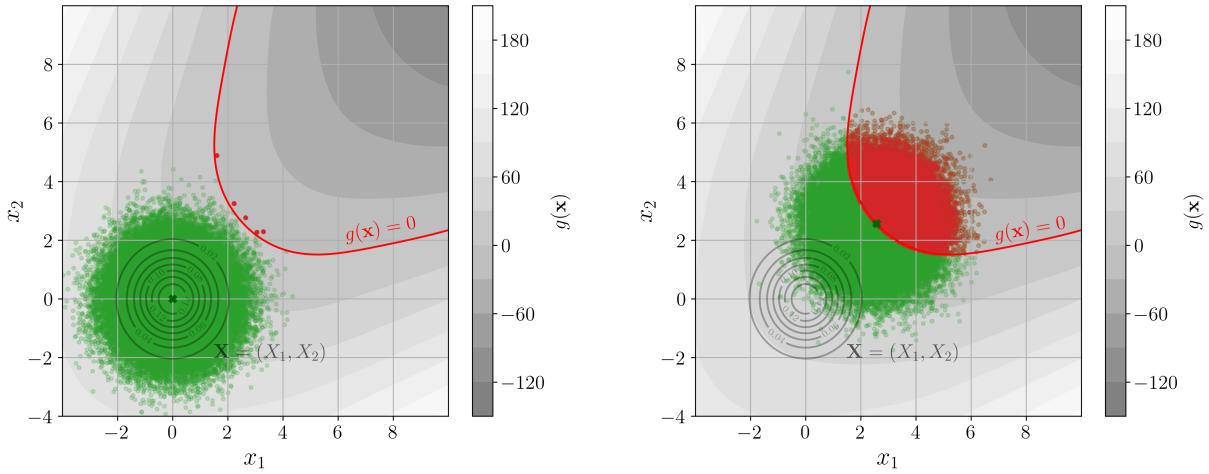
The quality of the variance reduction associated to this technique fully depends on the choice of the instrumental distribution. In fact, importance sampling can lead to higher variance than crude Monte Carlo when the instrumental distribution is poorly chosen (?). However, an optimal instrumental distribution h_opt theoretically gives the smallest variance by setting it equal to

zero in Eq. (1.47):

$$h_{\text{opt}}(\mathbf{x}) = \frac{\mathbb{1}_{\mathcal{F}_\mathbf{x}}(\mathbf{x}) f_\mathbf{X}(\mathbf{x})}{p_\text{f}}. \tag{1.48}$$

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. An important review of the use of importance sampling in the context of reliability analysis was proposed by **?**.

The most immediate solution is to combine the information provided by the results of FORM with importance sampling, simply called FORM-IS. In practice, the instrumental distribution is defined as the initial distribution centered on the design point resulting from FORM. Fig. 1.10 illustrates in the same two-dimensional case, the estimation by Monte Carlo and importance sampling centered on the design point. The points in red reached the failure domain and their number seems insufficient for Monte Carlo. Note that comparing the results from FORM and FORM-IS allows us to assess the nonlinearity of the LSF in the vicinity of the design point. This strategy is simple to implement, but it inherits the main drawbacks of FORM, such as the limits related to multiple failure areas (see the example illustrated in Fig. 1.9). Finally, other importance sampling schemes integrate adaptive mechanisms, progressively leading the sampling towards the failure domain (**?**).



(a) Monte Carlo ($n = 10^5$)

(b) Importance sampling ($n = 10^5$)

Figure 1.10 Illustration of a rare event estimation.

Adaptive importance sampling by cross-entropy

The *cross-entropy-based adaptive importance sampling* (CE-AIS) is an adaptive strategy, optimizing the IS variance reduction by searching for the best instrumental distribution within a parametric family. Let us consider the distribution $h_{\boldsymbol{\lambda}}$, belonging to the parametric family $\mathcal{H}_{\boldsymbol{\lambda}}$, defined as:

$$\mathcal{H}_{\boldsymbol{\lambda}} = \left\{ \mathbf{x} \longmapsto h_\mathbf{X}(\mathbf{x}|\boldsymbol{\lambda}) = h_{\boldsymbol{\lambda}}(\mathbf{x}), \quad \boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p) \in \mathcal{D}_{\boldsymbol{\lambda}} \subseteq \mathbb{R}^p \right\}. \tag{1.49}$$

The early work of **?** only included normal distributions to minimize the IS variance w.r.t. the parameter $\boldsymbol{\lambda}$, using Eq. (1.47) the optimization simplifies as:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \mathcal{D}_\lambda}{\arg\min}\, \mathbb{E}_{h_\lambda}\left[\mathbb{1}_{\mathcal{F}_\mathrm{x}}(\mathbf{X})\, w_\mathrm{X}(\mathbf{X})^2\right]. \tag{1.50}$$

However, this optimization strategy requires sampling with respect to the instrumental distribution at each optimization iteration, which was overcome by a different approach.

The "cross-entropy" (CE) method uses Kullback-Leibler (KL) divergence to optimize importance sampling. KL divergence is a dissimilarity measure between distributions, expressed between the parametric instrumental distribution $h_{\boldsymbol{\lambda}}$ and the optimal one h_opt:

$$D_\mathrm{KL}\left(h_\mathrm{opt} || h_{\boldsymbol{\lambda}}\right) = \int_{\mathcal{D}_\mathrm{x}} \log\left(\frac{h_\mathrm{opt}(\mathbf{x})}{h_{\boldsymbol{\lambda}}(\mathbf{x})}\right) h_\mathrm{opt}(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{1.51a}$$

$$= \int_{\mathcal{D}_\mathrm{x}} \log\left(h_\mathrm{opt}(\mathbf{x})\right) h_\mathrm{opt}(\mathbf{x})\, \mathrm{d}\mathbf{x} - \int_{\mathcal{D}_\mathrm{x}} \log\left(h_{\boldsymbol{\lambda}}(\mathbf{x})\right) h_\mathrm{opt}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{1.51b}$$

? symplify the expression of the optimization problem minimizing the KL divergence, which is most often convex and differentiable w.r.t. $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \mathcal{D}_\lambda}{\arg\min}\, D_\mathrm{KL}\left(h_\mathrm{opt} || h_{\boldsymbol{\lambda}}\right). \tag{1.52}$$

By injecting the expression in Eq. (1.51b), the optimization problem simply becomes a function of an expected value over the initial density f_X:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \mathcal{D}_\lambda}{\arg\max} \int_{\mathcal{D}_\mathrm{x}} \log\left(h_{\boldsymbol{\lambda}}(\mathbf{x})\right) h_\mathrm{opt}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \underset{\boldsymbol{\lambda} \in \mathcal{D}_\lambda}{\arg\max}\, \mathbb{E}_{f_\mathrm{X}}\left[\mathbb{1}_{\mathcal{F}_\mathrm{x}}(\mathbf{X}) \log\left(h_{\boldsymbol{\lambda}}(\mathbf{X})\right)\right]. \tag{1.53}$$

To directly estimate this expected value, the failure probability p_f should not be too rare, which allows to use of an empirical estimator of the expected value:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \mathcal{D}_\lambda}{\arg\max} \sum_{i=1}^{n} \mathbb{1}_{\mathcal{F}_\mathrm{x}}\left(\mathbf{x}^{(i)}\right) \log\left(h_{\boldsymbol{\lambda}}\left(\mathbf{x}^{(i)}\right)\right), \quad \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} f_\mathrm{X}. \tag{1.54}$$

Eventually, this optimization can be solved by canceling the gradient:

$$\sum_{i=1}^{n} \mathbb{1}_{\mathcal{F}_\mathrm{x}}\left(\mathbf{x}^{(i)}\right) \left[\nabla \log\left(h_{\boldsymbol{\lambda}^*}\right)\right]\left(\mathbf{x}^{(i)}\right) = \mathbf{0}. \tag{1.55}$$

According to **?**, this system of equations has a unique analytical solution when assuming that the instrumental distribution belongs to the "natural exponential family".

However, when dealing with rare probabilities, the empirical estimation does not draw enough points in the failure domain to get an accurate estimate. The adaptive version of this technique, called *multilevel cross-entropy*, gradually builds a set of intermediate levels, decreasing

towards the failure level (equal to zero). By working on a set of individually less rare events, the empirical estimation in Eq. (1.53) is made possible.

The algorithm starts by generating and evaluating an initial sample $\left\{g(\mathbf{X}_{[1]}^{(i)})\right\}_{i=1}^{n}$, on which a threshold level $q_{[1]}^{p_0}$ is computed as the empirical p_0-quantile. Using the samples below the first threshold $q_{[1]}^{p_0}$, a first instrumental distribution $h_{\boldsymbol{\lambda}_{[1]}^*}$ is optimized. At the next steps $k \in \{1, \ldots, k_\#\}$, the sample $\left\{\mathbf{X}_{[k]}^{(i)}\right\}_{i=1}^{n}$ is generated from the density $h_{\boldsymbol{\lambda}_{[k-1]}^*}$ and the rest of the process repeats until the estimated threshold level becomes negative, $q_{[k_\#]}^{p_0} \leq 0$.

The final instrumental density $h_{\boldsymbol{\lambda}_{[k_\#]}^*}$ is then considered for importance sampling as defined in Eq. (1.46):

$$\widehat{p}_{\mathrm{f}}^{\,\mathrm{CE-AIS}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq 0\}} \frac{f_{\mathbf{X}}\left(\mathbf{x}_{[k_\#]}^{(i)}\right)}{h_{\boldsymbol{\lambda}_{[k_\#]}^*}\left(\mathbf{x}_{[k_\#]}^{(i)}\right)}, \qquad \left\{\mathbf{X}_{[k_\#]}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} h_{\boldsymbol{\lambda}_{[k_\#]}^*}. \tag{1.56}$$

CE-AIS is widely used in rare event estimation, as it develops an adaptive technique while conserving the explicit IS variance given in Eq. (1.47). According to **?**, the successive instrumental distributions $h_{\boldsymbol{\lambda}_{[k]}^*}$ converge towards h_{opt} under a few hypotheses. The most important one is that the optimal density must belong to the parametric family considered, which should offer enough flexibility to describe a wide range of distributions.

When the failure domain is composed of multiple regions, different improvements of the CE-AIS were proposed. **?** proposed to optimize $h_{\boldsymbol{\lambda}^*}$ among a mixture of Gaussian distributions. This method was further studied by **?** and **?** using advanced mixtures in the standard space. However, when using mixtures, the optimization problem does not have an analytical expression anymore (**?**).

In the parametric framework, the family choice leads to a complicated trade-off between optimization complexity and flexibility allowed by the family. A similar mechanism is used by other importance sampling methods, inferring the optimal instrumental density by applying kernel density estimation to the points in the failure domain.

Nonparametric adaptive importance sampling

The use of multivariate kernel density estimation (KDE) to approximate the importance sampling optimal density h_{opt} was introduced in the context of structural reliability by **?**, latter followed by **?**. Let us first present the nonparametric importance sampling from **?**, considering the instrumental density $h_{[0]}$ (for now, $h_{[0]} \neq f_{\mathbf{X}}$), on which a sample $\left\{\mathbf{X}_{[1]}^{(i)}\right\}_{i=1}^{n}$ is generated. A first failure probability can be roughly estimated, assuming that enough samples lead to the failure domain:

$$\widehat{p}_{\mathrm{f}\,[1]} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x}\leq 0)\}}\left(\mathbf{x}_{[1]}^{(i)}\right) \frac{f_{\mathbf{X}}\left(\mathbf{x}_{[1]}^{(i)}\right)}{h_{[0]}\left(\mathbf{x}_{[1]}^{(i)}\right)} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq 0\}}\left(\mathbf{x}_{[1]}^{(i)}\right) w_{[1]}^{(i)}. \tag{1.57}$$

On this biased sample, another density can be fitted using KDE, using the previously defined $\widehat{p_{\mathrm{f}}}_{[1]}$ as a normalization term:

$$\widehat{h}_{[1]}(\mathbf{x}) = \frac{\det(\mathbf{H}_{[1]})^{-1/2}}{n\,\widehat{p_{\mathrm{f}}}_{[1]}} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq 0\}}\left(\mathbf{x}_{[1]}^{(i)}\right) w_{[1]}^{(i)}\, K\left(\mathbf{H}_{[1]}^{-1/2}(\mathbf{x}-\mathbf{x}_{[1]}^{(i)})\right). \tag{1.58}$$

Where the kernel K is commonly taken as the multivariate Gaussian-centered density with the covariance matrix $\mathbf{H}$. The tuning of $\mathbf{H}$ is usually done by minimizing an asymptotic mean integrated squared error (AMISE) criterion (**?**). In the previous expression, the normalization constant ensures building a probability density while the weights $w_{[1]}^{(i)}$, defined above, reflect the contribution of each point to $\widehat{p_{\mathrm{f}}}_{[1]}$. After performing this KDE, the estimated density can be used as instrumental density in Eq. (1.46).

As for the CE-IS methods, the risk is that barely any points sampled from the instrumental density $h_{[0]}$ hit the failure domain, leading to poor estimates. **?** proposed to couple an adaptive mechanism with a nonparametric inference of the optimal density. This method is further referred to as NAIS for *nonparametric adaptive importance sampling*. Later, the NAIS method was adapted by **?** to the reliability analysis problem, using a similar mechanism to the CE-AIS method.

In this framework, a series of intermediate thresholds are computed as empirical p_0-quantiles $q_{[1]}^{p_0} > \cdots > q_{[k_\#]}^{p_0}$ of the successive importance sampling steps. This algorithm is initiated by setting $h_{[0]} = f_{\mathrm{X}}$ and stops at the step $k_\#$, when $q_{[k_\#]}^{p_0} < 0$.

At the step k, the intermediate normalization constant is written as:

$$\widehat{p_{\mathrm{f}}}_{[k]} = \frac{1}{kn}\sum_{j=1}^{k}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq q_{[j]}^{p_0})\}}\left(\mathbf{x}_{[j]}^{(i)}\right) \frac{f_{\mathrm{X}}\left(\mathbf{x}_{[j]}^{(i)}\right)}{\widehat{h}_{[j-1]}\left(\mathbf{x}_{[j]}^{(i)}\right)} = \frac{1}{kn}\sum_{j=1}^{k}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq q_{[j]}^{p_0}\}}\left(\mathbf{x}_{[j]}^{(i)}\right) w_{[j]}^{(i)}, \tag{1.59}$$

with $\left\{\mathbf{X}_{[j]}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} h_{[j-1]}$. Then, an intermediate instrumental density is inferred by KDE on the samples exceeding the threshold $q_{[k]}^{p_0}$ such that:

$$\widehat{h}_{[k+1]}(\mathbf{x}) = \frac{\det(\mathbf{H}_{[k]})^{-1/2}}{kn\,\widehat{p_{\mathrm{f}}}_{[k]}} \sum_{j=1}^{k}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq q_{[j]}^{p_0}\}}\left(\mathbf{x}_{[j]}^{(i)}\right) w_{[j]}^{(i)}\, K\left(\mathbf{H}_{[k]}^{-1/2}(\mathbf{x}-\mathbf{x}_{[j]}^{(i)})\right). \tag{1.60}$$

The last instrumental density $\widehat{h}_{[k_\#]}$ is finally considered as an approximation of the optimal density for importance sampling introduced in Eq. (1.46):

$$\widehat{p_{\mathrm{f}}}^{\mathrm{NAIS}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x})\leq 0\}} \frac{f_{\mathrm{X}}\left(\mathbf{x}_{[k_\#]}^{(i)}\right)}{\widehat{h}_{[k_\#]}\left(\mathbf{x}_{[k_\#]}^{(i)}\right)}, \qquad \left\{\mathbf{X}_{[k_\#]}^{(i)}\right\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} h_{k_\#}. \tag{1.61}$$

Overall, the NAIS offers more flexibility to infer the optimal importance sampling density. This property might suit problems presenting a highly nonlinear limit state function. Then,

relying on importance sampling still provides an expression of the estimator's variance, by adapting Eq. (1.47) to the recurrent mechanism in NAIS. Because this approach depends on KDE, it inherits its drawbacks. As discussed in **?**, tuning the KDE can create numerical issues and KDE famously suffers from the curse of dimension. In practice, the performances of NAIS seriously decrease for problems in dimensions larger than ten.

Subset sampling

Although the concept of splitting already existed, the name of *subset sampling* (SS) was first introduced by **?** in the structural reliability community. This concept was generalized as a sequential Monte Carlo method under the name of "adaptive multilevel splitting", as reviewed by **?**.

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

$$p_{\mathrm{f}} = \mathbb{P}(\mathcal{F}_{\mathbf{x}}) = \mathbb{P}(\cap_{k=1}^{k_\#} \mathcal{F}_{[k]}) = \prod_{k=1}^{k_\#} \mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}). \tag{1.62}$$

From a practical point of view, the analyst tunes the algorithm[9] by setting the intermediary probabilities $\mathbb{P}(\mathcal{F}_{[k]}|\mathcal{F}_{[k-1]}) = p_0, \forall k \in \{1, \ldots, k_\#\}$. Then, the corresponding quantiles $q_{[1]}^{p_0} > \cdots > q_{[k_\#]}^{p_0}$ 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 $f_{\mathbf{X}}$. Following conditional subset samples are generated by *Monte Carlo Markov Chain* (MCMC) sampling of $f_{\mathbf{X}}(\mathbf{x}|\mathcal{F}_{[k-1]})$, using as seeds initialization points the $n = Np_0$ samples given by $\mathbf{A}_{[k],n} = \{\mathbf{X}_{[k-1]}^{(i)} \subset \mathbf{X}_{[k-1],N} | g(\mathbf{X}_{[k-1]}^{(i)}) > \widehat{q}_{[k-1]}^{p_0}\}_{i=1}^{n}$. This process is repeated until an intermediary quantile becomes negative: $\widehat{q}_{[k_\#]}^{p_0} < 0$. Finally, the failure probability is estimated by:

$$\widehat{p_{\mathrm{f}}}^{\mathrm{SS}} = p_0^{k_\#-1} \frac{1}{N} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{x}) \leq 0\}}(\mathbf{X}_{[k_\#],N}^{(i)}). \tag{1.63}$$

In practice, the subset sample size should be large enough to properly estimate intermediary quantiles, leading to the usual recommendation of $p_0 = 0.1$. Fig. 1.11 illustrates the consecutive subset samples moving towards the failure domain. At each step of the algorithm (corresponding to a color), a subset is generated and an intermediate quantile is estimated.

? also provide bounds to the coefficient of variation of $\widehat{p_{\mathrm{f}}}^{\mathrm{SS}}$. The first one results from a first-order Taylor expansion of Eq. (1.63) and is often considered as an upper bound. The second assumes the estimations of the conditional probabilities to be independent and tends to underestimate the coefficient of variation.

[9] An algorithmic presentation of the generic subset sampling method is given in Appendix C.

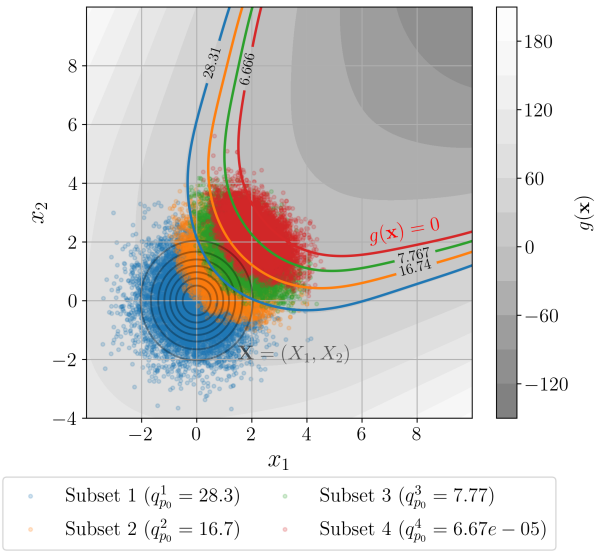


Figure 1.11 Illustration of a rare event estimation by subset sampling ($n = 4 \cdot 10^4, p_0 = 0.1$).

As discussed in (**?**), the efficiency of the SS method depends on the choice and tuning of the MCMC algorithm. The Metropolis–Hastings (MH) algorithm is widely used as MCMC algorithm for subset sampling, however, it quickly becomes inefficient as the dimension increases. Different improvements of the MH are made possible by working in the standard space, such as the "component-wise" (or "modified M–H"). More recently, alternative MCMC methods including physical system dynamics (e.g., Hamiltonian MCMC) showed promising results in high-dimension reliability problems (**?**).

The subset sampling is a versatile method, presenting consistent performances even for rare probabilities. Its flexibility allows it to deal with highly nonlinear LSF, but its drawbacks arise from the use of MCMC sampling. The convergence of MCMC is complex to control and depends on its tuning, in addition, the MCMC samples are dependent. Unlike the methods derived from importance sampling, the variance of $\widehat{p_f}^{SS}$ is only approximated.

1.5.3 Summary and discussion

This section introduced the generic formulation and the main methods for rare event estimation. Even if the problem is generic rare event estimation requires tailored solutions. Depending on the properties of the problem tackled some methods might outperform others. Beyond the one introduced previously, many more methods are worth mentioning in the field of reliability analysis, such as the *directional sampling* **?**, or *line sampling* (**?**), *moving particles* (**?**). **?** compares the advantages and drawbacks of the most commonly used one, with corresponding algorithmic descriptions and numerical benchmarks.

Overall, the main properties increasing the complexity of reliability problems are related to:

- the computational cost of the limit-state function evaluation;
- the strong nonlinearity of the limit-state function;

- the rareness of the failure event.

In regard to the methods, the estimation is made easier by algorithms with simple tuning or allowing to work in the physical space (avoiding a possibly complex iso-probabilistic transform). Considering all these elements the analyst may set up a sampling strategy, possibly coupled with the use of a surrogate model (further discussed in Section 1.7).

Nevertheless, the unified formulation of reliability analysis problems (see 1.31) is an opportunity for the community to share standardized benchmark problems. Following the well-accepted benchmark platform for optimization "Comparing Continuous Optimizers" (COCO) (**?**), an equivalent initiative was proposed for structural reliability. In 2019, the "black-box reliability challenge", was organized as a hackathon by the Dutch organization for applied scientific research (TNO) (**?**). This platform proposed a large catalog of reliability problems with their respective solutions. Most of them were encapsulated as a Python package called `otbenchmark`[10] (**?**), based on core `OpenTURNS` objects.

When working with computationally expensive numerical models, the direct use of rare event estimation methods is most often intractable. Many contributions were dedicated to the coupling of surrogate models with sampling methods for rare event estimation. **?** presented the results of a wide benchmark on the challenge from TNO, obtained by using surrogate models for reliability developed in the UQLab software (**?**).

In any case, risk assessment analysts should favor the methods offering convergence guarantees over punctual performance demonstrations. Finally, the robustness of the failure probability to the input uncertainty model is a major question, which was studied from probabilistic (**?**) and extra-probabilistic (**?**) frameworks.

`OpenTURNS` 4 (Rare event estimation)**.** The Python code available in Appendix D proposes a minimalistic `OpenTURNS` example to estimate rare event probabilities. Figures illustrating the present section may be reproduced, using the `OpenTURNS` scripts available on GitHub[11].

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 to 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

[10] https://github.com/mbaudin47/otbenchmark/

[11] https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/reliability.ipynb

the output, Two types of GSA methods exist in the literature, either proposing qualitative or quantitative approaches:

- *screening methods*: determines the non-influential variables in a UQ study (qualitative);
- *importance measures*: assess the contribution of inputs 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 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 **?**. Screening methods allow discriminating the non-influential variables, which can be considered afterward as determinist to simplify the problem.

Morris method

The Morris method (**?**) is a screening method historically used in engineering applications. It starts by mapping the input domain $\mathcal{D}_{\mathbf{X}}$ 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 design starts at a random node $\mathbf{x}^{(t)} = (x_1^{(t)}, \dots, x_j^{(t)}, \dots, x_d^{(t)})$ of the grid, and moves 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 trajectory t is expressed as a finite difference:

$$\mathrm{EE}_j^{(t)} = \frac{g(\mathbf{x}^{(t)}) - g(\mathbf{x}^{(t)} + \Delta_j)}{\Delta}. \tag{1.64}$$

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

$$\overline{\mathrm{EE}}_j = \frac{1}{n}\sum_{t=1}^{T} |\mathrm{EE}_j^{(t)}|, \qquad \widehat{\mathrm{Var}}(\mathrm{EE}_j) = \frac{1}{n-1}\sum_{t=1}^{T}\left(\mathrm{EE}_j^{(t)} - \overline{\mathrm{EE}}_j\right)^2. \tag{1.65}$$

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 **?** and further studied in **?**. As the Morris method, they study the mean value of local derivatives of the model output with regard to the inputs:

$$v_j = \int_{\mathcal{D}_{\mathbf{X}}} \left(\frac{\partial g(\mathbf{x})}{\partial x_j} \right)^2 f_{\mathbf{X}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \mathbb{E}\left[\left(\frac{\partial g(\mathbf{X})}{\partial X_j} \right)^2 \right]. \tag{1.66}$$

This continuous formulation does not require using OAT designs, which was proven to be more efficient when exploiting sampling methods such as quasi-Monte Carlo. The efficiency of the DGSMs for screening purposes was outlined in many papers (e.g., **?**). 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 (**?**).

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 us 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 an independent random vector $\mathbf{X}$. The output random variable $Y = g(\mathbf{X})$ can be decomposed, according to **?**, as:

$$Y = g(\mathbf{X}) = g_0 + \sum_{j=1}^{d} g_j(X_j) + \sum_{j<l}^{d} g_{jl}(X_j, X_l) + \cdots + g_{1\ldots d}(\mathbf{X}), \tag{1.67}$$

with the previous terms defined according to this recurrence:

$$g_0 = \mathbb{E}[g(\mathbf{X})] \tag{1.68a}$$
$$g_j(X_j) = \mathbb{E}[g(\mathbf{X})|X_j] - g_0 \tag{1.68b}$$
$$g_{jl}(X_j, X_l) = \mathbb{E}[g(\mathbf{X})|X_j, X_l] - g_j(X_j) - g_l(X_l) - g_0 \tag{1.68c}$$
$$\dots \tag{1.68d}$$

Sobol in **?** 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 FANOVA):

$$\mathrm{Var}(Y) = \sum_{j=1}^{d} V_j(Y) + \sum_{j<l}^{d} V_{jl}(Y) + \cdots + V_{1\dots d}(Y), \tag{1.69}$$

where the previous terms are defined in a recurrent way, in the same fashion as Eq. (1.68): $V_j(Y) = \mathrm{Var}\left(\mathbb{E}[Y|X_j]\right), V_{jl}(Y) = \mathrm{Var}\left(\mathbb{E}[Y|X_j, X_l]\right) - V_j(Y) - V_l(Y)$, 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.

$$S_j = \frac{V_j(Y)}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}\left(\mathbb{E}[Y|X_j]\right)}{\mathrm{Var}(Y)} \tag{1.70a}$$
$$S_{jl} = \frac{V_{jl}(Y)}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}\left(\mathbb{E}[Y|X_j, X_l]\right) - V_j(Y) - V_l(Y)}{\mathrm{Var}(Y)} \tag{1.70b}$$
$$\dots \tag{1.70c}$$

The generic definition of the Sobol' sensitivity indices associated with 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:

$$S_A = \frac{V_A(Y)}{\mathrm{Var}(Y)} = \frac{\sum_{B \subset A} (-1)^{|A|-|B|} \mathrm{Var}\left(\mathbb{E}[Y|X_B]\right)}{\mathrm{Var}(Y)}. \tag{1.71}$$

By using the functional decomposition of variance in Eq. (1.69), one can show that the Sobol' indices add up to one:

$$\sum_{A \in \mathcal{P}_d} S_A = 1. \tag{1.72}$$

The so-called *closed Sobol' index* associated to a subset of inputs $A \in \mathcal{P}_d$ (equivalent to the first-order Sobol' index of A) is defined as:

$$S_A^{\mathrm{clos}} = \sum_{A' \subset A} S_{A'} = \frac{\mathrm{Var}\left(\mathbb{E}[Y|\mathbf{X}_A]\right)}{\mathrm{Var}(Y)}. \tag{1.73}$$

Assessing Sobol' indices for every order becomes complex in medium to high dimensions. The *total Sobol' index* S_j^T associated with the variable j, see **?**, quantifies the share of output variance which is explained by all the interactions of the variable X_j:

$$S_j^T = 1 - \frac{\mathrm{Var}\left(\mathbb{E}[Y|\mathbf{X}_{-j}]\right)}{\mathrm{Var}(Y)} = \frac{\mathbb{E}\left[\mathrm{Var}(Y|\mathbf{X}_{-j})\right]}{\mathrm{Var}(Y)}, \tag{1.74}$$

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:

$$S_A^T = 1 - S_{A^{\complement}}^{\text{clos}} = 1 - \frac{\mathrm{Var}\left(\mathbb{E}[Y|\mathbf{X}_{A^{\complement}}]\right)}{\mathrm{Var}(Y)}, \quad A^{\complement} = \mathcal{P}_d \setminus A \tag{1.75}$$

By analyzing jointly the first and total Sobol' indices, one can get an indication about the 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, even if historically the *pick-freeze* scheme was the most popular. This method is based on two samples, but it often requires a prohibitive number of evaluations of 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 details in Chapter 3 of **?**. Alternatively, the surrogate models were exploited to estimate such sensitivity measures. Using an input-output dataset, the analyst may build a *polynomial chaos expansion* (PCE) surrogate model, which gives an explicit expression of the Sobol' indices (**?**). Authors such as **?** also studied the use of Gaussian processes for this purpose.

In the case of independent inputs, the first and total Sobol' indices are 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 input variables. When the inputs present a dependence structure, it becomes complicated to distinguish its effects from possible interactions. However, many authors tried to adapt Sobol's indices to this context. Chapter 5 of **?** reviews four of these approaches. For example, **?** 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 **?**, allowing to work with dependent inputs.

`OpenTURNS` 5 (Sobol' indices)**.** The Python code available in Appendix D gives a minimalistic `OpenTURNS` implementation of the Sobol' indices to assess global sensitivity analysis on the Ishigami analytical problem. Further scripts are also available on GitHub[12].

[12]https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/sensitivity_analysis.ipynb

Shapley effects

Shapley effects are an adaptation to GSA by **?** of the Shapley values from the cooperative games' theory (**?**). 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 game theory, Shapley values act as a rule on how to share the value created by a team between its members (players). The Shapley value allocated to the player X_j is given considering the indices $-\{j\} = \{1,\dots,d\}\backslash\{j\}$:

$$\varsigma_j = \sum_{A \subset -\{j\}} \binom{d-1}{\text{card}(A)}^{-1} \left(\text{val}(A \cup \{j\}) - \text{val}(A)\right), \tag{1.76}$$

where the value (or cost) function is denoted by $\text{val}(A)$, and A is a subset of $\{1,\dots,d\}$ with cardinality $\text{card}(A)$. 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:

$$Sh_j = \sum_{A \subset -\{j\}} \binom{d-1}{|A|}^{-1} \left(S^{\text{clos}}_{A\cup\{j\}} - S^{\text{clos}}_{A}\right). \tag{1.77}$$

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:

$$\sum_{j=1}^{d} Sh_j = 1. \tag{1.78}$$

In the case of independent inputs, the Shapley effects present properties related to the Sobol' indices. The following equation (see proof in **?**) reveal that the Shapley effects equally divide the interaction effects between the implicated variable:

$$S_j \leq Sh_j \leq S^T_j, \qquad Sh_j = \sum_{A \in \mathcal{P}_d, j \in A} \frac{S_A}{\text{card}(A)}. \tag{1.79}$$

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 **?**. Surrogate models were also coupled to estimate Shapley effects, using Gaussian processes in **?** and random forests in **?**.

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 output distributions. Comparing the entire distributions instead of their moments might be more robust in some cases (e.g., 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).

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 for GSA:

- directly using a dissimilarity measure to assess $\Delta(\mathbb{P}_Y, \mathbb{P}_{Y|X_j})$;
- 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 **??**. However, some f-divergences introduce estimation issues, and the resulting importance measures do not propose a functional decomposition of variance (also called FANOVA). Using kernel-based IPMs such as the maximum mean discrepancy (MMD), an alternative importance measure was proposed. The following section presents the *Hilbert-Schmidt Independence Criterion* (HSIC), which was initially introduced by **?** for dependence testing, and later adapted as a dependence measure in GSA by **?**.

Hilbert-Schmidt independence criterion

Let us first recall the definition of the maximum mean discrepancy (further discussed in Appendix D). This distance between two probability distributions π and ζ can be defined as the worst-case error for any function within a unit ball of a function space $\mathcal{H}$:

$$\mathrm{MMD}(\pi, \zeta) := \sup_{\|g\|_{\mathcal{H}(k)} \leq 1} \left| \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x}) \mathrm{d}\pi(\mathbf{x}) - \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x}) \mathrm{d}\zeta(\mathbf{x}) \right| \tag{1.80}$$

This quantity is a distance in the RKHS by taking a characteristic kernel (e.g., the Gaussian or Matérn kernel). After a calculation developed in Appendix D, an unbiased one-sample estimator of the squared-MMD was proposed by **?**, with a convergence rate of $O(n^{-1/2})$ in probability. Considering the two-samples $\left\{\pi^{(i)}\right\}_{i=1}^n \sim \pi$ and $\left\{\zeta^{(j)}\right\}_{j=1}^n \sim \zeta$:

$$\widehat{\mathrm{MMD}}^2(\pi, \zeta) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(\pi^{(i)}, \pi^{(j)}) - k(\pi^{(i)}, \zeta^{(j)}) - k(\zeta^{(i)}, \pi^{(j)}) + k(\zeta^{(i)}, \zeta^{(j)}), \tag{1.81}$$

In the context of GSA, the first option is to directly use this dissimilarity measure to define the unnormalized index:

$$S_j^{\mathrm{MMD}} = \mathrm{MMD}(\mathbb{P}_Y, \mathbb{P}_{Y|X_j}). \tag{1.82}$$

? remarked that the unormalized first order Sobol' indices are recovered by taking the linear kernel on the output $k_Y(y, y') = y\,y'$. Using this non-characteristic kernel (see the definition in Appendix D) brings us back to a moment-dependent importance measure.

Alternatively, the second option considers a couple of random variables (X_j, Y), with probability distributions $\mathbb{P}_{X_j}$ and $\mathbb{P}_Y$, and assumes the RKHS $\mathcal{H}$ induced by the tensor product kernel $k((x_j, y), (x'_j, y')) = k_{X_j}(x_j, x'_j)k_Y(y_j, y'_j)$. The *Hilbert-Schmidt independence criterion* (HSIC) measures the dependence between $\mathbb{P}_{X_j}$ and $\mathbb{P}_Y$ by expressing the MMD between $\mathbb{P}_{(X_j,Y)}$ and $\mathbb{P}_{X_j} \otimes \mathbb{P}_Y$:

$$\mathrm{HSIC}(X_j, Y) = \mathrm{MMD}^2(\mathbb{P}_{(X_j,Y)}, \mathbb{P}_{X_j} \otimes \mathbb{P}_Y). \tag{1.83}$$

This technique showed very good results for screening, and corresponding independence tests were studied for screening in **?**.

? proposed the functional decomposition of the two indices defined in Eq. (1.82) and Eq. (1.83), allowing to develop their respective normalized versions. Note that the HSIC decomposition requires a specific hypothesis on the structure of the kernel associated with the inputs.

1.6.4 Summary and discussion

This section introduced the GSA methods commonly used in uncertainty quantification. Either to reduce the dimension of a problem (screening) or to quantify the influence of inputs (with importance measures), GSA improves the understanding of an uncertainty quantification study. As for other steps of the generic UQ methodology, SA is made more complicated for computationally costly simulation models, hence the use of surrogate models. Additionally, the dependence between inputs still represents an important limit to interpreting GSA results.

Alongside rare event estimation, some literature is dedicated to the influence of random inputs on such tail statistics. The sensitivity is no longer qualified as "global" but becomes "goal-oriented". In the field of structural reliability, an overview of the reliability-oriented sensitivity analysis methods is presented in **?**. Several techniques derive from rare event estimation (e.g., the FORM importance factors **?**), or were adapted from GSA, like Sobol' indices (**?**), Target-HSIC (**?**), or Shapley effects (**?**).

Finally, sensitivity analysis may describe the effects of random inputs on the variation of the output, however, this study is done by assuming a model on the input uncertainties. The role of a regulatory agency auditing an uncertainty quantification approach for certification (i.e., a nuclear safety authority), might be to challenge the way to model the uncertainties on the inputs. In this case, various tools for *robustness analysis* exist to quantify the impact of mispecifying the random inputs on the quantity of interest studied. Among the methods to pertubate uncertainty models, some remain in the probabilistic framework, such as the "perturbed-law based indices" (PLI) (**?**), or on extra-probabilistic methods (**?**).

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 by $\widehat{g}_n(\cdot)$, replacing a costly numerical model $g(\cdot)$ over the input domain $\mathcal{D}_{\mathbf{X}}$. To do so, 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* written:

$$\{\mathbf{X}_n, \mathbf{y}_n\} = \left\{\mathbf{x}^{(i)}, y^{(i)}\right\}_{i=1}^{n} = \left\{\mathbf{x}^{(i)}, g\left(\mathbf{x}^{(i)}\right)\right\}_{i=1}^{n}. \tag{1.84}$$

A very large catalog of regression methods exists, here is a list of the most encountered ones in the field of UQ: generalized linear regression, polynomial chaos expansion (PCE) (**??**), support vector machine (SVM) (**?**), Gaussian processes (GP) (**?**), low-rank tensor approximations (**?**), and artificial neural network (ANN) (**?**). The following section will provide a short focus on Gaussian process regression.

Validating the accuracy and precision of a surrogate model is an important step to guarantee its fidelity with regard to the numerical model. When an m-sized input-output set is dedicated to validating the surrogate model, independently of the learning set, it is called *test set* and denoted by $\{\mathbf{X}_m, \mathbf{y}_m\} = \left\{\mathbf{x}^{(i)}, g\left(\mathbf{x}^{(i)}\right)\right\}_{i=1}^{m}$. 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 a certain cost).

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

$$\mathrm{R}^2(\widehat{g}_n) = 1 - \frac{\sum_{i=1}^{n}\left(y(\mathbf{x}^{(i)}) - \widehat{g}(\mathbf{x}^{(i)})\right)^2}{\sum_{i=1}^{n}\left(y(\mathbf{x}^{(i)}) - \overline{\mathrm{y}}_n\right)^2}, \tag{1.85}$$

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

$$\mathrm{Q}^2(\widehat{g}_n) = 1 - \frac{\mathrm{ISE}(\widehat{g}_n)}{\mathrm{Var}(g(\mathbf{X}))}, \tag{1.86}$$

where

$$\mathrm{ISE}(\widehat{g}_n) = \int_{\mathcal{D}_\mathrm{X}} \left(g(\mathbf{x}) - \widehat{g}(\mathbf{x})\right)^2 \mathrm{d}\mathbf{x}, \qquad \mathrm{Var}(g(\mathbf{X})) = \int_{\mathcal{D}_\mathrm{X}} \left(g(\mathbf{x}) - \int_{\mathcal{D}_\mathrm{X}} g(\mathbf{x})\,\mathrm{d}\mathbf{x}'\right)^2 \mathrm{d}\mathbf{x}. \tag{1.87}$$

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

$$\widehat{\mathrm{Q}}^2(\widehat{g}_n) = 1 - \frac{\sum_{i=1}^m \left(y(\mathbf{x}^{(i)}) - \widehat{g}(\mathbf{x}^{(i)})\right)^2}{\sum_{i=1}^m \left(y(\mathbf{x}^{(i)}) - \overline{\mathrm{y}}_m\right)^2}. \tag{1.88}$$

Note that for either criterion, 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 one to learn and test on the same sample. The most common cross-validation method is the *k-fold* validation, illustrated in Fig. 1.12. The idea is first to split the n-sized dataset into several equal parts, called folds. A first surrogate can be fitted on the entire datasets but the first fold, on which a validation criterion is estimated (i.e., performance metric). The operation is repeated for each fold, providing a virtual validation of 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, for example by adding a permutation or shuffling step. The "bagging" validation method (for "bootstrap aggregating") consists of a shuffled cross-validation repeated many times (**?**).

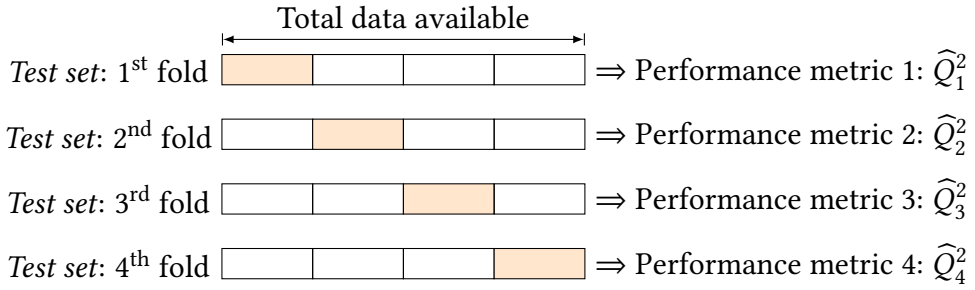


Figure 1.12 Illustration of a k-fold cross-validation (with $k = 4$)

1.7.2 General purposes surrogate model

In this section, a particular focus is dedicated to Gaussian process (GP) regression (also called kriging after the geostatician 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, the learning set should collect information over the entire domain uniformly. This 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 $\mathbf{X}_n$, the goal is to approximate the function $g(\cdot)$ by a scalar Gaussian process conditioned on a set of observations $\mathbf{y}_n = \left\{g\left(\mathbf{x}^{(i)}\right)\right\}_{i=1}^{n}$. Let us first define a prior structure G on the function approximated $g(\cdot)$, taken as a Gaussian process with a mean function $m(\cdot)$ and covariance function $k(\cdot,\cdot)$:

$$G \sim \mathrm{GP}(m(\cdot), k(\cdot,\cdot)), \tag{1.89}$$

with a:

- *trend model*: $m(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta}$, composed of a functional basis $\mathbf{f} = (f_1, \ldots, f_d)^\top$ and a vector of coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d)^\top$,
- *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 $\boldsymbol{\theta} \in \mathcal{D}_{\mathbf{X}}$.

The trend model of a GP defines its general tendency, while the covariance model influences its regularity. Gaussian process regression 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 **?** 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 $G(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$. Working with Gaussian variables allows us to easily write conditioning formulas between $G(\mathbf{x})$ and the observations $\mathbf{y}_n$. This Gaussian variable $G(\mathbf{x})$ conditioned on the observations $\mathbf{y}_n$ is sometimes called conditional posterior $G_n(\mathbf{x}) := (G(\mathbf{x})|\mathbf{y}_n) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$. The well-known "Kriging equations" (see e.g., **?**) offer its explicit expression:

$$\begin{cases} \eta_n(\mathbf{x}) & := \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{y}_n \\ s_n^2(\mathbf{x}) & := k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^\top(\mathbf{x}) \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \end{cases} \tag{1.90}$$

where $\mathbf{k}(\mathbf{x})$ is the column vector of the covariance kernel evaluations $[k(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, k(\mathbf{x}, \mathbf{x}^{(n)})]$ 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 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 maximum likelihood estimation or cross-validation.

The illustration in Fig. 1.13 is a typical one-dimensional representation of an ordinary kriging model. The mean of the conditioned process is plotted in red while its variability is represented by the many trajectories drawn on the process. In the simplest framework, the kriging model exactly interpolates the observations (black crosses).

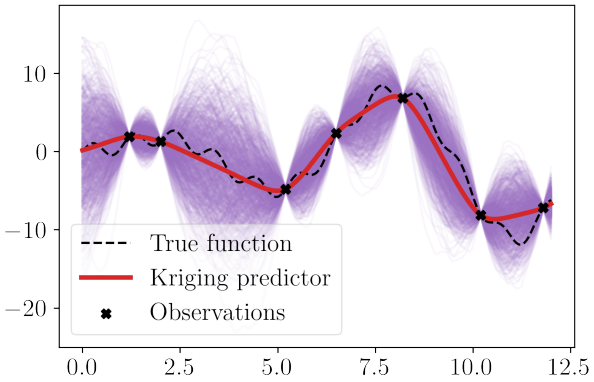


Figure 1.13 Illustration of an ordinary kriging model fitted on a limited set of observations ($n = 7$). The predictor is represented in and several trajectories of the conditioned Gaussian process are drawn and represented in purple.

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 **?** to confirm that the kriging variance is reliable. For a validation performed by holdout, and using an independent m-sized test set, the PVA is defined as:

$$\mathrm{PVA} = \left| \log \left(\frac{1}{m} \sum_{i=1}^{n} \frac{\left(y(\mathbf{x}^{(i)}) - \widehat{g}(\mathbf{x}^{(i)}) \right)}{s_n^2(\mathbf{x}^{(i)})} \right) \right|. \tag{1.91}$$

The smaller this quantity gets, 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 is a recurrent issue. Multiple techniques solve 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` 6 (Gaussian process regression)**.** The Python code available in Appendix D proposes a minimalistic `OpenTURNS` example to fit an ordinary kriging model and active learning models. Figures illustrating the present section may be reproduced, using the `OpenTURNS` scripts available on GitHub[13,14].

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 the area(s) of interest. An exploration-exploitation trade-off arises in active learning, mostly sorted 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 given computational cost. However, the "active learning" term is also used to handle big data frameworks in the machine learning community (**?**). When datasets become so large that learning methods do not scale in practice, the analyst needs to select a relevant 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 (**?**). 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. Factually, the learning criterion depends on the Gaussian process variance model. Numerous reviews of this field were proposed by **??** and numerical benchmarks presented in **?**.

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

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\, g(\mathbf{x}) \approx \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\, \widehat{g}(\mathbf{x}). \tag{1.92}$$

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 $G_n(\mathbf{x}) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$ is fitted using Eq. (4.11). The expected improvement, to be maximized, is then

[13] https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/surrogates.ipynb
[14] https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter1/active_learning.ipynb

written as:

$$\mathcal{A}^{\mathrm{EI}}(\mathbf{x};\mathbf{y}_n) = \mathbb{E}\left[\max\left(g_{\min} - G_n(\mathbf{x})\right)\right] \tag{1.93}$$

$$= \left(g_{\min} - \eta_n(\mathbf{x})\right)\Phi\left(\frac{g_{\min} - \eta_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x})\phi\left(\frac{g_{\min} - \eta_n(\mathbf{x})}{s_n(\mathbf{x})}\right), \tag{1.94}$$

where $g_{\min} = \min(\mathbf{y}_n)$, ϕ and Φ respectively stand for the PDF and the CDF of the standard Gaussian distribution. This learning criterion is relatively inexpensive and allows a progressive enhancement of the Gaussian process to solve the optimization problem with a limited number of calls to the true function.

Three iterations of the EGO algorithm are represented in Fig. 1.14 to minimize a function (dashed line), knowing a few observations (black crosses). After fitting an initial kriging model (in red), the corresponding expected improvement function is represented underneath it (green line). This learning criterion determines the location of the observation to be added to the learning set to enhance the surrogate w.r.t. to the optimization problem.

Bayesian optimization is an active research field, with different open problems such as constrained Bayesian optimization (**?**), or Bayesian optimization on stochastic functions (**?**). Similarly, active learning was also adapted for structural reliability problems.

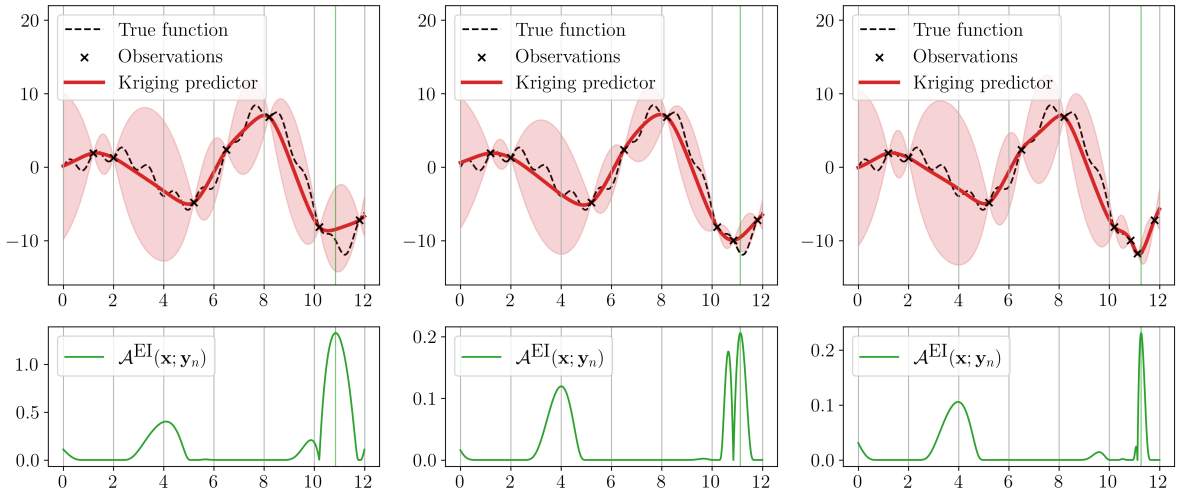


Figure 1.14 Illustration of the expected improvement learning criterion

Active kriging for reliability analysis

Rare event estimation often requires large amounts of evaluations of the limit-state function (becoming intractable for costly numerical models). Emulating this function by a surrogate model can drastically limit the number of calls to the 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 methods 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 binary classification. For example, active learning procedures using SVM classifiers were adapted to this specific goal (?).

