Robust certification of aerodynamic design by concentration-of-measure inequalities

Luc Bonnet

université
PARIS-SACLAY

Robust certification of aerodynamic design by concentration-of-measure inequalities

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 579, Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)
Spécialité de doctorat : Mécanique des fluides
Unité de recherche : Université Paris-Saclay, ONERA, Aérodynamique, Aéroélasticité, Acoustique, 92320, Châtillon, France
Référent : CentraleSupélec

Thèse présentée et soutenue à Paris-Saclay, le 13 Décembre 2021, par

Luc BONNET

Composition du jury :

Olivier Le Maître Directeur de recherche, CNRS	Président
Paola Cinnella Professeur des universités, Sorbonne Université	Rapporteur & Examinateur
Clémentine Prieur Professeur des universités, Université Grenoble Alpes	Rapporteur & Examinateur
Didier Clouteau Professeur des universités, CentraleSupélec	Examinateur
Éric Savin Ingénieur de recherche, ONERA	Directeur de thèse
Houman Owhadi Professeur, California Institute of Technology	Co-encadrant de thèse

Thèse de doctorat

NNT : 2021UPAST129

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Certification robuste d'un design aérodynamique par des inégalités de concentration de mesure

Mots clés : Concentration de la mesure, Théorie des probabilités, Certification, Mécanique des fluides, Aérodynamique

Résumé : La performance attendue d'un système peut généralement différer de sa performance opérationnelle en raison de la variabilité de certains paramètres. Habituellement, la phase de conception est divisée en deux phases différentes. La première phase consiste à déterminer la conception préoptimale. Grâce à l'utilisation de logiciels numériques, la meilleure conception possible est choisie. Il s'agit de tenir compte de certaines performances idéales à atteindre en formulant des hypothèses – qui peuvent être subjectives– traduisant la variabilité de certains paramètres. La deuxième phase consiste, quant à elle, à certifier par des expériences à taille réelle que la conception déterminée précédemment est valide. De cette façon, la conception post-optimale est spécifiée. Cette deuxième phase est la plus coûteuse ; c'est pourquoi les industriels, notamment dans le domaine aéronautique, cherchent à réduire le recours aux expériences à taille réelle. L'Optimal Uncertainty Quantification (OUQ) est un outil mathématique puissant qui peut être utilisé pour borner rigoureusement la probabilité de dépasser un seuil de performance donné pour des conditions opérationnelles ou des caractéristiques de système incertaines. Cet outil conduit à la résolution d'un problème d'optimisation sur l'ensemble des mesures de probabilités admissibles, permettant ainsi de ne pas à avoir à formuler des hypothèses subjectives qui peuvent être fortes. Ce problème d'optimisation est a priori un problème non-convexe et fortement contraint, dans un espace de dimension infinie. Ainsi, il est généralement difficile à résoudre sur le plan numérique. Néanmoins, il peut être réduit à un problème d'optimisation équivalent en dimension finie.

Ce travail de thèse porte, dans un premier temps, sur l'application de l'outil OUQ concernant notamment des cas issus du domaine de l'aéronautique permettant de borner rigoureusement la variation de fonctions de performance classiquement étudiées en aérodynamique, telles que la portance ou la traînée. Néanmoins, même après réduction du problème d'optimisation, cette approche demeure complexe et l'exactitude des résultats obtenus en est ainsi impactée.

Afin de pallier cette difficulté, un algorithme permettant d'assurer la validité de ces résultats numériques a été par la suite formulé. Cet algorithme est analogue à la régression isotonique mais diffère sur les hypothèses établies. La convergence de cet algorithme a été démontrée. L'algorithme a été ensuite validé sur des cas tests numériques ainsi que sur un cas aérodynamique. Quoi qu'il en soit, l'application de la méthode OUQ nécessite d'évaluer la fonction de performance du système plusieurs dizaines de milliers de fois. De fait, l'usage d'un modèle peu coûteux en termes de temps est requis.

Dans cette optique, plusieurs méthodes qui permettent de construire des modèles de substitution rapides à évaluer sont finalement présentées. Une première méthode fondée sur l'algorithme Kernel Flow, initialement appliquée à un problème de classification, est étendue à un problème de régression permettant la détermination du noyau d'un modèle de substitution de type processus gaussien. Puis, deux nouveaux algorithmes sont présentés. Ces algorithmes utilisent l'équivalence qui existe entre l'approche processus gaussien et l'approche par espace de Hilbert à noyau reproduisant (Reproducing Kernel Hilbert Space, RKHS) afin de déterminer le noyau adéquat. Ces différentes méthodes de modèles de substitution sont appliquées au travers de différents cas tests dont des cas du domaine aérodynamique.