The following paragraph introduces the most popular kriging-based learning criterion: the "deviation number" U (?). The reader may refer to ? for further active learning techniques dedicated to rare event estimation. More recently, ? and ? reviewed 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 $G_n(\mathbf{x}) \sim \mathcal{N}(\eta_n(\mathbf{x}), s_n^2(\mathbf{x}))$ is fitted using Eq. (4.11). The deviation number U is looking for points close to the limit-state function while presenting a high kriging variance. This criterion to be minimized is defined as:

$$\mathcal{A}^{\mathrm{U}}(\mathbf{x}; \mathbf{y}_n) = \frac{|y_{\mathrm{th}} - \eta_n(\mathbf{x})|}{s_n^2(\mathbf{x})}, \tag{1.95}$$

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

Fig. 1.15 reuses the same one-dimensional function as in Fig. 1.14 to create a rare event problem. In this case, the failure domain is defined for output values below the threshold y_{th}. Once again, three iterations of the AK algorithm are illustrated, with the corresponding learning criterion U (to minimize). In this simple case, the LSF is defined by the two intersections of the function with the threshold. Therefore, the AK method selects points near these intersections.

Unlike optimization problems, the surrogate is used for uncertainty propagation, meaning that the rare event estimation is the result of the approximation of the LSF (i.e., contour finding) and a sampling techniques. AK methods were coupled with most sampling techniques introduced in Section 1.5 (e.g., AK-MCS, AK-IS, AK-SS, etc.). As an agnostic strategy, ? recommend starting by applying an AK method (using the learning function U) paired with a subset sampling (taking an intermediary probability $p_0 = 0.2$).

The AK methods present the advantages of being easily implemented and interpreted, however, their learning criterion relies on a local approach. Alternatively, the *stepwise uncertainty reduction* (SUR) chooses iterative points by reducing the future expected uncertainty related to the quantity of interest (?). If this method was proven to be theoretically more consistent (?), its scaling ability is still a bottleneck.

1.7.4 Summary and discussion

This section brought attention to surrogate modeling in the context of computer experiments. Statistical learning in this framework is made specific by the capacity of the analyst to choose the repartition of the learning set and the small data constraint (mostly due to the costly numerical models manipulated). In this context, many methods are used, however, Gaussian processes became popular in UQ as they consider a prior structure of uncertainty that is conditioned

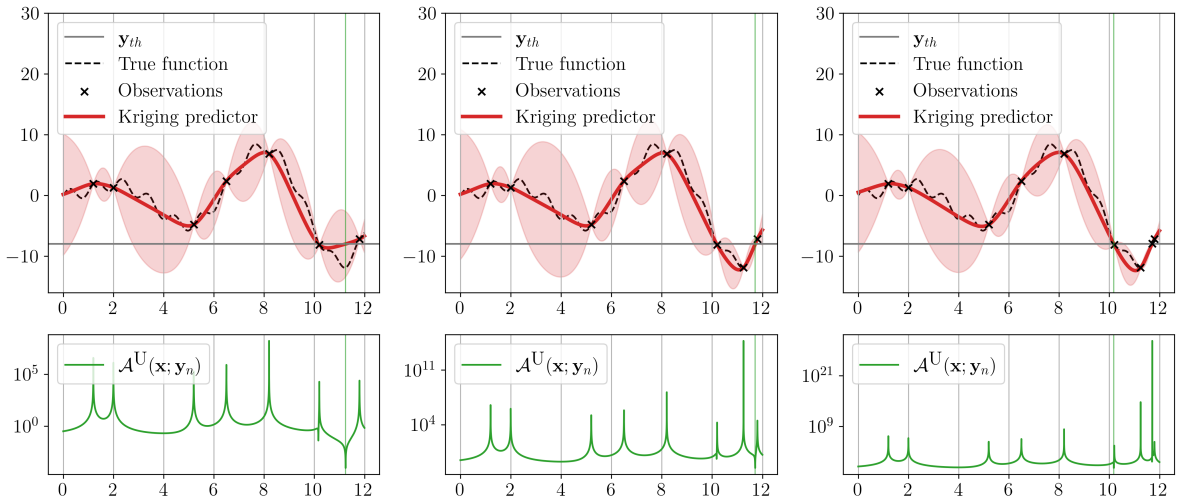


Figure 1.15 Illustration of the deviation number learning criterion

by observations (at the edge between a Bayesian and a frequentist approach). To enhance the learning for specific purposes (e.g., optimization or contour finding), active learning methods iteratively add learning points in the subdomain of interest. For some applications, the system studied might be modeled for different fidelities (each presenting different computational costs). Multi-fidelity surrogate modeling is an active field of research, associating observations from different fidelities to improve the learning (**?**). Such methods are relevant for models with a very high computational cost (typically in computer fluid dynamics).

In UQ, surrogate models are used for uncertainty propagation (step C) and inverse analysis (step C'). Surrogate modeling is made difficult when the functions present discontinuities (or strong nonlinearities), high dimension, stochasticity, or nonscalar inputs or outputs. To deal with high dimensional problems, unimportant inputs can be screened using sensitivity analysis (see Section 1.6.1), otherwise, model order reduction methods might be used [ref ?]. When the function is stochastic, different approaches allow fitting the function and its intrinsic variability (**???**).

Provided a strict validation process, surrogate models are a great opportunity for uncertainty quantification. However, many regulatory authorities are still reluctant to use surrogates, although their error is often much smaller than the modeling error (i.e., the error between the actual physical behavior and its numerical modeling).

1.8 Conclusion

This section gives a literature overview of the main steps in uncertainty quantification. From uncertainty modeling, uncertainty propagation, and sensitivity analysis to surrogate modeling. To ease the methodological presentation, all the illustrations from this section are reproducible using the Python/OpenTURNS scripts available on the GitHub repository mentioned earlier.

Finally this work, the numerical models exploited are supposed to be accurate, but they obviously carry some modeling uncertainty (**?**). In fact, prior to uncertainty quantification, the

model should be calibrated to make it match some physical information (e.g., measurements). Numerical model calibration is also called data assimilation when a stream of measured data is available.

The aim of this work is to apply the tools presented in this chapter to offshore wind turbine models, therefore the next chapter introduces the numerical models manipulated in this thesis.

Chapter 2

Introduction to wind turbine modeling and design

2.1 Introduction

Wind energy is a highly competitive industry with increasing regulations regarding its impact on ecosystems, land and sea use, landscapes, or air traffic management (**?**). During the long process of winning calls for tenders, obtaining construction permits, or through wind farm exploitation, an advanced technical understanding of such systems might offer a competitive advantage.

The operation of offshore wind turbines (OWTs) is 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 a primary role. Yet, many other types of solicitations affect the exploitation of offshore wind turbines e.g.: the 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 a given location should pass different validation procedures established by international standards such as the International Electrotechnical Commission (**?**). As wind turbine structures face a large number of stress cycles in their lifetime (up to 10^8 for 20 years of operation), this chapter will particularly focus on fatigue damage assessment.

The present thesis studied different steps of uncertainty quantification on two wind farm projects. First, the Teesside wind farm, operating since 2014 in the North Sea, second the theoretical wind farm of south Brittany, currently at the stage of calls for tenders.

Considering the standard uncertainty quantification diagram presented in Fig. 1, the material of this chapter is related to step A (problem specification) and step B (uncertainty quantification). It 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 describes the various sources of uncertainties considered in this thesis.

2.2 Metocean conditions simulation

In the atmosphere, the wind represents 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 by the intermediate of the well-known Coriolis effect. The wind is a highly variable resource, making its exploitation for energy production uncertain. This variability is 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). However, predictions at a

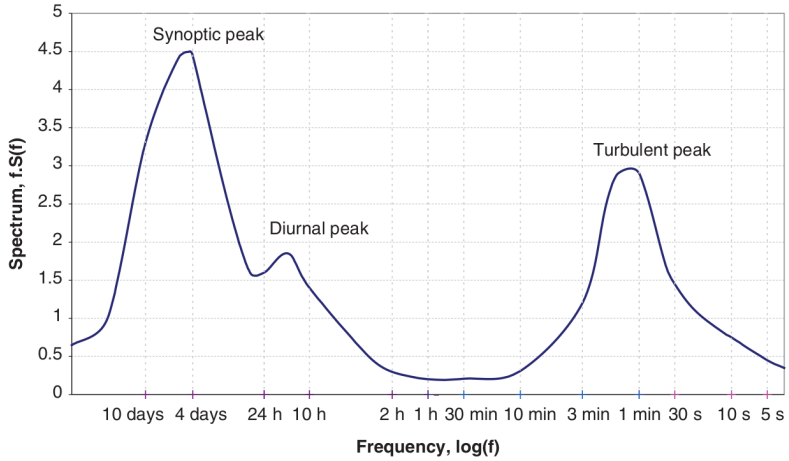


Figure 2.1 Wind spectrum from Brookhaven, USA (source: **?**)

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 **?** for timescales between a few seconds 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" peaks, 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 third peak describes the energy related to wind turbulence, which evolves in a range below ten minutes. Considering this typical energy distribution, wind behaviors are often referred to as "short-term" (for turbulent wind) and "long-term" (otherwise). In wind turbine simulation, ten-minute simulations became a common practice to fully consider turbulent winds.

Remark that the spectrum presented in the research paper of **?** (represented in Fig. 2.1) was built 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 more complicated. Additionally, wind resource assessment over decades is made more uncertain by climate change **?**, disrupting large weather trends and increasing the occurrence of extreme events.

2.2.1 Turbulent wind generation

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 $\overline{U}$ and standard deviation σ_U (**?**). 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. [This short-term / long-term modeling hypothesis is represented

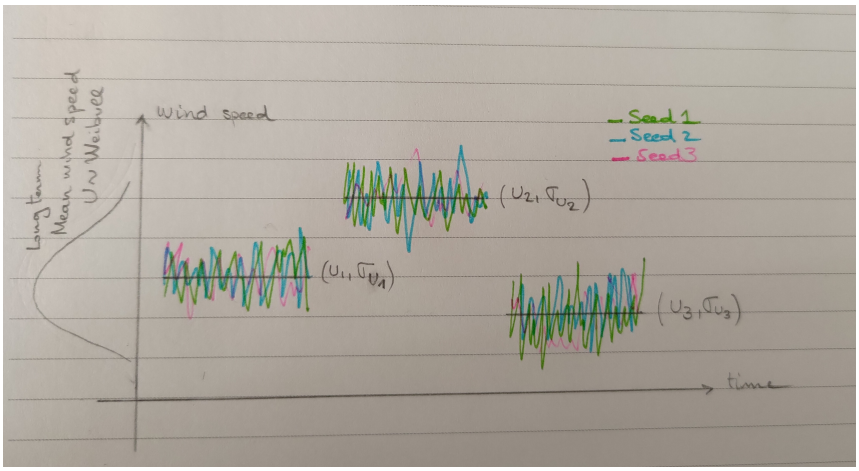


Figure 2.2 [Should we keep this representation? If so, it will be done properly.]

in Fig. ??.] Note that this assumption is based on the bimodal wind energy distribution observed in Fig. 2.1, which might vary at some specific locations.

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

$$I = \frac{\sigma_U}{\overline{U}}. \tag{2.1}$$

As the wind depends on differences between pressure, humidity, and 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 model to approximate vertical shear (**?**):

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

where $\overline{U}_0$ is a well-defined mean wind speed at the height z_0 (typically corresponding to a measurement height), z is the studied height (e.g., the turbine's hub height), and α is 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* (**?**) and the *Mann model* (**?**). In this thesis, the Kaimal spectrum as defined in **?** is used for turbulent wind generation over the Cartesian component $k \in \{u, v, w\}$:

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

such that f is the frequency, $\overline{U}$ is the longitudinal mean speed at hub-height, L_k are the Kaimal length scales, and σ_k standard deviations. Along with the Kaimal wind speed spectrum, a spatial

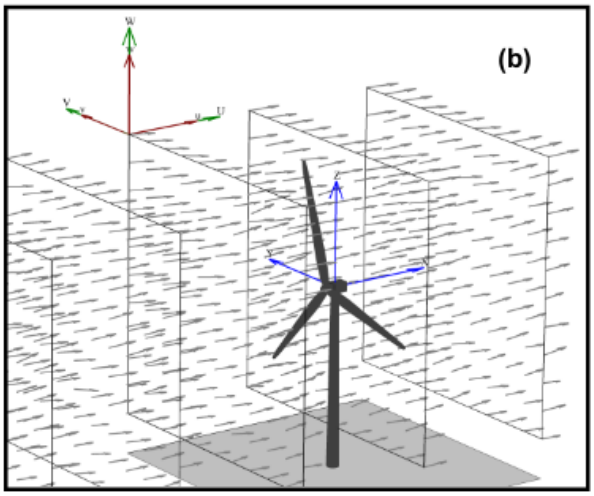


Figure 2.3 Example of a turbulent wind field generated by TurbSim (source: **?**)

coherence model is usually defined in the frequency domain. Each couple of nodes in the mesh are correlated, for example, using an exponential coherence model (see the complete definition in Annex C of **?**).

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) (**?**). TurbSim generates time realizations by adapting the spectral method proposed in **?** (relying on the inverse Fourier transforms of each axial 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-minute TurbSim simulation, considering a set of input long-term conditions.

In their recent review of the challenges in wind energy, (**?**) list some limits of the two spectral turbulence models recommended by the standards. First, their parameters were fitted using a restricted amount of data (**?**). Second, the spatial coherence models associated with Kaimal models showed differences with turbulence measured on-site (**?**). Finally, recent studies showed that the choice of spectral model impacts the resulting wind turbine loads (**?**). These approximations generally tend to overestimate wind flows, leading to conservative designs.

To ensure more realistic turbulent wind field generation, two research perspectives are actively explored. Authors recently developed hybrid methods, including measurement data to enhance spectral models (**?**). Alternatively, higher fidelity models were studied in this domain, see for example the use of vortex methods (**?**) and large eddy simulations (LES) (**??**). 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 for uncertainty quantification studies. When studying the wind resources at a wind farm scale, modeling wind energy losses induced by the turbines' wake becomes essential.

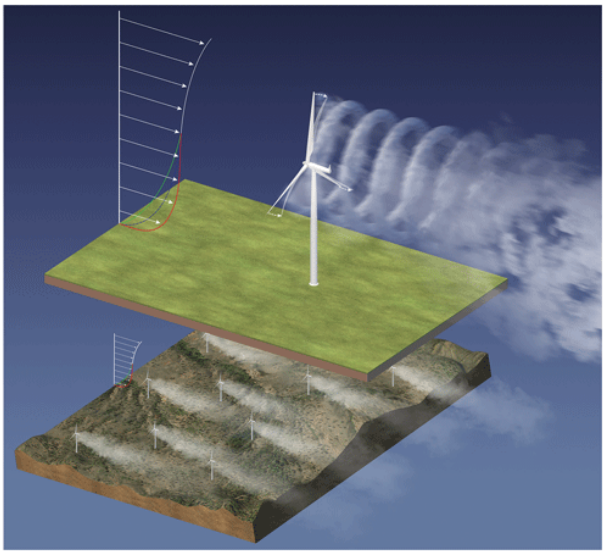


Figure 2.4 Illustration of the wake created downstream a wind farm (source: ?)

2.2.2 Wake modeling

The wake is caused by the extraction of the wind kinetic energy, 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 indeed 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 percent depending on the farm), and an increase in 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 (?). 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 ?). These models mostly rely on the equivalence between the thrust load and the turbine wind energy deficit. Since the seminal engineering model proposed by ?, multiple enhancements were proposed. A wide benchmark of the wake modeling solutions for different fidelities was performed in ? and ?. The optimal tuning of these engineering models was studied using measurements from a Doppler wind lidar in ?. Different software programs propose wake engineering models, such as FLORIS (developed by the NREL ?), 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 misalignments (making the control of turbines individually suboptimal) to optimize the global wake inside the farm (???).

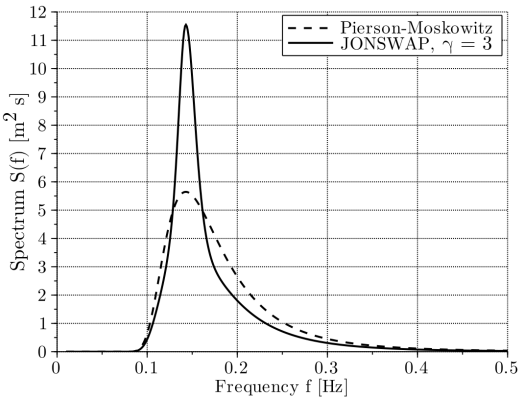


Figure 2.5 Peirson-Moskowitz and JONSWAP spectra at significant wave height $H_s = 3$ m and peak period $T_p = 7$ s (source: **?**)

2.2.3 Irregular wave generation

The propagation of wind-generated waves has long been studied in hydrodynamics under the prism of different theories such as Airy's, and Stokes'. Airy's wave theory (also referred to as the 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. Two standard statistics are used in oceanography to represent sea states and their corresponding wave spectra: the wave period T_p (with the corresponding 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 named JONSWAP, after the "Joint North Sea Wave Project" (**?**):

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

The JONSWAP spectrum is a corrected version of the Pierson-Moskowitz spectrum (developed in 1964), adding a peak enhancement factor γ^α. Further details regarding the numerical values to choose in Eq. (2.4) are given in **?**. An illustration of the two spectra is presented in Fig. 2.5, revealing the enhancement factor proposed in the JONSWAP model to better fit sea state measurements.

Swell waves are the result of weather conditions occurring far away from the location studied. Such waves usually present long wavelengths, allowing them to propagate over long distances with little dissipation. To take them into account, the unimodal wave spectrum introduced in Eq. (2.4) was improved. Different methods allow building a parametric bimodal distribution, with a mode in the low frequencies corresponding to the swell. **?** reviews different bimodal wave spectra and compares their adequacy with measured sea states.

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 has 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 nonlinearities and transient loading phases (**?**). These aspects happen to be essential in the design of OWTs (**?**). As an alternative, the behavior of OWT systems is also simulated in the time domain.

In the time domain, such systems may be models for different fidelities. The diagram in Fig. 2.7 illustrates the increasing complexities of two physics involved in OWT modeling (aerodynamics and structural dynamics). Generally, the computational cost increases with the model fidelity, making uncertainty quantification intractable at some point. In the present work, the numerical model of an OWT studied is actually a chain of three models executed sequentially (as illustrated in Fig. 2.6):

- **TurbSim**: a turbulent wind generator (see Section 2.2.1);
- **DIEGO**: a multiphysics wind turbine model in the time domain (see Section 2.3);
- **Fatigue assessment**: a post-precessing computing fatigue damage (see Section 2.3.5).

DIEGO is a numerical model developed by EDF R&D to simulate the aero-hydro-servo-elastic behavior of OWTs in the time domain. Different extensive code-to-code comparisons between DIEGO and other aero-hydro-servo-elastic models showed close results. Considering bottom-fixed OWT (**?**) or floating OWT (**??**), DIEGO was compared to "FAST" (developed by NREL), "HAWC2" (developed by DTU), "BLADED" (developed by DNV), and "DeepLines Wind" (developed by IFPEN).

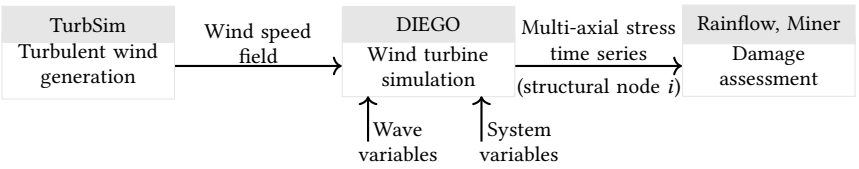


Figure 2.6 Chained numerical model of offshore wind turbine.

2.3.1 Aerodynamics of horizontal axis wind turbines

The blade element momentum theory mixes different concepts to compute the aerodynamic forces on the rotating blades of the wind turbine. In this coupled physics model, aerodynamics

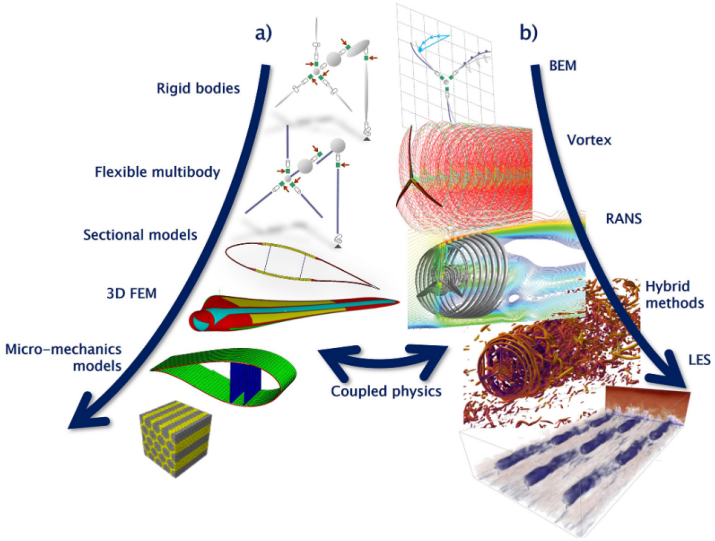


Figure 2.7 Hierarchy of structural (a) and aerodynamic (b) wind energy systems models (source: **?**)

affects the structural response and vice versa. To solve this problem, algorithms used in DIEGO first assess displacement of elementary blades, to recover the lift and drag coefficients. The elementary loads are then integrated over each blade and communicated to the structural model.

Momentum theory. At the core of wind turbine's aerodynamics, the concept of *momentum theory*, also called *actuator disk theory* assumes that the air stream passing through the rotor disk is bounded by a stream tube of circular surface (not mixing with the ambient air). Fig. 2.8 is a longitudinal representation of the actuator disk and the way it affects the air upstream and downstream of the rotor. The associated momentum theory assumes the conservation of airflow at any cross-section (of area A) during a time period. Passing through the actuator disk, the wind speed slows down and a drop in static pressure is at the origin of the wake. This pressure drop generates an axial force (called *axial thrust force*) and a torque on the actuator disk.

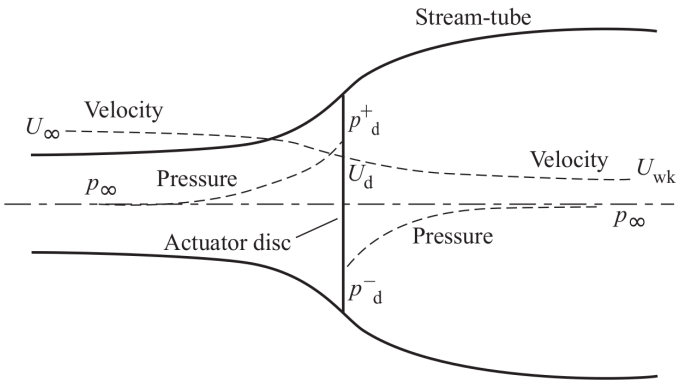


Figure 2.8 Actuator disk model of the energy extraction (source: **?**). Longitudinal evolution of the air pressure and wind speed along the wind stream.

Considering the upstream flow, the flow at the rotor disk and the airflow in the wake, respectively denoted by the subscripts $\{\infty, d, \text{wake}\}$, the following equality comes:

$$\rho A_\infty U_\infty = \rho A_d U_d = \rho A_{\text{wake}} U_{\text{wake}}, \tag{2.5}$$

where U is the wind speed, A the stream-tube area, and ρ the air density. The wind speed in at the rotor disk can be expressed using the induction factor a in the following expression:

$$U_d = U_\infty(1-a),\ 0 \le a \le 1. \tag{2.6}$$

Using the momentum theory and Bernoulli's incompressible flow equation, one can express the aerodynamic thrust T and power P (see **?**):

$$T = (p_d^+ - p_d^-)A_d = 2\rho A_d U_\infty^2 a(1-a) \tag{2.7a}$$

$$P = TU_d = 2\rho A_d U_\infty^3 a(1-a)^2 \tag{2.7b}$$

The widely used power coefficient (respectively thrust coefficient) is the ratio of the power captured by the turbine against the total kinetic wind power available in the stream tube:

$$C_P = \frac{P}{\frac{1}{2}\rho A_d U_\infty^3} = 4a(1-a)^2, \tag{2.8a}$$

$$C_T = \frac{T}{\frac{1}{2}\rho A_d U_\infty^2} = 4a(1-a). \tag{2.8b}$$

Betz's law is a theoretical limit value of the power coefficient, obtained by canceling the power coefficient gradient. To this day, no wind turbine has exceeded this limit value: $C_P^{\text{Betz}} = 0.593$ (**?**).

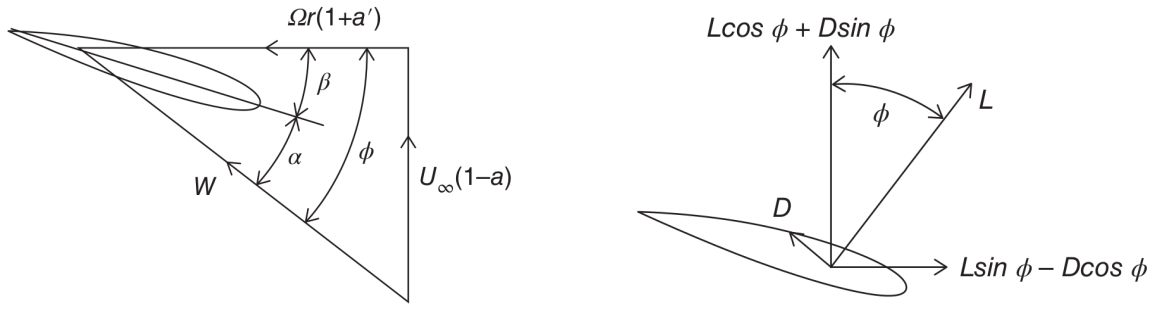
Blade element theory. Assuming a purely two-dimensional flow (meaning that the forces are only determined by the lift and drag coefficients), the blade element theory expresses the thrust $\mathrm{d}T$ and torque $\mathrm{d}Q$ applied on a blade element.

Let us consider a wind turbine with B blades, a pic length R and a pitch angle β. Assuming the blade element represented in Fig. 2.9 at the blade length r, with airfoil chord c, angle of attack α, lift C_L and drag C_D coefficients, lift L and drag D forces, and the axial and tangential induction factors a and a'. Under these assumptions, the axial thrust and torque exerted on a blade element are:

$$\mathrm{d}T = \frac{1}{2}\rho W^2 B c\,(C_L\cos(\varphi) + C_D\cos(\varphi))\,\mathrm{d}r, \tag{2.9a}$$

$$\mathrm{d}Q = \frac{1}{2}\rho W^2 B c\,(C_L\sin(\varphi) + C_D\sin(\varphi))\,\mathrm{d}r. \tag{2.9b}$$

Blade element momentum theory (BEMT) combines the results from blade element theory in Eq. (2.9) with the results from momentum theory in Eq. (2.7) to obtain the induction factors a



(a) Speed triangle

(b) Aerodynamic forces

Figure 2.9 Blade element forces. With the lift and drag forces L and D, the flow angle ϕ, the pitch angle β and the angle of attack α (source: **?**).

and a'. The resolution of this system of equations is often solved by iterative approaches (e.g., **?**). Global axial thrust over the blade is then computed by integrating the elementary loads over all the elements. Note that various corrections are applied to the BEMT model, for example, to take into account the non-homogeneous loss of momentum over the rotor disk. The BEMT also fails to model non-linear aerodynamic effects, occurring with sudden changes of angle of attack. Such effects are sometimes called "dynamic stall" and are represented in DIEGO by the Beddoes-Leishman model (see **?** for further details).

2.3.2 Hydrodynamics

Morison's equations are a widely-used semi-empirical model to assess the hydrodynamic forces on thin fixed structures such as offshore oil platforms and wind turbines. Considering a slender cylindrical structure of diameter D, a flow velocity $u(t)$, the drag and inertial coefficients C_m and C_D, the axial force (parallel to the flow direction) is given by:

$$F = C_m \rho \frac{\pi}{4} D^2 \frac{\mathrm{d}u}{\mathrm{d}t} + C_d \frac{1}{2} \rho D u |u|. \tag{2.10}$$

Standard values for the drag and inertial coefficients are often considered (**?**). DIEGO uses Morison's equation together with a first-order potential solution to perform hydrodynamical simulations in the time domain. An extended introduction to the hydrodynamics of fixed slender structures, as well as large floating structures, is given in Chapter 1 from (**?**).

To design floating structures, more complex wave-loading modeling should be considered. For this purpose, **?** reviews nonlinear theories applied to the fluid-structure interactions of FOWT and compares them to CFD results.

2.3.3 Control

To maximize their energy production under turbulent wind conditions, wind turbines rely on their control systems. This aspect of wind turbines is usually kept confidential by manufacturers,

as it gives them a competitive advantage. Nevertheless, the general control mode of a wind turbine depends on the wind speed. Two main ranges of operation are usually defined: first between the cut-in and rated wind speed, and second between the rated and cut-off wind speed. These characteristic wind speed values are given by the turbine manufacturer, for example, a turbine may present a cut-in at 4 $\mathrm{m\,s^{-1}}$, a rated at 13 $\mathrm{m\,s^{-1}}$ and a cut-off at 25 $\mathrm{m\,s^{-1}}$. Let us then recall the wind turbine power derived from the momentum theory:

$$P = \frac{1}{2}\rho A_d U_\infty^3 C_p(\lambda, \beta), \tag{2.11}$$

with the power coefficient C_p, as a function of the pitch angle β and the blade tip speed ratio λ, defined between the tangential speed on top of the blade and the wind speed: $\lambda = \frac{\Omega R}{U_\infty}$, for the rotation speed Ω and a rotor radius R.

Below the rated wind speed. The goal of the control system is to extract as much power as available. A control strategy among the family of the *maximum power point tracking* can be deployed (**?**). For example, the "power signal feedback" uses electromagnetic torque to control the power. This method first computes the maxima of the extracted power as a function of the rotation speed (using Eq. (2.11)), for different speed values. Then, for a measured wind rotation speed, the system can determine the reference maximal power. Considering this reference power, a controller (such as a proportional integral controller) intends to match the generated power with the reference by acting on the electromagnetic torque.

Above the rated wind speed. The control system switches to a *power limiting* mode by increasing the blades' pitch angles. By operating on the pitch, the rotation speed and the power produced are kept at their nominal values. This control is also often realized by a proportional integral system (**?**).

A more exhaustive description of wind turbine control systems is available in Chapter 8 from **?**. More recent strategies often consider the control at the farm scale. As explained earlier, the operation of one turbine affects the others via the effect of its wake. Moreover, since wind energy production becomes important in the electric mix, its production might be constrained to respect the stability of the grid (e.g., quality of the utility frequency). The work of **?** studied the optimal control of wind farms considering the effects of the wake and the grid restrictions.

2.3.4 Structural dynamics

The structural elements of modern wind turbines, such as the tower and the blades, compose a dynamic system subject to important elastic deformations. Modeling an operating wind turbine therefore requires rigid body dynamics and nonlinear elastic deformations. Altogether, various approaches were developed to model the structural dynamics of wind turbines: modal analysis, multibody methods and finite element methods (FEM). At the stage of preliminary designs, modal approaches can be used to represent the dynamics under linear assumptions

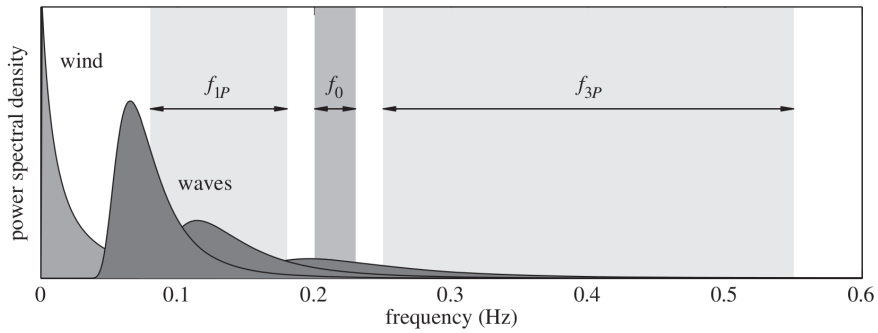


Figure 2.10 Illustration of a soft-stiff design strategy, placing the structure's natural frequency f_0 away from the wind and wave power spectra, and the rotor excitation frequencies f_{1P} and f_{3P} (source: **?**).

(**?**). Then, the tower's natural frequencies assessed by a modal analysis can be compared with the wind, waves, and the rotor's frequencies. As illustrated in Fig. 2.10, the structure's natural frequency (denoted by f_0) should not coincide with the main excitation frequencies to avoid critical dynamic resonance. In the case of a wind turbine, the rotor imbalance creates a first dynamic load of frequency f_{1P}, while the blades passing in front of the tower generate a second excitation of frequency f_{3P}. The *soft-stiff* design strategy places the structure's natural frequency between the two rotor frequencies (i.e., $f_{1P} < f_0 < f_{3P}$ as described in Fig. 2.10) and avoids main frequencies of the wind and waves.

However, modal analysis does not model transient loading phases and their corresponding non-linearities, which is crucial beyond early design. For a higher fidelity, simulations in the time domain using flexible multibody approaches are commonly used to describe the nonlinear dynamics (**??**). DIEGO implements such an approach by combining rigid multibody dynamics with a deflection model based on Lagrangian equations (**?**). Note that for floating wind turbine modeling, a preliminary step of rigid body dynamics is added to define the coordinate system of the floater. **?** reviews the state-of-the-art of numerical and experimental modeling techniques for multi-physics OWT systems.

2.3.5 Fatigue damage

Mechanical fatigue damage is an important phenomenon to consider when designing wind turbines. It refers to the progressive weakening of a material when subjected to cyclic or repeated loading, which may be significantly lower than the material's ultimate strength. Understanding the mechanisms behind mechanical fatigue damage is essential for designing durable and reliable structures. To quantify the fatigue damage on offshore wind turbine structures, standards (**?**) recommend simulating the stresses in the time domain and identifying a series of stress cycles. Then, the *stress-number of cycles curve* of a specific material (S-N curve) gives the number of cycles before failure at a given constant stress amplitude. As the stress cycles identified on the results of the OWT simulation are not constant, a linear aggregation method called *Miner's rule* gathers the elementary damages over the stress time series studied.

Stress cycle identification. Offshore wind turbine simulators as DIEGO, deliver a time-dependent stress tensor. To ease the manipulation of this tensor, the equivalent Von Mises stress is computed, turning a multiaxial stress into an equivalent uniaxial stress. One can also consider a "plane strain" hypothesis on the Cauchy stress tensor $\underline{\underline{\sigma}}$, which is expressed as:

$$\underline{\underline{\sigma}} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{pmatrix}. \tag{2.12}$$

This assumption simplifies the expression of the equivalent Von Mises stress:

$$\sigma_{\text{VM}} = \sqrt{\frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2\right] + 3\sigma_{12}^2}. \tag{2.13}$$

Stress cycles can now be identified on the equivalent Von Mises stress time series. The usual method to identify fatigue stress cycles is called *rainflow counting* (**?**). In this approach, fatigue stress cycles are only defined by their amplitude (also called "range") and mean value, regardless of their chronology. Rainflow counting returns a list of stress ranges identified and denoted by s in the following.

S-N curve. The S-N curve is also called the "Wöler curve" after the pioneer work of August Wöhler, who demonstrated that fatigue damage was at the origin of railway accidents in the mid-19$^{\text{th}}$ century (**?**). As a result of repeated fatigue experiments, this tool determines the number of similar stress cycles necessary to reach a fatigue ruin for a defined stress cycle amplitude. Its values depend on the material studied and on external conditions (i.e., in the offshore industry, the S-N curves distinguish the fatigue in the air vs. underwater).

A well-admitted simplification of the S-N curve is to consider it as log-linear[1] on two segments:

$$\log(N_{\text{c}}(s)) = \begin{cases} \log(a_1) - m_1 \log(s), & \text{for } s \in [s_{\min}, s_{\text{e}}] \\ \log(a_2) - m_2 \log(s), & \text{for } s \in [s_{\text{e}}, s_{\max}] \end{cases} \tag{2.14}$$

Where N_{c} is the predicted number of cycles to failure for stress range s, m is the negative inverse slope of the S-N curve, $\log(a)$ is the intercept of log N-axis by the S-N curve, $s_{\min}$ is the minimal (resp. maximal) stress range identified by the rainflow counting, and s_{e} is the stress range axis of the intersection of the two log-lines formed by the S-N curve.

The expression of this curve in two linear segments arises from the concept of endurance limit of a material, s_{e}, under which the effect of fatigue on a material should be considerably smaller. According to **?**, the S-N curve is altered for welded tubular joints by taking into account

[1] The logarithms related to the S-N curves in this document are logarithms in base 10.

the tube's thickness:

$$N_c(s) = \begin{cases} a_1 \left(s(\frac{t}{t_{ref}})^h \right)^{-m_1}, & \text{for } s \in [s_{min}, s_e] \\ a_2 \left(s(\frac{t}{t_{ref}})^h \right)^{-m_2}, & \text{for } s \in [s_e, s_{max}] \end{cases} \tag{2.15}$$

With t_{ref} the reference thickness (for tubular welded joints t_{ref} = 25 mm); t the plate thickness, and h the thickness exponent. The numerical values considered in the present work derive from Section 2.4.6 of **?**, reproduced in Table 2.1.

Table 2.1 S-N curve numerical values of welded tubular joints in different environmental conditions (source: **?**)

Environment	m_1	$log(a_1)$	m_2	$log(a_2)$	h
Air	3.0	12.48	5.0	16.13	0.25
Seawater with cathodic protection	3.0	12.18	5.0	16.13	0.25
Seawater free corrosion	3.0	12.03	3.0	12.03	0.25

Non-zero mean correction. Most S-N curves are built over zero mean stress cycles, however, different empirical models were developed to consider different stress mean s_m (**?**). The S-N curve becomes a three-dimensional envelope depending on the number of cycles N_c, the stress amplitude s, and the mean stress s_m. The "Goodman line" and the "Gerber parabola" are two models relating the stress amplitude s to the mean stress s_m:

$$\text{Goodman}: \quad \frac{s}{s_e} + \frac{s_m}{R_m} = 1 \tag{2.16}$$

$$\text{Gerber}: \quad \frac{s}{s_e} + \left(\frac{s_m}{R_m} \right)^2 = 1 \tag{2.17}$$

Where the material's yield stress is denoted by R_m and the endurance limit by s_e. The Haigh diagram represented in the Fig. 2.11 is a slice of the three-dimensional envelope for fixed values of fatigue endurance (i.e., number of cycles). By comparing the two models visually, the Goodman line is more conservative and is mostly used in the literature. Further discussion in the field of wind turbines was proposed in the early 2000s with a focus on the fatigue endurance of glass fiber materials (**?**). In the present work, the non-zero correction presented above is not considered as the values of mean stress were found to be negligible compared to the yield stress of the steel material studied.

Cumulative damage theory. A popular approach to assess the damage cumulated on a stress time series is to consider the fatigue contribution of each stress cycle according to the S-N curve. Palmgren-Miner's rule defines the *cumulative damage* d_c by summing the fatigue

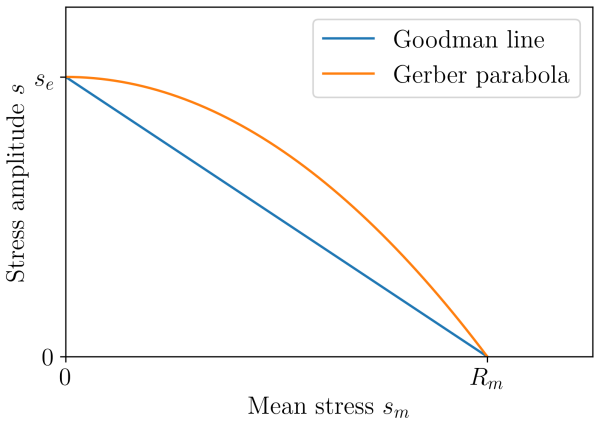


Figure 2.11 Illustration of the Haigh diagram representing the combination of stress mean and amplitude leading to the same fatigue endurance.

contributions of each stress cycle k, regardless of their order of appearance:

$$d_c = \sum_{j=1}^{k} \frac{1}{N_c\left(s^{(j)}\right)}. \tag{2.18}$$

In this theory, the material reaches fatigue ruin when the cumulative damage exceeds one. A common practice when using Palmgren-Miner's rule is to gather the stress cycles in a set of bins. This practice induces an integration error, which becomes significant as the number of bins is reduced. In the following, the cumulative damage is computed without binning, as defined in Eq. (2.18).

Spectral methods were also introduced to quantify fatigue damage in the late 80s. The main idea is to infer a PDF over the amplitudes of the stress cycles identified by rainflow counting, typically using a mixture of parametric distributions. From this PDF, one can derive the fatigue endurance and the cumulated damage (see further details in the review of **?**). In the context of wind turbine fatigue assessment, spectral approaches showed to be unsuited in some cases (e.g., for blades' fatigue **?**). Overall, fatigue estimation in the time domain does not represent an important computational effort compared to the simulation of the wind turbine's physics.

Nonlinear fatigue models were developed in the 90s (**?**) to take into account the order in which the loading cycles were applied to the structure. For offshore wind turbine applications, the recent work of **?** studied a probabilistic version of nonlinear fatigue models. Unfortunately, this refined approach requires a larger computational effort and calibration over experimental tests of various parameters (**?**).

2.4 Design and operation practices

The design and operation of offshore wind turbines are at the intersection of various engineering, environmental and social considerations. Regardless of the different bottom-fixed or floating technologies, OWTs are dynamically excited structures evolving in a harsh offshore environment.

To operate such assets over up to 25 years of lifespan, multiple aspects should be assessed, from soil modeling, studies of environmental impact, grid integration, manufacturing quality, port logistics, to marine growth management, and maintenance. This section resumes the main types of OWT technologies, as well as the main design and operation practices.

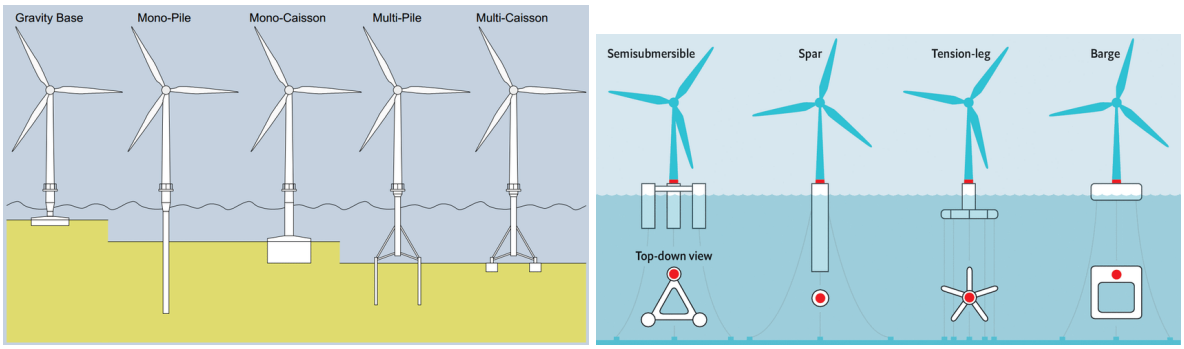
2.4.1 Types of technologies and preliminary design

The multiple OWTs technologies developed over the last two decades can be gathered into two groups: bottom-fixed or floating technologies. Fig. 2.12a and 2.12b respectively illustrate the different types of bottom-fixed or floating technologies. At this stage, the bottom-fixed solutions present more maturity while floating technologies are still transitioning from the phase of large demonstrators to industrial wind farms. In France, the current development of offshore wind energy has led to the construction of the two first industrial projects (both managed by EDF Renewables). On the coast of Saint-Nazaire, 80 bottom-fixed wind turbines were built on monopile foundations, altogether producing up to 480 MW. On the Mediterranean coast, the first French industrial floating project was recently installed 20 km offshore the coast of Marseille. This pilot project, called "*Provence grand large*", is composed of three turbines operating on so-called "tension-leg platforms", delivering 25 MW of nominal power.

In order to lift water depth limitations associated with bottom fixed technologies (technical limit around 60 meters), floating pilot projects have emerged across the world. However, the wind energy industry still tests different floating technologies in terms of cost efficiency and durability (as listed in **?**). An example of some farm projects with different types of technologies is described hereafter:

- **Semi-submersible**: a pilot project of three 10MW turbines called "*les éoliennes du Golfe du Lion*" in the south of France relies on a semi-submersible technology developed by the company Principle Power (**?**).
- **Tension-leg**: a pilot project of three 8MW turbines called "*Provence grand large*" exploits tension-leg platforms co-developed between the IFPEN national laboratory and the company SBM (**?**).
- **Barge**: a pilot project of three 10MW turbines called "EOLMED" uses the floater developped by the company Ideol (**?**).
- **Spar**: the Norwegian oil and gas company Equinor chose the spar technology (**?**) to equip its floating wind farm of 88MW, named "Hywind Tampen".

The turbines installed offshore over bottom-fixed foundations or floating structures present the same properties and components. As described in Fig. 2.13, the structure of a wind turbine is composed of blades made of composite materials, while the tower, the transition piece and the foundation (e.g., monopile) are made out of steel. The steels used for the foundation and the tower are typical structural steels (i.e., steels with low carbon concentration such as the S355).



(a) Bottom-fixed (b) Floating

Figure 2.12 Main offshore wind turbine technologies (sources: ??).

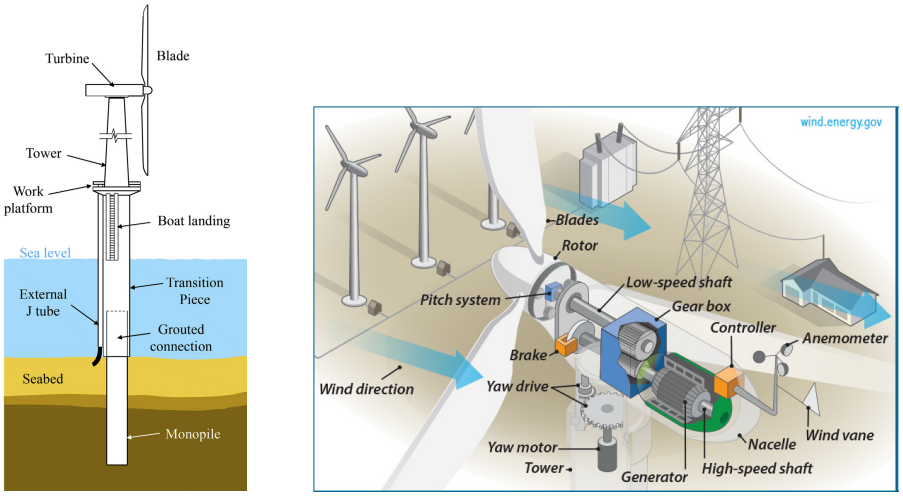
Inside the nacelle, the gearbox adapts the rotation speed to suit the energy conversion system (i.e., generator). To improve the reliability of the components, manufacturers offer without gearboxes, called "direct-drive". This technology is relevant offshore, as the maintenance constraints are higher. However, the corresponding generators used in this situation operate at a lower rotation speed. Adapting the generators significantly increases their weight and requires the use of larger permanent magnets increasing their price.

The construction of an offshore wind farm requires several years of project planning, administrative procedures, consultation with public opinion, and design. International standards define the recommended practices and requirements related to the design and operation of OWTs. Among them, the IEC 61400 is subdivided into many parts, including the general one (?) and other parts detailing specific topics. To validate the structural integrity of a wind turbine design, the standards recommend simulating the behavior of the OWT (using the methods described in Section 2.3) for many environmental conditions, called "design load cases" (DLC). As the environmental conditions depend on the site studied, the standards provide generic DLCs depending on a rough classification of the environmental conditions. Following the terminology in civil engineering, the structure is designed for "fatigue limit states" (FLS) and "ultimate limit states" (ULS) with respect to environmental conditions. For fatigue, advanced sampling methods relying on environmental data measured on-site will be introduced in Chapter 4. Beyond the main solicitations resulting from environmental loading, various aspects should be considered around offshore wind turbines.

2.4.2 Further design considerations

The present section focuses on different topics to be addressed ahead of, or during the design and operation of offshore wind turbines.

Soil modeling. The accurate geotechnical description of an offshore site plays an important role in the design and stability of bottom-fixed offshore wind turbines. The seabed soil properties are far from uniform in a wind farm, forcing the designer to adapt the foundations within a



(a) Monopile structure diagram (b) Major components in the nacelle

Figure 2.13 Diagrams of an offshore wind turbine structure (source: (?)) and nacelle (source: US ODE).

farm. Prior to the installation, geotechnical surveys and soil testing are conducted to assess parameters such as soil composition, density, strength, and seabed stability.

To model the dynamic behavior of foundations, certification companies adapted their methods from the oil & gas industry to offshore wind energy (?). For monopile foundations, the "$p-y$" method is often used to model soil-structure interactions. Assuming that these interactions are purely lateral, this method defines a set of non-linear lateral springs along the foundation's height. Together, the springs model the relation between the soil resistance "p" and the lateral displacement "y". Generally speaking, monopile foundations for OWT tend to be more rigid than for oil & gas platforms, as the cyclic loading on wind turbines induces more fatigue (see the case study presented in ?). However, various contributions in wind energy extended the use of $p-y$ curves to the case of multidirectional and irreversible displacements (?). In summary, geotechnical considerations are essential for offshore wind turbine design, and the variability of soil properties within a wind farm necessitates a tailored approach to foundation design. Finally, the consideration of uncertainties in this field is still an open research topic (?).

Marine growth. The bio-colonization of offshore structures and submarine cables is a significant concern in the maintenance and operation of OWTs. Elements exposed to the colonization of marine organisms, such as mussels, can cause several adverse effects. Firstly, the added weight increases the mass of the turbine and its foundation, potentially changing the dynamics of the systems and its structural integrity (??). Secondly, marine growth changes the surface's roughness of the submerged components, which can create fluctuating hydrodynamic loads and vibrations (?). To limit its impact on the reliability of OWTs, this phenomenon is addressed with regular preventive cleaning measures as part of the maintenance planning.

Global scour. The large-scale erosion of seabed sediment around bottom-fixed offshore wind turbine foundations, also called "global scour", poses different problems. The stability of the foundation is first reduced, potentially leading to tilting. Moreover, the load distribution changes, causing uneven stresses and increased fatigue. Finally, submarine cable exposure increases the risk of damage and electrical faults. As global scour is a critical element of the long-term OWT reliability, various mitigation measures are reviewed in **?**, including scour protection, and scour-robust foundation design.

Port logistics. In the installation and maintenance of such large-scale systems, port logistics plays an important role considering the international supply chain involved. The coordination, transportation and assembly of massive wind turbine components, foundations, and supporting infrastructure requires meticulous planning and execution. In accordance, the costs of handling operations and maintenance represent an important share of the *levelized cost of energy* (LCOE) (**?**).

In his review of OWT installation techniques, **?** describes the foundations' and components' installation processes depending on the OWT technology. Because of their large scale, most structural assembly (e.g., blades, or floater) are done on dedicated port docks, making the port choice critical. The assembled turbines are then transferred offshore with specialized vessels, such as installation jack-ups. Timing and synchronization are critical, as weather windows for handling operations can be limited.

Grid integration. Unlike traditional centralized energy production plants (i.e., nuclear and fossil), wind energy has a considerable impact on grid management. The intermittency of offshore wind generation is driven by variable wind conditions, which disrupts the electricity supply (**?**). Then, grid balancing becomes more complex as variable and distributed production sources are introduced. Wind turbine integration often requires more flexibility from the grid, resulting in grid infrastructure upgrades (e.g., energy storage) and advanced grid management.

Environmental impact and social acceptance. The fast development of offshore wind turbines in Europe raises questions regarding environmental and social impact. In their review, **?** showed that the installation and operation were shown to disrupt marine ecosystems. Further studies should be realized to better understand the reliance of the ecosystems on this change. This industry also affects other marine activities (e.g., fishing or tourism), and coastal landscapes, which need to be discussed during the regional marine spatial planning. Finally, social acceptance of offshore wind projects varies across Europe, often split between local disturbances and the regional economic activity generated.

Manufacturing quality. The manufacturing of structural wind turbine components is subject to several uncertainties that can affect the overall quality and performance of OWTs. For example, the manufacturing process of composite blades can lead to inconsistencies in the final

product. Imperfections in the composite material, like air pockets or delamination, can weaken the blades and reduce their lifespan. Additionally, variations in manufacturing processes can result in differences in blade weight, which impacts the turbine's performance. Regarding steel components, OWTs are mostly assembled by bolted and soldered joints. Inconsistent soldering, variations in material properties, and potential flaws in the joints can compromise the structural integrity of OWTs (**?**). These uncertainties in manufacturing quality can pose significant challenges in ensuring the reliability and longevity of the structures. Note that at the design phase, *stress concentration factors* are defined by standards to take into account the local change in material properties created by soldering. Rigorous quality control, material testing, and manufacturing standards are essential to maintain the safety and efficiency of wind energy installations.

Maintenance and end-of-life management. To ensure the continued performance and availability of wind turbines, advanced maintenance planning is essential. Maintenance activities involve inspections, repairs, component replacements, and addressing issues such as corrosion, or electrical faults. Preventive maintenance strategies (reviewed by **?**) minimize the asset's unavailability and extend its lifespan.

Once the wind farms reach their planned lifetime (typically between 20–25 years), the operator has the choice between decommissioning, "repowering", or "revamping" the assets. Usually, revamping implies an intermediate renovation of the WT. In most cases, the underperforming major components are replaced while the structural components are kept. Alternatively, repowering is a strategy for reusing the foundations of a wind farm to install brand-new turbines. This approach is often an opportunity to increase the scale and performance of the old turbines.

As the first generation of wind farms currently reach their end-of-life, an important problem arises from recycling large amounts of blades made out of composite materials. Different processes for recycling composite material are reviewed in **?**, including mechanical, pyrolysis and chemical techniques. However, recycling composites is a complex and energy-consuming operation, that needs to be further studied. The recent lifecycle study of floating OWT in the Mediterranean region by **?** showed that effective maintenance and proper decommissioning planning is essential for ensuring cost-effective yet durable lifecycle management.

2.5 Uncertain inputs

Following the general diagram of uncertainty quantification in Fig. 1, this section focuses on the definition of the uncertain inputs and their corresponding probabilistic model (step B). In our case, the generic term of "inputs" refers to the inputs to the wind turbine numerical model illustrated in Fig. 2.6, which will be considered as random afterward.

The random variables studied in this work are split into two groups (assumed independent) which are respectively called *environmental variables* and *system variables*. First, the environmental variables are a collection of variables characterizing the long-term metocean

conditions near a wind farm. Even if the associated random vector presents a complex dependence structure, this source of variability is well-defined after the wind potential measurement campaigns.

The second group of uncertain inputs is related to the wind turbine system. A wide range of uncertainties can be taken into account in such systems, such as material properties, manufacturing quality, soil conditions, control error, corrosion, marine growth, aerodynamic damping, etc. Among them, a restricted list of four variables is kept according to sensitivity analysis results from the literature and expert knowledge.

2.5.1 Environmental inputs

During the planning and operation of a wind farm project, the metocean conditions are studied using different sources of information. At the early stage, datasets generated by fine mesoscale numerical simulation can be used to assess the wind potential. This was typically the case during the call for tenders[2] issued by the French government regarding the construction of two floating offshore wind farms in the south coast of Brittany (of respectively 250 and 500 MW of nominal power). Open-access environmental data of the sea-states in this region were available, as a result of mesoscale simulations (**?**) realized by EDF R&D. In a second phase, the local conditions are measured using a meteorological mast with wind speed cup anemometers at different heights and wave boys are generally installed in the vicinity of the future farm. As a cheaper alternative to met masts, new measurement technologies such as floating LIDARs (standing for "light detection and ranging") were studied by **?**. Then, different adequation methods between the local measures and the data obtained by mesoscale simulations were reviewed in **?**. Finally, after the installation of the turbines, the acquisition system (usually called SCADA, for "supervisory control and data acquisition") measures wind conditions with a sampling period of ten minutes.

In the present work, two wind farm projects are partially studied: the Teesside wind farm, operating in the North Sea, and the south Brittany floating project, at the stage of tenders call. Table 2.2 summarizes the variables considered as random hereafter. The inference of such data will be discussed in Chapter 3 of this manuscript. Note that the environmental data resulting from the SCADA system of the Teesside wind farm is confidential, and will be represented as anonymized data in the following.

2.5.2 System inputs

As mentioned earlier, multiple parameters in a wind turbine system can be considered as uncertain. Our study focuses on the effects of uncertainties on fatigue damage over the structure. Therefore, the literature review of the sensitivity analysis on offshore wind turbine fatigue helped us narrow down a few system variables. **?** explored the sensitivity analysis of many variables on the fatigue of a wind turbine in Teesside. Even if the use of the Morris method is

[2] https://eolbretsud.debatpublic.fr/

Name	*Notation*	*Description*
Mean wind speed	U	10-min. average horizontal at 10m
Turbulence	σ_s	10-min. standard deviation
Wind direction	θ_{wind}	Wind directions
Significant wave height	H_s	Significant wave height per hour
Peak wave period	T_p	Peak 1-hour spectral wave period
Wave direction	θ_{wave}	Wave directions

Table 2.2 Marginal distributions of the environmental random variables

questionable, the results allowed us to screen out some variables. For example, the uncertainties related to the corrosion, the wind shear exponent, or the nacelle mass showed a limited impact on the fatigue. By crossing the conclusions of various research with the expert knowledge among partners from the HIPERWIND European project, the system variables considered uncertain in the following are summarized in Table 2.3. Each of them is assumed independent, with a marginal probabilistic model arising from the literature.

Name	*Notation*	*Marginal model*	*Description*
Soil coefficient	S	Normal ($\mu = 1., \sigma = 0.3$)	Applied to the soil stiffness matrix
Yaw misalignment	θ_{m}	Normal ($\mu = 0., \sigma = 0.3$)	Error in wind alignment
SN curve coefficient	a	Log-normal ($\mu = 1, \sigma = 0.3$)	See **?**
Critical damage	D_{cr}	Log-normal ($\mu = 1, \sigma = 0.3$)	See **?**

Table 2.3 Marginal distributions of the system random variables

2.5.3 Probabilistic fatigue assessment

The definition of a fatigue endurance model has a main impact on fatigue damage assessment. However, the S-N curves usually describing the endurance of a material are built on repeated laboratory experiments. Even if the need for random S-N curves has long been expressed in the field of fatigue experiments (**?**), their probabilistic description was better formalized in (**??**).

The models proposed in **?** are based on the experimental procedure used to build the S-N curves. For identical steel specimens, a cyclic loading with fixed amplitude is repeated until fatigue ruin. Because of variations in the material's microstructure, the fatigue endurance for the same cyclic solicitation is random. This variation is commonly assumed to follow a log-normal distribution in the literature. A probabilistic model of the S-N curve naturally comes:

$$\log(N_c(s,\omega)) = \log(a) - m\log(s) + \log(\varepsilon(\omega)) \tag{2.19a}$$

$$\Rightarrow N_c(s,\omega) = a\, s^{-m}\, \varepsilon(\omega), \tag{2.19b}$$

where $\log(\varepsilon(\omega)) \sim \mathcal{N}(0, \sigma_{N_c} = 0.2)$ is assumed when no measurement is available (according to Appendix F.5 from **?**).

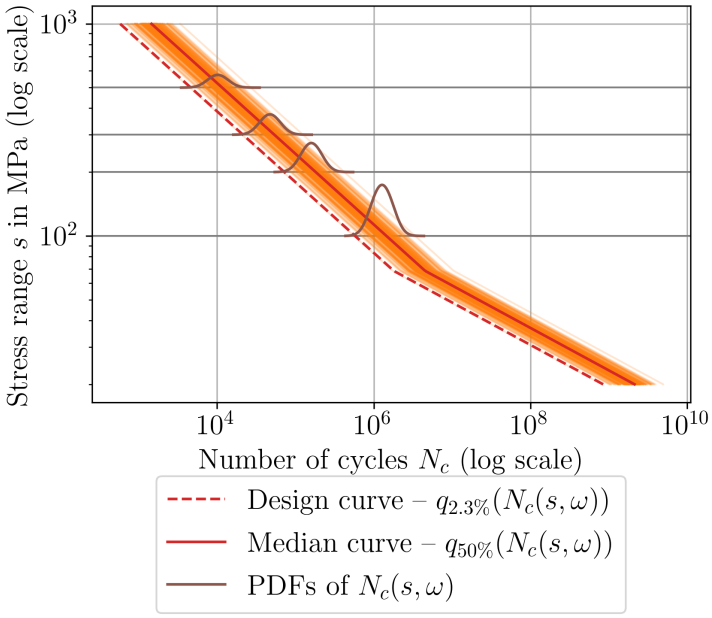


Figure 2.14 Illustration of a probabilistic S-N curve according to the model defined in **?**.

This uncertainty can be injected in the Miner-Palmgren rule defined in Eq. (2.18):

$$d_c(\omega) = \sum_{j=1}^{k} \frac{1}{N_c(s^{(j)},\omega)} = \sum_{j=1}^{k} \frac{1}{a\left(s^{(j)}\right)^{-m}\varepsilon(\omega)} = \frac{1}{\varepsilon(\omega)} \sum_{j=1}^{k} \frac{1}{a\left(s^{(j)}\right)^{-m}} \tag{2.20}$$

Then, this uncertainty can be assessed as a pure post-processing of fatigue damage results computed with a single deterministic S-N curve.

In wind energy standards, S-N curves for design are actually a conservative envelope of the measured fatigue endurance. Annex F.7 from **?** describes how to define a design S-N curve from fatigue measures. Assuming that the fatigue endurance follows a Gaussian distribution (on a logarithmic scale), the design S-N curve $N_c^{\text{design}}(s)$ is the curve at two standard deviations σ_{N_c} below the median curve.

Using the design S-N curve given in Section 2.4.6 **?** and the normality assumption, one can reconstruct the median S-N curve by taking:

$$q_{50\%}[\log(N_c(s,\omega))] = \log(N_c^{\text{design}}(s)) + 2\,\sigma_{N_c}. \tag{2.21}$$

Fig. 2.14 illustrates the design curve defined by DNV for tubular joints (in Section 2.4.6) and the reconstructed probabilistic model according to the previous assumptions.

2.6 Conclusion

This chapter proposed an overview of offshore wind turbine modeling and design. It introduced concepts related to the description and simulation of metocean conditions. The impact of the wake on the performance and on unsymmetrical fatigue loading was also explained. Then,

the different theories considered in OWT modeling were introduced such as aerodynamics, hydrodynamics, structural dynamics and control. A variety of software implementations exist for this purpose, but special attention was brought to DIEGO, a numerical model developed by EDF R&D. As a perspective, uncertainty quantification could benefit from the different fidelities proposed to model OWT systems presenting very nonlinear transient phases.

To understand the design of OWTs, the most common technologies of bottom-fixed and floating turbines were presented. Then, a focus on a few critical topics to be considered during design and exploitation was proposed. In light of the previous elements, the variables considered random in this work were listed with a particular focus on probabilistic fatigue.

This growing industry faces various challenges, for example, related to the important use of primary commodities, the cohabitation of offshore dynamic structures with an ecosystem, composite materials recycling, etc. Uncertainty quantification is a tool for understanding some of these problems, however, many uncertainties are hard to characterize and quantify. For example, manufacturing quality issues were revealed by Siemens Gamesa[3]regarding wrinkles on the surface of some blades.

[3]L. Pitel and R.Millard. (August 7, 2023). Siemens Energy warns of €4.5bn loss from ailing wind turbine division. *Finacial Times*. https://www.ft.com/content/df8947cd-4bab-46ff-804e-b28de4b5a0f0

Part II:

Contributions to uncertainty quantification and propagation

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

Voltaire

Chapter 3

Kernel-based uncertainty quantification

Parts of this chapter are adapted from the following references:

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).*

E. Vanem, E. Fekhari, N. Dimitrov, M. Kelly, A. Cousin and M. Guiton (2023). “A joint probability distribution model for multivariate wind and wave conditions”. In: *Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2023).*

3.1 Introduction

The main sources of solicitation in offshore design reside in the metocean conditions. To accurately verify a structural design against the joint wind and wave conditions, these random excitations must be carefully modeled. Offshore structures are usually certified against ultimate limit states (related to the occurrence of extreme metocean conditions) and fatigue limit states (related to the average fatigue over the metocean conditions). In this context, the probabilistic framework is typically used to model the joint distribution of random variables describing the metocean conditions (listed in Section 2.5).

Note that a given probabilistic model might describe well the central behavior of the environmental distribution but not its tail behavior (and vice-versa). Extreme value theory develops specific methods to model the far tails of distributions (**?**). Modeling the tails is not the priority in the present work since the focus is on mean fatigue estimation.

The environmental random variables studied present different particularities. First, an offshore wind turbine project leads to the collection of an important amount of metocean data (possibly merged with data from mesoscale simulations). Second, their dependence structure is complex, making the probabilistic modeling more complicated.

This chapter explores different aspects of the environmental conditions’ uncertainty quantification. The theory of some nonparametric copulas is introduced before their use in metocean conditions inference. A semiparametric approach is applied to the South Brittany data, mixing parametric modeling of the marginals with nonparametric modeling of the copula. To visually analyze multivariate distributions, the *copulogram* is a new tool that decomposes the marginal effects and the dependence structure of a joint distribution.

At the scale of a wind farm, each turbine perceives different metocean conditions as the wake of other turbines creates wind perturbations. To study this perturbation, an engineering wake model (see Section 2.2.2) was used to obtain one perturbed environmental distribution per turbine. This work applies a kernel-based discrepancy (the maximum mean discrepancy) to compare wake-induced perturbations. In a second phase, this discrepancy is used to gather wind

turbines perceiving similar perturbations. This clustering can be used to perform uncertainty propagation at the farm scale by considering a few turbines with are representative of a cluster.

3.2 Dependence modeling with nonparametric copula

In uncertainty quantification, the lack of knowledge can lead to rough assumptions regarding the dependence modeling. However, an accurate representation of the uncertain inputs is of prime importance. For example, the work of **?** demonstrates the influence of the dependence model on the estimation of rare event probabilities by studying the same problem with different copula models.

When inferring a probabilistic model over a multivariate dataset, one can decompose the problem into the fit of a set of marginals and the fit of a copula (see the Sklar Theorem 1). In the case of metocean conditions, the fit of the marginals is not problematic considering the amount of data available. However, the complex dependence structure appears to be more challenging. Different strategies to model the dependence for multivariate distributions are briefly summarized hereafter:

- **Vine copulas** (also known as pair copula) decompose the joint distribution as a product of conditioned bivariate copulas organized in a tree-like structure called a vine. This approach proved to be very efficient, but it requires the definition of the vine and the bivariate parametric copulas (Joe and Kurowicka, 2011).
- **Conditional modeling** defines the joint distribution as a product of univariate conditional distributions. In practice, the parameter of a marginal is defined as a function of other marginals (see e.g., **?**).
- **Multivariate KDE** is another way to capture the dependence together with marginal effects. As the dimension and the size of the dataset increase, this method becomes less tractable (**?**).
- **Nonparametric copulas** are methods uniformly approximating an empirical copula without any assumption on a dependence structure. They will be further described and used for metocean conditions inference in the present chapter.

Remark 2. The strategy referred to as "conditional modeling" can in fact be expressed as a copula (**?**) for continuous variables. For example, in the bivariate case of a continuous random vector $\mathbf{X} = (X_1, X_2)$ with PDF $f_{\mathbf{X}}(\mathbf{x})$, CDF $F_{\mathbf{X}}(\mathbf{x})$ and density copula c:

$$f_{\mathbf{X}}(\mathbf{x}) = f_{X_1}(x_1)\, f_{X_2|X_1}(x_2|x_1) = f_{X_1}(x_1)\, f_{X_2}(x_2)\, c\left(F_{X_1}(x_1), F_{X_2}(x_2)\right) \tag{3.1a}$$

$$\Leftrightarrow c\left(F_{X_1}(x_1), F_{X_2}(x_2)\right) = \frac{f_{X_1}(x_1)\, f_{X_2|X_1}(x_2|x_1)}{f_{X_1}(x_1)\, f_{X_2}(x_2)} \tag{3.1b}$$

The notions related to the copula theory are further introduced in the monographs of **???** while the key properties are introduced hereafter.

3.2.1 Preliminary definitions and properties

Let us consider a random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d$ defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Its probability distribution $\mathbb{P}_{\mathbf{X}}$ can be represented by a CDF $F_{\mathbf{X}}$ and PDF $f_{\mathbf{X}}$. The functional definition of a *d-dimensional copula* (or simply "d-copula") is a density function $C : [0,1]^d \mapsto [0,1]$ whose marginals are uniformly distributed on $[0,1]$.

Theorem 3 (Copula). *A function $C : [0,1]^d \mapsto [0,1]$ is a d-copula if, and only if, it presents the following properties:*

- *The function C is "grounded" (also called "anchored"): $C(u_1, \ldots, u_d) = 0$ if $u_j = 0, \forall j \in \{1, \ldots, d\}$;*
- *The marginals of C are uniform, then: $C(1, \ldots, u_j, \ldots, 1) = u_j, \forall j \in \{1, \ldots, d\}$;*
- *The function C is "d-increasing", meaning that for any hyperrectangle $A \subset [0,1]^d$, the corresponding volume induced by C is positive (see ? p.7).*

A copula is bounded by two functions according to the Fréchet-Hoeffding bounds.

Theorem 4 (Fréchet-Hoeffding bounds). *If a function $C : [0,1]^d \mapsto [0,1]$ is a d-copula, then it respects the following bounds for all $\mathbf{u} \in [0,1]^d$:*

$$W(\mathbf{u}) = \max(1 - d + u_1 + \cdots + u_d, 0) \leq C(\mathbf{u}) \leq M(\mathbf{u}) = \min(u_1, \ldots, u_d). \tag{3.2}$$

Where the upper bound M is still a copula while the lower bound W is only one for $d = 2$.

The rank transform plays an essential role in understanding copulas. Considering a continuous random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{x}}$ and the sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\} \sim \mathbf{X}$, its *ranks* $\mathbf{R}_n = \left\{\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(n)}\right\} \in \mathbb{N}^n$ correspond to the indexes of its order statistics:

$$r_j^{(i)} = n \widehat{F}_{X_j}(x_j^{(i)}) = \sum_{l=1}^{n} \mathbb{1}_{\left\{x_j^{(l)} \leq x_j^{(i)}\right\}}, \quad \forall j \in \{1, \ldots d\}, i \in \{1, \ldots n\}, \tag{3.3}$$

where $\widehat{F}_{X_j}$ stands for the marginal empirical CDF associated with the random variable X_j.

Theorem 5 (Rank-invariance). *Considering a random vector $\mathbf{X} = (X_1, \ldots, X_d)$, a set of mappings $\{r_j(\cdot)\}_{j=1}^d$, and the image random vector $\mathbf{R} = (r_j \circ X_1, \ldots, r_d \circ X_d)$. If the mappings are strictly increasing (which is the case for the rank transform introduced in Eq. (3.3)), then, the copula associated to $\mathbf{R}$ is invariant by transformation: $C_{\mathbf{X}} = C_{\mathbf{R}}$. A proof is presented in ? p. 57.*

Transforming in the ranks generally reduces the effect of outliers and ensures more robust estimates. The invariance by the rank transform of copulas allows the estimation of different *dependence measures* in the ranked space.

Spearman's rho. Is a well-known dependence measure, also called the "Spearman's rank correlation coefficient", which is defined for two random variables X_i, X_j as:

$$\rho^{\mathrm{S}}(X_i, X_j) = \frac{\mathrm{Cov}(r_i(X_i), r_j(X_j))}{\sigma_{r_i(X_i)}\, \sigma_{r_j(X_j)}}, \tag{3.4}$$

an equivalent definition exists, using the copula C between the joint distribution of X_i and X_j and the independent copula $\Pi(u_i, u_j) = u_i\, u_j$:

$$\rho^{\mathrm{S}}(X_i, X_j) = 12 \int_{[0,1]^2} C(u_i, u_j)\, \mathrm{d}u_i \mathrm{d}u_j - 3 = 12 \int_{[0,1]^2} (C(u_i, u_j) - \Pi(u_i, u_j))\, \mathrm{d}u_i \mathrm{d}u_j. \tag{3.5}$$

Kendall's tau. Also referred to as the "Kendall's rank correlation coefficient", is defined for a pair of random variables (X_i, X_j) and their respective independent copies (X_i', X_j') as:

$$\tau(X_i, X_j) = \mathbb{P}\Big((X_i - X_i')(X_j - X_j') > 0\Big) - \mathbb{P}\Big((X_i - X_i')(X_j - X_j') < 0\Big), \tag{3.6}$$

and can also be defined using the copula C between the joint distribution of the two random variables:

$$\tau(X_i, X_j) = 4 \int_{[0,1]^2} C(u_i, u_j)\, \mathrm{d}C(u_i, u_j) - 1 = 1 - 4 \int_{[0,1]^2} \frac{\partial C(u_i, u_j)}{\partial u_i} \frac{\partial C(u_i, u_j)}{\partial u_j}\, \mathrm{d}u_i \mathrm{d}u_j \tag{3.7}$$

These dependence measures fully rely on the copula and are both bounded between -1 and 1. Further properties and estimators of Spearman's rho and Kendall's tau are presented in **?** Section 2.4.

Upper/lower tail dependence. Considering the random vector $\mathbf{X} = (X_i, X_j)$ and the copula C underlying their joint distribution. The *upper/lower tail dependence* coefficients are defined as:

$$\lambda_U(X_i, X_j) = \lim_{\substack{u \to 1 \\ u<1}} \mathbb{P}\left(X_i > F_{X_i}^{-1}(u) | X_j > F_{X_j}^{-1}(u)\right) = \lim_{\substack{u \to 1 \\ u<1}} \left(2 - \frac{1 - C(u,u)}{1-u}\right) \tag{3.8a}$$

$$\lambda_L(X_i, X_j) = \lim_{\substack{u \to 0 \\ u>0}} \mathbb{P}\left(X_i \leq F_{X_i}^{-1}(u) | X_j \leq F_{X_j}^{-1}(u)\right) = \lim_{\substack{u \to 0 \\ u>0}} \left(\frac{C(u,u)}{1-u}\right) \tag{3.8b}$$

? further discusses the asymptotic limit and outlines the particular case of the bivariate Gaussian copula, for which the tail dependence measures are null, $\lambda_U = \lambda_L = 0$. Note that Kendall's tau and the tail dependence coefficients both have their associated plots, allowing us to compare the dependence of two distributions [add ref].

3.2.2 Empirical and checkerboard copula

The *empirical copula* was introduced by **?**, as an estimator of the copula C associated with the random vector $\mathbf{X}$. Since the normalized ranks are a creasing mapping that presents uniform marginals by construction, they are a natural empirical representation of the density copula. Considering a sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\} \sim \mathbf{X}$ with the respective ranks $\mathbf{R}_n = \left\{\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(n)}\right\}$, a definition of the empirical copula is:

$$C_n(u_1, \ldots, u_d) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \mathbb{1}\left\{\frac{r_j^{(i)}}{n} \leq u_j\right\}, \quad \mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d \tag{3.9}$$

Even if this function converges uniformly towards the copula C (according to the Glivenko-Cantelli theorem), it does not fulfill the conditions to be a copula (see e.g., **?**).

In this context, different methods may be applied to smooth the empirical copula into a genuine copula. This problem can be perceived as a functional approximation of the underlying copula C, which is unique for continuous variables (according to Sklar's Theorem 1). Let us consider a discretization of the unit hypercube as a grid:

$$G = \left\{\frac{0}{m_1}, \ldots, \frac{m_1}{m_1}\right\} \times \cdots \times \left\{\frac{0}{m_d}, \ldots, \frac{m_d}{m_d}\right\}, \quad \mathbf{m} = (m_1, \ldots, m_d) \in \mathbb{N}^d. \tag{3.10}$$

The *checkerboard copula* is a simple approximation of the empirical copula using the discretization G. This method is comparable to a multivariate histogram of the empirical density copula c_n (see the formal multivariate definition proposed by **?**). In the particular case for which $m_j = m, \forall j \in \{1, \ldots, d\}$, the checkerboard copula is called the "rook" copula, and expressed by Segers et al. (2017) as:

$$C_n^{\#m}(u_1, \ldots, u_d) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \min\left(\max(n u_j - r_j^{(i)} + 1, 0), 1\right). \tag{3.11}$$

This empirical copula has a low complexity (see **?**) and efficient results for large samples (**?**), however, its variance is comparable to the empirical copula for small-sized samples (Segers et al., 2017). It is proven to be a genuine copula and its asymptotic behavior was studied by various authors such as **??**. In the following, an approximation of the empirical copula with Bernstein polynomials is presented.

3.2.3 Empirical Bernstein and Beta copula

A few elements about Bernstein polynomials and their corresponding approximation are reminded before introducing the empirical Bernstein copula.

Bernstein polynomials and approximation

Let us first define the *Bernstein basis polynomial* of order $m \in \mathbb{N}$ as:

$$b_{m,t}(u) = \binom{m}{t} u^t (1-u)^{m-t}, \quad t \in \{0, \dots, m\}. \tag{3.12}$$

These polynomials present various interesting properties, such as their nonnegativity over $[0,1]$, being bounded by one, and offering a partition of unity on $[0,1]$ (**?**):

$$1 = \sum_{t=0}^{n} b_{m,t}(u)(x), \quad \forall x \in \mathbb{R}, \quad \forall n \in \mathbb{N}. \tag{3.13}$$

Bernstein's polynomials allow us to uniformly approximate any continuous and real-valued function defined on a compact set $f : [0,1]^d \mapsto \mathbb{R}$ (as they were used to demonstrate the Weierstrass approximation theorem). In the multivariate case, the *Bernstein approximation* of the function f can be written on a grid over the unit hypercube $G = \left\{\frac{0}{m_1}, \dots, \frac{m_1}{m_1}\right\} \times \cdots \times \left\{\frac{0}{m_d}, \dots, \frac{m_d}{m_d}\right\}, \mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}^d$, as:

$$B_{\mathbf{m}}(f)(\mathbf{u}) = \sum_{t_1=0}^{m_1} \cdots \sum_{t_d=0}^{m_d} f\left(\frac{t_1}{m_1}, \dots, \frac{t_d}{m_d}\right) \prod_{j=1}^{d} b_{m_j,t_j}(u_j), \quad \mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d. \tag{3.14}$$

The Bernstein polynomials approximate f such that $\lim_{m \to \infty} B_m(f) = f$ uniformly on $[0,1]$.

Bernstein polynomials for copula approximation

Copulas are continuous and bounded functions defined on a compact set (the unit hypercube). Therefore, they are good candidates to be approximated by Bernstein polynomials. The Bernstein approximation applied on an empirical copula C_n was introduced as *empirical Bernstein copula* (EBC) by Sancetta and Satchell (2004) for applications in economics and risk management:

$$B_{\mathbf{m}}(C_n)(\mathbf{u}) = \sum_{t_1=0}^{m_1} \cdots \sum_{t_d=0}^{m_d} C_n\left(\frac{t_1}{m_1}, \dots, \frac{t_d}{m_d}\right) \prod_{j=1}^{d} b_{m_j,t_j}(u_j), \quad \mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d. \tag{3.15}$$

In this expression, the evaluations of the empirical copula on the vertices of the grid are smoothed by the product of Bernstein polynomials. A respective approximation of the copula density can be directly expressed by deriving the previous formula. The EBC delivers a genuine copula, if and only if all the polynomial degrees $\{m_j\}_{j=1}^d$ are divisors of n (see Segers et al. 2017, Proposition 2.5).

In the particular case of regular grids, $\{m_j = m\}_{j=1}^d$, the EBC can be expressed as a mixture of beta distributions (Segers et al., 2017). Let us consider an n-sized rank sample, $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_d) \in \mathbb{N}^n$, and the degree m taken as divisor of n. Note that the r^{th} order statistic of an n-sized sample following a uniform $[0,1]$ is distributed according to the beta distribution $\mathcal{B}(r, n-r+1)$.

Considering these hypotheses, the EBC can be written as:

$$B_m(C_n)(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} F_{m,r_j^{(i)}}, \quad \mathbf{u} = (u_1,\dots,u_d) \in [0,1]^d, \tag{3.16}$$

where $F_{m,r}$ is the CDF of the beta distribution $\mathcal{B}(r, m-r+1)$ (also called the "regularized incomplete beta function"):

$$F_{m,r} = \sum_{t=r}^{m} \binom{m}{t} u^t (1-u)^{m-t}, \quad u \in [0,1], \quad r \in \{1,\dots,m\}. \tag{3.17}$$

Overall, the EBC is a very versatile tool which able to approximate complex dependence patterns. Moreover, Monte Carlo sampling on an EBC is straightforward and licit since it is a genuine copula. As a drawback, the estimation accuracy of this nonparametric method heavily relies on polynomial order tuning.

Asymptotic behavior of the empirical Bernstein copula

In practice, the choice of polynomial degree for an EBC leads to a challenging bias-variance tradeoff. For example, the particular case of $\{m = n\}$, introduced as the *empirical Beta copula* by Segers et al. (2017), tends to reduce the bias while increasing the variance. In this paper, the beta copula presents interesting results compared to the Bernstein or the checkerboard copula for small sample sizes (i.e., $n < 100$). Theoretically, the tuning of the degree was first optimized to minimize an "Asymptotic Mean Integrated Squared Error" (AMISE) of $B_{\mathbf{m}}(C_n)$:

$$\mathrm{AMISE}(B_{\mathbf{m}}(C_n)) = \mathbb{E}\left[\|B_{\mathbf{m}}(C_n) - C\|_2^2\right] = \mathbb{E}\left[\int_{\mathbb{R}^d} (B_{\mathbf{m}}(C_n)(\mathbf{u}) - C(\mathbf{u})\mathrm{d}\mathbf{u})^2\right]. \tag{3.18}$$

The seminal work of Sancetta and Satchell (2004) proves in Theorem 3 that:

- $B_{\mathbf{m}}(C_n)(\mathbf{u}) \to C(\mathbf{u})$ for any $u_j \in \,]0,1[$ if $\frac{m^{d/2}}{n} \to 0$, when $m, n \to \infty$.
- The optimal polynomial order in terms of AMISE is[1]: $m \lesssim m_{\mathrm{AIMSE}} = n^{2/(d+4)}, \forall u_j \in \,]0,1[$.

To illustrate the previous theorem, Fig. 6.1 represents the evolution of the m_{AMISE} for different dimensions and sample sizes (adapted from Lasserre 2022). In medium dimension, the values of m_{IMSE} tend towards one, which is equivalent to the independent copula. Therefore, high-dimensional problems should rather be divided into a product of smaller problems on which the EBC is tractable.

The polynomial order for EBC estimation is still a bottleneck that was studied over the years by different authors (see e.g., ???Segers et al. 2017). Meanwhile, other nonparametric approaches such as the "penalized Bernstein" and the "penalized B-spline" estimators were compared to the EBC and vine copulas in a benchmark realized by ?. The results showed that

[1]The sign $\lesssim$ stands for "less than or approximately".

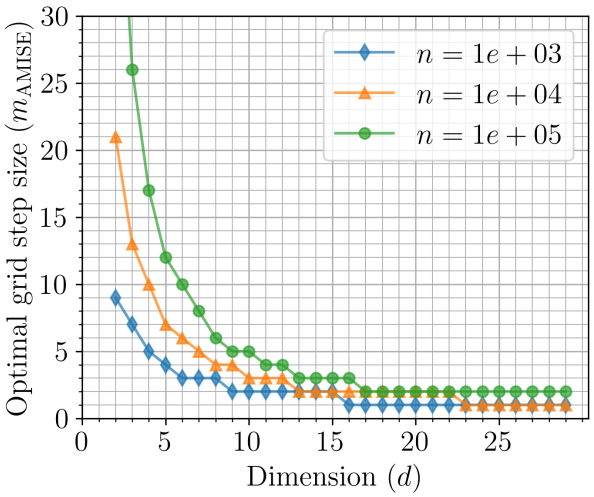


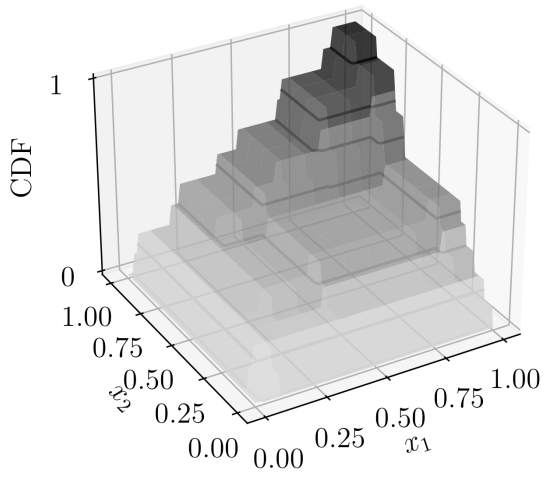
Figure 3.1 Evolution of m_{IMSE} for different dimensions and sample sizes.

the most performant methods vary depending on the problem studied (for different dimensions, sample sizes, and strength of dependence). Regarding tail dependence modeling, nonparametric approaches are generally limited, but recent contributions introduced Bootstrap procedures to better this aspect (**?**).

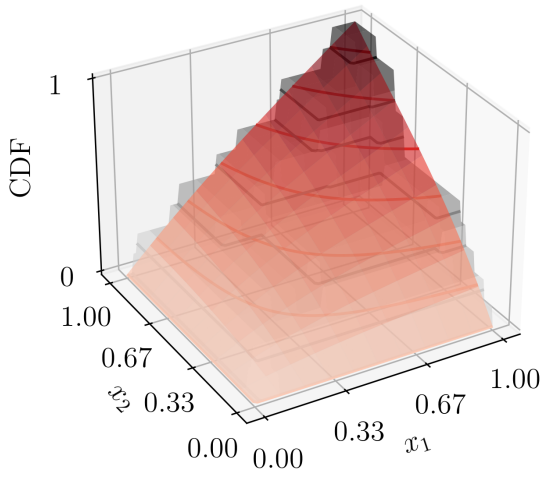
Illustrative example on a Clayton copula

Let us consider a bivariate Clayton copula C with parameter $\theta = 2.5$ (see **?**) A Monte Carlo sample with size $n = 10$ is generated on it, which is then used to build an empirical copula C_n as defined in Eq. (3.9). Fig. 3.2 (a) illustrates the empirical copula corresponding to the sample, with the shade of grey matching the CDF values. Then, the Bernstein approximation of the empirical copula (i.e., the EBC) is represented in Fig. 3.2 (b), (c), (d) according to the Eq. (6.4). The three versions of the EBC correspond to different polynomial orders, assuming that $(m_1 = m_2)$.

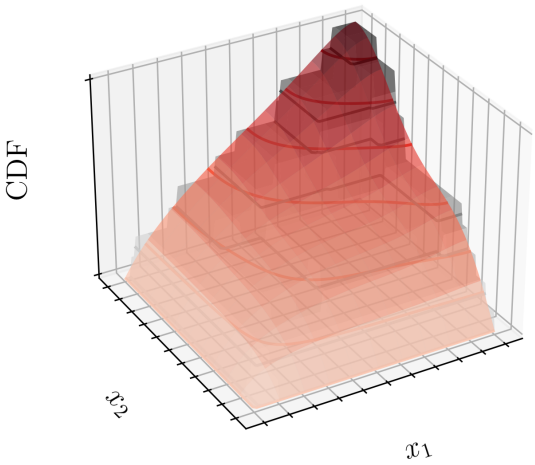
As the order increases, the bias between the EBC and the copula C tends to be reduced. Note that the second EBC in Fig. 3.2 (c) where $(m_1 = m_2 = n)$ is equivalent to the Beta copula. Moreover, increasing the order beyond the sample size definitely overfits the copula.



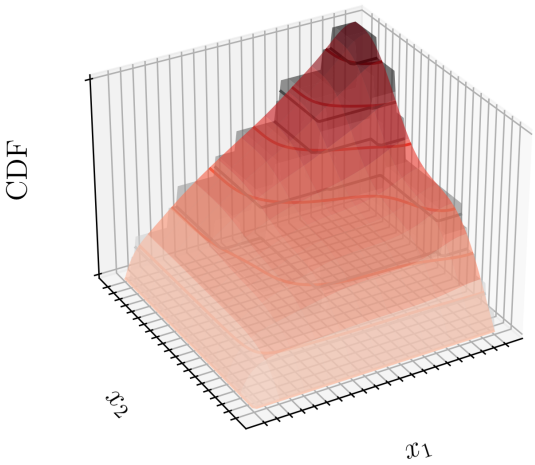
(a) Empirical copula C_n with $(n = 10)$.



(b) $B_{\mathbf{m}}(C_n)$ with $(m_1 = m_2 = 3)$.



(c) $B_{\mathbf{m}}(C_n)$ with $(m_1 = m_2 = 10)$.



(d) $B_{\mathbf{m}}(C_n)$ with $(m_1 = m_2 = 20)$.

Figure 3.2 Bernstein approximations of the empirical copula C_n (with size $n = 10$) of a Clayton copula (with parameter $\theta = 2.5$). The polynomial orders are assumed equal in the two dimensions $m_1 = m_2 \in \{3, 10, 20\}$.

3.3 *Copulogram*: a tool for multivariate data visualization

In statistics, data visualization offers a wide set of tools to analyze data. Multivariate data visualization is of great help in apprehending problems with dimensions higher than two. In the context of continuous variables, let us consider the n-sized sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\right\} \sim \mathbf{X} \in \mathcal{D}_{\mathbf{x}} \subseteq \mathbb{R}^d$. The marginal samples of $\mathbf{X}_n$ are denoted by $X_{n,j} = \{x_j^{(1)}, \dots, x_j^{(n)}\}, j \in \{1, \dots, d\}$.

Various techniques exist to represent multivariate data, such as the "parallel coordinate plot", also called "cobweb plot" (see e.g., **?**). For each sample $\mathbf{x}^{(i)} \in \mathbf{X}_n$, this plot draws a line passing by the values of $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_d^{(i)}]$. This representation was used in sensitivity analysis to illustrate the connections between a set of inputs and an output, however, it does not provide a good representation of the dependence structure between the inputs.

3.3.1 From the pairwise plot to the copulogram

Alternatively, the "pairwise plot", also named "generalized draftsman plot", was initially introduced by **?** to draw a matrix of scatter-plots between all the pairs of marginal samples $\{X_{n,i}, X_{n,j}\}, i \neq j \in \{1, \dots, d\}^2$. Because of the symmetry, the pairwise plot is usually represented on the lower triangle of the matrix. Later on, statisticians improved the pairwise plot by adding a histogram (or KDE) of the marginal samples $X_{n,j} j \in \{1, \dots, d\}$ on the diagonal. Additionally, the upper triangle was completed with the values of linear correlation for each pair of marginal samples $\{X_{n,i}, X_{n,j}\}, i \neq j$. This matrix of correlation coefficients is also known as "correlogram". Altogether, this matrix plot became known as the "scatter plot of matrices" (SPLOM).

However, the linear correlation coefficient is known to give a poor description of the dependence in nonlinear cases. When analyzing a continuous sample $\mathbf{X}_n \sim \mathbf{X}$, the Sklar theorem states that the dependence structure within the random vector $\mathbf{X}$ has a unique expression with its d-copula C. As mentioned in Section 3.2.1, the component-wise normalized ranks of the original sample $\mathbf{X}_n$ define the empirical copula density c_n (converging towards C as n increases).

To the best of our knowledge, the *copulogram* is a new multivariate data visualization tool improving the SPLOM by representing the empirical copula density c_n on the upper triangle of the matrix plot. This plot is an empirical decomposition of a multivariate sample in the vein of the Sklar theorem between marginals on the diagonal and copula on the upper triangle.

3.3.2 Implementation in a Python package

An open-source implementation is proposed in the python package `copulogram`. This code mostly relies on the Python package for data visualization `seaborn` (**?**). The developments are tracked and archived in a GitHub repository[2]and the package can be installed from the package-management system "PyPI".

Multiple visual options are offered by the `copulogram` package, as illustrated in the GitHub repository. For example, the user can represent the univariate samples on the diagonal or the bivariate samples in the triangles with kernel density estimation. Categorical variables

can be used to assign different colors depending on the data class. The colors can also vary depending on a continuous variable after defining a mapping between the values of this variable and a set of colors (also called colorbar).

Example #1: Iris flower dataset

The first example illustrates the copulogram on a widely used dataset in the machine learning community. The iris flower dataset was first introduced by Fisher and became a reference dataset for classification techniques. In the following lines of Python code, the dataset is loaded and the copulogram package is used to draw the new plot. The resulting copulogram applied to the iris flower data is represented in Fig. 3.3.

```
#!/usr/bin/python3
import seaborn as sns
import copulogram as cp
data = sns.load_dataset("iris")
copulogram = cp.Copulogram(data)
copulogram.draw(hue="species")
```

Since this data mostly presents linear dependencies, the copulogram is not very instructive. In other cases, the role of the dependence in the joint distribution is more important.

Example #2: Ishigami function

The Ishigami function is commonly used as a benchmark problem for global sensitivity analysis (GSA):

$$y = g(x_1, x_2, x_3) = \sin(x_1) + 7\,\sin(x_2)^2 + \frac{x_3^4 \sin(x_1)}{10}. \tag{3.19}$$

This uncertainty quantification problem considers an independent random input vector $\mathbf{X} = \prod_{j=1}^{3} X_j$. While the marginals in GSA benchmarks are usually assumed to be uniform, they will be considered Gaussian hereafter to distinguish the different elements of the joint distribution. Therefore, let us define $X_j \sim \mathcal{N}(0,1) \forall j \in \{1,2,3\}$. In this setup, the random inputs are independent but they each present interesting dependencies with the random output $Y = g(\mathbf{X})$.

A Monte Carlo sample with size $n = 10^3$ is generated, $\mathbf{X}_n \overset{\text{i.i.d.}}{\sim} \mathbf{X}$, and evaluated such that $Y_n = g(\mathbf{x}_n)$. The copulogram of the input-output sample $(\mathbf{X}_n, Y_n)$ is represented in Fig. 3.4. As expected, the scatter plots between the inputs in the upper triangle are uniform (representing an independent density copula).

In GSA, the work of **?** studied the discrepancy between the empirical density copula $c_n(X_j, Y), j \in \{1, \ldots, d\}$ and the independent copula to qualitatively assess the importance of X_j. In the same vein, the paper of **?** attempted to formalize a link between different GSA approaches based on copulas and to quantitative approaches as the Sobol' indices.

[2] GitHub repository: https://github.com/efekhari27/copulogram

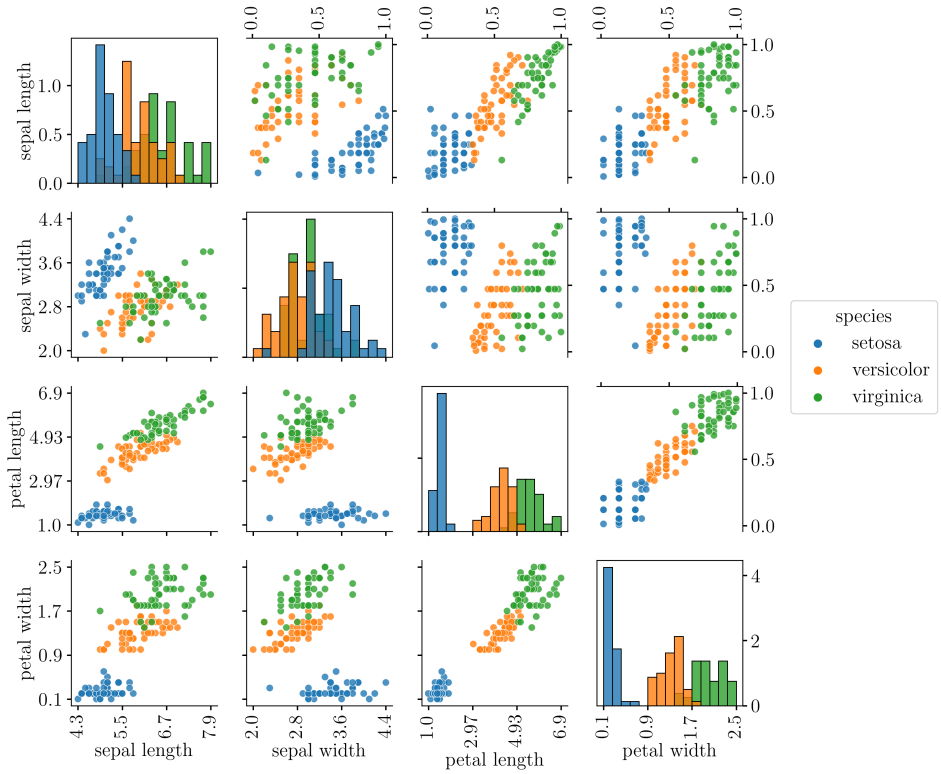


Figure 3.3 Copulogram of the iris flower dataset with colors assigned by the iris species.

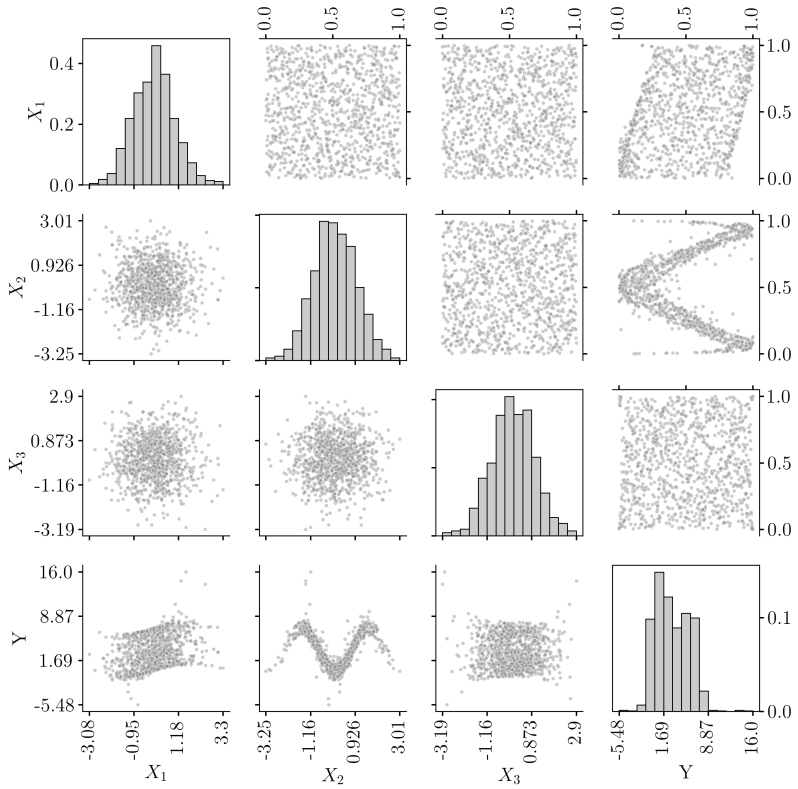


Figure 3.4 Copulogram of Monte Carlo sample (with size $n = 10^3$) of the inputs and outputs of the modified Ishigami problem.