Title : Robust certification of aerodynamic design by concentration-of-measure inequalities

Keywords : Concentration-of-measure inequalities, Probability theory, Certification, Fluid mechanics, Aerodynamic

Abstract : The expected performance of engineering systems can significantly differ from their operational performance due to the variability of some parameters they depend on. Usually, the design stage of these systems is divided into two different phases. The first phase consists in determining the pre-optimal design. Through the use of numerical simulations, for example, the best possible design is chosen. The issue is to take into account the ideal performances to be attained by formulating hypotheses—which can be subjective—characterizing the variability of different parameters. The second phase consists in certifying through full-scale experiments that the design achieved in the first phase is valid. In this way, the post-optimal design is specified. This second phase is the most expensive one; that is why manufacturers, especially in the aeronautical sector, are trying to reduce the number of full-scale experiments. Optimal Uncertainty Quantification (OUQ) is a powerful mathematical tool that can be used to rigorously bound the probability of exceeding a given performance threshold for uncertain operational conditions or system characteristics. This approach leads to the solution of an optimization problem on a set of admissible probability measures, thus avoiding the need to make subjective assumptions that can be significant. The optimization setting is a non-convex and strongly constrained problem, in a space of infinite dimension. Thus, it is generally difficult to solve numerically. Nevertheless, it can be reduced to an equivalent finite-dimensional optimization problem.

This thesis work focuses, in a first step, on the application of the OUQ tool concerning, in particular, cases from the aeronautical domain, in order to rigorously bound the variations of performance functions classically studied in aerodynamics, such as the lift or the drag. Nevertheless, even after reduction of the optimization problem, this approach remains complex and the accuracy of the obtained results is thus impacted.

In order to overcome this difficulty, an algorithm has been formulated that allows to ensure the validity of these numerical results. This algorithm is analogous to isotonic regression but differs from the assumptions made. Its convergence has been demonstrated, and it has been validated on simple validation test cases as well as on a more complex aerodynamic case. Nevertheless, the application of the OUQ framework requires to evaluate the performance function of the system several tens of thousands of times. Therefore, the use of a time-consuming model is not feasible and a computationally cheap model is required.

With this in mind, several methods are finally presented for the construction of surrogate models that are simple to evaluate. A first approach based on the Kernel Flow algorithm initially applied to a classification problem is extended to a regression problem for the determination of the kernel (covariance) function in Gaussian process regression. Then, two new algorithms are presented. These algorithms use the equivalence between the Gaussian process regression approach and the Reproducing Kernel Hilbert Space (RKHS) approach to determine the appropriate kernel function. These different constructions of surrogate models are applied to validation test cases and more complex objective functions pertaining to aerodynamic applications.

Remerciements

Je souhaite tout d'abord remercier mon directeur de thèse Eric Savin pour m'avoir si bien accompagné à tous points de vue, je le remercie pour sa disponibilité que ce soit lors de mon stage de fin d'études comme lors de ma thèse. Je souhaite remercier Houman Owhadi qui, par ses remarques, m'a permis de bien comprendre les tenants et les aboutissants de la méthode de certification présentée dans mon manuscrit de thèse. Merci à lui ainsi qu'à Diana Bohler qui tous deux m'ont permis d'effectuer un séjour de deux mois à Caltech. Je tiens à remercier Tim Sullivan pour m'avoir accueilli trois mois au Zuse Institute de Berlin. Ces deux séjours ont été pour moi très enrichissants autant du point de vue personnel que professionnel. Je remercie aussi Jean-Luc Akian pour son expertise mathématique.

Je tiens à remercier Olivier Le Maître qui a accepté de présider ma soutenance de thèse. Je remercie aussi les deux rapporteurs Clémentine Prieur et Paola Cinnella pour leurs commentaires constructifs. Je remercie Didier Clouteau pour avoir accepté de participer à mon jury de soutenance de thèse.

Tous mes remerciements vont également à l'ONERA et plus particulièrement au département DAAA où j'ai effectué mes travaux de thèse. Je remercie Vincent Couaillier pour m'avoir accueilli au sein de l'unité NFLU à Châtillon. Je souhaite aussi remercier, pour leur accueil chaleureux et leur cordialité, tous les membres de l'unité NFLU, que ce soit les membres permanents ou les doctorants, avec lesquels j'ai discuté lors des repas et des pauses.

Enfin, mes remerciements vont à ma famille et à mes proches qui m'ont soutenu toutes ces années, ainsi qu'à mes anciens professeurs qui ont cru en moi.

> "Jeunes gens, ayons bon courage ! Si rude qu'on nous veuille faire le présent, l'avenir sera beau."
> Victor Hugo, Préface d'*Hernani* (1830)