3.4 Semiparametric inference of the South Brittany metocean conditions

Metocean conditions have been long studied in coastal and offshore engineering. Inferring multivariate probabilistic models on metocean data became essential in wind energy.

Numerous approaches are proposed in the literature to fit a model on environmental data. Among them, let us mention the use of parametric methods as the conditional modeling (e.g., **??**), or the construction of vine copulas (e.g., **???**). Nonparametric methods as the KDE were also applied in this context (e.g., **?**). The nonparametric techniques generally struggle to model the distributions' tails, even if the tails are essential to qualify structures for ultimate events. However, they are highly flexible and often easier to implement than parametric methods.

In this section, a semiparametric inference strategy is presented, composing some well-known parametric models for the marginals (e.g., Weibull distribution for the wind speed), with a highly flexible dependence modeling by the EBC. A metocean dataset is used to showcase the empirical Bernstein copula and its representation by the copulogram. This dataset from the ANEMOC (Digital Atlas of Ocean and Coastal Sea States atlas, **?**) gathers 32 years of preprocessed data (at an hourly resolution) from a location off the coast of South Brittany, France. A subset of 10^4 points is randomly selected among the ANEMOC data, which will be used to realize the semiparametric inference. The full code developed for this inference study is available on a GitHub repository[3].

3.4.1 Inference of the marginals

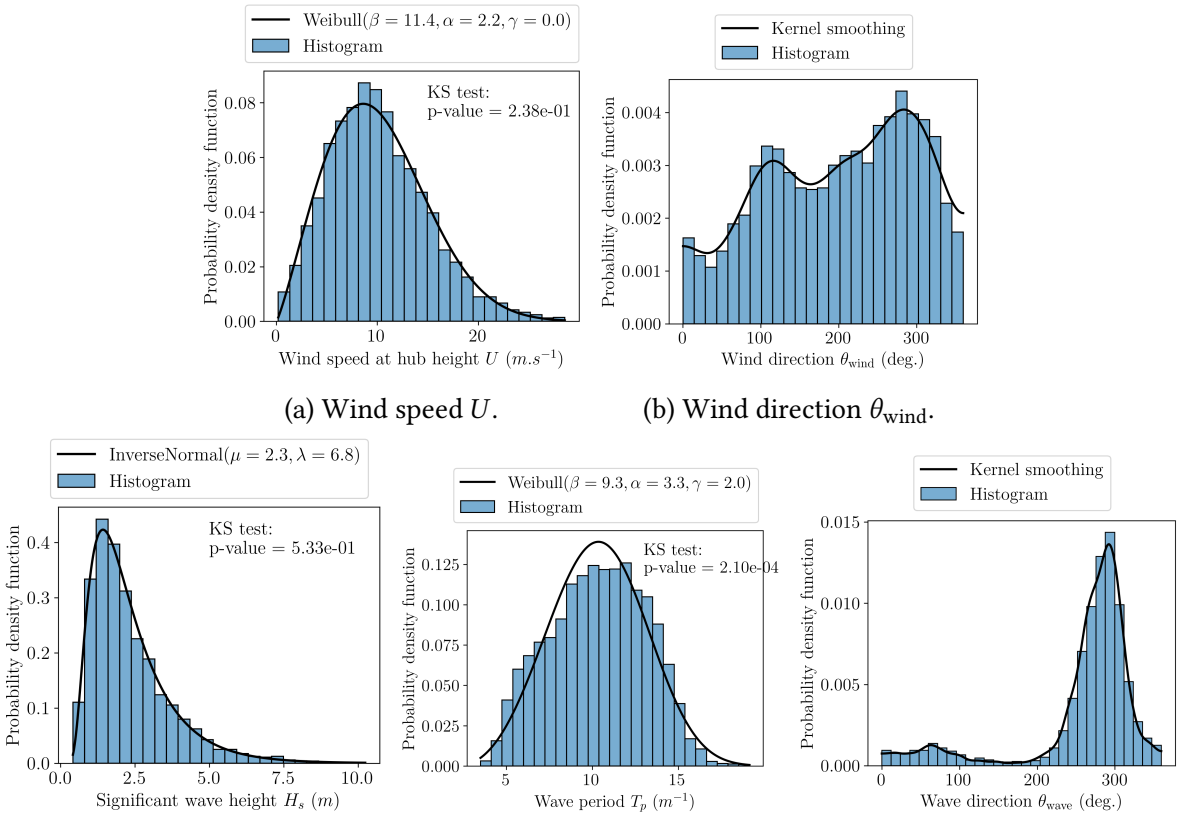
The variables studied to describe the environmental conditions match the ones defined in Table 2.2. Unfortunately, the turbulence is provided by the ANEMOC database and is therefore not fitted. A straightforward inference is performed on the data, resulting in the models presented in Table 3.1. The wind and wave directions are fitted by KDE to catch their multimodal behavior while the other variables by MLE on various parametric models. Note that some variations of KDE with kernels specific to circular data could be interesting to ensure the continuity of the model at the bounds (**?**).

The results of the marginals' inferences, plotted in Fig. 3.5 against histograms, are visually satisfying. Statistical testing is not necessary in our case since the actual topic of discussion is related to the inference of the dependence. Considering these marginals, a study of the copula inference can be developed.

[3]GitHub repository: https://github.com/efekhari27/thesis/blob/main/numerical_experiments/chapter3/south_brittany_inference.ipynb

Name	*Notation*	*Fitted model*	*KS p-value (α= 5%)*
Wind speed	U	Weibull ($\beta = 11.4, \alpha = 2.2, \gamma = 0$)	0.238
Wind direction	θ_{wind}	KDE	–
Significant wave height	H_s	Inverse Normal ($\mu = 2.3, \lambda = 6.8$)	0.533
Wave period	T_p	Weibull ($\beta = 9.3, \alpha = 3.3, \gamma = 2$)	0.00021
Wave direction	θ_{wave}	KDE	–

Table 3.1 Marginal inference results of the South Brittany metocean data.



(a) Wind speed U. (b) Wind direction θ_{wind}.

(c) Significant wave height H_s. (d) Wave period T_p. (e) Wave direction θ_{wave}.

Figure 3.5 Marginal inference results of the South Brittany metocean data.

3.4.2 Nonparametric inference of the dependence

The aim of this section is to complete the set of marginals fitted previously by the inference of a copula. A nonparametric estimation using the empirical Bernstein copula is studied in this section. To validate the goodness-of-fit of the EBC, the ANEMOC dataset (with size $N = 10^4$) is randomly split into two: a learning set $\mathbf{X}_n$ and a validation set $\mathbf{X}'_n$. A joint distribution $\widehat{F_{\mathbf{X}}}$ is then build using $\mathbf{X}_n$, by combining the marginals fitted earlier with an empirical copula $B_{\mathbf{m}}(C_n)$, such that:

$$\widehat{F_{\mathbf{X}}}(\mathbf{x}) = B_{\mathbf{m}}(C_n)\left(\widehat{F_{X_1}}(x_1), \ldots, \widehat{F_{X_d}}(x_d)\right). \tag{3.20}$$

Where $\widehat{F_{X_j}}$ stand for the model of the marginal j, just inferred in Section 3.4.1. While C_n is the empirical copula associated with the sample $\mathbf{X}_n$, and all the polynomial orders of the EBC are equal, $\mathbf{m} = \{m, \ldots, m\}, m \in \mathbb{N}$.

Then, one could compare a sample $\widehat{\mathbf{X}_n}$, generated from the fitted joint distribution $\widehat{F_{\mathbf{X}}}$, with the learning set $\mathbf{X}_n$. However, to prevent an overfit from the semiparametric model, the comparison is rather done between $\mathbf{X}_n$ and the independent validation set $\mathbf{X}'_n$. The statistic used is the maximum mean discrepancy, $\mathrm{MMD}(\widehat{\mathbf{X}_n}, \mathbf{X}'_n)$, initially introduced for multivariate two-sample testing (a specific presentation of the MMD and its estimation is developed in Appendix B). For a given fitted joint distribution, this procedure is repeated 100 times to take into account the sampling variability.

In Fig. 3.6, MMD distributions are represented for different values of the EBC polynomial order, with $m \in \{5, 10, 20, 50, 100, 1000\}$. The smaller the values of this dissimilarity measure, the closer the samples should be. Even if further developments could be implemented to improve the MMD estimation, these results are sufficient to set the EBC tuning at $m = 100$. Considering this setup, Fig. 3.7 represents the copulogram of a sample $\widehat{\mathbf{X}_n}$ (in red), side by side with the copulogram of the learning set $\mathbf{X}_n$ (in blue). This semiparametric approach offers a lot of flexibility, which is essential when inferring such complete dependence structures. As with any nonparametric method, it should be used with caution when inferring distributions' tails.

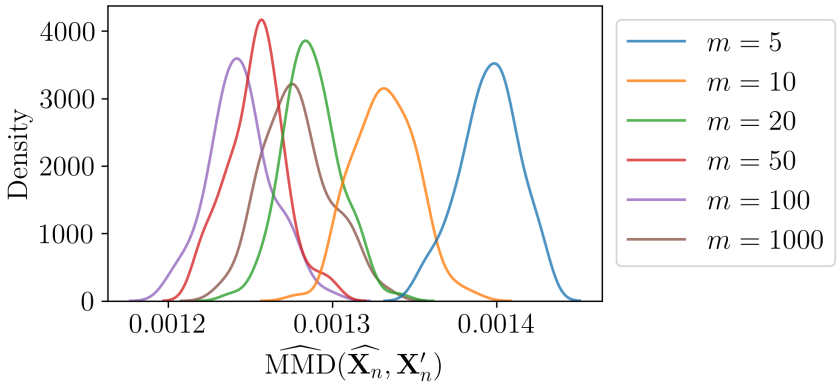
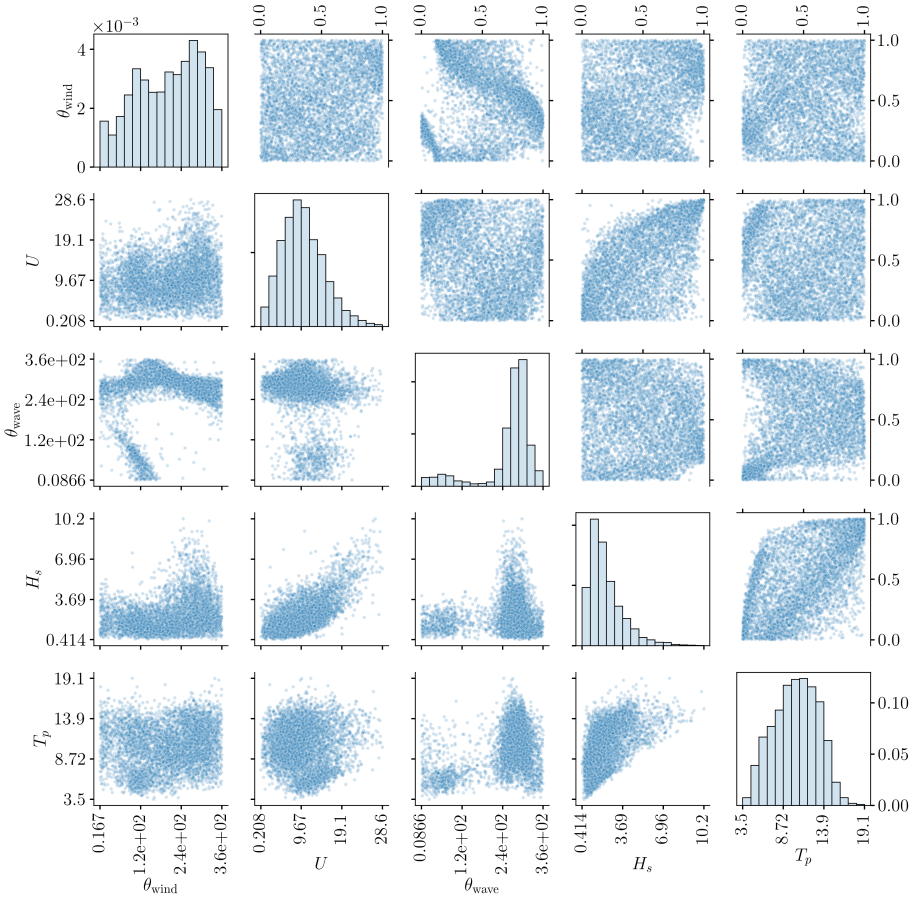
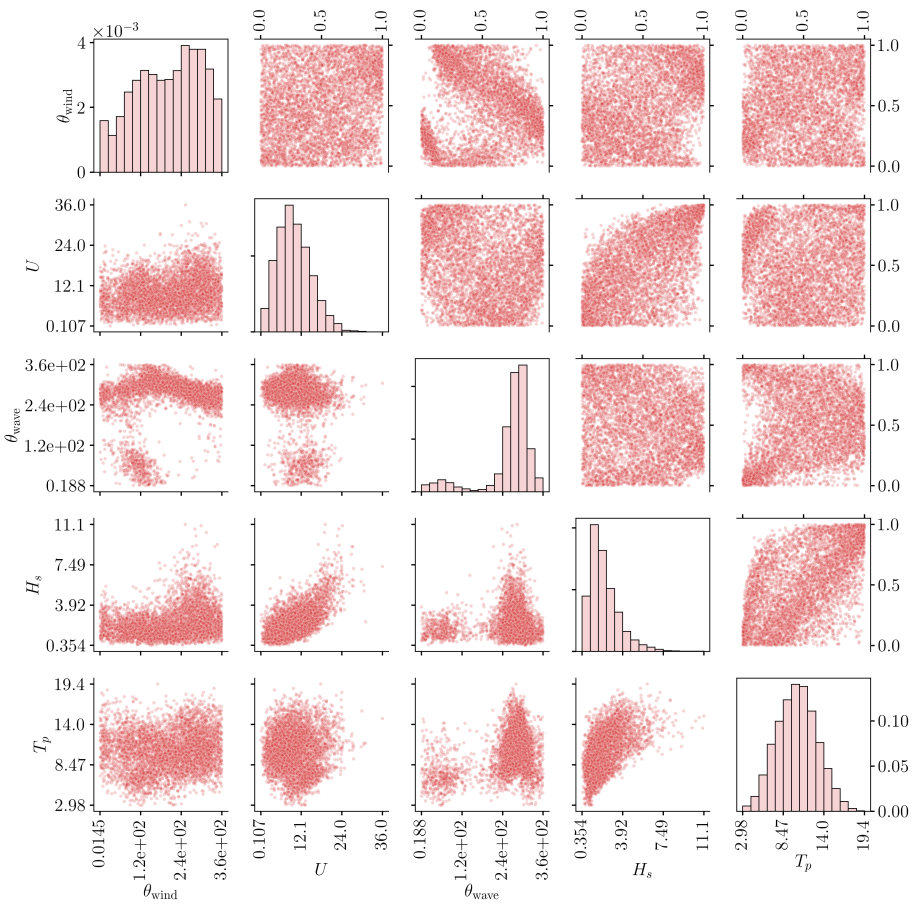


Figure 3.6 Empirical distributions of the maximum mean discrepancy between the validation sample $\mathbf{X}'$ and the sample $\widehat{\mathbf{X}_n} \overset{\text{i.i.d.}}{\sim} \widehat{F_{\mathbf{X}}}$ (repeated for 100 samples $\widehat{\mathbf{X}_n}$).



(a) South Brittany ANEMOC data (size $n = 5000$).



(b) Monte Carlo sample (size $n = 5000$), generated from the semiparametric model fitted on the ANEMOC data (EBC $B_{\mathbf{m}}(C_n)$ with $\{m_j = 100\}_{j=1}^{d}$).

Figure 3.7 Copulogram of the South Brittany metocean data.

3.4.3 Summary and discussion

In this section, a semiparametric inference strategy was illustrated on a metocean dataset from a site off the coast of South Brittany, France. This approach is pragmatic and offers a lot of flexibility. In our case, the unimodal marginals are fitted by MLE while the multimodal ones are fitted by KDE. Considering the complexity of the dependence, the EBC showed interesting results for large-size samples. Its capacity to extrapolate in the tails could be further studied (**?**) but this tool is an appropriate solution for general inference (for example needed for fatigue assessment).

In the monograph of Joe and Kurowicka (2011), the use of nonparametric methods is briefly discussed p.250. The author recommends using nonparametric copulas when the marginals are well-behaved, but the dependence structure is nonlinear.

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

After defining a probabilistic model based on ambient metocean data, the present section studies the impact of the wake on the wind conditions within a farm. The wake arises from extracting kinetic energy from the wind, leading to a decrease in wind speed and an increase in turbulence downstream of the turbines. In a wind farm, the wake mostly depends on the turbines' layout, the ambient wind speed, and the ambient turbulence intensity. The resulting heterogeneous wind field in a wind farm can be simulated by numerical models with different fidelities (as discussed in Section 2.2.2).

In our case, simplified wake models (sometimes called "dynamic wake meandering", or "engineering" models, see e.g., **?**) are used to simulate the wind speed deficit and the added turbulence at each turbine. To recover a wake-perturbed wind distribution at each turbine, the ambient wind distribution is propagated through a wake model. Having different wind distributions naturally impacts the loading and should be considered during fatigue assessment at a farm scale. Such heterogeneity is mostly considered by international standards via empirical coefficients (also called "effective turbulence"). **?** compares the effective turbulence approach with dynamic wake meandering models and studies their impact on loading.

In practice, fatigue assessment on a turbine represents a computational effort. As a consequence of wake modeling, each turbine presents a different wind distribution, which implies repeating a fatigue assessment for every turbine. To make this computation tractable at a farm scale, the present section aims at building clusters of turbines similarly affected by the wake. Then, fatigue assessment can be computed on a few turbines, each representing a cluster of turbines facing similar wake-modified wind loading. The maximum mean discrepancy (MMD) is used as a statistical dissimilarity measure to compare the perturbed distributions induced by the wake.

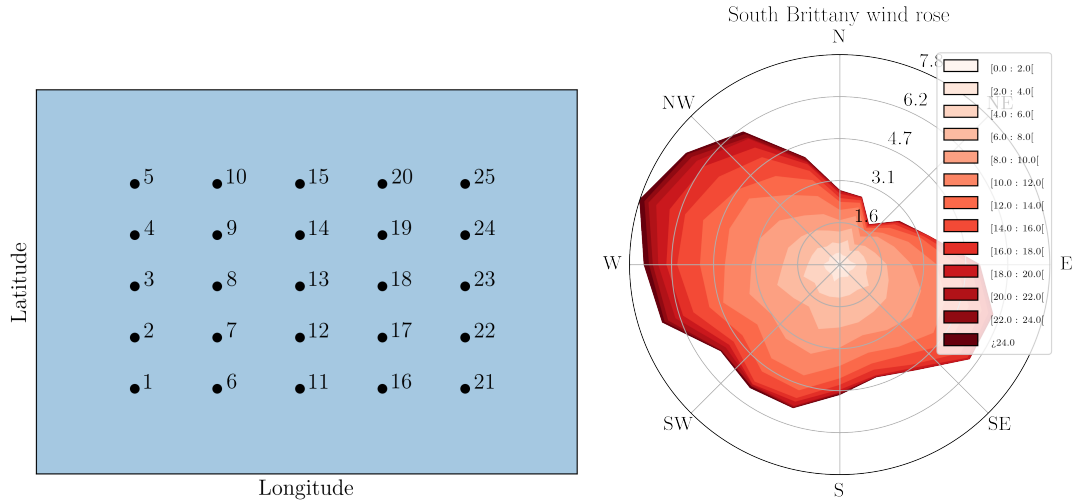


Figure 3.8 South Brittany wind farm layout, the vertical direction does not represent the exact north (left). South Brittany wind rose from the ANEMOC data (right).

This new approach is applied to a theoretical wind farm representing the recent call for tenders off the coast of South Brittany, France. The turbines considered are a modified version of the floating offshore wind turbine (FOWT) IEA-15MW (described in ?). Figure 3.8 illustrates the layout of the 25 FOWTs modeled in the following, the coordinates are normalized by the rotor's diameter D. The spacing between turbines in this regular layout is equivalent to seven rotor diameters in the dominant wind direction and five rotor diameters in the orthogonal direction (i.e., crosswind).

To develop this approach on the South Brittany farm, the present section is structured as follows: first, the wake model and its corresponding uncertainty propagation are presented, then MMDs are estimated between the resulting empirical joint wind distributions, and finally, a simple clustering gathers turbines perceiving similar wakes and defines their representative turbines. In the end, the 25 FOWT are split into four groups, whose representatives can be used for the fatigue assessment of the whole group.

3.5.1 Uncertainty propagation on a wake model

Wake model definition When simulating the wake effect of floating wind turbines, different studies (either using dynamic wake meandering ? or LES ?), showed the importance of modeling the floaters' position (translation and rotation). Therefore, an engineering wake model based on the Farmshadow™ software (developed by IFPEN) was coupled with a hydro-static calculation to predict the floater's position, as well as the wind speed and turbulence intensity. In this model, the floaters are considered to be rigid, and all degrees of freedom are considered (surge, sway and heave for the three translations and roll, pitch and yaw for the three rotations).

FarmShadow™ uses engineering wake models to simulate the wind field throughout the whole farm, starting from the most upstream WT and working downwards. More precisely, the model used includes the "super-gaussian" approach for speed deficit (?), waked-induced

turbulence according to **?**, and superimposition following the linear sum approach defined by **?**. Further assumptions regarding the hydro-statics loading, the effect of the mooring lines, and the wake model are defined in **?**.

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

$$g : \mathbb{R}^3 \longrightarrow \mathbb{R}^{3n_{WT}} \tag{3.21}$$

$$\mathbf{x} \longmapsto g(\mathbf{x}) = (\mathbf{x}'_1, .., \mathbf{x}'_{n_{WT}}) \tag{3.22}$$

The uncertainties associated with the ambient wind conditions are represented by a random vector $\mathbf{X}$ following the distribution f_0. A parametric model has been fitted in **?** using conditional probability density functions to capture the dependence structure, but the semi-parametric inference proposed in Section 3.4 could have been an alternative. Note that the missing turbulence intensity in the ANEMOC data from South Brittany was assumed to follow a lognormal distribution. The random vector $\mathbf{X}$ gathers the following random variables:

- Mean wind speed (U) is the 10-min average horizontal wind speed at hub height.
- Wind turbulence intensity (TI) is the 10-min wind speed turbulence intensity at hub height.
- Wind direction (θ_{wind}) is the 10-min average wind direction.

In the following, the wind orientation θ_{wind} is supposed to be unaffected by the wake. The uncertainty propagation through the wake models provides a set of perturbed environmental distributions $f'_l, l \in (1, .., n_{WT})$. In practice, a Monte Carlo sample $\mathbf{X}_n = \mathbf{x}^{(1)}, .., \mathbf{x}^{(n)} \sim \mathbf{X}$ (with size n=6000) is generated and evaluated by the wake model. Since the model has a low computational cost, Monte Carlo sampling was affordable while giving strong convergence guarantees.

Fig. 3.9 illustrates the perturbation of the wind distributions for three WT differently affected by the wake depending on their position in the farm (see Figure 3.8). One can notice that the distribution of WT 25 (in orange) is very close to the ambient distribution (in black), as expected since this WT is located on the edge of the farm and facing the dominant wind direction. Meanwhile, the distribution of WT 13 (in red) seems more affected by the wake, by getting higher wind turbulence with lower wind speed. This analysis can be completed with the two marginals in Fig. 3.10a and Fig. 3.10b, both describing the ambient marginal distributions (in black) and wake-disturbed distributions. In general, a small wind speed deficit is indicated by the small shifts of the probability density functions to the left on Fig. 3.10a. Also, a small added turbulence is noticeable with shifts of the PDFs to the right on Fig. 3.10b. A tool is needed to quantify the wind perturbations induced by the wake.

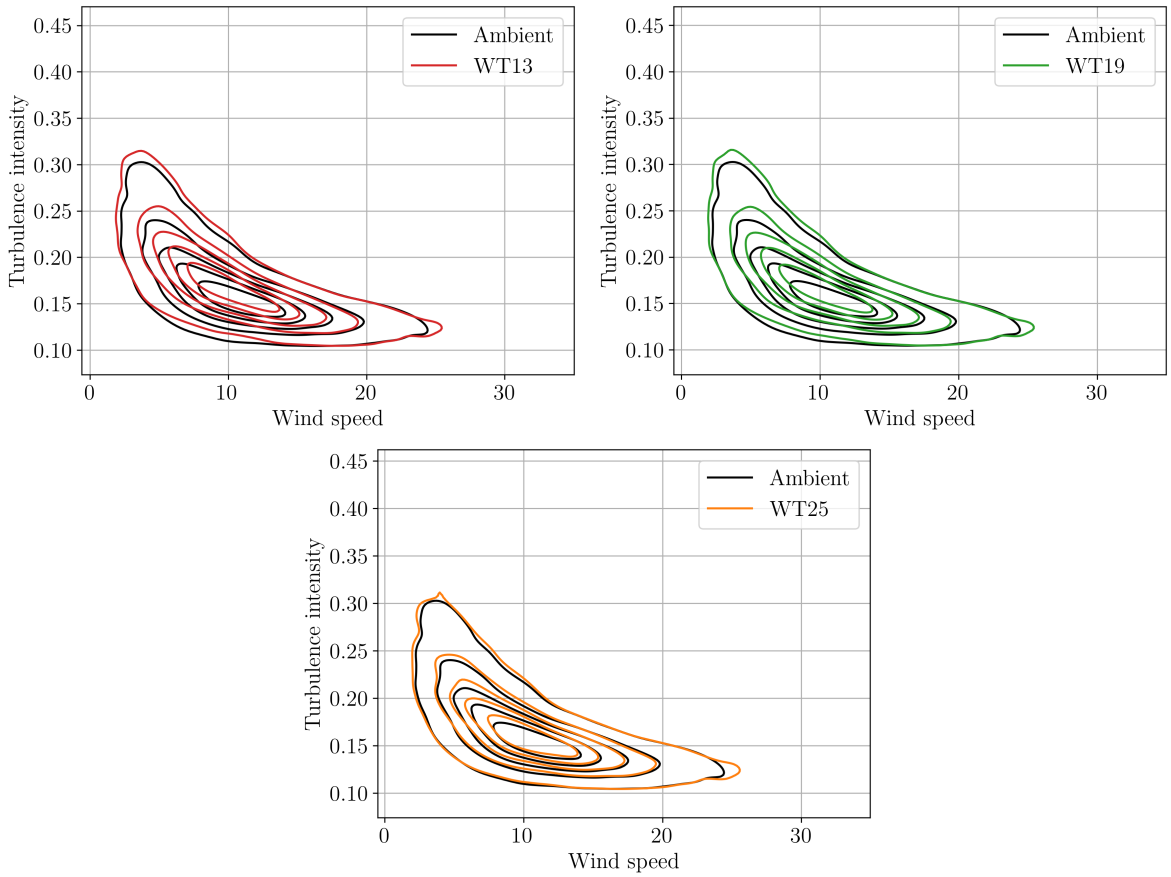
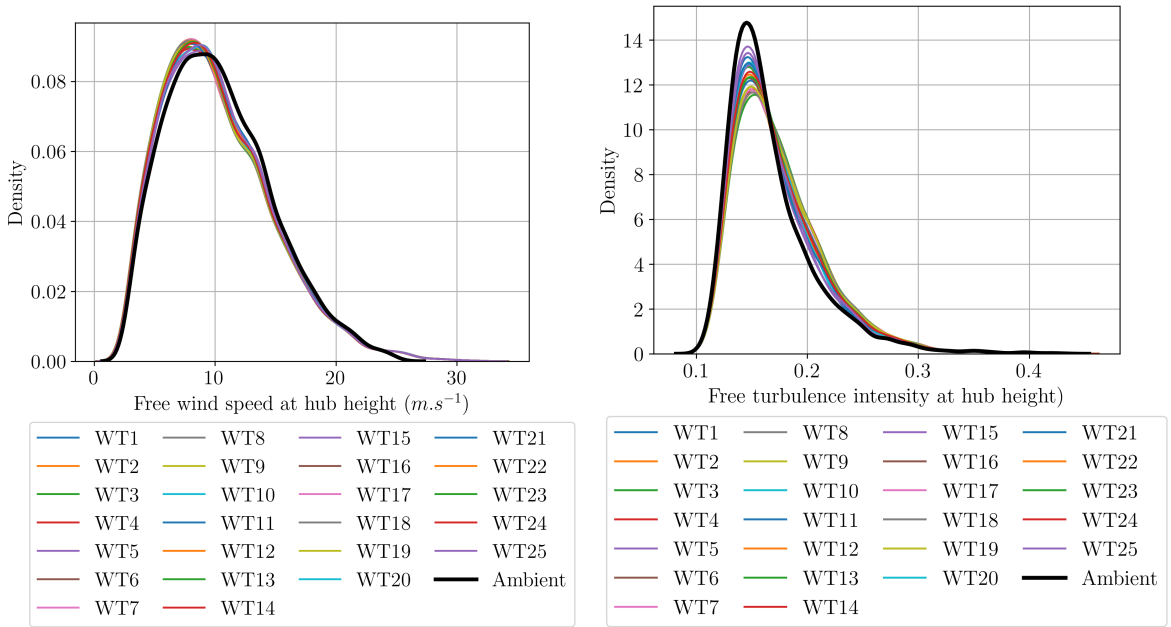


Figure 3.9 Joint distributions of the wake-perturbed wind conditions at WT 13, 19, and 25 (in color) compared with the ambient wind conditions (in black).



(a) Wind speed.

(b) Turbulence intensity.

Figure 3.10 Ambient (in black) and wake-perturbed (in color) distributions of wind distributions.

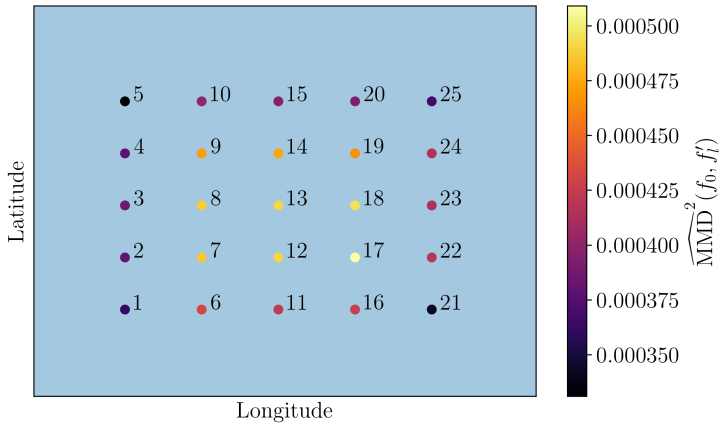


Figure 3.11 South Brittany layout and wake effects measured by the squared MMD on wind conditions. Note that the vertical direction does not represent the north direction.

3.5.2 Statistical metric of wake-induced perturbations

The maximum mean discrepancy was introduced by **?** as a statistic for two-sample testing. After further work on this tool, authors such as **?** showed that the MMD is a distance between two distributions embedded in a specific function space. The MMD becomes a metric for specific kernels called "characteristic kernels", which offer the following property: $\mathrm{MMD}(\pi, \zeta) = 0 \iff \pi = \zeta$. The squared MMD has been used for multiple purposes and is further presented in Appendix B In the following, the idea is to compare the ambient wind distribution f_0 to the wake-perturbed wind conditions f_l' at the WT l using the squared MMD.

Application to the South Brittany wind farm project Once the joint perturbed distributions of each WT are evaluated on a large Monte Carlo sample, their squared MMD with the ambient wind conditions can be computed. Fig. 3.11 represents the squared MMD for each WT to quantify the wake-induced perturbation. The values of squared MMD presented in this figure are estimated between two Monte Carlo samples with size $n = 6000$. A verification of their convergence is realized in terms of coefficient of variation.

3.5.3 Clustering the wake-induced perturbations

The aim of this section is to use the MMD as a metric to define clusters of turbines getting similar wind conditions. Instead of comparing the wake-perturbed distributions with the ambient one, let us compare all the pairs of wake-perturbed distributions. Considering all the WT $\{1, \ldots, n_{WT}\}$ in the farm, and their respective wake-perturbed distributions $\{f_l'\}_{l=1}^{n_{WT}}$, let us define the symmetric matrix D of MMD between every pair of perturbed distributions such that $D_{i,j} = \widehat{\mathrm{MMD}}^2(f_i', f_j')$.

Different unsupervised clustering techniques, such as hierarchical or centroid-based methods were compared in **?**. The "k-medoids" (**?**) are a variation of the well-known k-means that selects

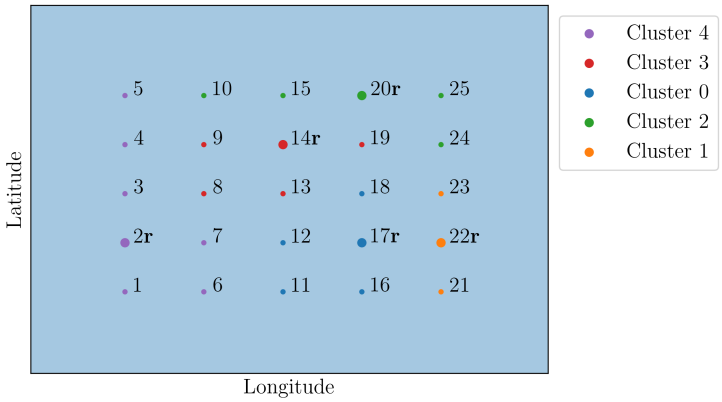


Figure 3.12 K-medoids clustering solution for five clusters. The representative elements of the clusters are tagged with the mention “r”.

actual data points as centers (i.e., medoids). In our case, this method not only gathers turbines with the same wind conditions but also determines a representative turbine for each cluster.

Assuming a final number of clusters equal to five, Fig. 3.12 represents the clusters defined by the k-medoid method applied to the matrix D. Let us notice that the results are rather coherent with the main wind orientation illustrated in the wind rose (see Fig. 3.8). Interestingly, the conclusion that emerged from comparing the wake-perturbed distributions to the ambient one (see Fig. 3.11) is different from the ones obtained by comparing pairs of wake-perturbed distributions. Finally, the representative turbines are tagged with the mention “r” and could be used to perform a fatigue analysis.

3.5.4 Summary and discussion

This section studied the impact of the wake on the wind conditions in a farm. The ambient wind conditions were propagated on a low-fidelity wake model of a floating wind farm. Even if higher-fidelity models could simulate this phenomenon more accurately, their huge computational cost would not allow uncertainty propagation (e.g., several days for LES certain solver, with intensive parallelization). The present model has a reasonable error and has a short execution (i.e., about a few minutes on a regular computer). The resulting wake-perturbed distributions at each turbine were compared to each other using a kernel-based dissimilarity measure for distributions. Using this scalar metric of perturbations created by the wake, clusters were built. In the context of fatigue assessment at the farm scale, the computed clusters may reduce the computational effort by assessing one turbine per cluster.

3.6 Conclusion

In this chapter, different aspects of uncertainty quantification were applied to define the metocean conditions in a wind farm. First, to infer a joint environmental distribution, a semiparametric approach was applied to a large dataset from a site in South Brittany, France. This

semiparametric approach combines parametric models of the marginals with nonparametric models of the copula such as the highly flexible empirical Bernstein copula. Second, to take into account the heterogeneous wind conditions inside a farm, a dynamic wake meandering model was developed for the South Brittany farm. Then, the ambient joint environmental distribution was propagated through this model to recover a wake-perturbed environmental distribution per turbine.

The copulogram is a new data visualization tool based on the empirical copula, aiming at improving the description of nonlinear dependencies in datasets.

The maximum mean discrepancy is a kernel-based dissimilarity measure between multivariate distributions which was used throughout this chapter. Either for testing the goodness-of-fit of the semiparametric model fitted using the EBC, or as a measure of the perturbation occasioned by the wake on the wind conditions. In this context, the MMD allowed building clusters of turbines witnessing the same wind conditions. As perspectives, the robustness of this metric to the choice of kernel could be further investigated as well as the goodness-of-fit of the EBC over distributions' tails. After defining the probabilistic model of the environmental conditions, this uncertainty is ready to be propagated in a multiphysics wind turbine model as DIEGO.

Chapter 4

Kernel-based central tendency estimation

This chapter is adapted from the following reference:

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.

4.1 Introduction

As a sustainable and renewable energy source, offshore wind turbines (OWT) are likely to take a growing share of the global electric mix. However, to be more cost-effective, wind farm projects tend to move further from the coast, exploiting stronger and steadier wind resources. Going further offshore, wind turbines are subject to more severe and uncertain environmental conditions (i.e., wind and waves). In such conditions, their structural integrity should be certified. To do so, numerical simulation and probabilistic tools have to be used. In fact, according to **?**, for new environmental conditions or new turbine models, international standards such as **?** from the International Electrotechnical Commission and **?** from Det Norske Veritas recommend performing over 2×10^5 simulations distributed over a grid. However, numerical simulations are computed by a costly hydro-servo-aero-elastic wind turbine model, making the design process time-consuming. In the following, the simulated output cyclic loads studied are aggregated over the simulation period to assess the mechanical fatigue damage at hot spots of the structure. To compute the risks associated with wind turbines throughout their lifespan, one can follow the steps of the universal framework for the treatment of uncertainties presented in the introduction of this manuscript Fig. 1. After specifying the problem (Step A), one can quantify the uncertainties related to site-specific environmental conditions represented by the random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^d, d \in \mathbb{N}^*$ (Step B). Then, one can propagate them through the OWT simulation model (Step C) denoted by $g : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}, \mathbf{X} \mapsto Y = g(\mathbf{X})$, and estimate a relevant quantity of interest $\psi(Y) = \psi(g(\mathbf{X}))$ (e.g., a mean, a quantile, a failure probability). An accurate estimation of the quantity of interest $\psi(Y)$ relies on both a relevant quantification of the input uncertainty and an efficient sampling method.

Regarding Step B, when dealing with uncertain environmental conditions, a specific difficulty often arises from the complex dependence structure such variables may exhibit. Here, two cases may occur: either measured data are directly available (i.e., the "given-data" context) or a theoretical parametric form for the joint input probability distribution can be postulated. Such existing parametric joint distributions often rely on prior data fitting combined with expert knowledge. For example, several parametric approaches have been proposed in the literature to derive such formulations, ranging from fitting conditional distributions **?**) to using vine copulas (**?**). When a considerable amount of environmental data is available, nonparametric approaches such as the empirical Bernstein copula were studied in Chapter 3 to capture complex dependence structures. Alternatively, an idea is to directly use the data as an empirical representation of input uncertainties in order to avoid an additional inference error.

Step C usually focuses on propagating the input uncertainties in order to estimate the quantity of interest. Depending on the nature of $\psi(Y)$, one often distinguishes between two types of uncertainty propagation: a central tendency estimation (e.g., focusing on the output mean value or the variance) and a tail estimation (e.g., focusing on a high-order quantile or a failure probability). When uncertainty propagation aims at central tendency estimation, the usual methods can be split into two groups. First, those relying on sampling, i.e., mainly Monte Carlo sampling (**?**), quasi-Monte Carlo sampling (**?**), geometrical subsampling (**?**), or deterministic quadrature rules (**?**). All these methods estimate the quantity directly on the numerical simulator's outputs. Second, those that rely on the use of surrogate models (or metamodels, see Fig. 1) to emulate the costly numerical model by a statistical model. Among a large panel of surrogates, one can mention, regarding wind energy applications, the use of polynomial chaos expansions (**??**), Gaussian process regression (**????**), or artificial neural networks (**?**). When uncertainty propagation aims at studying the tail of the output distribution such as in risk or reliability assessment, one usually desires to estimate a quantile or a failure probability. In the wind energy literature, failure probability estimation has been largely studied, e.g., in time-independent reliability assessment (**???**) or regarding time-dependent problems (**?**Lataniotis, 2019).

During the overall process described in Fig. 1, modelers and analysts often need to determine whether inputs are influential or not in order to prioritize their effort (in terms of experimental data collecting, simulation budget, or expert elicitation). Sometimes, they want to get a better understanding of the OWT numerical models' behavior or to enhance the input uncertainty modeling. All these questions are intimately related to the topic of sensitivity analysis (**??**) and can be seen as an "inverse analysis" denoted by Step C' in Fig. 1. In the wind energy literature, one can mention, among others, some works related to Spearman's rank correlation analysis and the use of the Morris method in **??**. Going to variance-based analysis, the direct calculation of Sobol' indices after fitting a polynomial chaos surrogate model has been proposed in many works (e.g., in **?**) while the use of distributional indices (e.g., based on the Kullback–Leibler divergence) has been investigated by **?**.

The present chapter focuses on the problem of uncertainty propagation, and more specifically, on the mean fatigue damage estimation (i.e., $\psi(Y) = \mathbb{E}[g(\mathbf{X})]$). Such a problem is usually encountered, by engineers, during the design phase. Most of the time, current standards as well as common engineering practices make them use regular grids (**?**). Altogether, one can describe three alternative strategies: (i) direct sampling on the numerical model (e.g., using Monte Carlo), (ii) sampling on a static surrogate model (e.g., using Gaussian process regression), or (iii) using an "active learning" strategy (i.e., progressively adding evaluations of the numerical model to enhance the surrogate model fitting process). In practice, fitting a surrogate model in the context of OWT fatigue damage can be challenging due to the nonlinearity of the code. Moreover, the surrogate model validation procedure complexifies the process. Finally, active learning strategies restrict the potential number of parallel simulations, which limits the use of HPC facilities. Thus, the main contribution of this chapter is to explore different ways to

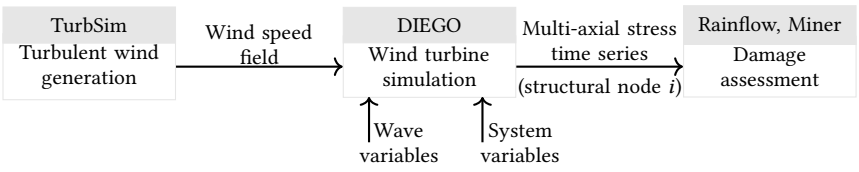


Figure 4.1 Diagram of the chained OWT simulation model.

propagate uncertainties by directly evaluating the numerical model (i.e., without any surrogate model) with a relevant tradeoff between computational cost and accuracy. In the specific context of wind turbine fatigue damage, this work shows how to propagate uncertainties arising from a complex input distribution through a costly wind turbine simulator. The proposed work consists of evaluating the advantages and limits of kernel herding as a tool for given-data, fast, and fully-distributable uncertainty propagation in OWT simulators. Additionally, this sampling method is highly flexible, allowing one to complete an existing design of experiments. Such a property can be crucial in practice when the analyst is asked to include some specific points to the design (e.g., characteristic points describing the system's behavior required by experts or by standards, see **?**).

The present chapter is organized as follows. Section 4.2 will present the industrial use case related to a wind farm operating in Teesside, UK. Then, Section 4.3 will introduce various kernel-based methods for central tendency estimation. Section 4.4 will analyze the results of numerical experiments obtained on both analytical and industrial cases. Finally, the last section will present some discussions and draw some conclusions.

4.2 Treatment of uncertainties on the Teesside wind farm

An OWT is a complex system interacting with its environment. To simulate the response of this system against a set of environmental solicitations, multi-physics numerical models are developed. In the present chapter, the considered use case consists of a chain of three numerical codes executed sequentially. As illustrated in Fig. 4.1, a simulation over a time period is the sequence of, first, a turbulent wind speed field generation, then a wind turbine simulation (computing various outputs including mechanical stress), and finally, a post-processing phase to assess the fatigue damage of the structure.

4.2.1 Numerical simulation model

This subsection generally describes the modeling hypotheses considered in the industrial use case, further details regarding wind turbines modeling a provided in Chapter 2 of this manuscript. The first block of the chain consists of a turbulent wind field simulator called "TurbSim" (developed by **?** from the National Renewable Energy Laboratory, USA) that uses, as a turbulence model, a Kaimal spectrum (**?**) (as recommended by the **?**). Moreover, to extrapolate the wind speed vertically, the shear is modeled by a power law. Since the wind field generation

Table 4.1 Teesside Offshore Wind turbine datasheet

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

shows inherent stochasticity, each 10-minute long simulation is repeated with different pseudo-random seeds and one averages the estimated damage over these repetitions. This question has been widely studied by some authors, (e.g., **?**), who concluded that the six repetitions recommended by the **?** may be insufficient to properly average this stochasticity. Thus, in the following, the simulations are repeated eleven times (picking an odd number also directly provides the median value over the repetitions). This number of repetitions was chosen to suit the maximum number of simulations and the storage capacity of the generated simulations.

As a second block, one finds the "DIEGO" software (for "Dynamique Intégrée des Éoliennes et Génératrices Offshore"[1]) which is developed by EDF R&D (**?**) to simulate the aero-hydro-servo-elastic behavior of OWTs. It takes the turbulent wind speed field generated by TurbSim as input and computes the dynamical behavior of the system (including the multiaxial mechanical stress at different nodes of the structure). For the application of interest here, the control system is modeled by the open-source DTU controller (**?**), and no misalignment between the wind and the OWT is assumed. As for the waves, they are modeled in DIEGO using a JONSWAP spectrum (named after the 1975 Joint North Sea Wave Project). The considered use case here consists of a DIEGO model of a Siemens SWT 2.3MW bottom-fixed turbine on a monopile foundation (see the datasheet in Table 4.1), currently operating in Teesside, UK (see the wind farm layout and wind turbine diagram in Fig. 4.2). Although wind farms are subject to the wake effect, affecting the behavior and performance of some turbines in the farm, this phenomenon is not considered in this chapter. To avoid numerical perturbations and reach the stability of the dynamical system, our simulation period is extended to 1000 seconds and the first 400 seconds are cropped in the post-processing step. This chained OWT numerical simulation model has been deployed on an EDF R&D HPC facility to benefit from parallel computing speed up (a single simulation on one CPU takes around 20 minutes).

4.2.2 Measured environmental data

During the lifespan of a wind farm project, environmental data is collected at different phases. In order to decide on the construction of a wind farm, meteorological masts, and wave buoys are usually installed on a potential site for a few years. After its construction, each wind turbine is equipped with monitoring instruments (e.g., cup anemometers). In total, five years of wind data have been collected on the turbines which are not affected by the wake on this site. Their acquisition system (usually called "SCADA" for "Supervisory Control And Data Acquisition")

[1] In English, "Integrated Dynamics of Wind Turbines and Offshore Generators".

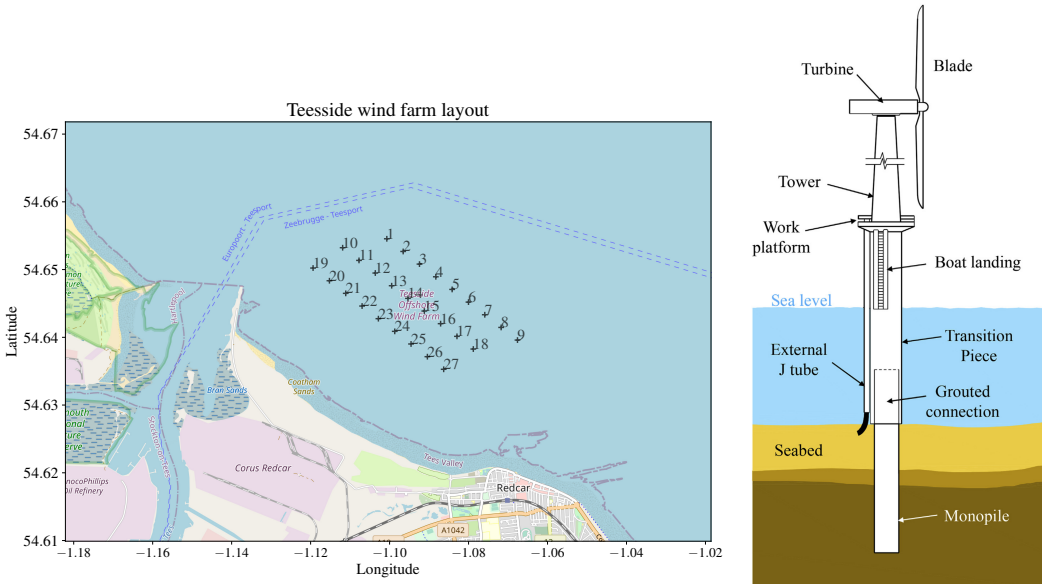


Figure 4.2 Teesside wind farm layout (left). Monopile OWT diagram (?) (right)

Variable	Notation	Unit	Description
Mean wind speed	U	$\mathrm{m.s}^{-1}$	10-min. average horizontal wind speed
Wind turbulence	σ_U	$\mathrm{m.s}^{-1}$	10-min. wind speed standard deviation
Wind direction[2]	θ_{wind}	deg.	10-min. average wind direction
Significant wave height	H_s	m	Significant wave height
Peak wave period	T_p	s	Peak 1-hour spectral wave period
Wave direction	θ_{wave}	deg.	10-min. average wave direction

Table 4.2 Description of the environmental data.

has a sampling period of ten minutes. The wave data arise from a buoy placed in the middle of the farm. These data describe the physical features listed in Table 4.2. A limitation of the present study is that its controller-induced uncertainty (like wind misalignment) is not considered.

The Teesside farm is located close to the coast, making the environmental conditions very different depending on the direction (see the wind farm layout in Fig. 4.2). Since measures are also subject to uncertainties, a few checks were made to ensure that the data were physically consistent. Truncation bounds were applied since this study is focused on central tendency estimation (i.e., mean behavior) rather than extreme values. In practice, this truncation only removes extreme data points (associated with storm events). In addition, a simple trigonometric transform is applied to each directional feature to take into account their cyclic structure. Finally, the remaining features are rescaled (i.e., using a min-max normalization).

Teesside's environmental data is illustrated by its copulogram in Fig. 4.3, a graphical tool presented in Section 3.3 to visualize multivariate data. The copulogram exhibits the marginals

[2]Note that the two directional variables could be replaced by a wind-wave misalignment variable for a bottom-fixed technology, however, our framework can be directly transposed to floating models.

with univariate kernel density estimation plots (in the diagonal), and the dependence structure with scatter plots in the normalized rank space (in the upper triangle). Looking at data in the rank space instead of the initial space allows one to observe the ordinal associations between variables. The scatter plots of normalized ranks are actually a representation of the empirical copula density. Two independent variables will present a uniformly distributed scatter plot in the rank space. In the lower triangular matrix, the scatter plots are set in the physical space, merging the effects of the marginals and the dependencies (as in the usual visualization offered by the matrix plot). Since the dependence structure is theoretically modeled by an underlying copula, this plot is called *copulogram*, generalizing the well-known "correlogram" to nonlinear dependencies. It gives a synthetic and empirical decomposition of the dataset that was implemented in a new open-source Python package named `copulogram`[3].

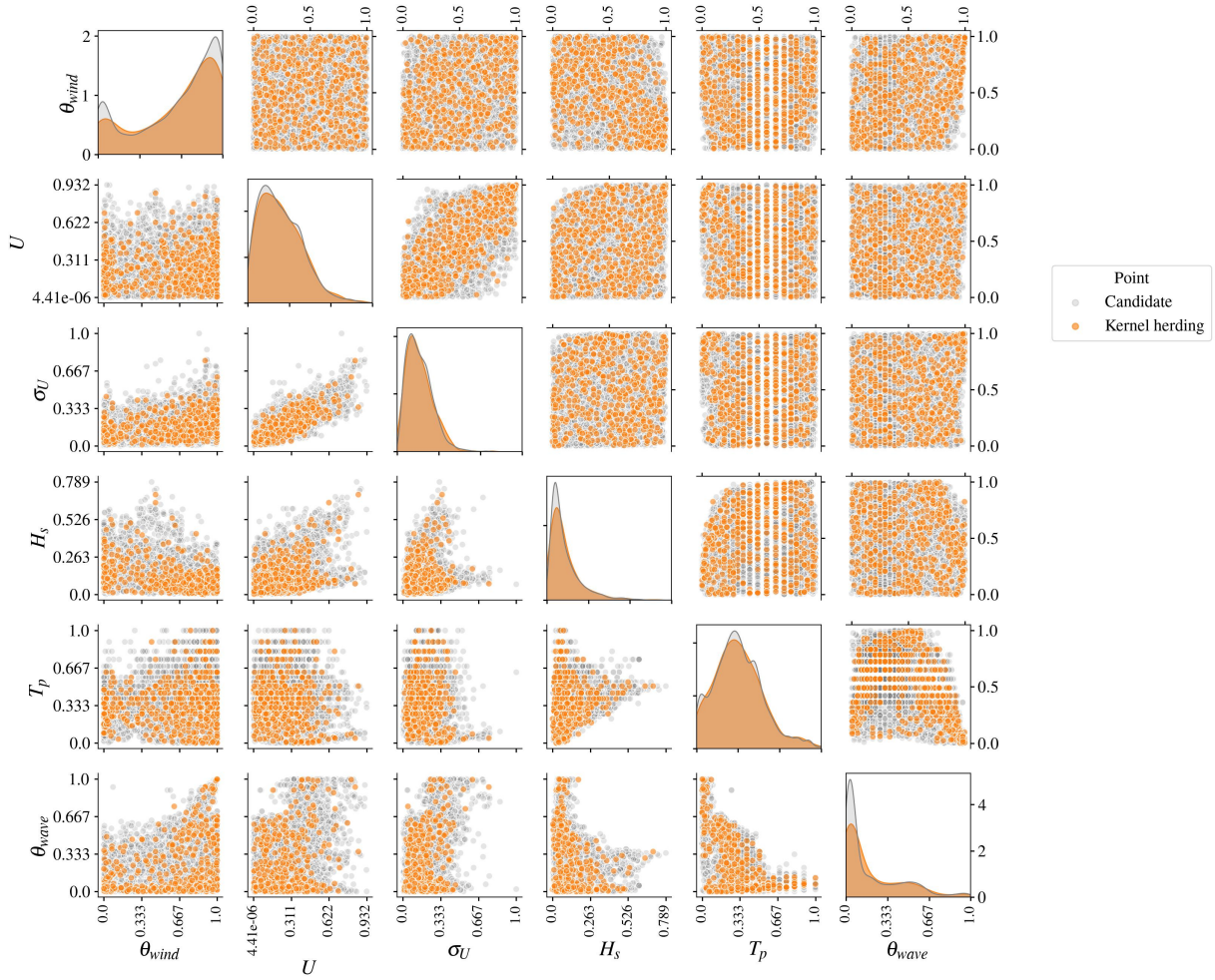


Figure 4.3 Copulogram of the Teesside measured data ($N = 10^4$ in grey), kernel herding subsample ($n = 500$ in orange). Marginals are represented by univariate kernel density estimation plots (diagonal), the dependence structure with scatter plots in the rank space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.

[3]GitHub repository: https://github.com/efekhari27/copulogram

On Fig. 4.3, a large sample $\mathcal{S} \subset \mathcal{D}_{\mathbf{X}}$ (with size $N = 10^4$) is randomly drawn from the entire Teesside data (with size $N_{\text{Teesside}} = 2 \times 10^5$), and plotted in grey. In the same figure, the orange matrix plot is a subsample of the sample $\mathcal{S}$, selected by kernel herding, a method minimizing some discrepancy measure with the sample $\mathcal{S}$ that will be presented in Section 4.3). For this example, generating the kernel herding subsample takes under one minute, which is negligible compared with the simulation time of OWT models. Visually, this orange subsample seems to be representative of the original sample both in terms of marginal distributions and dependence structure. In the following study, the large samples $\mathcal{S}$ will be considered as an empirical representation of the distribution of the random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}}$, with probability density function $f_{\mathbf{X}}$, and called *candidate set*. Kernel herding allows direct subsampling from this large and representative dataset, instead of fitting a joint distribution and generating samples from it. Indeed, fitting a joint distribution would introduce an additional source of error in the uncertainty propagation process. Note that a proper parametric model fit would be challenging for complex dependence structures such as the one plotted on Fig. 4.3. As examples of works that followed this path, one can mention the work of **?** who built a parametric model of a similar multivariate distribution using vine copulas.

For a similar purpose and to avoid some limits imposed by the parametric framework, a nonparametric approach coupling empirical Bernstein copula fitting with kernel density estimation of the marginals is proposed in subSection 4.2.3.

4.2.3 Non parametric fit with empirical Bernstein copula

Instead of directly subsampling from a dataset such as the one from Fig. 4.3, one could first infer a multivariate distribution and generate a sample from it. However, accurately fitting such complex multivariate distributions is challenging. The amount of available data is large enough to make nonparametric inference a viable option.

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

1. A set of univariate marginal distributions to describe the behavior of the individual variables;

2. A function describing the dependence structure between all variables, called a *copula*.

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

$$F(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right). \tag{4.1}$$

It allows us to divide the problem of fitting a joint distribution into two independent problems: fitting the marginals and fitting the copula. The empirical Bernstein copula is a nonparametric copula approximation method introduced and applied to similar data in Chapter 3. Provided a large enough learning set $\mathbf{X}_n$ (over five years in the present case), the EBC combined with

kernel density estimation for the marginals fits well the environmental joint distribution related to the dataset in Fig. 4.3. Moreover, the densities of the EBC are available in an explicit form, making Monte Carlo or quasi-Monte Carlo generation easy. As discussed in Chapter 3, this method is sensitive to the chosen polynomial orders $\{m_j\}_{j=1}^{d}$ and the learning set size.

4.2.4 Fatigue assessment

As described in Fig. 4.1, a typical DIEGO simulation returns a 10-minute multiaxial stress time series at each node $i \in \mathbb{N}$ of the 1D meshed structure. Since classical fatigue laws are established for uniaxial stresses, the first step is to compute one equivalent Von Mises stress time series at each structural node. The present section recalls the main concepts but fatigue assessment is further discussed in Section 2.3.5.

The foundation and the tower of an OWT are a succession of tubes with various sections connected by bolted or welded joints. Our work focuses on the welded joints at the mudline level, identified as a critical area for the structure. This hypothesis is confirmed in the literature by different contributions, see e.g., the results of **?** in Figure 12, or **?**. At the top of the turbine, the fatigue is commonly studied at the blade root, which was not studied here since the blades in composite material have different properties (see e.g., **?**). Note that the OWT simulations provide outputs allowing us to similarly study any node along the structure (without any additional computational effort).

To compute fatigue in a welded joint, the external circle of the welding ring is discretized for a few azimuth angles $\theta \in \mathbb{R}_+$ (see the red points in the monopile cross-section on the right in Fig. 4.4). The equivalent Von Mises stress time series is then reported on the external welding ring for an azimuth θ. According to **?** and our own experience, the most critical azimuth angles are roughly aligned with the main wind and wave directions (whose distributions are illustrated in Fig. 4.4). Looking at these illustrations, the wind and wave conditions have a very dominant orientation, which is explained by the closeness of the wind farm to the shore. Then, it is assumed that azimuth angles in these directions will be more solicited, leading to higher fatigue damage. To assess fatigue damage, rainflow counting (**?**) first identifies the stress cycles and their respective amplitudes (using the implementation of the ASTM E1049-85 rainflow cycle counting algorithm from the Python package `rainflow`[4]). For each identified stress cycle of amplitude, $s \in \mathbb{R}_+$, the so-called "Stress vs. Number of cycles" curve (also called the "SN curve" or "Wöhler curve") allows one to estimate the number N_c of similar stress cycles necessary to reach fatigue ruin. The SN curve, denoted by $W(\cdot)$ is an affine function in the log-log scale with slope $-m$ and y-intercept a:

$$N_c(s) = as^{-m},\ a \in \mathbb{R}_+,\ m \in \mathbb{R}_+. \tag{4.2}$$

Finally, a usual approach to compute the damage is to aggregate the fatigue contribution of each stress cycle identified using Miner's rule. Damage occurring during a 10-minute operating

[4]https://github.com/iamlikeme/rainflow

time is simulated and then scaled up to the OWT lifetime. More details regarding damage assessment and the Wöhler curve used are available in Section 2.4.6 from (**?**). For a realization $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$ of environmental conditions, at a structural node i, an azimuth angle θ; $k \in \mathbb{N}$ stress cycles of respective amplitude $\{s_{i,\theta}^{(j)}(\mathbf{x})\}_{j=1}^{k}$ are identified. Then, Miner's rule (**?**) defines the damage function $g_{i,\theta}(\mathbf{x}) : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}_+$ by:

$$g_{i,\theta}(\mathbf{x}) = \sum_{j=1}^{k} \frac{1}{N_{\mathrm{c}}\left(s_{i,\theta}^{(j)}(\mathbf{x})\right)}. \tag{4.3}$$

As defined by the DNV standards for OWT fatigue design (**?**), the quantity of interest in the present chapter is the "mean damage" $d_{\mathrm{c}}^{i,\theta}$ (also called "cumulative damage"), computed at a node i, for an azimuth angle θ:

$$d_{\mathrm{c}}^{i,\theta} = \mathbb{E}[g_{i,\theta}(\mathbf{X})] = \int_{\mathcal{D}_{\mathbf{X}}} g_{i,\theta}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{4.4}$$

To get a preview of the distribution of this output random variable $g_{i,\theta}(\mathbf{X})$, a histogram of a large Monte Carlo simulation ($N_{\mathrm{ref}} = 2000$) is represented in Fig. 4.5 (with a log scale). In this case, the log-damage histogram presents a little asymmetry but it is frequently modeled by a normal distribution (see e.g., **?**).

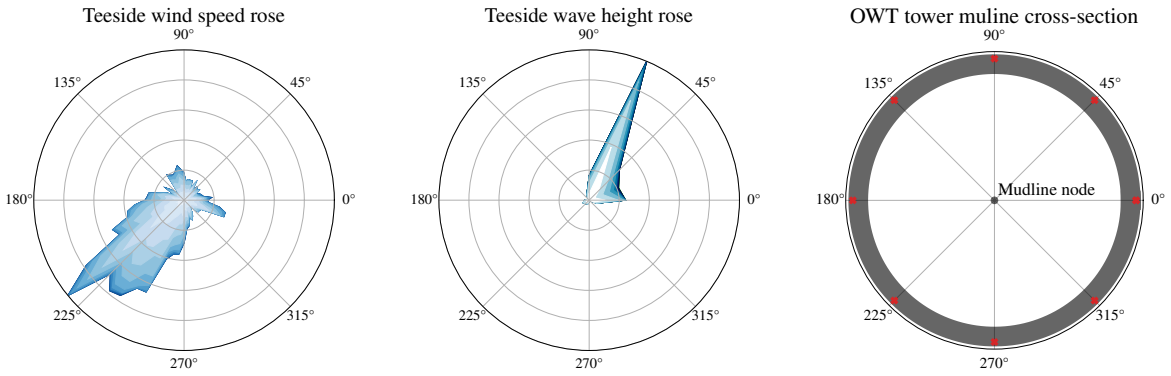


Figure 4.4 Angular distribution of the wind and waves with a horizontal cross-section of the OWT structure and the mudline. Red crosses represent the discretized azimuths for which the fatigue is computed

4.3 Numerical integration procedures for mean damage estimation

The present section explores different methods aiming at approximating the integral of a function against a probability measure. In the case of OWT mean damage estimation, these methods can be used for defining efficient design load cases. This problem is equivalent to

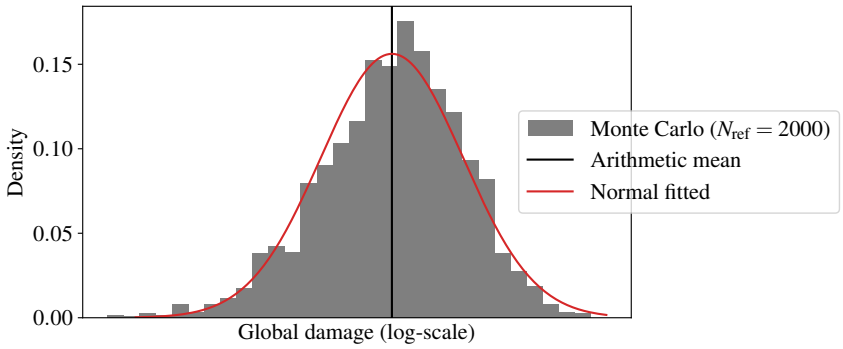


Figure 4.5 Histogram of the log-damage, at mudline, azimuth 45 deg. (Monte Carlo reference sample)

the central tendency estimation of $\mathbf{Y} = g(\mathbf{X})$, the image of the environmental random variable $\mathbf{X}$ by the damage function $g(\cdot) : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}$ (see e.g., Eq. (4.4)). Considering a measurable space $\mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^d, d \in \mathbb{N}^*$, associated with a known probability measure π, this section studies the approximation of integrals of the form $\int_{\mathcal{D}_{\mathbf{X}}} g(\mathbf{x}) \mathrm{d}\pi(\mathbf{x})$.

4.3.1 Quadrature rules and quasi-Monte Carlo methods

Numerical integration authors may call this generic problem *probabilistic integration* (Briol et al., 2019). In practice, this quantity of interest is estimated on an n-sized set of damage realizations $\mathbf{y}_n = \left\{g(\mathbf{x}^{(1)}), \ldots, g(\mathbf{x}^{(n)})\right\}$ of an input sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\}$. A weighted arithmetic mean of the realizations $\left\{g(\mathbf{x}^{(1)}), \ldots, g(\mathbf{x}^{(n)})\right\}$ is called a *quadrature rule* with a set of unconstrained weights $\mathbf{w}_n = \{w_1, \ldots, w_n\} \in \mathbb{R}^n$:

$$I_\pi(g) := \int_{\mathcal{D}_{\mathbf{X}}} g(\mathbf{x}) \mathrm{d}\pi(\mathbf{x}) \approx \sum_{i=1}^{n} w_i g(\mathbf{x}^{(i)}). \tag{4.5}$$

Our numerical experiment framework often implies that the function g is costly to evaluate, making the realization number limited. For a given sample size n, our goal is to find a set of tuples $\left\{\mathbf{x}^{(i)}, w_i\right\}_{i=1}^{n}$ (i.e., quadrature rule), giving the best approximation of our quantity. In the literature, a large panel of numerical integration methods has been proposed to tackle this problem. For example, **?** recently developed a quadrature rule based on arbitrary sample sets which was applied to a similar industrial OWT use case.

Alternatively, sampling methods rely on generating a set of points $\mathbf{X}_n$ drawn from the input distribution to compute the arithmetic mean of their realizations (i.e., uniform weights $\left\{w_i = \frac{1}{n}\right\}_{i=1}^{n}$). Among them, low-discrepancy sequences, also called "quasi-Monte Carlo" sampling (e.g., Sobol', Halton, Faure sequences) are known to improve the standard Monte Carlo convergence rate and will be used as a deterministic reference method in the following numerical experiments (Leobacher and Pillichshammer, 2014).

Quantization of probability measures and quadrature When dealing with probabilistic integration such as Eq. (4.5), a quadrature rule is a finite representation of a continuous measure π by a discrete measure $\zeta_n = \sum_{i=1}^n w_i \delta(\mathbf{x}^{(i)})$ (weighted sum of Dirac distributions at the design points $\mathbf{X}_n$). In the literature, this procedure is also called *quantization* of a continuous measure π. Overall, numerical integration is a particular case of probabilistic integration against a uniform input measure. For uniform measures, the Koksma-Hlawka inequality (Morokoff and Caflisch, 1995) provides a useful upper bound on the absolute error of a quadrature rule:

$$\left| \int_{[0,1]^d} g(\mathbf{x}) \mathrm{d}\mathbf{x} - \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}^{(i)}) \right| \leq V(g) D_n^*(\mathbf{X}_n). \tag{4.6}$$

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

4.3.2 Kernel herding sampling

Quasi-Monte Carlo sampling methods widely rely on a metric of uniformity, called *discrepancy*. To go beyond uniform measures, Appendix B introduces a kernel-based discrepancy, generalizing the discrepancy concept to non-uniform measures. This tool, named the maximum mean discrepancy (MMD) allows comparing multivariate distributions by embedding them in a specific function space. In this manuscript, the MMD was employed as a tool for statistical testing and quantifying the perturbations of distributions in Chapter 3, and for sensitivity analysis in Section 1.6.

Herin, the MMD is used to build a quadrature rule by sampling from a known measure. In other words, to quantize a known target measure π by a design sample $\mathbf{X}_n$. For practical reasons, the design construction is done sequentially. Sequential strategies have also been used to learn and validate regression models for statistical learning (see **?**). Moreover, since each realization is supposed to be obtained at the same unitary cost, the quadrature weights are first fixed as uniform during the construction of the design $\mathbf{X}_n$.

Kernel herding (KH), proposed by **?**, is a sampling method that offers a quantization of the measure π by minimizing a squared MMD when adding points iteratively. With a current design $\mathbf{X}_n$ and its corresponding discrete distribution with uniform weights $\zeta_n = \frac{1}{n} \sum_{i=1}^n \delta(\mathbf{x}^{(i)})$, a KH iteration is as an optimization of the following criterion, selecting the point $\mathbf{x}^{(n+1)} \in \mathcal{D}_{\mathbf{X}}$:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x} \in \mathcal{D}_{\mathbf{X}}}{\arg\min} \left\{ \mathrm{MMD} \left(\pi, \frac{1}{n+1} \left(\delta(\mathbf{x}) + \sum_{i=1}^n \delta(\mathbf{x}^{(i)}) \right) \right)^2 \right\}. \tag{4.7}$$

In the literature, two formulations of this optimization problem can be found. The first one uses the Frank-Wolfe algorithm (or "conditional gradient algorithm") to compute a linearization of the problem under the convexity hypothesis (see **?** and **?** for more details). The second one is a straightforward greedy optimization. Due to the combinatorial complexity, the greedy formulation is tractable for sequential construction only. Let us develop the MMD criterion from Eq. (B.7):

$$\mathrm{MMD}\left(\pi, \frac{1}{n+1}\left(\delta(\mathbf{x}) + \sum_{i=1}^{n}\delta(\mathbf{x}^{(i)})\right)\right)^2 = \varepsilon_\pi - \frac{2}{n+1}\sum_{i=1}^{n+1} P_\pi\left(\mathbf{x}^{(i)}\right) + \frac{1}{(n+1)^2}\sum_{i,j=1}^{n+1} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \tag{4.8a}$$

$$= \varepsilon_\pi - \frac{2}{n+1}\left(P_\pi(\mathbf{x}) + \sum_{i=1}^{n} P_\pi\left(\mathbf{x}^{(i)}\right)\right) \tag{4.8b}$$

$$+ \frac{1}{(n+1)^2}\left(\sum_{i,j=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) + 2\sum_{i=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) - k(\mathbf{x}, \mathbf{x})\right). \tag{4.8c}$$

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

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{\frac{1}{n+1}\sum_{i=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) - P_\pi(\mathbf{x})\right\} = \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{\frac{n}{n+1}P_{\zeta_n}(\mathbf{x}) - P_\pi(\mathbf{x})\right\}. \tag{4.9}$$

Remark 3. For the sequential and uniformly weighted case, the formulation in Eq. (4.9) is almost similar to the Frank-Wolfe formulation. Our numerical experiments showed that these two versions generate very close designs, especially as n becomes large. **?** express the Frank-Wolfe formulation in the sequential and uniformly weighted case as follows:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{P_{\zeta_n}(\mathbf{x}) - P_\pi(\mathbf{x})\right\}. \tag{4.10}$$

Remark 4. In practice, the optimization problem is solved by a brute-force approach on a fairly dense finite subset $\mathcal{S} \subseteq \mathcal{D}_\mathrm{X}$ of candidate points with size $N \gg n$ that emulates the target distribution, also called the "candidate set". This sample is also used to estimate the target potential $P_\pi(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$.

The diagram illustrated in Fig. 4.6 summarizes the main steps of a kernel herding sampling algorithm. One can notice that the initialization can either be done using a median point (maximizing the target potential) or from any existing design of experiments. This second configuration is practical when the analyst must include some characteristic points in the design (e.g., points with a physical interpretation).

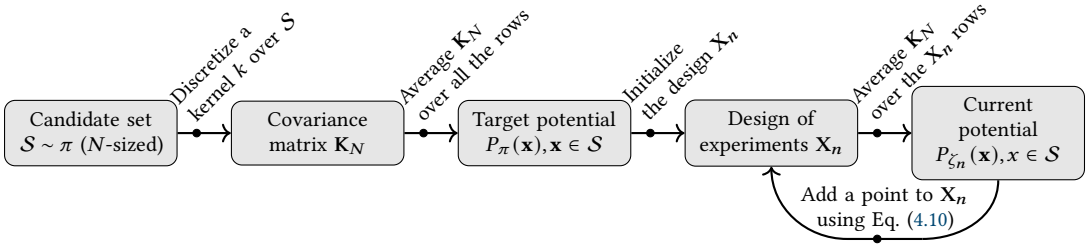


Figure 4.6 Greedy kernel herding algorithm

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

Table 4.3 Kernels considered in the following numerical experiments.

As explained previously, choosing the kernel defines the function space on which the worst-case function is found (see Eq. (B.5)). Therefore, this sampling method is sensitive to the kernel's choice. A kernel is defined, both by the choice of its parametric family (e.g., Matérn, squared exponential) and the choice of its tuning. The so-called "support points" method developed by **?** is a special case of kernel herding that uses the characteristic and parameter-free "energy-distance" kernel (introduced by **?**). In the following numerical experiments, the energy-distance kernel will be compared with an isotropic tensor product of a Matérn kernel (with regularity parameter $\nu = 5/2$ and correlation lengths θ_i), or a squared exponential kernel (with correlation lengths θ_i) defined in Table 4.3. Since the Matérn and squared exponential kernels are widely used for Gaussian process regression (**?**), they were naturally picked to challenge the energy-distance kernel. The correlation lengths for the squared exponential and Matérn kernels are set using the heuristic given in **?**, $\theta_i = n^{-1/d}, i \in \{1,\dots,d\}$.

Fig. 4.7 represents the covariance structure of the three kernels. One can notice that the squared exponential and Matérn $\nu = 5/2$ kernels are closer to one another than they are to the energy-distance. In fact, as ν tends to infinity, the Matérn kernel tends toward the squared exponential kernel (which has infinitely differentiable sample paths, see **?**). For these two stationary kernels, the correlation length controls how fast the correlation between two points decreases as their distance from one another increases.

Meanwhile, the energy distance is not stationary (but still positive and semi-definite). Therefore, its value does not only depend on the distance between two points but also on the norm of each of the points. Interestingly, the energy-distance kernel is almost similar to the kernel used by Hickernell (1998) to define a widely-used space-filling metric called the centered L^2-discrepancy. A presentation of these kernel-based discrepancies from the design of experiment point of view is also provided in Chapter Two from Fang et al. (2018).

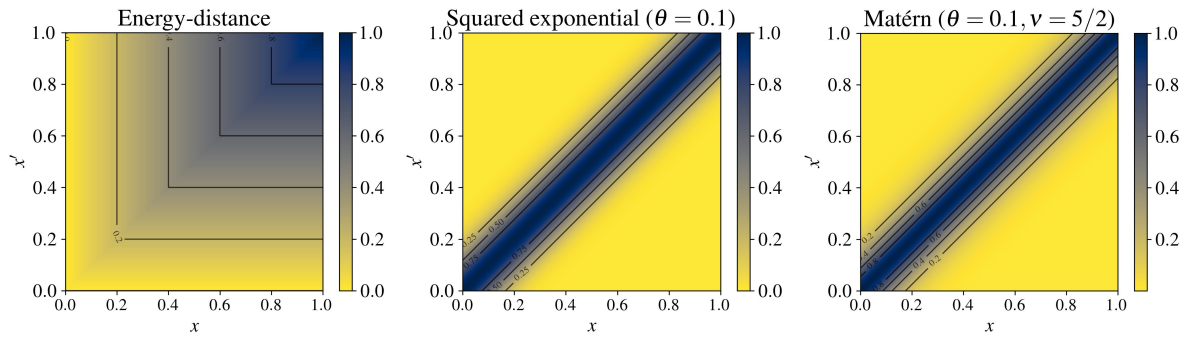


Figure 4.7 Kernel illustrations (left to right: energy-distance, squared exponential, and Matérn 5/2)

To illustrate the kernel herding sampling of a complex distribution, Fig. 4.8 shows three nested samples (orange crosses for different sizes $n \in \{10, 20, 40\}$) of a mixture of Gaussian distributions with complex nonlinear dependencies (with density represented by the black isoprobability contours). In this example, the method seems to build a parsimonious design between each mode of the distribution (by subsampling directly without any transformation). The candidate set (in light grey) was generated by a large quasi-Monte sample of the underlying Gaussian mixture. In this two-dimensional case, this candidate set is sufficient to estimate the target potential P_π. However, the main bottleneck of kernel herding is the estimation of the potentials, which becomes costly in high dimensions.

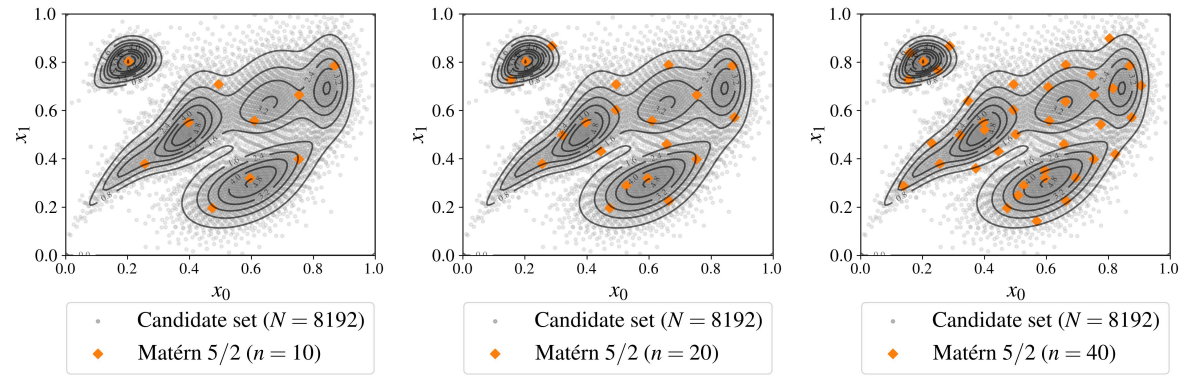


Figure 4.8 Sequential kernel herding for increasing design sizes ($n \in \{10, 20, 40\}$) built on a candidate set of $N = 8196$ points drawn from a complex Gaussian mixture π

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

4.3.3 Bayesian quadrature

Gaussian processes for Bayesian quadrature Kernel methods and Gaussian processes present a lot of connections and equivalences, thoroughly reviewed by **?**. In numerical integration, Gaussian processes have been used to build quadrature rules in the seminal paper of **?**, introducing the concept of *Bayesian quadrature* (BQ). Let us recall the probabilistic integration problem $I_\pi(g) = \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x}) \mathrm{d}\pi(\mathbf{x})$ (stated in Eq. (4.5)). From a general point of view, this quantity could be generalized by composing g with another function ψ (e.g., other moments, quantiles, exceedance probabilities). The quantity of interest then becomes, $I_\pi(\psi(g))$, for example when ψ is a monomial, it gives a moment of the output distribution.

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

$$\begin{cases} \eta_n(\mathbf{x}) & := \mathbf{k}_n^\top(\mathbf{x}) \mathbf{K}_n^{-1} \mathbf{y}_n \\ s_n^2(\mathbf{x}) & := k_n(\mathbf{x},\mathbf{x}) - \mathbf{k}_n^\top(\mathbf{x}) \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}) \end{cases} \tag{4.11}$$

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

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

$$I_\pi(G_n) = \int_{\mathcal{D}_\mathbf{X}} (G(\mathbf{x})|\mathbf{y}_n) \mathrm{d}\pi(\mathbf{x}) = \int_{\mathcal{D}_\mathbf{X}} G_n(\mathbf{x}) \mathrm{d}\pi(\mathbf{x}) \,. \tag{4.12}$$

Fig. 4.9 provides a one-dimensional illustration of the Bayesian quadrature of an unknown function (dashed black curve) against a given input measure π (with corresponding grey distribution at the bottom). For an arbitrary design, one can fit a Gaussian process model, interpolating the function observations (black crosses). Then, multiple trajectories of this conditioned Gaussian process G_n are drawn (orange curves) whilst its mean function, also called "predictor", is represented by the red curve. Therefore, the input measure π is propagated through the conditioned Gaussian process to obtain the random variable $I_\pi(G_n)$, with distribution represented on the right plot (brown curve). Again on the right plot, remark how the mean of this posterior distribution (brown line) is closer to the reference output expected value (dashed black line) than the arithmetic mean of the observations (black line). This plot was inspired by the paper of **?**.

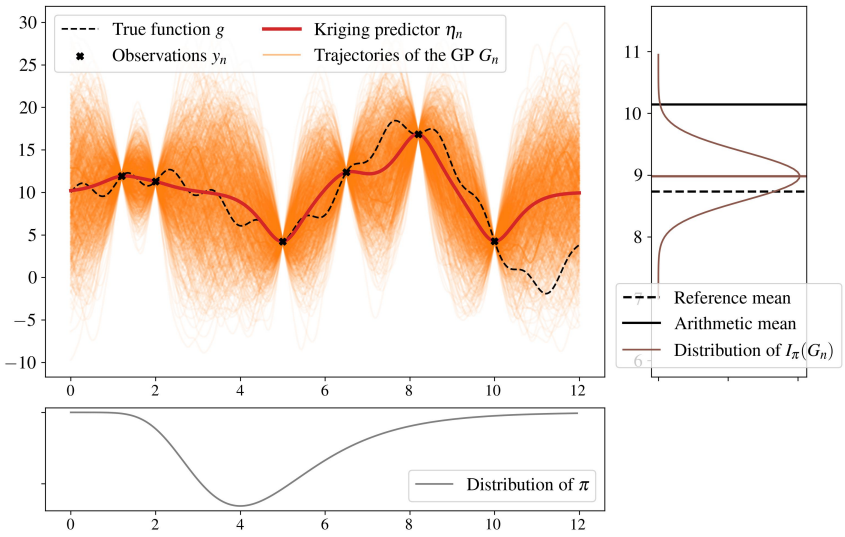


Figure 4.9 Bayesian quadrature on a one-dimensional case

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

$$\overline{y}_n^{\mathrm{BQ}} := \mathbb{E}\left[I_\pi(G_n)|\mathbf{y}_n\right] = \int_{\mathcal{D}_\mathbb{X}} \eta_n(\mathbf{x})\mathrm{d}\pi(\mathbf{x}) = \left[\int_{\mathcal{D}_\mathbb{X}} \mathbf{k}_n^\top(\mathbf{x})\mathrm{d}\pi(\mathbf{x})\right]\mathbf{K}_n^{-1}\mathbf{y}_n = P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}\mathbf{y}_n, \tag{4.13}$$

$$\left(\sigma_n^{\mathrm{BQ}}\right)^2 := \mathrm{Var}\left(I_\pi(G_n)\right) = \iint_{\mathcal{D}_{\mathbb{X}^2}} k_n(\mathbf{x},\mathbf{x}')\mathrm{d}\pi(\mathbf{x})\mathrm{d}\pi(\mathbf{x}') = \varepsilon_\pi - P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}P_\pi(\mathbf{X}_n)^\top. \tag{4.14}$$

Where $P_\pi(\mathbf{X}_n)$ is the row vector of potentials $\left[\int k_n(\mathbf{x},\mathbf{x}^{(1)})\mathrm{d}\pi(\mathbf{x}),\ldots,\int k_n(\mathbf{x},\mathbf{x}^{(n)})\mathrm{d}\pi(\mathbf{x})\right]$, and ε_π is given in Eq. (1). As in the one-dimensional example presented in Fig. 4.9, the expected value of $I_\pi(G_n)$ expressed in Eq. (4.13) is a direct estimator of the quantity of interest Eq. (4.5). The so-called "Bayesian quadrature estimator" appears to be a simple linear combination of the observations by taking the row vector of "optimal weights" as:

$$\mathbf{w}_{\mathrm{BQ}} := P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1} \tag{4.15}$$

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

expressed in closed-form in Eq. (4.14). The expressions in Eq. (4.13) and Eq. (4.14) were extended to Gaussian processes in the case of constant and linear trends in **?**. In the following numerical experiments, the expression with a hypothesis of constant trend β_n is used, which leads to:

$$\mathbb{E}\left[I_\pi(G_n)\right] = \beta_n + P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}\left(\mathbf{y}_n - \beta_n \mathbf{1}_n\right). \tag{4.16}$$

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

$$\overline{y}_n^{\mathrm{BQ}} \in \left[\overline{y}_n^{\mathrm{BQ}} - 2\sigma_n^{\mathrm{BQ}}, \overline{y}_n^{\mathrm{BQ}} + 2\sigma_n^{\mathrm{BQ}}\right]. \tag{4.17}$$

Variance-based Bayesian quadrature rule The link between the posterior variance and the squared MMD has been first made by **?** in their Proposition 1: the expected variance in the Bayesian quadrature $\mathrm{Var}\left(I_\pi(G_n)\right)$ is the MMD between the target distribution π and $\zeta_n = \sum_{i=1}^n \mathbf{w}_{\mathrm{BQ}}^{(i)} \delta(\mathbf{x}^{(i)})$. The proof is reproduced below (as well as in Proposition 6.1 from **?**):

$$\begin{aligned}
\mathrm{Var}\left(I_\pi(G_n)\right) &= \mathbb{E}\left[\left(I_\pi(G_n) - I_{\zeta_n}(G_n)\right)^2\right] && \text{(4.18a)}\\
&= \mathbb{E}\left[\left(\langle G_n, P_\pi\rangle_{\mathcal{H}(k)} - \left\langle G_n, P_{\zeta_n}\right\rangle_{\mathcal{H}(k)}\right)^2\right] && \text{(4.18b)}\\
&= \mathbb{E}\left[\left\langle G_n, P_\pi - P_{\zeta_n}\right\rangle_{\mathcal{H}(k)}^2\right] && \text{(4.18c)}\\
&= \|P_\pi - P_{\zeta_n}\|_{\mathcal{H}(k)}^2 && \text{(4.18d)}\\
&= \mathrm{MMD}(\pi, \zeta_n)^2. && \text{(4.18e)}
\end{aligned}$$

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

Remark 5. Every quadrature method introduced in this section has been built without any observation of the possibly costly function g. Therefore, they cannot be categorized as active learning approaches. Contrarily, **?** presents a set of methods for BQ with transformations (i.e., adding a positivity constraint on the function g), which are truly active learning methods.

4.4 Numerical experiments

This section presents numerical results computed on two different analytical toy cases, respectively in dimension 2 (toy case #1) and dimension 10 (toy case #2), with easy-to-evaluate functions $g(\cdot)$ and associated input distributions π. Therefore, reference values can easily be computed with great precision. For each toy case, a large reference Monte Carlo sample ($N_{\text{ref}} = 10^8$) is taken. This first benchmark compares the mean estimation of toy cases given by a quasi-Monte Carlo technique (abbreviated by QMC in the next figures) which consists herein using a Sobol' sequence, and kernel herding with the three kernels defined in Table 4.3. Notice that the quasi-Monte Carlo designs are first generated on the unit hypercube and then, transformed using the generalized Nataf transformation to follow the target distribution (**?**). Additionally, the performances of kernel herding for both uniform and optimally-weighted Eq. (4.16) estimators are compared.

All the following results and methods (i.e., the kernel-based sampling and BQ methods) have been implemented in a new open-source Python package named `otkerneldesign`[5]. This development mostly relies on the open source software OpenTURNS[6] ("Open source initiative for the Treatment of Uncertainties, Risks'N Statistics") devoted to uncertainty quantification and statistical learning (**?**). Finally, note that the numerical experiments for the toy cases are available in the Git repository named `ctbenchmark`[7].

4.4.1 Illustration on analytical toy-cases

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

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

The 10-dimensional "GSobol function" (toy case #2) with a set of coefficient $\{a_i = 2\}_{i=1}^{10}$ has an effective dimension equal to 10 and belongs to the hardest class C from **?**. In the case of the two-dimensional Gaussian mixture problem, the complexity is carried by the mixture of

[5] https://efekhari27.github.io/otkerneldesign/master/index.html
[6] https://openturns.github.io/www/
[7] https://github.com/efekhari27/ctbenchmark

Table 4.4 Analytical toy-cases

Toy-case #1	$dim = 2$	$g_1(\mathbf{x}) = x_1 + x_2$	Gaussian mixture from Fig. 4.8
Toy-case #2	$dim = 10$	$g_2(\mathbf{x}) = \prod_{i=1}^{10} \frac{\lvert 4x_i - 2\rvert + a_i}{1+a_i}, \{a_i = 2\}_{i=1}^{10}$	Gaussian $\mathcal{N}(\mathbf{0.5}, \mathbf{I}_{10})$

Gaussian distributions with highly nonlinear dependencies. Probabilistic integration results are presented in Fig. 4.10 (toy case #1) and Fig. 4.11 (toy case #2). Kernel herding samples using the energy-distance kernel are in red, while quasi-Monte Carlo samples built from Sobol' sequences are in grey. Convergences of the arithmetic means are plotted on the left and MMDs on the right. The respective BQ estimators of the means are plotted in dashed lines.

Remark 6. Different kernels are used in these numerical experiments. First, the generation kernel is used by the kernel herding algorithm to generate designs (with the heuristic tuning defined in Section 4.3.2). Second, the BQ kernel allows computation of the optimal weights (arbitrarily set up as a Matérn 5/2 with the heuristic tuning). Third, the evaluation kernel must be common to allow a fair comparison of the computed MMD results (same as the BQ kernel).

Results analysis for toy case #1. Convergence plots are provided in Fig. 4.10. KH consistently converges faster than quasi-Monte Carlo in this case, especially for small sizes in terms of MMD. BQ weights tend to reduce the fluctuations in the mean convergence, which ensures better performance for any size. Overall, applying the weights enhances the convergence rate.

Results analysis for toy case #2. Convergence plots are provided in Fig. 4.11. Although quasi-Monte Carlo is known to suffer the "curse of dimensionality", KH does not outperform it drastically in this example. In fact, KH with uniform weights performs worse than quasi-Monte Carlo while optimally-weighted KH does slightly better. Moreover, the results confirm that $\text{MMD}_{\text{BQ}} < \text{MMD}_{\text{unif}}$ for all our experiments. The application of optimal weights to the quasi-Monte Carlo sample slightly improves the estimation in this case. Note that the prediction interval around the BQ estimator is not plotted for the sake of readability.

In these two toy cases, the MMD is shown to quantify numerical integration convergence well, which illustrates the validity of the inequality given in Eq. (B.4c), similar to the Koksma-Hlawka inequality (as recalled in Eq. (4.6)).

4.4.2 Application to the Teesside wind turbine fatigue estimation

Let us summarize the mean damage estimation strategies studied in this chapter. The diagram represented in Fig. 4.12 describes the different workflows computed. The simplest workflow is represented by the grey horizontal sequence. It directly subsamples a design of experiments from a large and representative dataset (previously referred to as candidate set). This workflow simply estimates the mean damage by computing an arithmetic average of the outputs.

Alternatively, one can respectively fit a joint distribution and sample from it. In our case, this distribution is only known empirically via the candidate set. Since its dependence structure is complex (see Fig. 4.3), a parametric method might fit the distribution poorly (and therefore lead

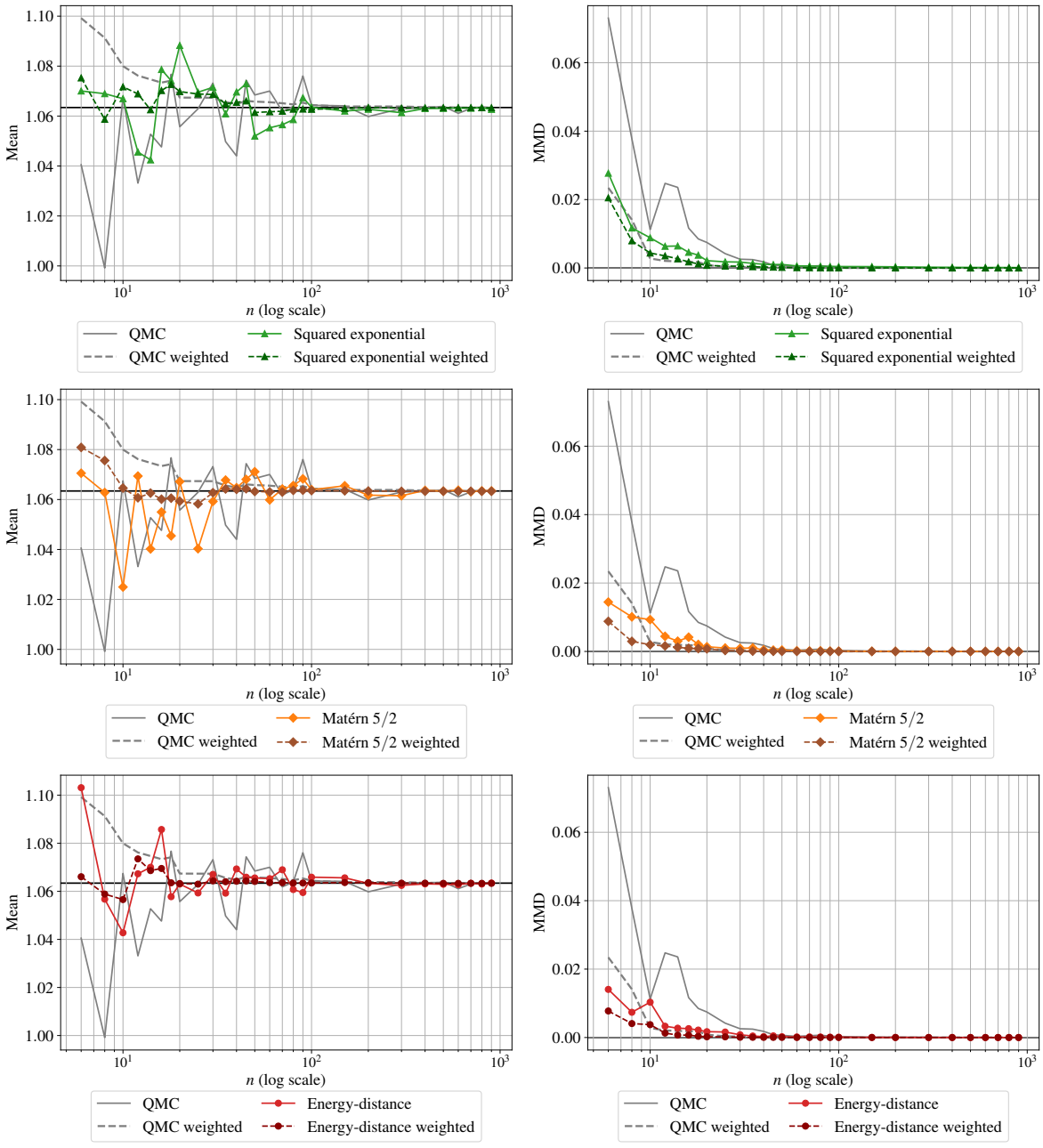


Figure 4.10 Analytical benchmark results on the toy-case #1

to a poor estimation of the quantity). Then, a nonparametric fit using the empirical Bernstein copula (introduced in Section 4.2.3) coupled with a kernel density estimation on each marginal is applied to the candidate set (with the EBC parameter $m = 100 > m_{\text{MISE}}$ to avoid bias, see Lasserre (2022) p.117). The sampling on this hybrid joint distribution is realized with a quasi-Monte Carlo method. A Sobol' low-discrepancy sequence generates a uniform sample in the unit hypercube, which can then be transformed according to a target distribution. Remember that quasi-Monte Carlo sampling is also sensitive to the choice of a low-discrepancy sequence, each presenting different properties (e.g., Sobol', Halton, Faure, etc.). Finally, the estimation by

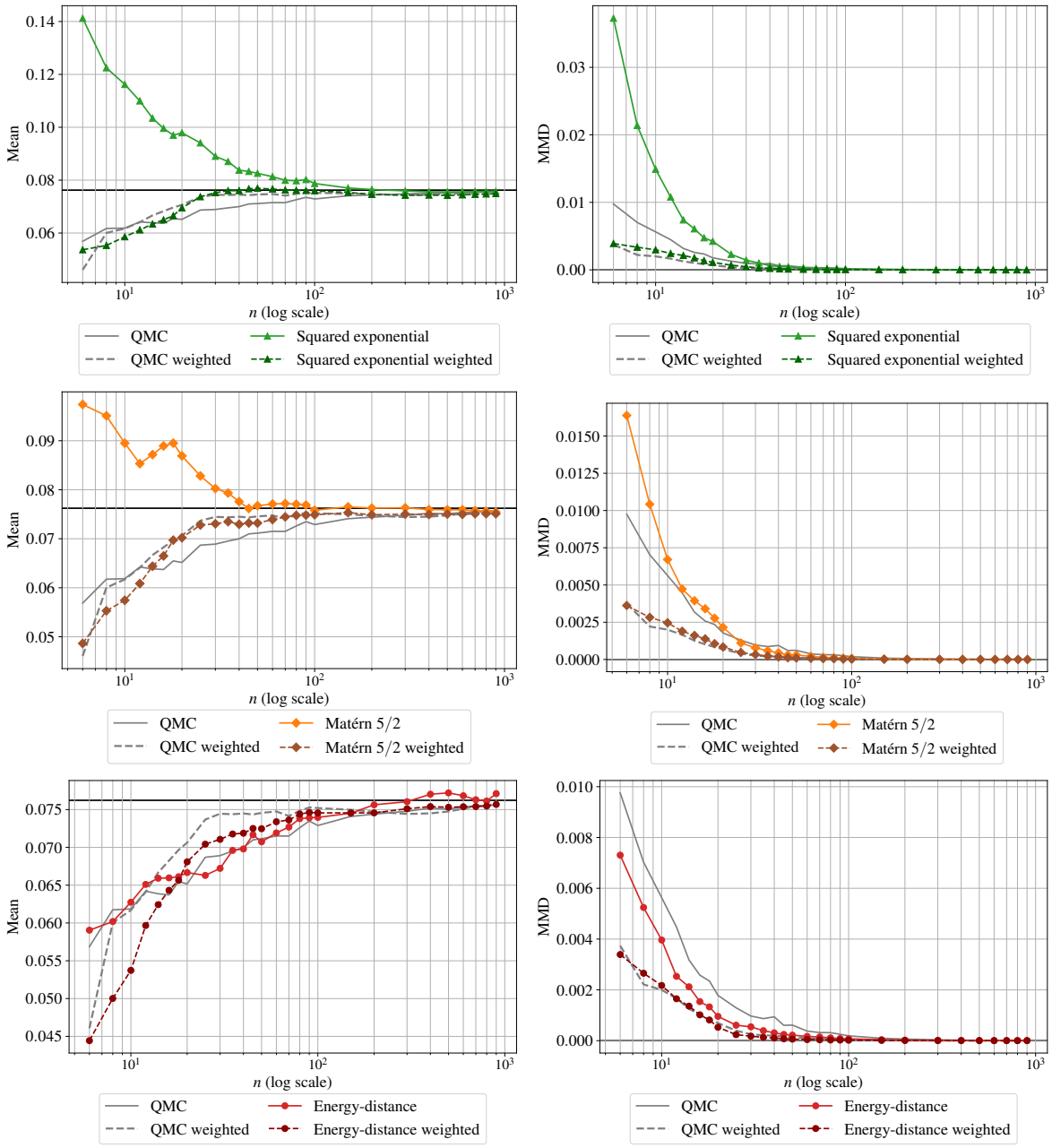


Figure 4.11 Analytical benchmark results on the toy-case #2

an arithmetic mean can be replaced by an optimally weighted mean. To do so, optimal weight must be computed, using the formulas introduced in Eq. (4.15).

The copulogram in Fig. 4.13 illustrates the intensity of the computed damages, proportionally to the color scale. Note that the numerical values of the damage scale are kept confidential since it models the state of an operating asset. Before analyzing the performance of the KH on this industrial application, let us notice that the copulogram Fig. 4.13 seems to be in line with the global sensitivity analysis presented in **?** and **?**. In particular, the fact that the scatter plot of mean wind speed vs. turbulence wind speed is the main factor explaining the variance of the output $Y = g(\mathbf{X})$. Judging from these references, the numerical model does not seem to have

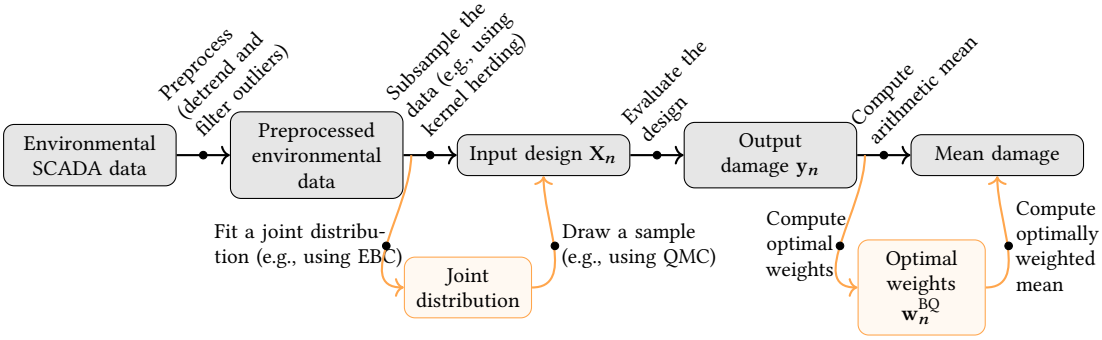


Figure 4.12 Mean damage estimation workflows for the industrial use case. The orange parts represent optional alterations to the workflow: the first one is an alternative to input data subsampling where the underlying distribution is sampled from, the second one improves mean damage calculation by using optimal weights over the output data

a highly effective dimension, however, the input dependence structure is challenging and the damage assessment induces strong nonlinearities (see Eq. (4.2)).

The results presented are compared in the following to a large reference Monte Carlo sample (size 2000) with a confidence interval computed by bootstrap (see Fig. 4.14). This reference is represented by a horizontal line intersecting with the most converged Monte Carlo estimation. Once again, the mean damage scale is hidden for confidentiality reasons, but all the plots are represented for the same vertical scale. The performance of the KH is good: it quickly converges towards the confidence interval of the Monte Carlo obtained with the reference sample. In addition, the Bayesian quadrature estimator also offers a posteriori prediction interval, which can reassure the user. The BQ prediction intervals are smaller than the ones obtained by bootstrap on the reference Monte Carlo sample.

To provide more representative results, note that a set of scale parameters is computed with a kriging procedure to define the kernel used to compute BQ intervals. Since other methods do not generate independent samples, bootstrapping them is not legitimate. Contrarily to the other kernels, we observe that the energy-distance kernel presents a small bias with the MC reference for most of the azimuth angles computed in this experiment. Meanwhile, combining nonparametric fitting with quasi-Monte Carlo sampling also delivers good results as long as the fitting step does not introduce a bias. In our case, any potential bias due to poor fitting would be the result of a poorly tuned empirical Bernstein copula. Fortunately, **?** formulated recommendations regarding how to tune empirical Bernstein copulas. We follow these recommendations in the present work.

4.5 Conclusion

Wind energy assets are subject to highly uncertain environmental conditions. Uncertainty propagation through numerical models is performed to ensure their structural integrity (and

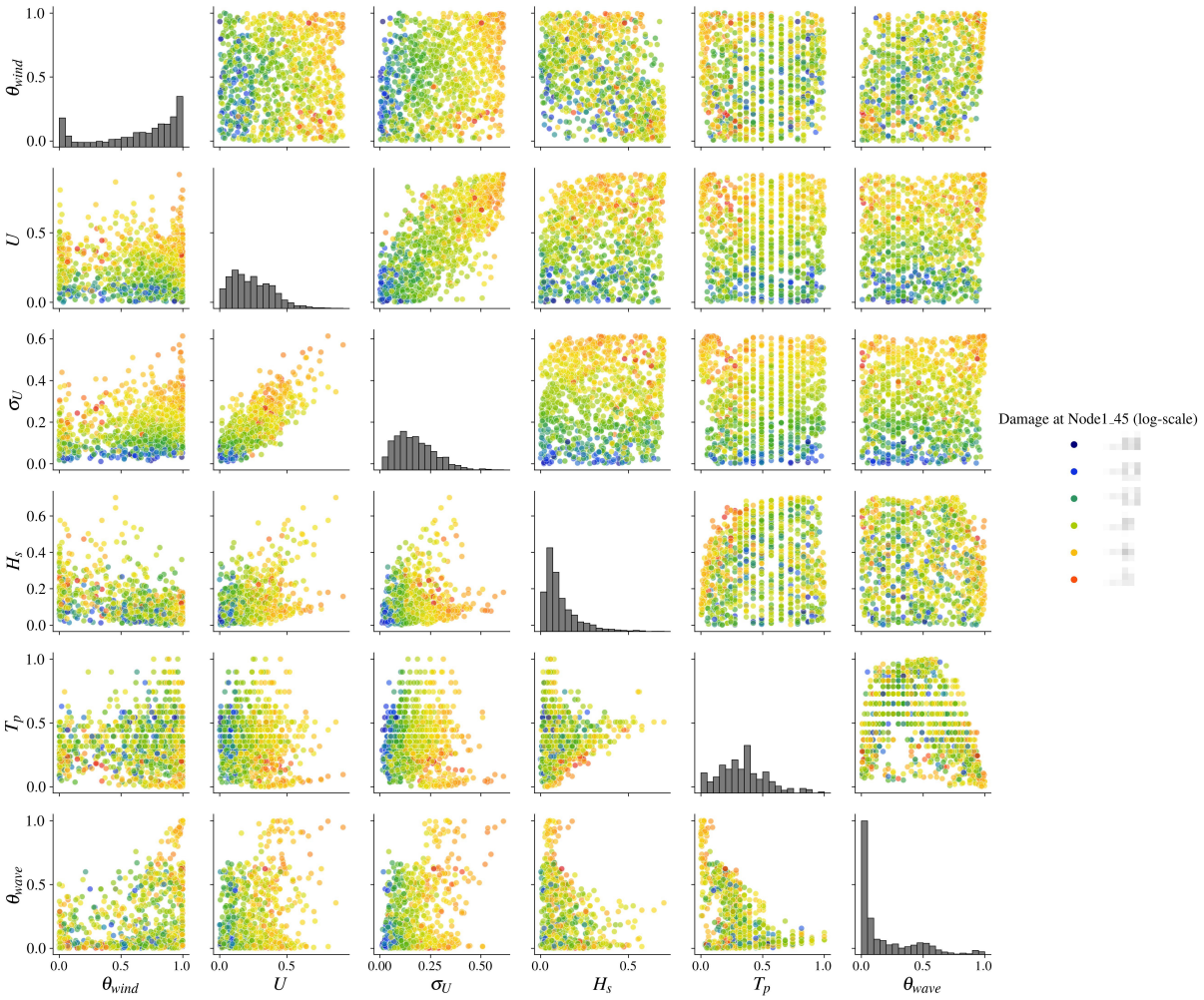


Figure 4.13 Copulogram of the kernel herding design of experiments with corresponding outputs in color (log-scale) on the Teesside case ($n = 10^3$). The color scale ranges from blue for the lowest values to red for the largest. Marginals are represented by histograms (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.

energy production). For this case, the method recommended by the standards (regular grid sampling) is intractable for even moderate-fidelity simulators. In practice, such an approach can lead to poor uncertainty propagation, especially when facing simulation budget constraints.

In the present chapter, a real industrial wind turbine fatigue estimation use case is investigated, considering site-specific data. As a perspective, other sites with different environmental conditions could be studied. This use case induces two practical constraints: first, usual active learning methods are hard to set up on such a model (mainly due to the nonlinearity of the variable of interest), and they restrict the use of high-performance computing facilities; second, the input distribution of the environmental conditions presents a complex dependence structure which is hard to infer with common parametric approaches.

In this work, the association of kernel herding sampling with Bayesian quadrature for central tendency estimation is explored theoretically and numerically. This method fits with

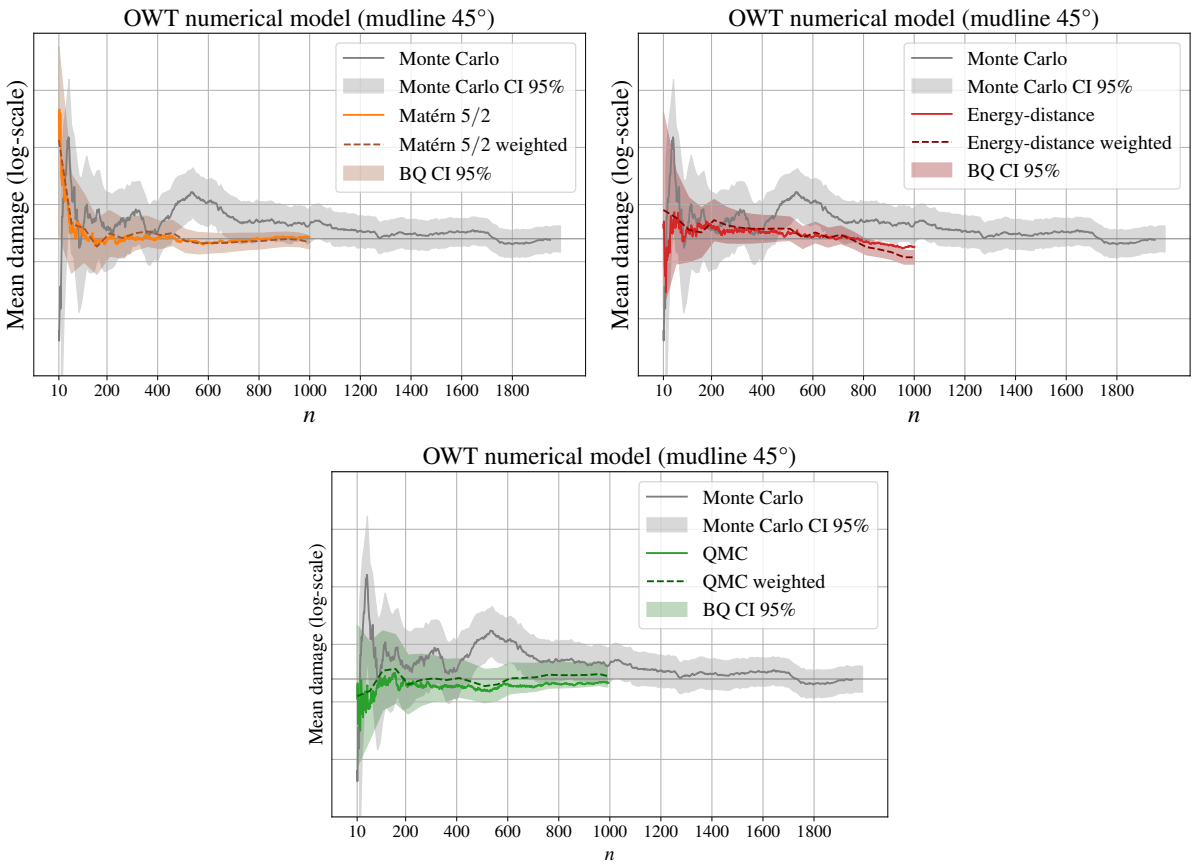


Figure 4.14 Mean estimation convergence (at the mudline, azimuth $\theta = 45$ deg.) on the Teesside case. Monte Carlo confidence intervals are all computed by bootstrap

the practical constraints induced by the industrial use case. To be more specific, the kernel herding method easily subsamples the relevant points directly from a given dataset (here, from the measured environmental data). Moreover, the method is fully compatible with intensive high-performance computer use. Moreover, the present work outlined an upper bound based on the maximum mean discrepancy (MMD) on numerical integration absolute error. Kernel herding and Bayesian quadrature both aim at finding the quadrature rule minimizing the MMD, and therefore the absolute integration error. The numerical experiments confirm that the MMD is an appropriate criterion since it leads to results being better or equivalent to quasi-Monte Carlo sampling. Finally, the proposed numerical benchmark relies on a Python package, called `otkerneldesign`, which implements the methods and allows anyone to reproduce the results.

The limits of the proposed method are reached when the input dimension of the problem increases, requiring a larger candidate set and therefore a larger covariance matrix. Moreover, the numerical experiments show that the method can be sensitive to the choice of the kernel and its tuning (although good practices can be derived). From a methodological viewpoint, further interpretation of the impact of the different kernels could be explored. Besides, extensions of kernel herding sampling for quantile estimation could be investigated, in a similar fashion

as the work on randomized quasi-Monte Carlo for quantiles proposed by Kaplan et al. (2019). Kernel herding could also be used to quantize conditional distributions, using the so-called "conditional kernel mean embedding" concept reviewed by **?**. Finally, regarding the industrial use case, the next step should be to perform a reliability analysis by considering another group of random variables (related to the wind turbine) or to explore the possibilities offered by reliability-oriented sensitivity analysis in the context of kernel-based indices, as studied in **?**.

Chapter 5

Kernel-based surrogate models validation

This chapter is adapted from the following reference:

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.

5.1 Introduction

The development of methods to validate and certify the predictivity of supervised learning models is essential to the industry. Estimating the predictivity of these models can either be done by cross-validation or using a suitably selected test sample (as introduced in Section 1.7). Both in a given-data context (i.e., machine learning) or a simulated data context (i.e., computer experiment), guarantees on the validation procedure are increasingly asked. Certain risk-averse industries (e.g., nuclear) impose to establish these guarantees from independent test sets, i.e., datasets that have not been used either to train or to select the learning model (**???**). Using the prediction residuals on this test set, an independent evaluation of the proposed learning model can be done, enabling the estimation of relevant performance metrics, such as the mean-squared error for regression problems, or the misclassification rate for classification problems.

The present chapter introduces methods to choose a "good" test set, either within a given dataset or within the input space of the model, as recently motivated in **??**. The construction of test sets is studied as an uncertainty propagation of the learning model's error, on which an average error may be estimated using the Bayesian quadrature methods introduced in Chapter 4 for mean estimation.

A first choice concerns the size of the test set. No optimal choice exists, and, when only a finite dataset is available, classical machine learning (ML) handbooks (**??**) provide different heuristics on how to split it, e.g., 80%/20% between the training and test samples, or 50%/25%/25% between the training, validation (used for model selection) and test samples. This point is not formally addressed in the following (see **?** for a numerical study of this issue). A second issue concerns how the test sample is picked within the input space. The simplest, and most common way to build a test sample is to extract an independent Monte Carlo sample (**?**). For small test sets, these randomly chosen points may fall too close to the training points or leave large areas of the input space unsampled, and a more constructive method to select points inside the input domain is therefore preferable. Similar concerns motivate the use of space-filling designs when choosing a small set of runs for computationally expensive computer experiments on which a model will be identified (**??**).

When the test set must be a subset of an initial dataset, the problem amounts to selecting a certain number of points within a finite collection of points. A review of classical methods for solving this issue is given in **?**. For example, the CADEX and DUPLEX algorithms (**??**) can sequentially extract points from a database to include them in a test sample, using an inter-point distance criterion.

Several algorithms have also been proposed for the case where points need to be added to an already existing training sample. When the goal is to assess the quality of a model learned using a known training set, one may be tempted to locate the test points the furthest away from the training samples, such that, in some sense, the union of the training and test sets is space-filling. As this chapter shows, test sets built in this manner do enable a good assessment of the quality of models learned with the training set if the observed residuals are appropriately weighted. Moreover, the incremental augmentation of a design can be useful when the assessed model turns out to be of poor quality, or when an additional computational budget is available after a first study (**??**). Different empirical strategies have been proposed for incremental space-filling design (**???**), which basically entail the addition of new points in the zones poorly covered by the current design. **?** have recently proposed an improvement of the CADEX algorithm, called the "fully-sequential space-filling" (FSSF) design. **?** also proposed an improved version of such design enforcing boundary avoidance. Although they are developed for different purposes, nested space-filling designs (**?**) and sliced space-filling designs (**?**) can also be used to build sequential designs.

This work provides insights into these subjects in two main directions: *(i)* definition of a new predictivity criteria through an optimal weighting of the test points residuals, and *(ii)* use of test sets built by incremental space-filling algorithms, namely FSSF, support points **?** and kernel herding **?**, the latter two algorithms being typically used to provide a representative sample of a desired theoretical or empirical distribution. Besides, this chapter presents a numerical benchmark analysis comparing the behavior of the three algorithms on a selected set of test cases and an industrial case.

This chapter is organized as follows. Section 5.2 defines the predictivity criterion considered and proposes different methods for its estimation. Section 5.3 presents the algorithms used for test-point selection. Our numerical results are presented in Sections 5.4 and 5.5: in Section 5.4 a test set is freely chosen within the entire input space, while in Section 5.5 an existing data set can be split into a training sample and a test set. Finally, Section 5.6 concludes and outlines some perspectives.

5.2 Predictivity assessment criteria for an ML model

In this section, a new criterion to assess the predictive performance of a model is proposed, enhancing a standard model quality metric by suitably weighting the errors observed on the test set. Let us denote by $\mathcal{D}_{\mathbf{x}} \subset \mathbb{R}^d$ the space of the input variables $\mathbf{x} = (x_1, \ldots, x_d)$ of the model. Then let $y(\mathbf{x}) \in \mathbb{R}$ (resp. $y(\mathbf{x}') \in \mathbb{R}$) be the observed output at point $\mathbf{x} \in \mathcal{D}_{\mathbf{x}}$ (resp. $\mathbf{x}' \in \mathcal{D}_{\mathbf{x}}$). Considering the training sample denoted by $(\mathbf{X}_m, \mathbf{y}_m)$, with $\mathbf{y}_m = [y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(m)})]^\top$. The test sample is denoted by $(\mathbf{X}_n, \mathbf{y}_n) = (\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)}))_{1 \leq i \leq n}$. Remember that the intersection between these two samples is empty, $\mathbf{X}_m \cap \mathbf{X}_n = \varnothing$.

5.2.1 The predictivity coefficient

Let us denote by $\eta_m(\mathbf{x})$ the prediction at point $\mathbf{x}$ of a model learned using $(\mathbf{X}_m, \mathbf{y}_m)$ (??). A classical measure for assessing the predictive ability of η_m, in order to evaluate its validity, is the *predictivity coefficient*. Considering the probability measure π that weights how comparatively important it is to accurately predict y over the different regions of $\mathcal{D}_{\mathbf{x}}$. For example, the input could be a random vector with a known distribution: in that case, this distribution would be a reasonable choice for π. The true (i.e., ideal) value of the predictivity is defined as the following normalization of the Integrated Square Error (ISE):

$$Q^2_\pi(\eta_m) = 1 - \frac{\mathrm{ISE}_\pi(\eta_m)}{\mathrm{Var}_\pi(g(\mathbf{X}))}, \tag{5.1}$$

where

$$\mathrm{ISE}_\pi(\eta_m) = \int_{\mathcal{D}_{\mathbf{x}}} [y(\mathbf{x}) - \eta_m(\mathbf{x})]^2 \, \mathrm{d}\pi(\mathbf{x}),$$
$$\mathrm{Var}_\pi(g(\mathbf{X})) = \int_{\mathcal{D}_{\mathbf{x}}} \left[y(\mathbf{x}) - \int_{\mathcal{D}_{\mathbf{x}}} y(\mathbf{x}') d\pi(\mathbf{x}') \right]^2 \mathrm{d}\pi(\mathbf{x}).$$

The ideal predictivity $Q^2_{\mathrm{ideal}}(\pi)$ is usually estimated by its empirical version calculated over the test sample $(\mathbf{X}_n, \mathbf{y}_n)$ (see ?, p. 32):

$$\widehat{Q}^2_n = 1 - \frac{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)}) \right]^2}{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_n \right]^2}, \tag{5.2}$$

where $\overline{y}_n = (1/n) \sum_{i=1}^n y(\mathbf{x}^{(i)})$ denotes the empirical mean of the observations in the test sample. Note that the calculation of $\widehat{Q}^2_n$ only requires access to the predictor $\eta_m(\cdot)$. To compute $\widehat{Q}^2_n$, one does not need to know the training set which was used to build $\eta_m(\cdot)$. This estimator $\widehat{Q}^2_n$ is the *coefficient of determination* (also called "Nash-Sutcliffe criterion" ?), which is a standard notion in regression studies (??). It compares the prediction errors obtained with the model η_m with those obtained when prediction equals the empirical mean of the observations. Thus, the closer $\widehat{Q}^2_n$ is to one, the more accurate the surrogate model is (for the test set considered). On the contrary, $\widehat{Q}^2_n$ close to zero (negative values are possible too) indicates poor prediction abilities, as there is little improvement compared to a prediction by the simple empirical mean of the observations. The next section shows how a suitable weighting of the residual on the training sample may be key to improving the estimation of $\widehat{Q}^2_n$.

5.2.2 Weighting the test sample

The simplest way to estimate the $\mathrm{ISE}_\pi(\mathbf{X}_m, \mathbf{y}_m)$ (present on the numerator of the predictivity coefficient) is by computing the arithmetic mean of the squared residuals evaluated on the test set $\mathbf{X}_n$. Writing the equivalent discrete measure $\xi_n = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}^{(i)}}$, with $\delta_{\mathbf{x}}$ the Dirac measure at $\mathbf{x}$,

this estimate can be expressed as:

$$\mathrm{ISE}_{\xi_n}(\eta_m) = \frac{1}{n}\sum_{i=1}^{n}\left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})\right]^2 .$$

When the points $\mathbf{x}^{(i)}$ of the test set $\mathbf{X}_n$ are distant from the points of the training set $\mathbf{X}_m$, the squared prediction errors $|y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})|^2$ tend to represent the worst possible error situations, and $\mathrm{ISE}_{\xi_n}(\eta_m)$ tends to overestimate $\mathrm{ISE}_{\pi}(\eta_m)$. In this section, a statistical model for the prediction errors is assumed in order to be able to quantify this potential bias when sampling the residual process, enabling its subsequent correction.

Let us assume that the prediction error $\delta_m(\mathbf{x}) = y(\mathbf{x}) - \eta_m(\mathbf{x})$ is a realization of a Gaussian Process (GP) with mean $\widehat{\delta}_m(\mathbf{x})$ and covariance kernel $\sigma^2 K_{|m}$, $\delta_m(\mathbf{x}) \sim \mathrm{GP}(\widehat{\delta}_m(\mathbf{x}), \sigma^2 K_{|m})$

$$\begin{cases} \widehat{\delta}_m(\mathbf{x}) = \mathbf{k}_m^\top(\mathbf{x})\mathbf{K}_m^{-1}(\mathbf{y}_m - \boldsymbol{\eta}_m), \\ \sigma^2 K_{|m}(\mathbf{x},\mathbf{x}') = \mathbb{E}[\delta_m(\mathbf{x})\delta_m(\mathbf{x}')] = \sigma^2\left[K(\mathbf{x},\mathbf{x}') - \mathbf{k}_m^\top(\mathbf{x})\mathbf{K}_m^{-1}\mathbf{k}_m(\mathbf{x}')\right] . \end{cases} \tag{5.3}$$

Where $\boldsymbol{\eta}_m = [\eta_m(\mathbf{x}^{(1)}), \ldots, \eta_m(\mathbf{x}^{(m)})]^\top$, $\mathbf{k}_m(\mathbf{x})$ is the column vector $[K(\mathbf{x},\mathbf{x}^{(1)}), \ldots, K(\mathbf{x},\mathbf{x}^{(m)})]^\top$, and $\mathbf{K}_m$ is the $m \times m$ covariance matrix whose element (i,j) is given by $\{\mathbf{K}_m\}_{i,j} = K(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$, with K a positive definite kernel. Note that in the case of a learning model η_m which interpolates the observations $\mathbf{y}_m$, the errors observed at the learning points $\mathbf{X}_m$ equal zero, leading finally to the posterior $\mathrm{GP}(0, \sigma^2 K_{|m})$ for $\delta_m(\mathbf{x})$.

The prediction model error above allows us to study how well $\mathrm{ISE}_{\pi}(\eta_m)$ is estimated using a test set $\mathbf{X}_n$. The expected squared error when estimating $\mathrm{ISE}_{\pi}(\eta_m)$ by $\mathrm{ISE}_{\xi_n}(\eta_m)$, is defined as $\overline{\Delta}^2(\xi_n, \pi; \eta_m)$:

$$\begin{aligned} \overline{\Delta}^2(\xi_n,\pi;\eta_m) &= \mathbb{E}\left[\left(\mathrm{ISE}_{\xi_n}(\eta_m) - \mathrm{ISE}_{\pi}(\eta_m)\right)^2\right] \\ &= \mathbb{E}\left[\left(\int_{\mathcal{D}_\mathbf{x}} \delta_m^2(\mathbf{x})\,\mathrm{d}(\xi_n - \pi)(\mathbf{x})\right)^2\right] \\ &= \mathbb{E}\left[\int_{\mathcal{D}_\mathbf{x}^2} \delta_m^2(\mathbf{x})\delta_m^2(\mathbf{x}')\,\mathrm{d}(\xi_n - \pi)(\mathbf{x})\mathrm{d}(\xi_n - \pi)(\mathbf{x}')\right] . \end{aligned}$$

Tonelli's theorem gives

$$\overline{\Delta}^2(\xi_n,\pi;\eta_m) = \int_{\mathcal{D}_\mathbf{x}^2} \mathbb{E}[\delta_m^2(\mathbf{x})\delta_m^2(\mathbf{x}')]\,\mathrm{d}(\xi_n - \pi)(\mathbf{x})\mathrm{d}(\xi_n - \pi)(\mathbf{x}') .$$

Since $\mathbb{E}\left[U^2V^2\right] = 2\,(\mathbb{E}[UV])^2 + \mathbb{E}\left[U^2\right]\mathbb{E}\left[V^2\right]$ for any one-dimensional normal-centered random variables U and V. The expression can then be written as:

$$\overline{\Delta}^2(\xi_n,\pi;\eta_m) = \int_{\mathcal{D}_{\mathbf{x}}^2} 2\,\mathbb{E}[\delta_m(\mathbf{x})\delta_m(\mathbf{x}')]^2 + \mathbb{E}[\delta_m^2(\mathbf{x})]\,\mathbb{E}[\delta_m^2(\mathbf{x}')]\,\mathrm{d}(\xi_n-\pi)(\mathbf{x})\mathrm{d}(\xi_n-\pi)(\mathbf{x}') \tag{5.4a}$$

$$\overline{\Delta}^2(\xi_n,\pi;\eta_m) = \int_{\mathcal{D}_{\mathbf{x}}^2} \overline{K}_{|m}(\mathbf{x},\mathbf{x}')\,\mathrm{d}(\xi_n-\pi)(\mathbf{x})\mathrm{d}(\xi_n-\pi)(\mathbf{x}') \tag{5.4b}$$

$$\overline{\Delta}^2(\xi_n,\pi;\eta_m) = \sigma^2\,\mathrm{MMD}^2_{\overline{K}_{|m}}(\xi_n,\pi). \tag{5.4c}$$

Interestingly, the last expression is equivalent to the maximum mean discrepancy (previously introduced in this manuscript and further defined in Appendix B) between π and ξ_n for a kernel $\overline{K}_{|m}(\cdot,\cdot)$. Note that σ^2 only appears as a multiplying factor in Eq. (5.4b), with the consequence that σ^2 does not impact the choice of a suitable ξ_n. The resulting kernel $\overline{K}_{|m}(\cdot,\cdot)$ is differently defined whether *(i)* the learning model $\eta_m(\mathbf{x})$ interpolates $\mathbf{y}_m$ or not *(ii)*:

$$\begin{cases} (i) \Rightarrow \overline{K}_{|m}(\mathbf{x},\mathbf{x}') & := 2\,K_{|m}^2(\mathbf{x},\mathbf{x}') + K_{|m}(\mathbf{x},\mathbf{x})K_{|m}(\mathbf{x}',\mathbf{x}') \\ (ii) \Rightarrow \overline{K}_{|m}(\mathbf{x},\mathbf{x}') & := 2\left[K_{|m}(\mathbf{x},\mathbf{x}') + 2\,\widehat{\delta}_m(\mathbf{x})\widehat{\delta}_m(\mathbf{x}')\right]K_{|m}(\mathbf{x},\mathbf{x}') \\ & \quad + \left[\widehat{\delta}_m^2(\mathbf{x}) + K_{|m}(\mathbf{x},\mathbf{x})\right]\left[\widehat{\delta}_m^2(\mathbf{x}') + K_{|m}(\mathbf{x}',\mathbf{x}')\right]. \end{cases} \tag{5.5}$$

The main idea is to replace ξ_n, uniform on $\mathbf{X}_n$, by a nonuniform measure ζ_n supported on $\mathbf{X}_n$, $\zeta_n = \sum_{i=1}^n w_i \delta_{\mathbf{x}^{(i)}}$ with weights $\mathbf{w}_n = (w_1,\ldots,w_n)^\top$ chosen such that the estimation error $\overline{\Delta}^2(\xi_n,\pi;\eta_m)$, and thus $\mathrm{MMD}^2_{\overline{K}_{|m}}(\zeta_n,\pi)$, is minimized. The squared MMD for the kernel $\overline{K}_{|m}$ between π and the weighted measure ζ_n can be expressed as:

$$\mathrm{MMD}^2_{\overline{K}_{|m}}(\zeta_n,\pi) = \varepsilon_{\overline{K}_{|m},\pi} - 2\,\mathbf{w}_n^\top P_{\overline{K}_{|m},\pi}(\mathbf{X}_n) + \mathbf{w}_n^\top \overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\mathbf{w}_n, \tag{5.6}$$

where $P_{\overline{K}_{|m},\pi}(\mathbf{X}_n)$ is the vector of potentials $\left[\int_{\mathcal{D}_{\mathbf{x}}} \overline{K}_{|m}(\mathbf{x},\mathbf{x}^{(1)})\mathrm{d}\pi(\mathbf{x}),\ldots,\int_{\mathcal{D}_{\mathbf{x}}} \overline{K}_{|m}(\mathbf{x},\mathbf{x}^{(n)})\mathrm{d}\pi(\mathbf{x})\right]^\top$, and $\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)$ is the $n\times n$ covariance matrix such that $\{\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\}_{i,j} = \overline{K}_{|m}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$, $\forall i,j = 1,\ldots,n$, and $\varepsilon_{\overline{K}_{|m},\pi} = \int_{\mathcal{D}_{\mathbf{x}}^2} \overline{K}_{|m}(\mathbf{x},\mathbf{x}')\,\mathrm{d}\pi(\mathbf{x})\mathrm{d}\pi(\mathbf{x}')$. The squared MMD defined in Eq. (5.6) is minimized for the following optimal weights $\mathbf{w}_n^*$:

$$\mathbf{w}_n^* = \overline{\mathbf{K}}_{|m}^{-1}(\mathbf{X}_n)\mathbf{p}_{\overline{K}_{|m},\pi}(\mathbf{X}_n)\,. \tag{5.7}$$

Therefore, an optimally weighted estimator of the predictivity coefficient supported on the test set $\mathbf{X}_n$ is defined as:

$$\widehat{Q}^2_{n*} = 1 - \frac{\sum_{i=1}^n w_i^* \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})\right]^2}{\frac{1}{n}\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_n\right]^2}, \tag{5.8}$$

with $\overline{y}_n = (1/n)\sum_{i=1}^n y(\mathbf{x}^{(m+i)})$. Notice that the weights w_i^* do not depend on the variance parameter σ^2 of the GP model. Moreover, this approach does not constrain the weights, which

works better in our experience than the different constrained versions (e.g., non-negativity, summing to one) studied in **?**.

Remark 7. The optimal estimator $\widehat{Q}^2_{n*}$ focused on the weighting numerator of the coefficient of determination defined in Eq. (5.2). However, the variance estimator on the denominator can also be optimally weighted. Let us write an alternative version of $\widehat{Q}^2_{n*}$

$$\widehat{Q}'^2_n = 1 - \frac{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})\right]^2}{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_m\right]^2}, \tag{5.9}$$

which compares the performance on the test set of η_m and $\overline{y}_m = (1/m)\sum_{i=1}^m y(\mathbf{x}^i)$. Using similar developments as previously it is possible to also apply a weighting procedure to the denominator of $\widehat{Q}'^2_n$, in order to make it resemble its integral version $V'_\pi(\mathbf{y}_m) = \int_{\mathcal{D}_\mathbf{x}} \left[y(\mathbf{x}) - \overline{y}_m\right]^2 \mathrm{d}\pi(\mathbf{x})$ (see **?**).

5.3 Test-set construction

In the previous section, the test set was assumed as given, and a method was proposed to estimate the $\mathrm{ISE}_\pi(\eta_m)$ (with the learning model η_m, built on $\mathbf{X}_m$) by an optimally weighted sum of the residuals. The objective in this section is to construct a test set of size n, denoted by $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\} \subset \mathcal{D}_\mathbf{x}$. To evaluate the predictivity of a model learned on a training set $\mathbf{X}_m$, a strategy is to place the test points the furthest away from the training set, to obtain a space-filling design when gathering leaning and test set. The sampling methods used for building the test set should then be space-filling and incremental. The advantage of using an iterative construction is that it can be stopped as soon as the predictivity estimation is considered sufficiently accurate In case the conclusion is that model predictions are not reliable enough, the full design $\mathbf{X}_{m+n} = \mathbf{X}_m \cup \mathbf{X}_n$ and the associated observations $\mathbf{y}_{m+n}$ can be used to update the model. This updated model can then be tested at additional design points, elements of a new test set to be constructed. This section introduces different space-filling methods, later compared for test set construction.

5.3.1 Fully-Sequential Space-Filling design

The Fully-Sequential Space-Filling forward-reflected (FSSF-fr) algorithm (**?**) relies on the CADEX algorithm (**?**) (also called the "coffee-house" method **?**). It constructs a sequence of nested designs in a bounded set $\mathcal{D}_\mathbf{x}$ by sequentially selecting a new point $\mathbf{x}$ as far away as possible from the $\mathbf{x}^{(i)}$ previously selected. New inserted points are selected within a set of candidates $\mathcal{S}$ which may coincide with $\mathcal{D}_\mathbf{x}$ or be a finite subset of $\mathcal{D}_\mathbf{x}$ (which simplifies the implementation, only this case is considered here). The improvement of FSSF-fr when compared to CADEX is that new points are selected *at the same time* far from the previous design points as well as far from the boundary of $\mathcal{D}_\mathbf{x}$.

The algorithm is as follows:

1. Choose $\mathcal{S}$, a finite set of candidate points in $\mathcal{D}_{\mathbf{x}}$, with size $N \gg n$ in order to allow a fairly dense covering of $\mathcal{D}_{\mathbf{x}}$. When $\mathcal{D}_{\mathbf{x}} = [0,1]^d$, **?** recommends taking $\mathcal{S}$ equal to the first $N = 1\,000\,d + 2\,n$ points of a Sobol sequence in $\mathcal{D}_{\mathbf{x}}$.

2. Choose the first point $\mathbf{x}^{(1)}$ randomly in $\mathcal{S}$ and define $\mathbf{X}_1 = \{\mathbf{x}^{(1)}\}$.

3. At iteration i, with $\mathbf{X}_i = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i)}\}$, select

$$\mathbf{x}^{(i+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i}{\arg\max} \left[\min \left(\min_{j \in \{1,\ldots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|, \sqrt{2}\, d\, \mathrm{dist}(\mathbf{x}, R(\mathbf{x})) \right) \right], \tag{5.10}$$

   where $R(\mathbf{x})$ is the symetric of $\mathbf{x}$ with respect to its nearest boundary of $\mathcal{D}_{\mathbf{x}}$, and set $\mathbf{X}_{i+1} = \mathbf{X}_i \bigcup \mathbf{x}^{(i+1)}$.

4. Stop the algorithm when $\mathbf{X}_n$ has the required size.

The role of the reflected point $R(\mathbf{x})$ is to avoid selecting $\mathbf{x}^{(i+1)}$ too close to the boundary of $\mathcal{D}_{\mathbf{x}}$, which is a major problem with standard coffee-house, especially when $\mathcal{D}_{\mathbf{x}} = [0,1]^d$ with d large. While the standard coffee-house (greedy packing) algorithm simply uses $\mathbf{x}^{(i+1)} \in \arg\max_{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i} \min_{j \in \{1,\ldots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|$. The factor $\sqrt{2}d$ in Eq. (5.10) proposed in **?** sets a balance between distance to the design $\mathbf{X}_i$ and distance to the boundary of $\mathcal{D}_{\mathbf{x}}$. Another scaling factor, depending on the target design size n is proposed in **?**.

FSSF-fr is entirely based on geometric considerations and implicitly assumes that the selected set of points should cover $\mathcal{D}_{\mathbf{x}}$ evenly.

However, in the context of uncertainty quantification, the distribution π of the model inputs is frequently not uniform. It is then desirable to select a test set representative of π. This can be achieved through the inverse transform of the CDF: FSSF-fr constructs $\mathbf{X}_n$ in the unit hypercube $[0,1]^d$, and an "isoprobabilistic" transform $T : [0,1]^d \to \mathcal{D}_{\mathbf{x}}$ is then applied to the points in $\mathbf{X}_i$, T being such that, if $\boldsymbol{U}$ is a random variable uniform on $[0,1]^d$, then $T(\boldsymbol{U})$ follows the target distribution π. The transformation can be applied to each input separately when π is the product of its marginals but is more complicated in other cases, see (**?**, Chap. 4). Note that FSSF-fr operates in the bounded set $[0,1]^d$ even if the support of π is unbounded. The other two algorithms presented in this section are able to directly choose points representative of a given distribution π and do not need to resort to such a transformation.

5.3.2 Support points

Support points (**?**) are such that their associated empirical distribution ξ_n has minimum Maximum-Mean-Discrepancy (MMD) with respect to π for the energy-distance kernel of Székely and Rizzo (**?**),

$$K_E(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \left(\|\mathbf{x}\| + \|\mathbf{x}'\| - \|\mathbf{x} - \mathbf{x}'\| \right). \tag{5.11}$$

The squared MMD between ξ_n and π for the distance kernel equals

$$\mathrm{MMD}^2_{K_E}(\xi_n, \pi) = \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{x}^{(i)} - \zeta\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| - \mathbb{E}\|\zeta - \zeta'\|, \tag{5.12}$$

where ζ and ζ' are independently distributed with π (see **?**). A key property of the energy-distance kernel is that it is characteristic (**?**), meaning that for any couple of probability distributions π and ξ on $\mathcal{D}_\mathbf{x}$, $\mathrm{MMD}^2_{K_E}(\pi, \xi)$ equals zero if and only if $\pi = \xi$. This MMD then defines a norm in the space of probability distributions. Compared to more heuristic methods for solving quantization problems, support points benefit from the theoretical guarantees of MMD minimization in terms of convergence of ξ_n to π as $n \to \infty$.

As $\mathbb{E}\|\mathbf{x}^{(i)} - \zeta\|$ is not known explicitly, in practice π is replaced by its empirical version π_N for a given large-size sample $(\mathbf{x}'^{(k)})_{k=1\ldots N}$. The support points $\mathbf{X}^s_n$ are then given by

$$\mathbf{X}^s_n \in \underset{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}}{\arg\min}\left(\frac{2}{nN}\sum_{i=1}^{n}\sum_{k=1}^{N}\|\mathbf{x}^{(i)} - \mathbf{x}'^{(k)}\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\right). \tag{5.13}$$

The function to be minimized can be written as a difference of functions convex in $\mathbf{X}_n$, which yields a difference-of-convex program. In **?**, a majorization-minimization procedure, efficiently combined with resampling, is applied to the construction of large designs (up to $n = 10^4$) in high dimensional spaces (up to $d = 500$). The examples treated clearly show that support points are distributed in a way that matches π more closely than Monte Carlo and quasi-Monte Carlo samples.

The method was used to split a dataset into a training set and a test set in **?**, where the N points $\mathbf{X}_N$ in Eq. (5.13) are those from the dataset. Then $\mathbf{X}^s_n$ gives the test set and the other $N - n$ points are used for training. There is a serious additional difficulty though, as choosing $\mathbf{X}^s_n$ among the dataset corresponds to a difficult combinatorial optimization problem. A possible solution is to perform the optimization in a continuous domain $\mathcal{D}_\mathbf{x}$ and then choose $\mathbf{X}^s_n$ that corresponds to the closest points in $\mathbf{X}_N$ (for the Euclidean distance) to the continuous solution obtained (**?**).

The direct determination of support points through Eq. (5.13) does not allow the construction of a nested sequence of test sets. One possibility would be to solve Eq. (5.13) sequentially, one point at a time, in a continuous domain, and then select the closest point within $\mathbf{X}_N$ as the one to be included in the test set. A different approach can be used, based on the greedy minimization of the MMD Eq. (5.12) for the candidate set $\mathcal{S} = \mathbf{X}_N$: at iteration i, the algorithm chooses

$$\mathbf{x}^s_{i+1} \in \underset{\mathbf{x}\in\mathcal{S}}{\arg\min}\left(\frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x} - \mathbf{x}'^{(k)}\| - \frac{1}{i+1}\sum_{j=1}^{i}\|\mathbf{x} - \mathbf{x}^{(j)}\|\right). \tag{5.14}$$

The method requires the computation of the $N(N-1)/2$ distances $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$, $i, j = 1, \ldots, N$, $i \neq j$, which hinders its applicability to large-scale problems (a test case with $N = 1\,000$ is

presented in Section 5.5). Note that **?** only studies the split of a given data set into a learning and test set while this chapter builds support points on the input space $\mathcal{D}_{\mathbf{x}}$

Greedy MMD minimization can be applied to other kernels than the distance kernel Eq. (5.11), see **??**. In the next section the closely related method of kernel herding is recalled (KH) (**?**), after its presentation in Chapter 4 of the present manuscript.

5.3.3 Kernel herding

As introduced in Section 5.3.3, **?** proposed a linearization of the MMD minimization using the Frank-Wolfe algorithm. Let us define a positive definite kernel K on $\mathcal{D}_{\mathbf{x}} \times \mathcal{D}_{\mathbf{x}}$, and consider $\xi_i = (1/i) \sum_{j=1}^{i} \delta_{\mathbf{x}^{(i)}}$ as the empirical measure for $\mathbf{X}_i$. In the sequential and uniformly weighted case, this iteration i of kernel herding is expressed as a difference of potentials:

$$\mathbf{x}_{i+1} \quad \in \quad \underset{\mathbf{x} \in \mathcal{S}}{\arg\min} \left[P_{\xi_i}(\mathbf{x}) - P_{\pi}(\mathbf{x}) \right], \tag{5.15}$$

with $\mathcal{S} \subseteq \mathcal{D}_{\mathbf{x}}$ a given candidate set and $P_{\xi_i}(\mathbf{x}) = (1/i) \sum_{j=1}^{i} K(\mathbf{x}, \mathbf{x}^{(j)})$.

Once the targeted measure substituted by an empirical measure π_N based on a sample $(\mathbf{x}'^{(k)})_{k=1\dots N}$ then complete estimation becomes: $P_{\pi_N}(\mathbf{x}) = (1/N) \sum_{k=1}^{N} K(\mathbf{x}, \mathbf{x}'^{(k)})$, which gives

$$\mathbf{x}_{i+1} \quad \in \quad \underset{\mathbf{x} \in \mathcal{S}}{\arg\min} \left[\frac{1}{i} \sum_{j=1}^{i} K(\mathbf{x}, \mathbf{x}^{(j)}) - \frac{1}{N} \sum_{k=1}^{N} K(\mathbf{x}, \mathbf{x}'^{(k)}) \right].$$

When K is the energy-distance kernel Eq. (5.11) the greedy support points from Eq. (5.14) are recovered with a factor $1/i$ instead of $1/(i+1)$ in the second sum.

The candidate set $\mathcal{S}$ in Eq. (5.15) is arbitrary and can be chosen as in Section 5.3.1. A neat advantage of kernel herding over support points is that the potential $P_{\pi}(\mathbf{x})$ is sometimes explicitly available. When $\mathcal{S} = \mathbf{X}_N$, this avoids the need to calculate the $N(N-1)/2$ distances $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$ and thus allows application to very large sample sizes. This is the case in particular when $\mathcal{D}_{\mathbf{x}}$ is the cross product of one-dimensional sets $\mathcal{X}_{x_i}$, $\mathcal{D}_{\mathbf{x}} = \mathcal{X}_{x_1} \times \cdots \times \mathcal{X}_{x_d}$, π is the product of its marginals $\pi_{[i]}$ on the $\mathcal{X}_{x_i}$, K is the product of one-dimensional kernels $K_{[i]}$, and the one-dimensional integral in $P_{\pi_{[i]}}(x)$ is known explicitly for each $i \in \{1, \dots, d\}$. Indeed, for $\mathbf{x} = (x_1, \dots, x_d) \in \mathcal{D}_{\mathbf{x}}$, $P_{\pi}(\mathbf{x}) = \prod_{i=1}^{d} P_{\pi_{[i]}}(x_i)$ (see **?**). When K is the product of Matérn kernels with regularity parameter $5/2$ and correlation lengths θ_i, $K(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} K_{5/2,\theta_i}(x_i - x'_i)$, with

$$K_{5/2,\theta}(x - x') = \left(1 + \frac{\sqrt{5}}{\theta}|x - x'| + \frac{5}{3\theta^2}(x - x')^2\right) \exp\left(-\frac{\sqrt{5}}{\theta}|x - x'|\right), \tag{5.16}$$

the one-dimensional potentials are given in Appendix B for $\pi_{[i]}$ uniform on $[0,1]$ or $\pi_{[i]}$ the standard normal $\mathcal{N}(0,1)$. When no observation is available, which is the common situation at the design stage, the correlation lengths have to be set to heuristic values. The values of the correlation lengths empirically show a significant influence over the design. A reasonable choice for $\mathcal{D}_{\mathbf{x}} = [0,1]^d$ is $\theta_i = n^{-1/d}$ for all i, with n the target number of design points (see **?**).

5.3.4 Numerical illustration

As a first numerical illustration, the FSSF-fr (denoted FSSF in the following), support points and kernel herding algorithms were applied in the situation where a given initial design of size m has to be completed by a series of additional points $\mathbf{x}^{(m+1)}, \ldots, \mathbf{x}^{(m+n)}$. The objective is to obtain a full design $\mathbf{X}_{m+n}$ that is a good quantization of a given distribution π.

Figures 5.1 and 5.2 correspond to π uniform on $[0,1]^2$ and π the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, with $\mathbf{I}_2$ the 2-dimensional identity matrix, respectively. All methods are applied to the same candidate set $\mathcal{S}$.

The initial designs $\mathbf{X}_m$ are chosen in the class of space-filling designs, well suited to initialize sequential learning strategies (**?**). When π is uniform, the initial design is a maximin Latin hypercube design (introduced in Section 1.4.2) with $m = 10$ and the candidate set is given by the $N = 2^{12}$ first points S_N of a Sobol sequence in $[0,1]$. When π is normal, the inverse probability transform method is first applied to S_N and $\mathbf{X}_m$ (this does not raise any difficulty here as π is the product of its marginals). The candidate points $\mathcal{S}$ are marked in gray on Fig. 5.1 and Fig. 5.2 and the initial design is indicated by the red crosses. The index i of each added test point $\mathbf{x}^{(m+i)}$ is indicated (the font size decreases with i). In such a small dimension ($d = 2$), a visual appreciation gives the impression that the three methods have comparable performance. However, FSSF tends to choose points closer to the boundary of $\mathcal{S}$ than the other two, and the support points seem to sample more freely the holes of $\mathbf{X}_m$ than kernel herding, which seems to be closer to a space-filling continuation of the training set. In the next section, these designs are used for estimating the quality of the predictivity metric.

5.4 Numerical results I: construction of a training set and a test set

This section presents numerical results obtained on three different test cases, in dimension 2 (test cases 1 and 2) and 8 (test case 3), for which $y(\mathbf{x}) = f(\mathbf{x})$ with $f(\mathbf{x})$ has an easy to evaluate analytical expression, see Section 5.4.1. This allows a good estimation of Q_π^2 (see Eq. (5.1)) by a large Monte Carlo sample (with size $M = 10^6$), which will serve as a reference when assessing the performance of each of the other estimators.

The validation designs are built by FSSF, support points and kernel herding, presented in Sections 5.3.1, 5.3.2, and 5.3.3, and the performances obtained are compared for each one, considering the uniform and the weighted estimator of Section 5.2.2.

5.4.1 Test cases

The training design $\mathbf{X}_m$ and the set $\mathcal{S}$ of potential test set points are as in Section 5.3.4. For test cases 1 and 3, π is the uniform measure on $\mathcal{D}_\mathbf{x} = [0,1]^d$, with $d = 2$ and $d = 8$, respectively; $\mathbf{X}_m$ is a maximin Latin hypercube design in $\mathcal{D}_\mathbf{x}$, and $\mathcal{S}$ corresponds to the first N points S_N

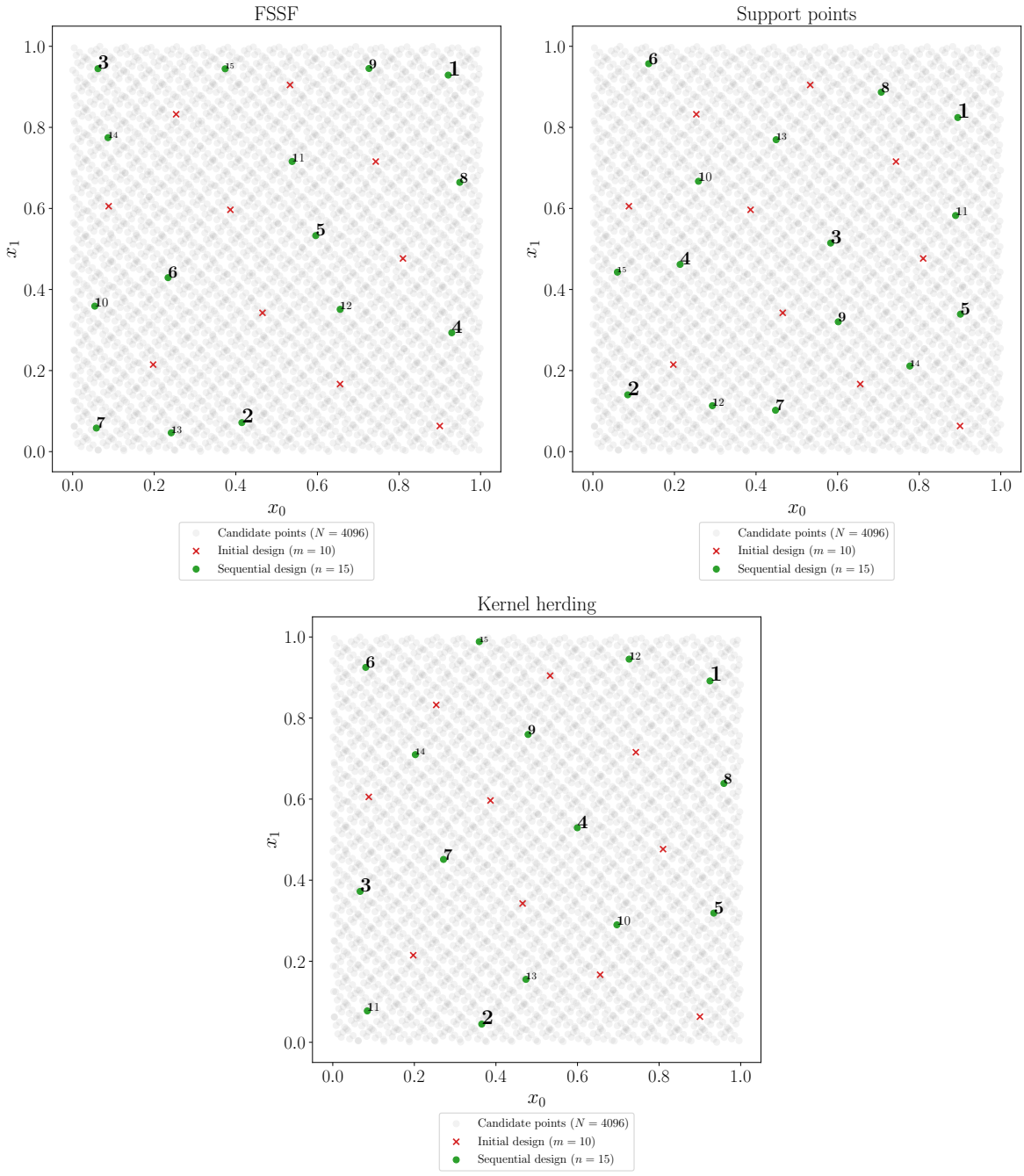


Figure 5.1 Additional points (ordered, green) complementing an initial design (red crosses), π is uniform on $[0,1]$, the candidate points are in gray.

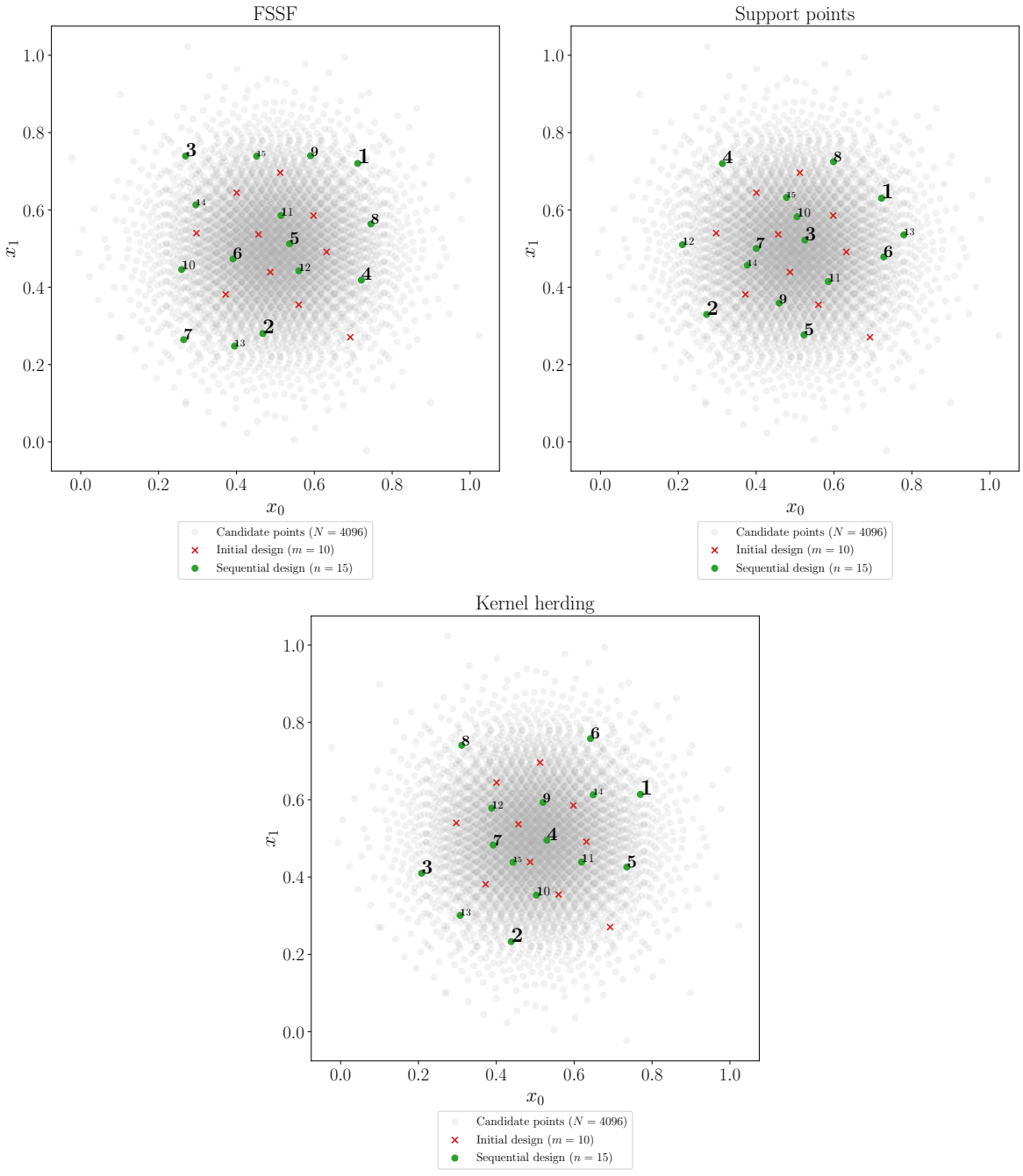


Figure 5.2 Additional points (ordered, green) complementing an initial design (red crosses), π normal, the candidate points are in gray.

of Sobol' sequence in $\mathcal{D}_{\mathbf{x}}$, complemented by the 2^d vertices. In the second test case, $d = 2$, π is the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, and the sets $\mathbf{X}_m$ and $\mathbf{S}_N$ must be transformed as explained in section 5.3.1. There are $N = 2^{14}$ candidate points for test cases 1 and 2 and $N = 2^{15}$ for test case 3 (this value is rather moderate for a problem in dimension 8, but using a larger N yields numerical difficulties for support points; see Section 5.3.2).

For each test case, a GP regression model is fitted to the m observations using ordinary Kriging (**?**) (a GP model with constant mean), with an anisotropic Matérn kernel with regularity parameter $5/2$, and the correlation lengths θ_i are estimated by maximum likelihood via a truncated Newton algorithm. All calculations were done using the Python package OpenTURNS for uncertainty quantification (**?**). The kernel used for kernel herding is different and corresponds to the tensor product of one-dimensional Matérn kernels Eq. (5.16), so that the potentials $P_\pi(\cdot)$ are known explicitly (see Appendix B); the correlations lengths are set to $\theta = 0.2$ in test cases 1 and 3 ($d = 2$) and to $\theta = 0.7$ in test case 3 ($d = 8$).

Assuming that a model is classified, in terms of the estimated value of its predictivity index Q^2 as "poor fitting" if $Q^2 \in [0.6, 0.8]$, "reasonably good fitting", when $Q^2 \in (0.8, 0.9]$, and "very good fitting" if $Q^2 > 0.9$, for each test case, three different sizes m of the training set are selected such that the corresponding models cover all three possible situations. For all test cases, the impact of the size n of the test set is studied in the range $n \in \{4, \ldots, 50\}$.

Test case 1. This test function is $f_1(\mathbf{x}) = h(2x_1 - 1, 2x_2 - 1)$, $(x_1, x_2) \in \mathcal{D}_{\mathbf{x}} = [0,1]^2$, with

$$h(u_1, u_2) = \frac{\exp(u_1)}{5} - \frac{u_2}{5} + \frac{u_2^6}{3} + 4u_2^4 - 4u_2^2 + \frac{7u_1^2}{10} + u_1^4 + \frac{3}{4u_1^2 + 4u_2^2 + 1}.$$

Color-coded 3d and contour plots of f_1 for $\mathbf{X} \in \mathcal{D}_{\mathbf{x}}$ are shown on the left panel of Fig. 5.3, showing that the function is rather smooth, even if its behavior along the boundaries of $\mathcal{D}_{\mathbf{x}}$, in particular close to the vertices, may present difficulties for some regression methods. The size of the training set for this function is $m \in \{5, 15, 30\}$.

test case 2. The second test function, plotted in the right panel of Fig. 5.3 for $\mathbf{x} \in [0,1]^2$, is

$$f_2(\mathbf{x}) = \cos\left(5 + \frac{3}{2}x_1\right) + \sin\left(5 + \frac{3}{2}x_1\right) + \frac{1}{100}\left(5 + \frac{3}{2}x_1\right)\left(5 + \frac{3}{2}x_2\right).$$

Training set sizes for this test case are $m \in \{8, 15, 30\}$.

test case 3. The third function is the so-called "gSobol" function, defined over $\mathcal{D}_{\mathbf{x}} = [0,1]^8$ by

$$f_3(\mathbf{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + a_i}{1 + a_i}, \qquad a_i = i^2.$$

This parametric function is very versatile as both the dimension of its input space and the coefficients a_i can be freely chosen. The sensitivity to input variables is determined by the a_i:

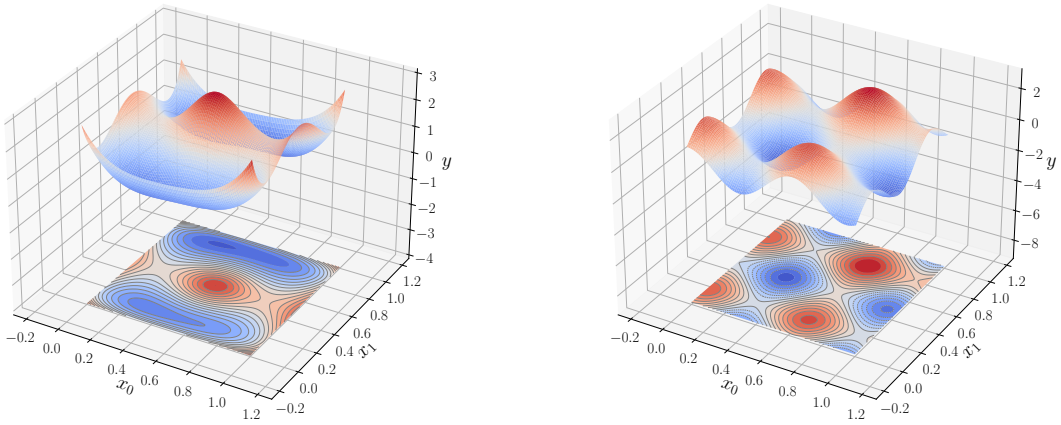


Figure 5.3 Left: $f_1(\mathbf{x})$ (test case 1); right: $f_2(\mathbf{x})$ (test case 2); $\mathbf{x} \in \mathcal{D}_{\mathbf{x}} = [0,1]^2$.

the larger a_i is, the less f is sensitive to x_i. Larger training sets are considered for this test case: $m \in \{15, 30, 100\}$.

5.4.2 Results and analysis

The numerical results obtained in this section are presented in Figures 5.4, 5.5, and 5.6. Each figure corresponds to one of the test cases and gathers three sub-figures, corresponding to test sets with sizes m yielding poor (left), reasonably good (right) or very good (bottom) fittings.

The baseline value of Q^2_{MC}, calculated with 10^6 Monte Carlo points, is indicated by the black diamonds (the black horizontal lines). Assuming that the error of Q^2_{MC} is much smaller than the errors of all other estimators, this section compares the ability of the methods to approximate Q^2_{MC}. For each sequence of nested test sets ($n \in \{4, \ldots, 50\}$), the observed values of $\widehat{Q}^2_n$ (equation Eq. (5.2)) and Q^2_{n*} (equation Eq. (5.8)), are plotted as the solid and dashed lines, respectively.

The figures also show the value Q^2_{LOO} obtained by Leave-One-Out (LOO) cross-validation, which is indicated at the left of each figure by a red diamond (values smaller than 0.25 are not shown). Note that, contrarily to the other methods considered, for LOO the test set is not disjoint from the training set, and thus the method does not satisfy the conditions set in the Introduction. As the complete model-fitting procedure is repeated for each training sample of size $m-1$, including the maximum-likelihood estimation of the correlation lengths of the Matérn kernel, the closed-form expressions of **?** cannot be used, making the computations rather intensive. The three figures show, and as expected, that the Q^2_{LOO} tends to under-estimate Q^2_{ideal}: by construction of the training set, LOO cross-validation relies on model predictions at points $\mathbf{x}^{(i)}$ far from the other $m-1$ design points used to build the model, and thus tends to systematically overestimate the prediction error at $\mathbf{x}^{(i)}$. The underestimation of Q^2_{ideal} can

be particularly severe when m is small, the training set is then necessarily sparse; see Fig. 5.4 where $Q^2_{LOO} < 0.3$ for $m = 5$ and 15.

Let us first concentrate on the non-weighted estimators (solid curves). The two MMD-based constructions, support points (in orange) and kernel herding (in blue), generally produce better validation designs than FSSF (green curves), leading to values of $\widehat{Q}^2_n$ that approach Q^2_{ideal} quicker as n increases. This is particularly noticeable for "good" and "very good" models (central and rightmost panels of all three figures). This supports the idea that test sets should complement the training set $\mathbf{X}_m$ by populating the holes it leaves in $\mathcal{D}_{\mathbf{x}}$ while at the same time being able to mimic the target distribution π, this second objective being more difficult to achieve for FSSF than for the MMD-based constructions.

A comparison of the two MMD-based estimators reveals that support points tend to underestimate ISE, leading to an over-confident assessment of the model predictivity, while kernel herding displays the expected behavior, with a negative bias that decreases with n. The reason for the positive bias of estimates based on support points designs is not fully understood, but may be linked to the fact that support points tend to place validation points at "mid-range" from the designs (and not at the furthest points like FSSF or kernel herding), see central and rightmost panels in Figure 5.1, and thus residuals at these points are themselves already better representatives of the local average errors.

Let us consider now the impact of the GP-based weighting of the residuals when estimating Q^2 (by Q^2_{n*}), which is related to the relative training-set/validation-set geometry (the manner in which the two designs are entangled in ambient space). The improvement resulting from applying residual weighting is apparent on all panels of the three figures, the dashed curves lying closer to Q^2_{ideal} than their solid counterparts; see in particular kernel herding (blue curve) in Fig. 5.4 and FSSF (green curve) in Fig. 5.5. Unexpectedly, the estimators based on support points seem to be rather insensitive to residual weighting, the dashed and solid orange curves being most of the time close to each other (and in any case, much closer than the green and blue ones). While the reason for this behavior deserves a deeper study, the fact that the support point designs – see Figure 5.1 – sample in a better manner the range of possible training-to-validation distances, being in some sense less space-filling than both FSSF and kernel herding, is again a plausible explanation for this weaker sensitivity to residual weighting.

Consider now a comparison of the behavior across test cases. Setting aside the strikingly singular situation of test case 2, for which kernel herding displays a pathological (bad) behavior for the "very good" model, and all methods present an overall good behavior, the details of the tested function do not seem to play an important role concerning the relative merits of the estimators and validation designs.

Let us finally observe how the methods behave for models of distinct quality (m leading to poor, good or very good models), comparing the three panels in each figure. On the left panels, m is too small for the model η_m to be accurate, and all methods and test-set sizes are able to detect this. For models of practical interest (good and very good), the test sets generated with support points and kernel herding allow a reasonably accurate estimation of Q^2 with a few

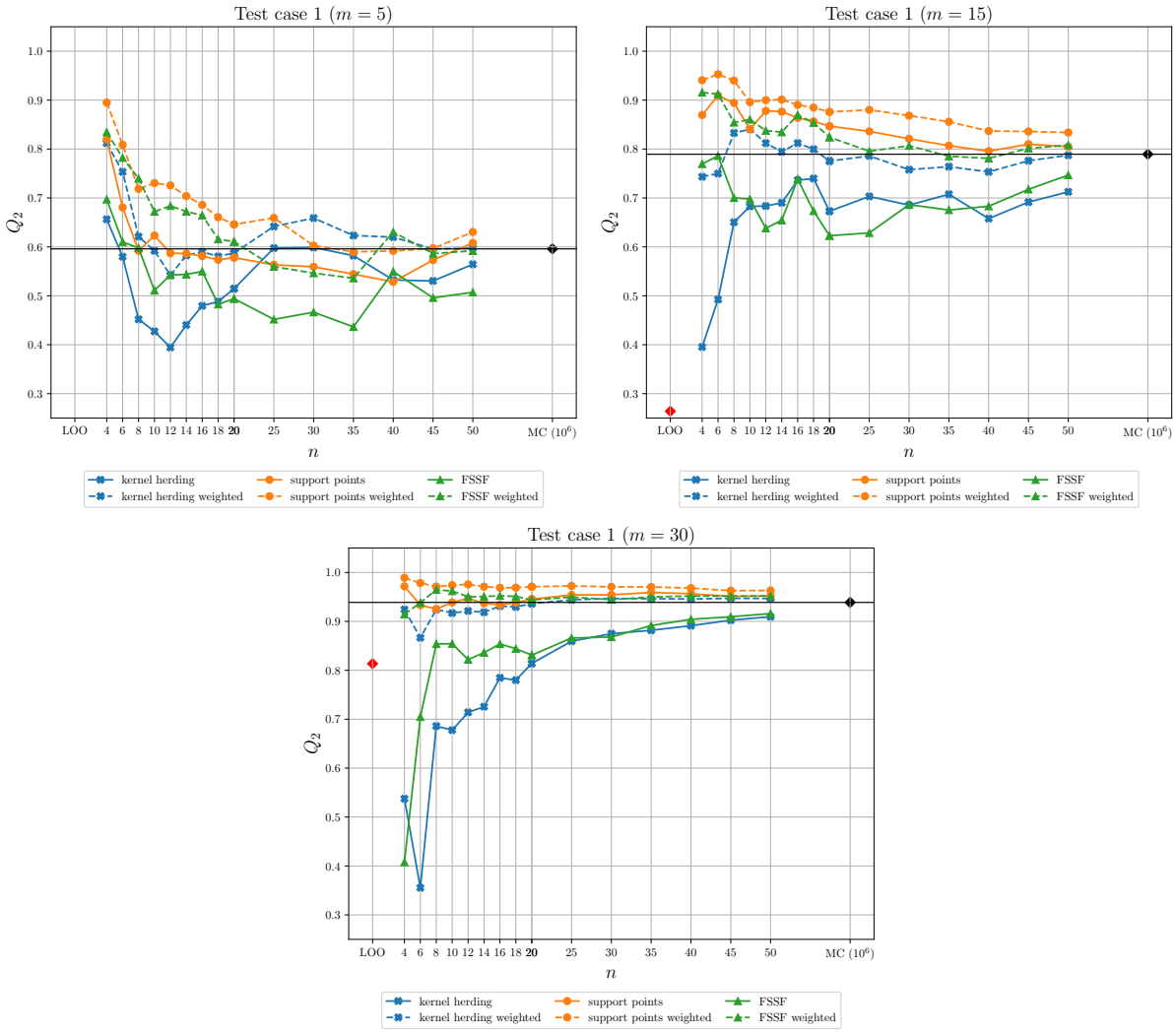


Figure 5.4 test case 1: predictivity assessment of a poor (left), good (right) and very good (bottom) model with kernel herding, support points and FSSF test sets.

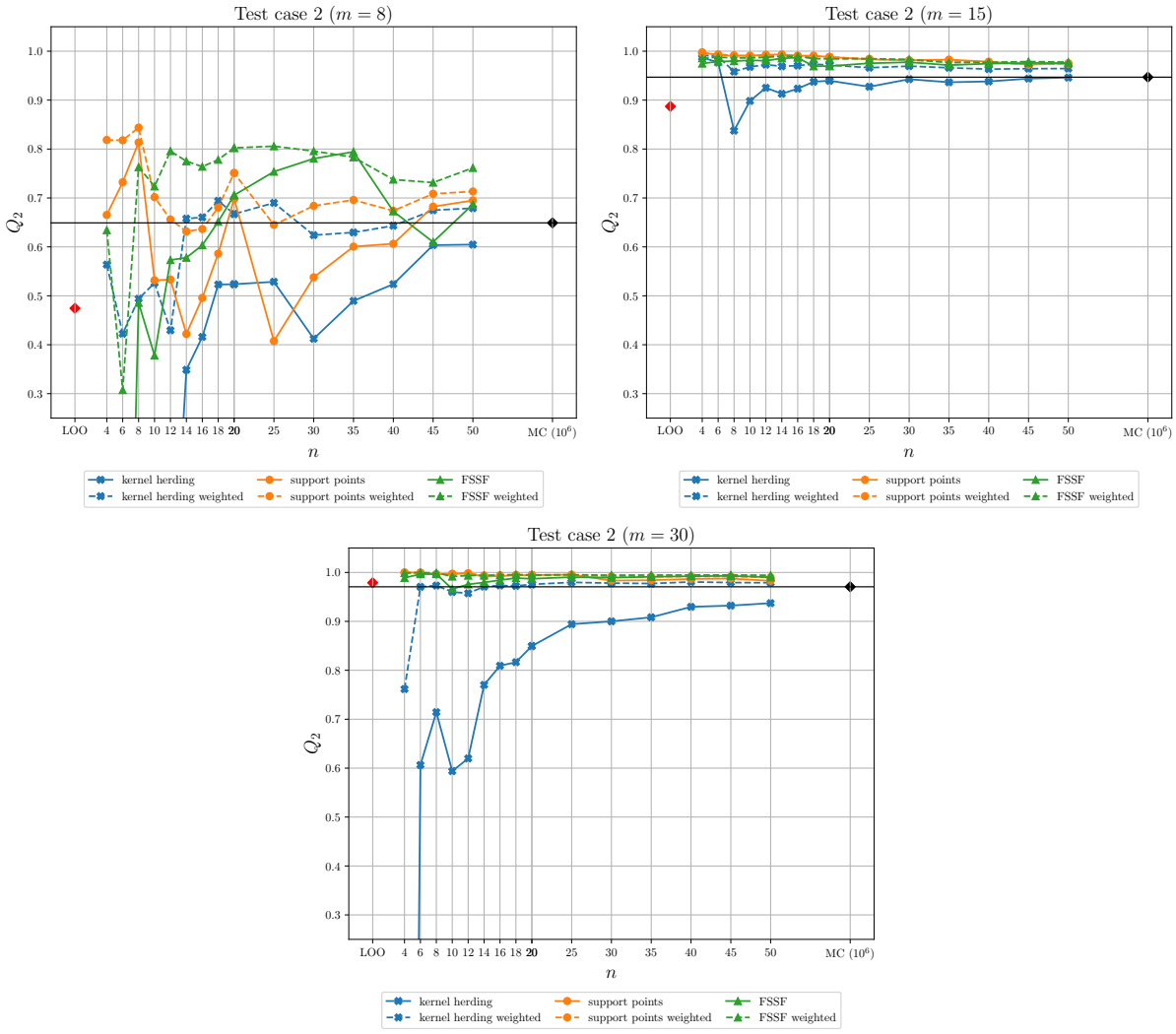


Figure 5.5 test case 2: predictivity assessment of a poor (left), good (right) and very good (bottom) model with kernel herding, support points and FSSF test sets.

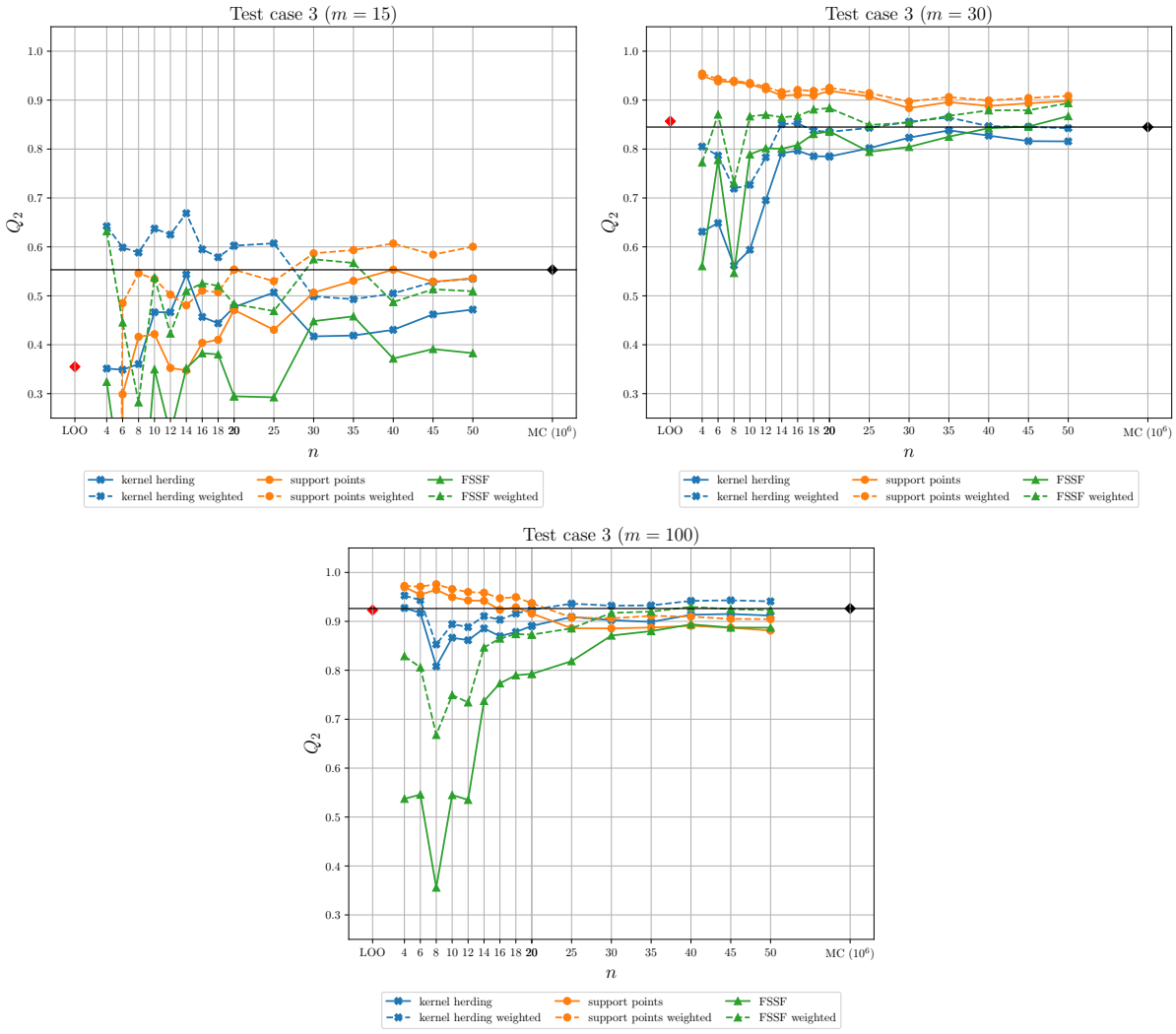


Figure 5.6 test case 3: predictivity assessment of a poor (left), good (right) and very good (bottom) model with kernel herding, support points and FSSF test sets.

points. Note, incidentally, that except for test case 2 (where the interplay with a non-uniform measure π complicates the analysis), it is in general easier to estimate the quality of the very good model (right-most panel) than that of the good model (central panel), indicating that the expected complexity (the entropy) of the residual process should be a key factor determining how large the validation set must be. In particular, it may be that larger values of m allow for smaller values of n.

5.5 Numerical results II: splitting a dataset into a training set and a test set

This section illustrates the performance of the different designs and estimators considered in this chapter when applied in the context of an industrial application, to split a given dataset of size N into training and test sets, with m and n points respectively, $m+n=N$. In contrast with **?**, the observations $y(\mathbf{x}^{(i)})$, $i=1,\dots,N$, are not used in the splitting mechanism, meaning that it can be performed before the observations are collected and that there cannot be any selection bias related to observations (indeed, the use of observation values in an MMD-based splitting criterion may favor the allocation of the most different observations to different sets, training versus validation).

An ML model is fitted to the training data, and the data collected on the test set are used to assess the predictivity of the model. The influence of the ratio $r_n = n/N = 1-m/N$ on the quality assessment is investigated. Random Cross-Validation (RCV) is also considered, where n points are chosen at random among the N points of the dataset: for each n, there are $\binom{N}{n}$ possible choices, and $R=1\,000$ designs were randomly selected among them. A model is fitted on the m complementary points ($m=N-n$), which yields an empirical distribution of Q^2 values for each ratio n/N considered.

5.5.1 Industrial test case CATHARE

The test case corresponds to the computer code CATHARE2 (for "Code Avancé de ThermoHydraulique pour les Accidents de Réacteurs à Eau"), which models the thermal-hydraulic behavior inside nuclear pressurized water reactors (**?**). The studied scenario simulates a hypothetical large-break loss of primary coolant accident for which the output of interest is the peak cladding temperature (**??**). The complexity of this application lies in the large run-time of the computer model (of the order of twenty minutes) and in the high dimension of the input space: the model involves 53 input parameters z_i, corresponding mostly to constants of physical laws, but also coding initial conditions, material properties and geometrical modeling. The z_i were independently sampled according to normal or log-normal distributions. These characteristics make this test case challenging in terms of construction of a surrogate model and validation of its predictivity.

In the following, an existing Monte Carlo sample $\mathbf{Z}_N$ of $N = 1\,000$ points in $\mathbb{R}^{53}$ is used, that corresponds to 53 independent random input configurations (see **?** for details). The output of the CATHARE2 code at these N points is also available. To reduce the dimensionality of this dataset, a sensitivity analysis (**?**) screens the inputs that do not impact the output significantly. This dimension-reduction step relies on the Hilbert-Schmidt Independence Criterion (HSIC), which is known as a powerful tool to perform input screening from a single sample of inputs and output values without reference to any specific ML regression model (**?**). HSIC-based statistical tests and their associated p-values are used to identify (with a 5%-threshold) inputs on which the output is significantly dependent (and therefore, also those of little influence). They were successfully applied to similar datasets from thermal-hydraulic applications in **??**. The screened dataset only includes 10 influential inputs, over which the candidate set $\mathbf{X}_N$ used for the construction of the test-set $\mathbf{X}_n$ (and therefore of the complementary training set $\mathbf{X}_{N-n}$) is defined. The marginal distributions are shown as histograms along the axes of the plots.

To include RCV in the methods to be compared, many (here, $R = 1\,000$) different models η_m must be constructed for each considered design size m. Since Gaussian Process regression proved to be too expensive for this purpose, the comparatively cheaper Partial Least Squares (PLS) method (**?**) is used. For each given training set, the model obtained is a sum of monomials in the 10 input variables. Note that models constructed with different training sets may involve different monomials and have different numbers of monomial terms.

5.5.2 Benchmark results and analysis

Fig. 5.7 compares various ways of extracting an n-point test set from an N-point dataset to estimate model predictivity, for different splitting ratios $n/N \in \{0.1, 0.15, 0.2, \ldots, 0.9\}$.

Consider RCV first. For each value of $r_n = n/N$, the empirical distribution of Q^2_{RCV} obtained from $R = 10^3$ random splittings of $\mathbf{X}_N$ into $\mathbf{X}_m \cup \mathbf{X}_n$ is summarized by a boxplot. Depending on r_n, three behaviors are roughly distinguished. For $0.1 \leq r_n \lesssim 0.3$ the distribution is bi-modal, with the lower mode corresponding to unlucky test-set selections leading to poor performance evaluations. When $0.3 \lesssim n/N \lesssim 0.7$, the distribution looks uni-modal, revealing a more stable performance evaluation. Note that this is (partly) in line with the recommendations discussed in section 5.1. For $r_n \gtrsim 0.7$, the variance of the distribution increases with r_n: many unlucky training sets lead to poor models. Note that the median of the empirical distribution slowly decreases as r_n increases, which is consistent with the intuition that the model predictivity should decrease when the size of the training set decreases.

For completeness, the red diamond represented on the left of Fig. 5.7 the value of Q^2_{LOO} computed by LOO cross-validation. In principle, being computed using the entire dataset, this value should establish an upper bound on the quality of models computed with smaller training sets. This is indeed the case for small training sets (rightmost values in the figure), for which the predictivity estimated by LOO is above the majority of the predictivity indexes calculated. But at the same time, LOO cross-validation tends to overestimate the errors, which explains the higher predictivity estimated by some other methods when $m = N - n$ is large enough.

Compare now the behavior of the two MMD-based algorithms of Section 5.3, $\widehat{Q}^2_n$ (unweighted) and Q^2_{n*} (weighted) are plotted using solid and dashed lines, respectively, for both kernel herding (in blue) and support points (in orange). FSSF test sets are not considered, as the application of an iso-probabilistic transformation imposes knowledge of the input distribution, which is not known for this example. Compare first the unweighted versions of the two MMD-based estimators. For small values of the ratio r_n, $0.1 \lesssim r_n \lesssim 0.45$, the relative behavior of support points and kernel herding coincides with what was observed in the previous section, support points (solid orange line) estimating a better performance than kernel herding (solid blue line), which, moreover, is close to the median of the empirical distribution of Q^2_{RCV}. However, for $r_n \geq 0.5$, the dominance is reversed, support points estimating a worse performance than kernel herding.

As r_n increases up to $r_n \lesssim 0.7$ the solid orange and blue curves crossover, and it is now $\widehat{Q}^2_n$ for kernel herding that approximates the RCV empirical median, while the value obtained with support points underestimates the predictivity index. Also, note that for (irrealistic) very large values of r_n both support points and kernel herding estimate lower Q^2 values, which are smaller than the median of the RCV estimates.

Let us now focus on the effect of residual weighting, i.e., in estimators Q^2_{n*} which use the weights computed by the method of Section 5.2.2, shown in dashed lines in Figure 5.7. First, note that while kernel herding weighting leads, as in the previous section, to higher estimates of the predictivity (compare solid and dashed blue lines), this is not the case for support points (solid and dashed orange curves), which, for small split ratios, produces smaller estimates when weighting is introduced. In the large r_n region, the behavior is consistent with what was previously presented, weighting inducing an increase of the estimated predictivity. It is remarkable – and rather surprising – that Q^2_{n*} for support points (the dashed orange line) does not present the discontinuity of the uncorrected curve.

The sum $\sum_{i=1}^{n} w_i^*$ of the optimal weights of support points and kernel herding Eq. (5.7) is shown in Fig. 5.8 (orange and blue curves, respectively). The slow increase with n/N of the sum of kernel-herding weights (blue line) is consistent with the increase of the volume of the input region around each validation point when the size of the training set decreases. The behavior of the sum of weights is more difficult to interpret for support points (orange line) but is consistent with the behavior of Q^2_{n*} on Fig. 5.7. Note that the energy-distance kernel Eq. (5.11) used for support points cannot be used for the weighting method of Section 5.2.2 as K_E is not positive definite but only conditionally positive definite. A full understanding of the observed curves would require a deeper analysis of the geometric characteristics of the designs generated by the two MMD methods, in particular of their interleaving with the training designs, which is not compatible with the space constraints of this manuscript.

While a number of unanswered points remain, in particular how deeply the behaviors observed may be affected by the poor predictivity resulting from the chosen PLS modeling methodology, the example presented in this section shows that the construction of test sets via MMD minimization and estimation of the predictivity index using the weighted estimator Q^2_{n*}

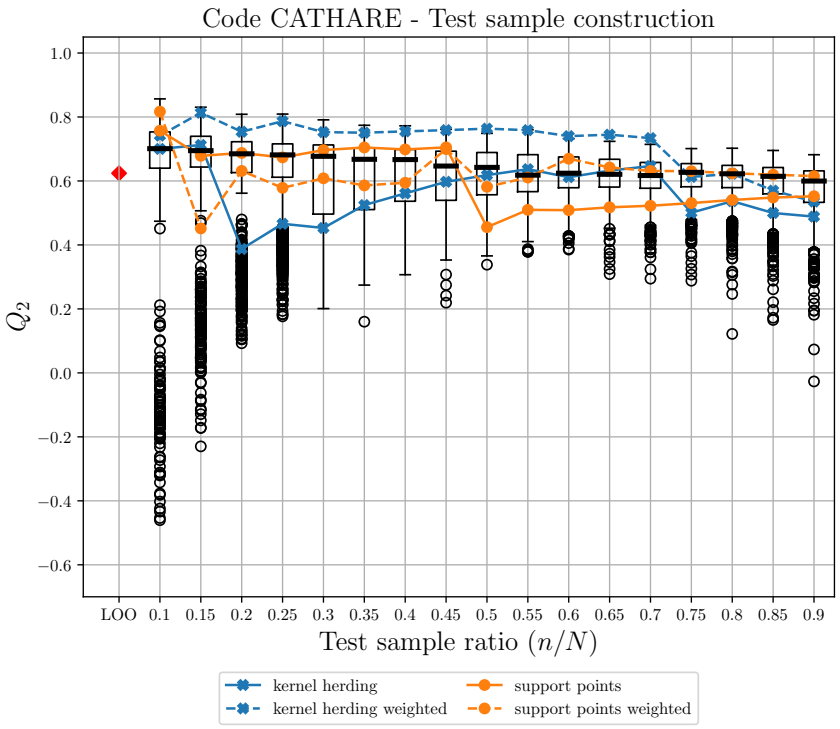


Figure 5.7 test case CATHARE: estimated Q^2. The box plots are for random cross-validation, and the red diamond (left) is for Q^2_{LOO}.

is promising as an efficient alternative to RCV: at a much lower computational cost, it builds performance estimates based on independent data the model developers may not have access to. Moreover, kernel herding proved, in the examples studied in this manuscript, to be a more reliable option for designing the test set, exhibiting a behavior that is consistent with what is expected, and very good estimation quality when the residuals over the design points are appropriately weighted.

5.6 Conclusion

Our study shows that ideas and tools from the design of experiment framework can be transposed to the problem of test-set selection. This chapter explored approaches based on support points, kernel herding and FSSF, considering the incremental construction of a test set (*i*) either as a particular space-filling design problem, where design points should populate the holes left in the design space by the training set, or (*i*) from the point of view of partitioning a given dataset into a training set and a test set.

A numerical benchmark has been performed for a panel of test cases of different dimensions and complexity. Additionally to the usual predictivity coefficient, a new weighted metric (see ?) has been proposed and shown to improve the assessment of the predictivity of a given model for a given test set.

This weighting procedure appears very efficient for interpolators, like Gaussian process regression models, as it corrects the bias when the points in the test set used to predict the

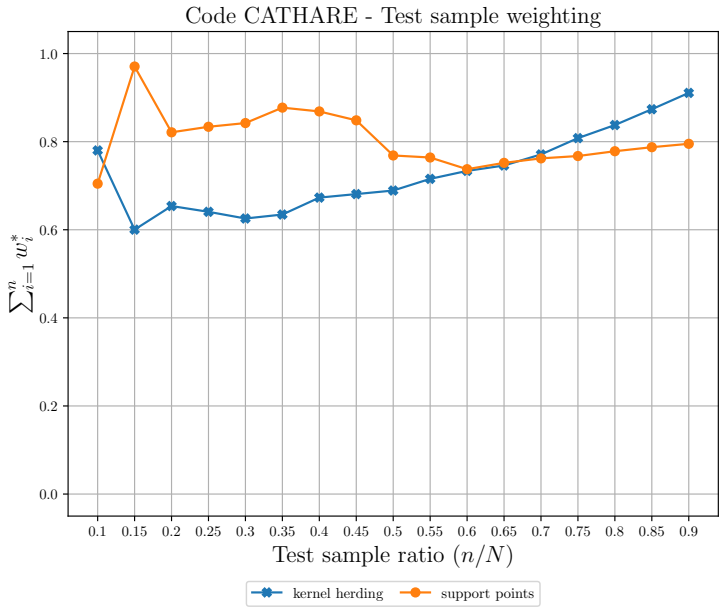


Figure 5.8 test case CATHARE: sum of the weights Eq. (5.7).

errors are far from the training points. For the first three test cases (Section 5.4), pairing one iterative design method with the weight-corrected estimator of the predictivity coefficient Q^2 shows promising results as the estimated Q^2 characteristic is close to the true one even for test-sets of moderate size.

Weighting can also be applied to models that do not interpolate the training data. For the industrial test case of Section 5.5, the true Q^2 value is unknown, but the weight-corrected estimation Q^2_{n*} of Q^2 is close to the value estimated by Leave-One-Out cross-validation and to the median of the empirical distribution of Q^2 values obtained by random k-fold cross-validation. At the same time, estimation by Q^2_{n*} involves a much smaller computational cost than cross-validation methods and uses a dataset fully independent of the one used to construct the model.

To each of the design methods considered to select a test set a downside can be attached. FSSF requires knowledge of the input distribution to be able to apply an iso-probabilistic transformation if necessary; it tends to select many points along the boundary of the candidate set considered. Support points require the computation of the $N(N-1)/2$ distances between all pairs of candidate points, which implies important memory requirements for large N; the energy-distance kernel on which the method relies cannot be used for the weighting procedure. Finally, the efficient implementation of kernel herding relies on analytical expressions for the potentials P_π, see Appendices A and B, which are available for particular distributions (like the uniform and the normal) and kernels (like Matérn) only. The great freedom in the choice of the kernel K gives a lot of flexibility, but at the same time implies that some non-trivial decisions have to be made; also, the internal parameters of K, such as its correlation lengths, must be specified. Future work should go beyond empirical rules of thumb and study the influence of these choices.

Numerical tests were only computed with independent inputs. Kernel herding and support points are both well suited for probability measures not being equal to the product of their marginals, which is a frequent case in real datasets. Note only incremental constructions were considered, as they allow to stop the validation procedure as soon as the estimation of the model predictivity is deemed sufficiently accurate, but it is also possible to select several points at once, using support points (**?**), or MMD minimization in general (**?**).

Further developments around this work could be as follows. Firstly, the incremental construction of a test set could be coupled with the definition of an appropriate stopping rule, in order to decide when it is necessary to continue improving the model (possibly by supplementing the initial design with the test set, which seems well suited to this). The $\mathrm{MMD}_{\overline{K}_{|m}}(\zeta_n^*, \pi)$ of Section 5.2.2 could play an important role in the derivation of such a rule. Secondly, the approach presented gives equal importance to all the d inputs. However, it seems that inputs with a negligible influence on the output should receive less attention when selecting a test set. A preliminary screening step that identifies the important inputs would allow the test-set selection algorithm to be applied to these variables only. For example, when a $\mathbf{X}_N \subset \mathbb{R}^d$ dataset is to be partitioned into $\mathbf{X}_m \cup \mathbf{X}_n$, one could use only $d' < d$ components to define the partition, but still use all d components to build the model and estimate its (weighted) Q^2. Note, however, that this would imply a slight violation of the conditions mentioned in the introduction, as it renders the test set dependent on the function observations.

Finally, in some cases, the probability measure π is known up to a normalizing constant. The use of a Stein kernel then makes the potential $P_{K,\pi}$ identically zero (**?**), which would facilitate the application of kernel herding. Also, more complex problems involve functional inputs, like temporal signals, images, or categorical variables; the application of the methods presented to kernels specifically designed for such situations raises challenging issues.

Part III:

Contributions to rare event estimation

La résignation est un suicide quotidien.

H. Balzac

Chapter 6

Nonparametric rare event estimation

6.1 Introduction

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

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

The present work proposes a new rare event estimation method, adopting the same sequential structure as SS while using a strictly different sampling mechanism to generate conditional

samples. This method intends to fit the intermediary conditional distributions with a nonparametric tool called the *Empirical Bernstein Copula*. Contrarily to SS, the proposed method named "Bernstein adaptive nonparametric conditional sampling" (BANCS), generates i.i.d. samples of the intermediary conditional distributions. For instance, a practical use of such i.i.d. samples can be to estimate dedicated reliability-oriented sensitivity indices (see, e.g., Chabridon et al. (2021); **?**).

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

6.1.1 Background

Subset sampling

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

$$p_{\mathrm{f}} = \mathbb{P}(\mathcal{F}_{\mathbf{x}}) = \mathbb{P}(\cap_{k=1}^{k_\#} \mathcal{F}_{[k]}) = \prod_{k=1}^{k_\#} \mathbb{P}(\mathcal{F}_{[k]} | \mathcal{F}_{[k-1]}). \tag{6.1}$$

From a practical point of view, the analyst tunes the algorithm by setting the intermediary probabilities $\mathbb{P}(\mathcal{F}_{[k]}|\mathcal{F}_{[k-1]}) = p_0, \forall k \in \{1, \ldots, k_\#\}$. Then, the corresponding quantiles $q_{[1]}^{p_0} > \cdots > q_{[k_\#]}^{p_0}$ 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 $f_{\mathbf{X}}$. Following conditional subset samples are generated by MCMC sampling of $f_{\mathbf{X}}(\mathbf{x}|\mathcal{F}_{[k-1]})$, using as seeds initialisation points the $n = Np_0$ samples given by $\mathbf{A}_{[k],n} = \{\mathbf{X}_{[k-1]}^{(j)} \subset \mathbf{X}_{[k-1],N} | g(\mathbf{X}_{[k-1]}^{(j)}) > \widehat{q}_{[k-1]}^{\alpha}\}_{j=1}^{n}$. This process is repeated until an intermediary quantile exceeds the threshold: $\widehat{q}_{[k_\#]}^{p_0} < y_{\mathrm{th}}$. Finally, the failure probability is estimated by:

$$p_{\mathrm{f}} \approx \widehat{p}_{\mathrm{f}}^{\mathrm{SS}} = p_0^{k_\#-1} \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{\{g(\mathbf{x}) \leq y_{\mathrm{th}}\}}(\mathbf{X}_{[k_\#],N}^{(j)}). \tag{6.2}$$

In practice, the subset sample size should be large enough to properly estimate intermediary quantiles, which leads **?** to recommend setting $p_0 = 0.1$. SS efficiency depends on the proper choice and tuning of the MCMC algorithm (**?**). Our work uses the SS implementation from `OpenTURNS`[1] (**?**) which integrates a component-wise Metropolis-Hastings algorithm. As an alternative to generating samples on a conditional distribution by MCMC, one could try to fit this conditional distribution.

[1] https://openturns.github.io/www/index.html

Multivariate modeling using copulas

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

1. A set of univariate marginal distributions to describe the behavior of the individual variables;

2. A function describing the dependence structure between all variables, called a copula.

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

$$F(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_p(x_d)\right). \tag{6.3}$$

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

6.2 Bernstein adaptive nonparametric conditional sampling

6.2.1 Empirical Bernstein copula

Copulas are continuous and bounded functions defined on a compact set (the unit hypercube). Bernstein polynomials allow to uniformly approximate as closely as desired any continuous and real-valued function defined on a compact set (Weierstrass approximation theorem). Therefore, they are good candidates to approximate unknown copulas. This concept was introduced as *empirical Bernstein copula* (EBC) by Sancetta and Satchell (2004) for applications in economics and risk management. Later on, Segers et al. (2017) offered further asymptotic studies. Formally, the multivariate Bernstein polynomial for a function $C : [0,1]^d \to \mathbb{R}$ on a grid over the unit hypercube $G := \left\{\frac{0}{m_1}, \ldots, \frac{m_1}{m_1}\right\} \times \cdots \times \left\{\frac{0}{m_d}, \ldots, \frac{m_d}{m_d}\right\}, \mathbf{m} = (m_1, \ldots, m_d) \in \mathbb{N}^d$, writes:

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

with $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$, 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. (6.4) to the so-called "empirical copula".

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

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

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

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

Then, Sancetta and Satchell (2004) prove in their Theorem 3 that:

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

Let us remark that in the special case $m = n$, also called the "Beta copula" in Segers et al. (2017), the bias is very small while the variance gets large. To illustrate the previous theorem, Lasserre (2022) represents the evolution of the m_{IMSE} for different dimensions and sample sizes (see Fig. 6.1). In high dimension, the values of m_{IMSE} tend towards one, which is equivalent to the independent copula. Therefore, high-dimensional problems should be divided into a product of smaller problems on which the EBC is tractable. Provided a large enough learning set $\mathbf{X}_n$, KDE fitting of marginals combined with EBC fitting of the copula delivers good results even on complex dependence structures. Moreover, EBC provides an explicit expression, making a Monte Carlo generation of i.i.d. samples simple. In the following, this nonparametric tool is used to fit the intermediary conditional distributions present in subset sampling.

6.2.2 Bernstein adaptive nonparametric conditional sampling algorithm

This new method reuses the main idea from SS while employing a different approach to generate conditional samples. Instead of using MCMC sampling, the conditional distribution is firstly fitted by a nonparametric procedure, before sampling on this nonparametric model. As described in Algorithm 1, conditional sampling is done on a distribution composed by merging marginals

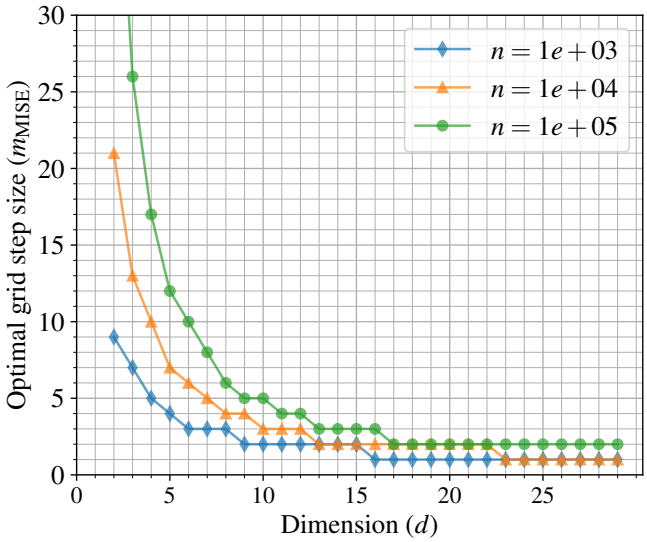


Figure 6.1 Evolution of m_{IMSE} for different dimensions and sample sizes.

$\{\widehat{F}_i\}_{i=1}^d$ fitted by KDE, with a copula $B_{\mathbf{m}}(C_n)$ fitted by EBC. Fig. 6.2 illustrate the nonparametric fit and conditional sampling in BANCS method on a two-dimensional reliability problem (later introduced as "toy-case #1"). At iteration k, after estimating the intermediary quantile $\widehat{q}_{[k]}^{p_0}$, a nonparametric model is fitted on $\mathbf{A}_{[k+1],n}$ and used to generate the next N-sized subset sample $\mathbf{X}_{[k+1],N}$. Note that the BANCS method does not require iso-probabilistic transform.

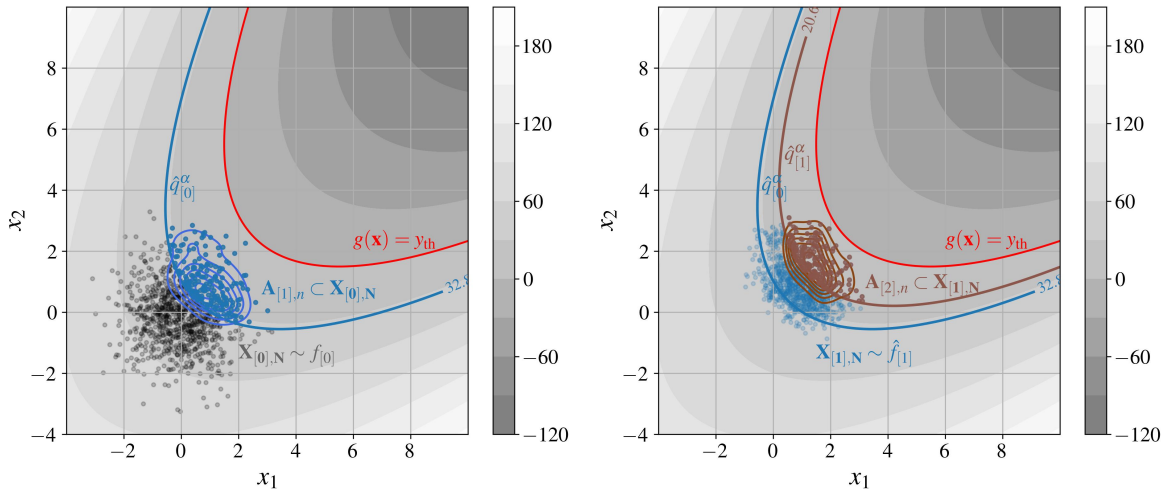


Figure 6.2 BANCS on toy-case #1: illustration of conditional sampling and nonparametric fit at the first and second iterations.

As discussed in the previous section, EBC fitting is tuned by the Bernstein polynomial of order m, implying a bias-variance tread off. In Fig. 6.2, conditional distributions fitted by EBC (blue and brown isolines) seem to present a slight bias since they overlay the quantiles. However, reducing this bias implies decreasing the tuning parameter m, until $m = 1$, which is equivalent to an independent copula. Tools to control the goodness of fit of nonparametric conditional

Algorithm 1 Bernstein adaptive nonparametric conditional sampling (BANCS).

▷ Inputs:

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

▷ Algorithm:

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

$\widehat{p}_{\text{f}}$, estimate of p_{f}

distributions are also available. As an example, let us consider the fitted conditional distribution at the first iteration (visible in Fig. 6.2). Its quantile-quantile plot in Fig. 6.3 shows a good fit of the two marginals by KDE. Then, the goodness of fit of copulas can be evaluated by Kendall's plot, represented in Fig. 6.4. This fit is also good, even if a slight bias is again visible.

6.3 Numerical experiments

In the following analytical numerical experiments, the intermediary probabilities were set to $p_0 = 0.1$, allowing a fair comparison with subset sampling. Then, the subset sample size is set to $N = 10^4$, in order to get a reasonable sample size $n = Np_0 = 10^3$ to perform the nonparametric fitting. EBC tuning is setup to minimize the MISE in Eq. (6.6): $m = 1 + n^{\frac{2}{d+4}}$. In order to take into account the variability of the method's results, each experiment is repeated 100 times, allowing the computation of a coefficient of variation $\widehat{\delta} = \frac{\sigma_{\widehat{p_{\text{f}}}}}{\mu_{\widehat{p_{\text{f}}}}}$. Note that an implementation of the BANCS method and the following numerical experiments are available in a Git repository[2].

[2] https://github.com/efekhari27/icasp14

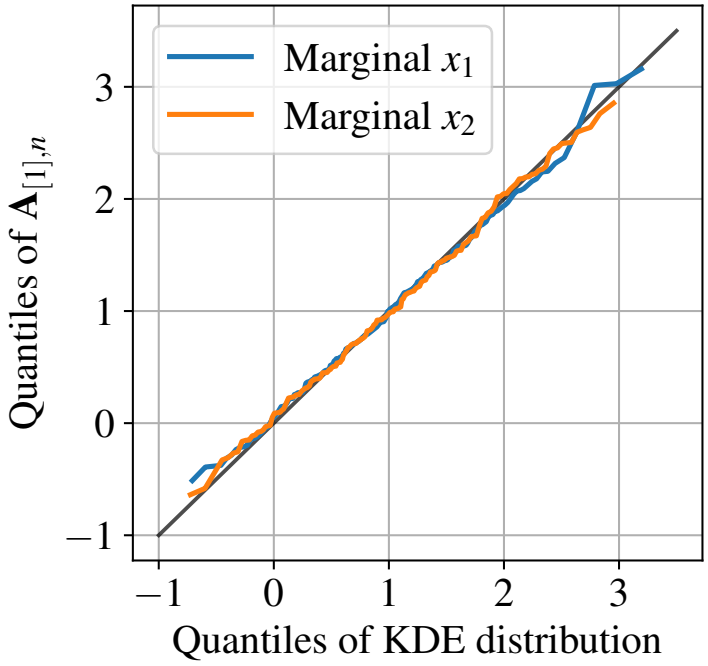


Figure 6.3 QQ-plot for KDE of marginals of the conditional distribution from Fig. 6.2.

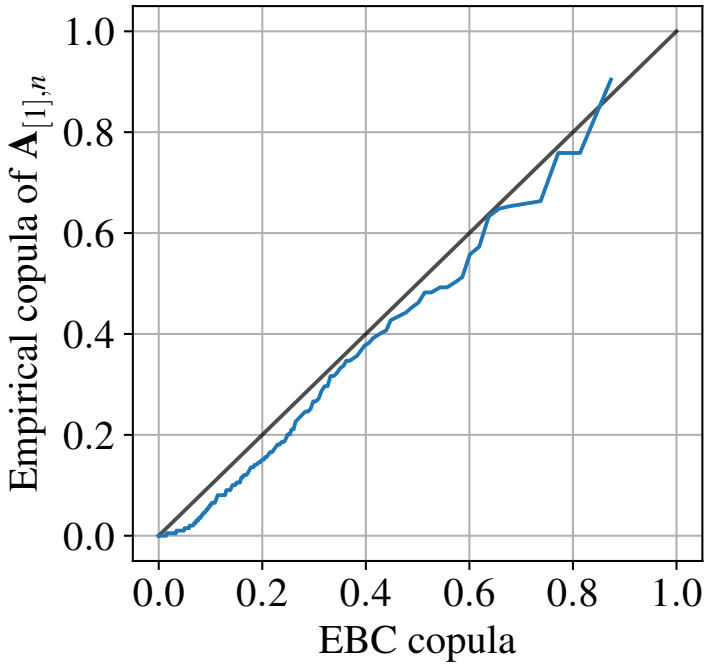


Figure 6.4 Kendall plot for EBC on the copula of a conditional distribution from Fig. 6.2.

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

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

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

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

$$g_2(\mathbf{x}) = \min \begin{pmatrix} 5 + 0.1(x_1 - x_2)^2 - \frac{(x_1+x_2)}{\sqrt{2}} \\ 5 + 0.1(x_1 - x_2)^2 + \frac{(x_1+x_2)}{\sqrt{2}} \\ (x_1 - x_2) + \frac{9}{\sqrt{2}} \\ (x_2 - x_1) + \frac{9}{\sqrt{2}} \end{pmatrix}, \tag{6.8}$$

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

Toy-case #3: high-dimensional reliability problem Let us define the higher-dimensional reliability problem (proposed by **?**), considering the following function $g_3 : \mathbb{R}^7 \to \mathbb{R}$:

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

with the input random vector $\mathbf{X} = (X_1, \ldots, X_7)$, following a product of normal distributions defined in **?**. The reliability problem consists in evaluating: $p_{\mathrm{f},3} = \mathbb{P}(g_3(\mathbf{X}) \leq 0) = 8.10 \times 10^{-3}$.

6.3.1 Results analysis

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

The variance obtained with the repetitions is quite large. Although, part of it is due to the fact that the algorithm might compute a different total number of subsets (e.g., toy-case #1 is either solved in four or five subsets). Overall, considering the EBC tuning from Eq. (6.6), BANCS performs worst than SS on toy-cases #1 and #2 but performs as well as SS on the toy-case #3. This might be due to the fact that toy-case #3 has a higher input dimension. However, one can note that SS coefficient of variation is computed by an approximation, tending to underestimate the true coefficient of variation (see e.g., **?**).

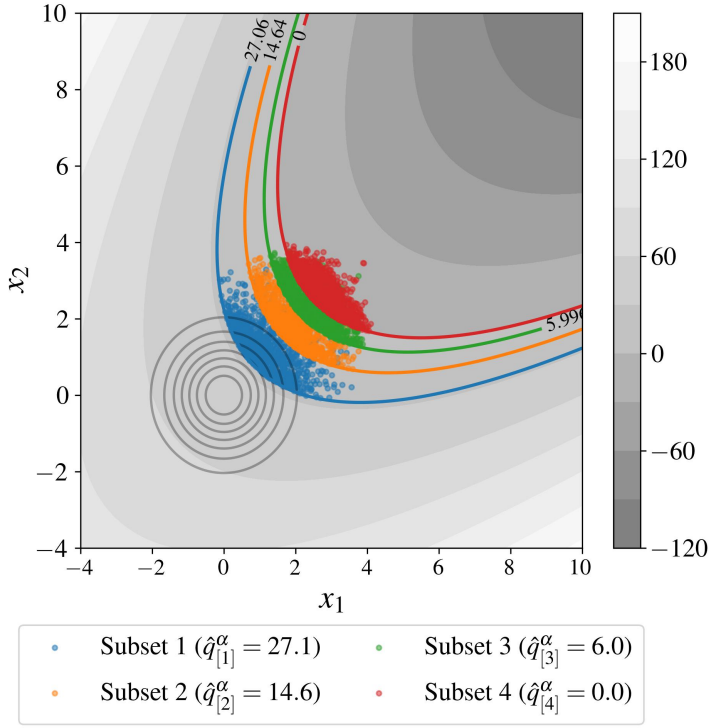


Figure 6.5 BANCS sampling steps on toy-case #1.

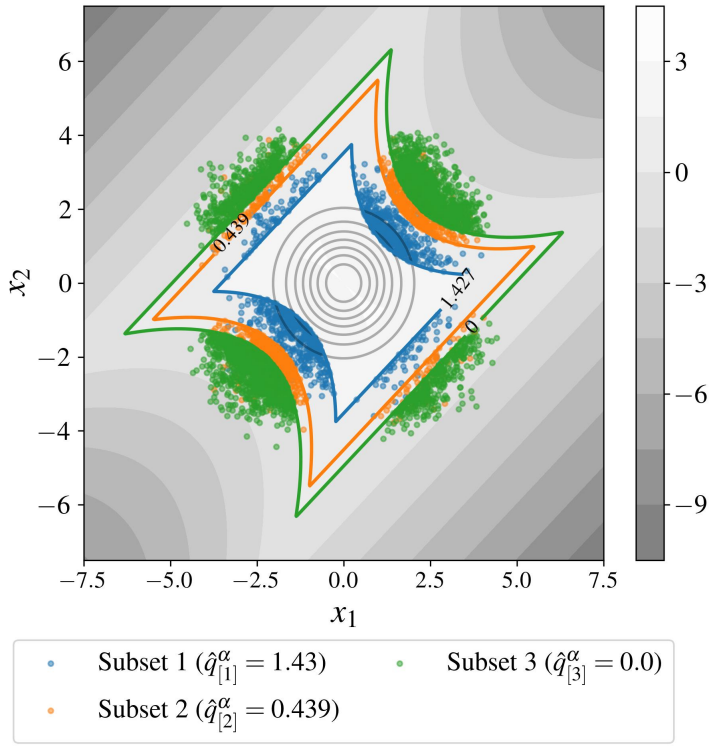


Figure 6.6 BANCS sampling steps on toy-case #2.

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

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

6.4 Application to wind turbine fatigue reliability

6.5 Conclusion

Subset Simulation uses MCMC sampling to generate its intermediary conditional samples. However, MCMC algorithms tends to be complex to tune and does not generate i.i.d. conditional samples. In this work, a new method is proposed, replacing MCMC sampling with a simpler procedure. An intermediary conditional distribution is first fitted by a nonparametric approach, mixing kernel density estimation for fitting the marginals and Empirical Bernstein Copula (EBC) for fitting the copula. Then, the resulting allows to perform direct Monte Carlo sampling. This method is named "Bernstein adaptive nonparametric conditional sampling" (BANCS) and is applied to three toy-cases (two 2-dimensional and one 7-dimensional) and compared with SS.

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

Chapter 7

Sequential reliability oriented sensitivity analysis

On risk levels: https://onlinelibrary.wiley.com/doi/epdf/10.1002/we.2610 [Partie plus prospective]

7.1 Introduction

7.2 PLI + application on the case

7.3 HSIC for GSA

7.4 HSIC for TSA & CSA

7.5 Sequential ROSA

7.6 Application to wind turbine fatigue reliability

7.7 Conclusion

Conclusion and perspectives

Bibliography

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 industrialsystems*. PhD thesis, Université Bourgogne Franche-Comté.

Briol, F., Oates, C., Girolami, M., Osborne, M., and Sejdinovic, D. (2019). Probabilistic Integration: A Role in Statistical Computation? *Statistical Science*, 34:1 – 22.

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.

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.

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112.

Fang, K., Liu, M.-Q., Qin, H., and Zhou, Y.-D. (2018). *Theory and application of uniform experimental designs*, volume 221. Springer.

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.

Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322.

Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Chapman and Hall.

Joe, H. and Kurowicka, D. (2011). *Dependence modeling: vine copula handbook*. World Scientific.

Joseph, V., Gul, E., and Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, 102.

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.

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.

L'Ecuyer, P. (2018). *Randomized quasi-Monte Carlo: An introduction for practitioners*. Springer.

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.

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.

Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122(2):218–230.

Owen, A. (2013). *Monte Carlo theory, methods and examples*.

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.

Segers, J., Sibuya, M., and Tsukahara, H. (2017). The empirical beta copula. *Journal of Multivariate Analysis*, 155:35–51.

Sun, F., Wang, Y., and Xu, H. (2019). Uniform projection designs. *The Annals of Statistics*, 47:641 – 661.

Trefethen, L. (2008). Is Gauss quadrature better than Clenshaw–Curtis? *SIAM review*, 50(1):67–87.

Warnock, T. (1972). Computational investigations of low-discrepancy point sets. In *Applications of number theory to numerical analysis*, pages 319–343. Elsevier.

Appendix A

Univariate distribution fitting

This appendix recalls the main methods to infer a univariate distribution considering a n-sized i.i.d sample $X_n = \left\{x^{(1)}, \dots, x^{(n)}\right\} \in \mathbb{R}^n$. 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 moment's method aims at looking for a parametric distribution with density $f_X(\boldsymbol{\theta})$, whose first moments (e.g., $m(\boldsymbol{\theta})$ and $\sigma^2(\boldsymbol{\theta})$) match the empirical moments of the sample X_n (e.g., $\widehat{m}_{X_n}$ and $\widehat{\sigma}^2$). After computing the empirical moments:

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

one can solve the system of equations $\left(m(\boldsymbol{\theta}) = \widehat{m}_n;\ \sigma^2(\boldsymbol{\theta}) = \widehat{\sigma}_n^2\right)$ to determine the optimal set of parameters $\boldsymbol{\theta}$ in this situation. Some families of distributions are more suited to this method (i.e.,) 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 $f_X(\boldsymbol{\theta})$. This metric is the *likelihood* function, defined as:

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

with the PDF taking the set of parameters $\boldsymbol{\theta}$ written: $f_X(x^{(i)};\boldsymbol{\theta})$. 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 $\widehat{\boldsymbol{\theta}}^*$ of parameters minimizing the following expression:

$$\widehat{\boldsymbol{\theta}}^* = \underset{\boldsymbol{\theta}\in\mathcal{D}_{\boldsymbol{\theta}}}{\arg\min} \left(-\sum_{i=1}^{n} \ln(f_X(x^{(i)};\boldsymbol{\theta})) \right). \tag{A.3}$$

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

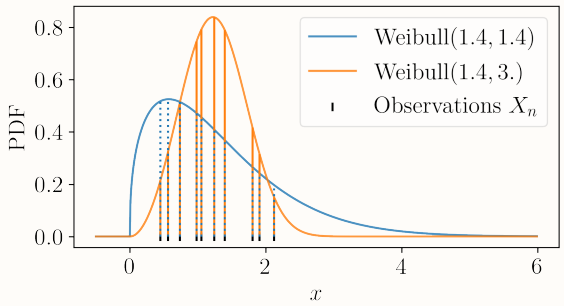


Figure A.1 Adequation of two different Weibull models using their likelihood with a sample of observations (black crosses).

A.2 Main nonparametric methods

Empirical CDF and histogram

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

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

A histogram consists of 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 tunning 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 $k : \mathbb{R} \longrightarrow \mathbb{R}_+$ and a scaling parameter $h > 0$, also called bandwidth, the kernel density estimator is defined as:

$$\widehat{f_X}(x) = \frac{1}{nh} \sum_{i=1}^{n} k\left(\frac{x - x^{(i)}}{h}\right) \tag{A.5}$$

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 (**?**).

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

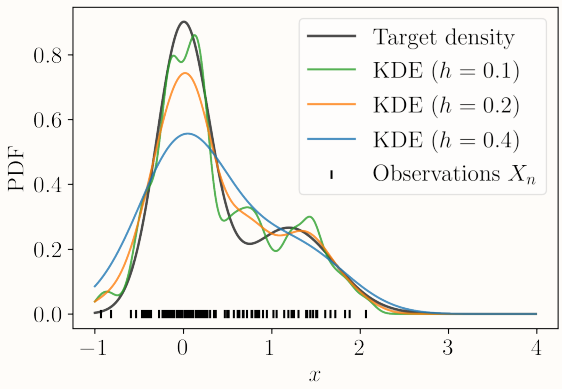


Figure A.2 Fit of a bimodal density by KDE using different tuning parameters.

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 remember that more flexible models will require more data to provide a robust estimation.

The AIC and BIC are expressed as follows:

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

with the likelihood $\mathcal{L}(\boldsymbol{\theta}|X_n)$ 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 $\left\{\alpha^{(i)}\right\}_{i=1}^{n} = \left\{\widehat{F}_X(x^{(i)})\right\}_{i=1}^{n}$. The following Fig. A.3 is a QQ-plot of the model fitted in [Example xx]. The closer the scatter plot gets to the first bisector line the better the fit is.

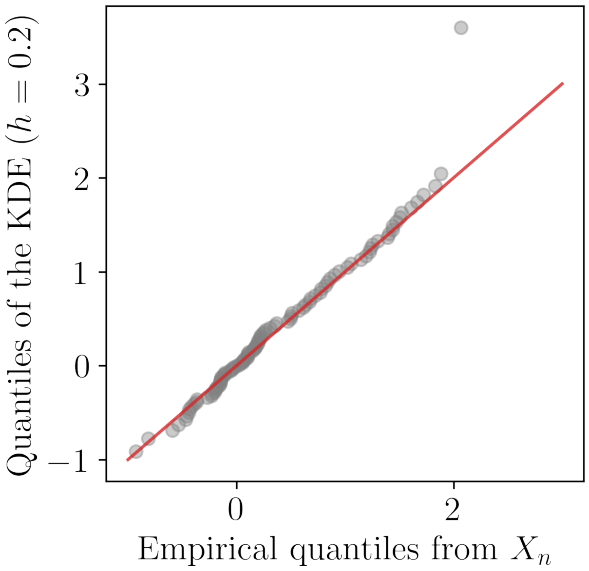


Figure A.3 QQ-plot between the data from Example 2 and a KDE model.

Appendix B

Dissimilarity measures between probability distributions

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

B.1 Csizár f-divergences

[General definition]

[Numerous examples depending on the function choosen: 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]

B.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.]

B.2.1 Kernel discrepancy

Quasi-Monte Carlo sampling methods widely rely on a uniformity metric, called *discrepancy*. This section first presents the link between discrepancy and numerical integration. Then it introduces a kernel-based discrepancy, generalizing the concept to non-uniform measures. This tool is eventually used to build a sequential quadrature rule by subsampling from a finite dataset.

Reproducing kernel Hilbert space and kernel mean embedding To generalize the Koksma-Hlawka inequality to any probability measure, let us assume that the integrand g lives

in a specific function space $\mathcal{H}(k)$. $\mathcal{H}(k)$ is a *reproducing kernel Hilbert space* (RKHS), which is an inner product space of functions $g : \mathcal{D}_X \to \mathbb{R}$. Considering a symmetric and positive definite function $k : \mathcal{D}_X \times \mathcal{D}_X \to \mathbb{R}$, later called a "reproducing kernel" or simply a "kernel", an RKHS verifies the following axioms:

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

Note that it can be shown that every positive semi-definite kernel defines a unique RKHS (and vice versa) with a feature map ϕ, such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}(k)}$. This framework allows us to embed a continuous or discrete probability measure in an RKHS, as illustrated in Fig. B.1. For any measure π, let us define its *kernel mean embedding* (**?**), also called "potential" $P_\pi(\mathbf{x})$ in **?**, associated with the kernel k as:

$$P_\pi(\mathbf{x}) := \int_{\mathcal{D}_X} k(\mathbf{x}, \mathbf{x}') \mathrm{d}\pi(\mathbf{x}'). \tag{B.1}$$

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

$$P_{\zeta_n}(\mathbf{x}) = \int_{\mathcal{D}_X} k(\mathbf{x}, \mathbf{x}') \mathrm{d}\zeta_n(\mathbf{x}') = \sum_{i=1}^n w_i k(\mathbf{x}, \mathbf{x}^{(i)}). \tag{B.2}$$

The potential $P_\pi(\mathbf{x})$ of the targeted measure π will be referred to as "target potential" and the potential $P_{\zeta_n}(\mathbf{x})$ associated with the discrete distribution ζ_n called "current potential" when its support is the current design $\mathbf{X}_n$. When $P_{\zeta_n}(\mathbf{x})$ is close to $P_\pi(\mathbf{x})$, it can be interpreted as ζ_n being an adequate quantization or representation of π (which leads to a good estimation of a quantity such as $I_\pi(g)$ from Eq. (4.5)). Potentials can be computed in closed forms for specific pairs of distribution and associated kernel. Summary tables of some of these cases are detailed in **?** (section 3.4), **?** (section 4), and extended in **?**. However, in most cases, the target potentials must be estimated on a large and representative sample, typically a large quasi-Monte Carlo sample of π.

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

$$\varepsilon_\pi := \int_{\mathcal{D}_X} P_\pi(\mathbf{x}) \mathrm{d}\pi(\mathbf{x}) = \iint_{\mathcal{D}_X^2} k(\mathbf{x}, \mathbf{x}') \, \mathrm{d}\pi(\mathbf{x}) \mathrm{d}\pi(\mathbf{x}'). \tag{B.3}$$

Finally, using the reproducing property and writing the Cauchy-Schwarz inequality on the absolute quadrature error leads to the following inequality, similar to the Koksma-Hlawka inequality Eq. (4.6) (see Briol et al. (2019)):

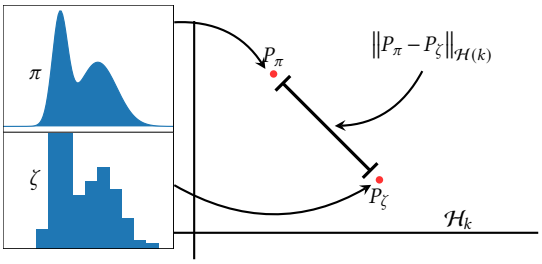


Figure B.1 Kernel mean embedding of a continuous and discrete probability distribution

$$\left|\sum_{i=1}^{n} w_i g(\mathbf{x}^{(i)}) - \int_{\mathcal{D}_\mathrm{X}} g(\mathbf{x})\mathrm{d}\pi(\mathbf{x})\right| = \left|\left\langle g, P_{\zeta_n}(\mathbf{x})\right\rangle_{\mathcal{H}(k)} - \left\langle g, P_\pi(\mathbf{x})\right\rangle_{\mathcal{H}(k)}\right| \tag{B.4a}$$

$$= \left|\left\langle g, \left(P_{\zeta_n}(\mathbf{x}) - P_\pi(\mathbf{x})\right)\right\rangle_{\mathcal{H}(k)}\right| \tag{B.4b}$$

$$\leq \|g\|_{\mathcal{H}(k)} \left\|P_\pi(\mathbf{x}) - P_{\zeta_n}(\mathbf{x})\right\|_{\mathcal{H}(k)}. \tag{B.4c}$$

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

$$\mathrm{MMD}(\pi,\zeta) := \sup_{\|g\|_{\mathcal{H}(k)}\leq 1} \left|\int_{\mathcal{D}_\mathrm{X}} g(\mathbf{x})\mathrm{d}\pi(\mathbf{x}) - \int_{\mathcal{D}_\mathrm{X}} g(\mathbf{x})\mathrm{d}\zeta(\mathbf{x})\right| \tag{B.5}$$

According to the inequality in Eq. (B.4c), $\mathrm{MMD}(\pi,\zeta) = \left\|P_\pi - P_\zeta\right\|_{\mathcal{H}(k)}$, meaning that the MMD fully relies on the difference of potentials. Moreover, **?** defines a kernel as "characteristic kernel" when the following equivalence is true: $\mathrm{MMD}(\pi,\zeta) = 0 \Leftrightarrow \pi = \zeta$. This property makes the MMD a metric on $\mathcal{D}_\mathrm{X}$. The squared MMD has been used for other purposes than numerical integration: e.g., statistical testing (**?**), and global sensitivity analysis (**?**). It can be written as follows:

$$\mathrm{MMD}(\pi,\zeta)^2 = \left\|P_\pi(\mathbf{x}) - P_\zeta(\mathbf{x})\right\|^2_{\mathcal{H}(k)} \tag{B.6a}$$

$$= \left\langle \left(P_\pi(\mathbf{x}) - P_\zeta(\mathbf{x})\right), \left(P_\pi(\mathbf{x}) - P_\zeta(\mathbf{x})\right)\right\rangle_{\mathcal{H}(k)} \tag{B.6b}$$

$$= \left\langle P_\pi(\mathbf{x}), P_\pi(\mathbf{x})\right\rangle_{\mathcal{H}(k)} - 2\left\langle P_\pi(\mathbf{x}), P_\zeta(\mathbf{x})\right\rangle_{\mathcal{H}(k)} + \left\langle P_\zeta(\mathbf{x}), P_\zeta(\mathbf{x})\right\rangle_{\mathcal{H}(k)} \tag{B.6c}$$

$$= \iint_{\mathcal{D}_\mathrm{X}^2} k(\mathbf{x},\mathbf{x}')\,\mathrm{d}\pi(\mathbf{x})\mathrm{d}\pi(\mathbf{x}') - 2\iint_{\mathcal{D}_\mathrm{X}^2} k(\mathbf{x},\mathbf{x}')\,\mathrm{d}\pi(\mathbf{x})\mathrm{d}\zeta(\mathbf{x}') + \iint_{\mathcal{D}_\mathrm{X}^2} k(\mathbf{x},\mathbf{x}')\,\mathrm{d}\zeta(\mathbf{x})\mathrm{d}\zeta(\mathbf{x}'). \tag{B.6d}$$

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

$$\mathrm{MMD}(\pi, \zeta_n)^2 = \varepsilon_\pi - \frac{2}{n}\sum_{i=1}^{n} P_\pi\left(\mathbf{x}^{(i)}\right) + \frac{1}{n^2}\sum_{i,j=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right). \tag{B.7}$$

Maximum discrepancy measure

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

$$\mathrm{MMD}(\mu, \zeta) := \sup_{\|g\|_{\mathcal{H}} \leq 1} \left| \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}) - \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x}) \mathrm{d}\zeta(\mathbf{x}) \right| = \left\| P_\mu(\mathbf{x}) - P_\zeta(\mathbf{x}) \right\|_{\mathcal{H}}. \tag{B.8}$$

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 $g : \mathcal{D}_\mathbf{X} \to \mathbb{R}$. It verifies the following axioms, considering a symmetric and positive definite function $k : \mathcal{D}_\mathbf{X} \times \mathcal{D}_\mathbf{X} \to \mathbb{R}$, later called a "reproducing kernel" or simply a "kernel":

- The "feature map" $\phi : \mathcal{D}_\mathbf{X} \to \mathcal{H}(k); \phi(\mathbf{x}) = k(\cdot, \mathbf{x}) \in \mathcal{H}(k), \forall \mathbf{x} \in \mathcal{D}_\mathbf{X}$.
- The "reproducing property": $\langle g, k(\cdot, \mathbf{x})\rangle_{\mathcal{H}(k)} = g(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}_\mathbf{X}, \forall g \in \mathcal{H}(k)$.

Every positive semi-definite kernel defines a unique RKHS (and vice versa) with a feature map ϕ, such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}')\rangle_{\mathcal{H}(k)}$. Moreover, **?** defines a kernel as "characteristic kernel" when the following equivalence is true: $\mathrm{MMD}_k(\mu, \zeta) = 0 \Leftrightarrow \mu = \zeta$. This property makes the MMD a metric on $\mathcal{D}_\mathbf{X}$.

Then, a probability measure has a representation in the RKHS through its *kernel mean embedding* (**?**), also called "potential" $P_\mu(\mathbf{x})$ in **?**, defined as:

$$P_\mu(\mathbf{x}) := \int_{\mathcal{D}_\mathbf{X}} k(\mathbf{x}, \mathbf{x}') \mathrm{d}\mu(\mathbf{x}'). \tag{B.9}$$

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

$$\mathrm{MMD}_k(\mu, \zeta)^2 = \int_{\mathcal{D}_\mathbf{X}} P_\mu(\mathbf{x})\, \mathrm{d}\mu(\mathbf{x}) - 2\int_{\mathcal{D}_\mathbf{X}} P_\mu(\mathbf{x})\, \mathrm{d}\zeta(\mathbf{x}) + \int_{\mathcal{D}_\mathbf{X}} P_\zeta(\mathbf{x})\, \mathrm{d}\zeta(\mathbf{x}). \tag{B.10}$$

[Add a sentence on estimation]

Analytical computation of potentials for Matérn kernels

As for tensor-product kernels, the potential is the product of the one-dimensional potentials, we only consider one-dimensional input spaces.

For μ the uniform distribution on $[0,1]$ and K the Matérn kernel $K_{5/2,\theta}$ with smoothness $\nu = 5/2$ and correlation length θ, see (5.16), we get

$$P_{K_{5/2,\theta},\mu}(x) = \frac{16\theta}{3\sqrt{5}} - \frac{1}{15\theta}(S_\theta(x) + S_\theta(1-x)),$$

where

$$S_\theta(x) = \exp\left(-\frac{\sqrt{5}}{\theta}x\right)\left(5\sqrt{5}x^2 + 25\theta x + 8\sqrt{5}\theta^2\right).$$

The expressions $P_{K_{\nu,\theta},\mu}(x)$ for $\nu = 1/2$ and $\nu = 3/2$ can be found in **?**.

When μ is the standard normal distribution $\mathcal{N}(0,1)$, the potential $P_{K_{5/2,\theta},\mathcal{N}(0,1)}$ is $P_{K_{5/2,\theta},\mathcal{N}(0,1)}(x) = T_\theta(x) + T_\theta(-x)$, where

$$\begin{aligned} T_\theta(x) &= \frac{1}{6}\left(\frac{5}{\theta^2}x^2 + \left(3 - \frac{10}{\theta^2}\right)\frac{\sqrt{5}}{\theta}x + \frac{5}{\theta^2}\left(\frac{5}{\theta^2} - 2\right) + 3\right) \\ &\quad \times \operatorname{erfc}\left(\frac{\frac{\sqrt{5}}{\theta} - x}{\sqrt{2}}\right)\exp\left(\frac{5}{2\theta^2} - \frac{\sqrt{5}}{\theta}x\right) + \frac{1}{3\sqrt{2\pi}}\frac{\sqrt{5}}{\theta}\left(3 - \frac{5}{\theta^2}\right)\exp\left(-\frac{x^2}{2}\right). \end{aligned}$$

Appendix C

Advanced rare event estimation algorithms

C.1 Subset simulation (SS)

C.2 Nonparametric adaptive importance sampling (NAIS)

Appendix D

Uncertainty quantification practice with OpenTURNS

[Add short introduction to the motivation]

[Should I print the results?]

OpenTURNS 7 (Bivariate distribution). The following Python code proposes a minimalistic OpenTURNS example of a probabilistic uncertainty modeling.

```
#!/usr/bin/python3
import openturns as ot
# Build multivariate distribution from marginals and copula
copula=ot.GumbelCopula(2.0)
marginals=[ot.Uniform(1.0, 2.0), ot.Normal(2.0, 3.0)]
distribution=ot.ComposedDistribution(marginals, copula)
# Compute first moments
mean_vector=distribution.getMean()
covariance_matrix=distribution.getCovariance()
# Compute CDF (respectively PDF)
x_cdf=distribution.computeCDF([1.5, 2.5])# x=[1.5, 2.5]
a_quantile=distribution.computeQuantile([0.9])# alpha=0.9
```

OpenTURNS 8 (Numerical integration). The following Python code presents a minimalistic OpenTURNS example to build multivariate quadrature rules.

```
#!/usr/bin/python3
import openturns as ot
marginals=[ot.Exponential(1.0), ot.Uniform(-1.0, 1.0)]
distribution=ot.ComposedDistribution(marginals)
# Build a 2D Gaussian quadrature
n_marginal=[4, 4] # Number of nodes per marginal
g_quad=ot.GaussProductExperiment(distribution, n_marginal)
g_nodes, weights=g_quad.generateWithWeights()
# Build a Monte Carlo design
n=16
mc_nodes=distribution.getSample(n)
# Build a quasi-Monte Carlo design
sequence=ot.HaltonSequence(2) # d=2
qmc_experiment=ot.LowDiscrepancyExperiment(sequence, distribution, n)
qmc_nodes=qmc_experiment.generate()
```

OpenTURNS 9 (Design of experiments). The following Python code is a minimalistic OpenTURNS example to build an LHS and an LHS optimized w.r.t. to a space-filling metric (here the L2-centered discrepancy) using the simulated annealing algorithm.

```
#!/usr/bin/python3
import openturns as ot
marginals=[ot.Uniform(0.0, 1.0), ot.Uniform(0.0, 1.0)]
distribution=ot.ComposedDistribution(marginals)
# Build a LHS
n=10
LHS_exp=ot.LHSExperiment(distribution, n)
LHS_design=LHS_exp.generate()
# Build an optimized LHS using L2-centered discrepancy
LHS_exp=ot.LHSExperiment(distribution, n)
SF_metric=ot.SpaceFillingC2()
SA_profile=ot.GeometricProfile(10., 0.95, 20000)
LHS_opt=ot.SimulatedAnnealingLHS(LHS_exp, SF_metric, SA_profile)
LHS_opt.generate()
LHS_design=LHS_opt.getResult().getOptimalDesign()
```

OpenTURNS 10 (Rare event estimation). The following Python code proposes a minimalistic OpenTURNS implementation of rare event estimation algorithms.

```
#!/usr/bin/python3
import openturns as ot
marginals=[ot.Normal(0.0, 1.0), ot.Exponential(1.0)]
distribution=ot.ComposedDistribution(marginals)
# Build a limit-state function and failure event
g=ot.SymbolicFunction(["x1", "x2"], ["(x1 - x2) ^ 2"])
X=ot.RandomVector(distribution)
Y=ot.CompositeRandomVector(g, X)
th=0.0
failure_event=ot.ThresholdEvent(Y, ot.LessOrEqual(), th)
# Estimate pf using FORM
starting_p=distribution.getMean()
FORM_algo=ot.FORM(ot.Cobyla(), failure_event, starting_p)
FORM_algo.run()
FORM_results=FORM_algo.getResult()
design_point=FORM_results.getStandardSpaceDesignPoint()
FORM_pf=FORM_results.getEventProbability()
# Estimate pf using Monte Carlo
MC_exp=ot.MonteCarloExperiment()
MC_algo=ot.ProbabilitySimulationAlgorithm(failure_event, MC_exp)
MC_algo.run()
MC_results=MC_algo.getResult()
MC_pf=MC_results.getProbabilityEstimate()
MC_pf_confidence=MC_results.getConfidenceLength(0.95)
# Estimate pf using importance sampling
aux_distribution=ot.Normal(design_point, [1.0, 1.0])
standard_event=ot.StandardEvent(failure_event)
IS_exp=ot.ImportanceSamplingExperiment(aux_distribution)
IS_algo=ot.ProbabilitySimulationAlgorithm(standard_event, IS_exp)
IS_algo.run()
IS_results=IS_algo.getResult()
IS_pf=IS_results.getProbabilityEstimate()
IS_pf_confidence=IS_results.getConfidenceLength(0.95)
# Estimate pf using subset sampling
SS_algo=ot.SubsetSampling(failure_event)
SS_algo.run()
SS_results=SS_algo.getResult()
SS_pf=SS_results.getProbabilityEstimate()
SS_pf_confidence=SS_results.getConfidenceLength(0.95)
```

OpenTURNS 11 (Sobol' indices). The following Python code gives a minimalistic OpenTURNS implementation of the Sobol' indices to assess global sensitivity analysis on the Ishigami analytical problem.

```
#!/usr/bin/python3
import openturns as ot
g=ot.SymbolicFunction(
    ['x1', 'x2', 'x3'],
    ['sin(x1) + 7.0 * sin(x2)^2 + 0.1 * x3^4 * sin(x1)']
    )
X=ot.ComposedDistribution([ot.Uniform(-3.14, 3.14)] * 3)
size=1000
# Generate samples and evaluate their images
sie=ot.SobolIndicesExperiment(im.distributionX, size)
input_design=sie.generate()
output_design=im.model(input_design)
# Four estimators : Saltelli, Martinez, Jansen, and Mauntz-Kucherenko
SA=ot.JansenSensitivityAlgorithm(input_design, output_design, size)
sobol_first_order=SA.getFirstOrderIndices()
sobol_tolal=SA.getTotalOrderIndices()
```

OpenTURNS 12 (Gaussian process regression). The following Python code gives a minimalistic OpenTURNS implementation of an ordinary kriging model fitting.

```
#!/usr/bin/python3
import openturns as ot
g=ot.SymbolicFunction(['x'], ['x * sin(x) + sin(6 * x)'])
x_train=ot.Uniform(0., 12.).getSample(7) # n=7
y_train=g(x_train)
basis=ot.ConstantBasisFactory(1).build() # d=1
cov_model=ot.MaternModel([1.], 1.5)
algo=ot.KrigingAlgorithm(x_train,y_train,cov_model,basis)
algo.run()
kriging_results=algo.getResult()
kriging_predictor=kriging_results.getMetaModel()
```

Appendix E

Résumé étendu de la thèse

E.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 **?**. 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 (**?**).

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) (**?**). 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 à 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 (**?**).

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 (**?**). Une présentation des outils mathématiques utilisés dans ce domaine est proposée par **?**. Cette approche apporte une meilleure compréhension d'un système, ce qui contribue à une prise de décision plus robuste.

La Figure E.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 (**?**)). 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 (**?**).

- **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.

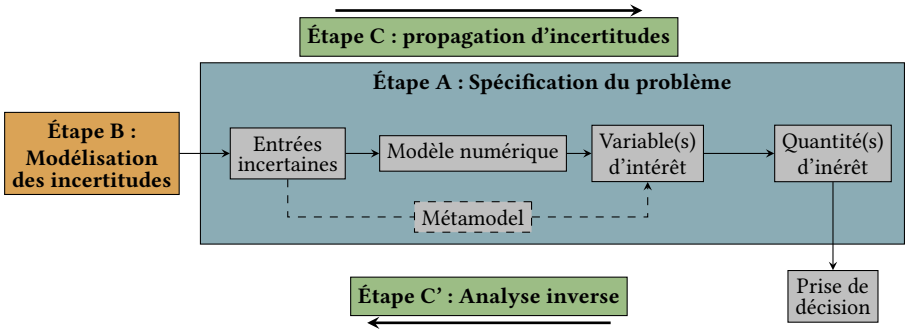
- **Étape C – Propagation des incertitudes.** Lors de cette étape, les entrées incertaines sont propagées au travers du modèle de simulation numérique. Dès lors, la sortie du modèle numérique (habituellement de type scalaire) devient également incertaine. L'objectif est alors d'estimer une quantité d'intérêt, 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 œuvre 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 :



?figurename? E.1 Schéma générique de la quantification des incertitudes (**?**, 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'échantillonage 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.

E.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 relateront 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 (**?**). 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'échantillonnage 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 (**??**).

Ce chapitre aborde également les principales méthodes d'analyse de sensibilité globale (**?**). 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 (**?**). 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 (**?**). Depuis, leur utilisation s'est étendue à la résolution de problèmes de fiabilité des structures (**?**).

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 (**?**)). 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 **?**. 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 **??** 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 (**?**Lataniotis, 2019; **?**; **?**; **?**).

Dans ce contexte, ce chapitre liste les paramètre 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).

E.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) **?** est une mesure de dissimilarité entre des lois de probabilité basée sur des noyaux qui est utilisée dans des contextes différents (tests statistiques **?**, analyse de sensibilité **?**, échantillonnage **?**, 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) **?**. 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 prédiction 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 **?**. 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 environnemen-

tales 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'échantillonnage 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'échantillonnage basées sur des approches géométriques (**?**) ou sur des méthodes à noyaux (**??**) 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 **?**. 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*" **?** 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, une probabilité d'événement rare). L'analyse de sensibilité globale **?** 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*", **?**) 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 Indepencence Criterion*, pour évaluer la dynamique de la sensibilité fiabiliste **?**.

E.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.

E.4.1 Communications et publications dans revues à comité de lecture

Book Chap.	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.
Jour. Pap.	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.
Int. Conf	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Numerical Studies of Bayesian Quadrature Applied to Offshore Wind Turbine Load Estimation". In: *SIAM Conference on Uncertainty Quantification (SIAM UQ22)*, Atlanta, USA. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *22nd Annual Conference of the European Network for Business and Industrial Statistics (ENBIS 2022)*, Trondheim, Norway. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Efficient techniques for fast uncertainty propagation in an offshore wind turbine multi-physics simulation tool". In: *Proceedings of the 5th International Conference on Renewable Energies Offshore (RENEW 2022)*, Lisbon, Portugal. (Paper & Talk) E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). "Bernstein adaptive nonparametric conditional sampling: a new method for rare event probability estimation"[1]. In: *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)*, Dublin, Ireland. (Paper & Talk) E. Vanem, E. Fekhari, N. Dimitrov, M. Kelly, A. Cousin and M. Guiton (2023). "A joint probability distribution model for multivariate wind and wave conditions". In: *Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2023)*, Melbourne, Australia. (Paper) 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 (Paper)
Nat. Conf.	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Kernel-based quadrature applied to offshore wind turbine damage estimation". In: *Proceedings of the Mascot-Num 2022 Annual Conference (MASCOT NUM 2022)*, Clermont-Ferrand, France (Poster) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2023). "Rare event estimation using nonparametric Bernstein adaptive sampling". In: *Proceedings of the Mascot-Num 2023 Annual Conference (MASCOT-NUM 2023)*, Le Croisic, France (Talk)
Invited Lec.	Le Printemps de la Recherche 2022, Nantes, France. "Traitement des incertitudes pour la gestion d'actifs éoliens". (Talk) Journées Scientifiques de l'Eolien 2024, Saint-Malo, France. "Evaluation probabiliste de la fiabilité en fatigue des structures éoliennes en mer". (Talk)

[1] Cette contribution a été récompensée par le "CERRA Student Recognition Award"

E.4.2 Développements informatiques open source

`otkerneldesign`[2]

- 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`[3]

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

`ctbenchmark`[4]

- 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`[5]

- 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