Contents

Introduction

The greening of air transport and the improvement of its cost efficiency are major challenges facing the aviation industry and the regulatory government agencies, both at the national and international levels. The Advisory Council for Aviation Research and Innovation in Europe (ACARE)[1] for instance produced a Strategic Research and Innovation Agenda[2] (SRIA) in 2012 that defined the path to reach the ambitious goal set out in 2011 by a European group of personalities with the publication of Flightpath 2050.[3] SRIA specified that aircraft technologies in 2020 shoul have reduced their fuel consumption and CO_2 emissions by 50% per passenger-kilometer, NOx emissions by 80%, and perceived noise by 50% in comparison to year 2000. More recently, ACARE has set new environmental goals towards year 2050: a 75% reduction in CO_2 emissions per passenger-kilometer, a 90% reduction in NOx emissions, and a 65% reduction in perceived noise emission relative to the capabilities of typical new aircraft in 2000. The Clean Sky project,[4] launched in 2008, aims at developing technologies and demonstrators that will allow future aircraft to target the ACARE environmental goals. Following the success of Clean Sky I, Clean Sky II has been launched in 2014 with the objective of reaching the 2050 environmental goals set by ACARE. In the United States, the Continuous Lower Energy, Emissions and Noise (CLEEN) program[5] initiated by the Federal Aviation Administration (FAA) aims at accelerating the development of new aircraft and engine technologies, and advancing sustainable alternative jet fuels. This five-year program in collaboration with, among others, Boeing, Rolls-Royce and Pratt & Whitney, was first launched in 2010. It was sought to develop new technologies in order to reduce noise, emissions, and fuel burn. For instance, a 40% reduction in fuel burn was targeted at the end of the program in comparison to year 2000 best-in-class in-service aircraft. In 2020, building upon the success of the first two CLEEN programs, the FAA initiated a third CLEEN program, which continues efforts to achieve the same targets.

Either in Europe or the United States, the objectives of reduction of noise, emissions, and fuel burn are in common and at the heart of designing new aircraft for the next decades. Whether the safe and economic operation of such a complex system can be certified with sufficient confidence to warrant continued operation is a fundamental question in engineering. These objectives rely for a significant part on design processes based on virtual prototyping through the intensive use of numerical simulation tools and computers for the prediction of loads, stresses, temperatures, emissions, noise levels, efficiency, *etc.* The potential to achieve them is strongly dependent on the robustness of the virtual prototyping chain, that is, its ability to

[1] ACARE, *Advisory Council for Aviation Research and Innovation in Europe*. http://www.acare4europe.org

[2] Strategic Research and Innovation Agenda, ACARE.

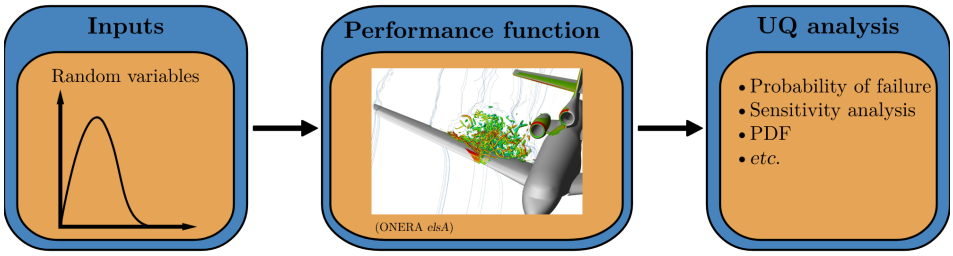
[3] Flightpath 2050: Europe's vision for aviation.

[4] Clean Sky I: 2008-2017. Clean Sky II: 2014-2024. http://www.cleansky.eu

[5] CLEEN I: 2010-2015. CLEEN II: 2015-2020. CLEEN III: 2020-2025.

cope with errors during execution, and with erroneous inputs.

The certification of the performance of an aircraft, or a system more generally, is often formulated using statistical concepts, where the objective functions are the mean and standard deviation of the objective functions assigned to deterministic or worst-case scenarios. However, it is not enough to probe the system by performing a limited number of so-called "hero calculations", or even computing the mean performance and design margins by means of extensive sampling. Instead, it is essential to be able to predict the system performance with rigorously quantified uncertainties. By rigorous we specifically mean that the performance measures (outputs) assigned to the design processes are certified by mathematically provable bounds, of which sharpness is not jeopardized by the aggregate of multiple sources of uncertainties. In this context, certification is understood as the process of guaranteeing that the probability of exceeding a given threshold a is below an acceptable tolerance ε, which is typically small. That is, one wants to certify that $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a] \leq \varepsilon$, where F is the performance function and $\mathbf{X}$ are the random variables following the probability measure $\mu^\dagger$, which influence the performance F. The probability $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a]$ will be called "probability of failure" in this work. The acceptable tolerance ε vastly differs from one application to another: $\varepsilon = 10^{-9}$ in the aviation industry (crash of an aircraft) [3, 13], $\varepsilon = 0$ in the seismic design of nuclear power plants [43, 50], $\varepsilon = 0.5$ for bridges, *etc.* In practical cases, the performance function F and/or the probability measure $\mu^\dagger$ may not be known, or only partially. For robust certification, it is thus desirable to develop concepts and methods by which the variability or uncertainty affecting the operational or embedded inputs of the design processes are rigorously taken into account and quantified. Determining how likely certain outputs are if some aspects of the system are not exactly known is the main topic of the so-called science of Uncertainty Quantification (UQ). An UQ analysis can be described as in Figure 1. Given some input random variables $\mathbf{X}$ following the probability measure $\mu^\dagger$, determine some output functions of interest such as the probability of failure, *i.e.* $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a]$, or the probability density function (PDF) through the performance function F.



Process of an UQ analysis.

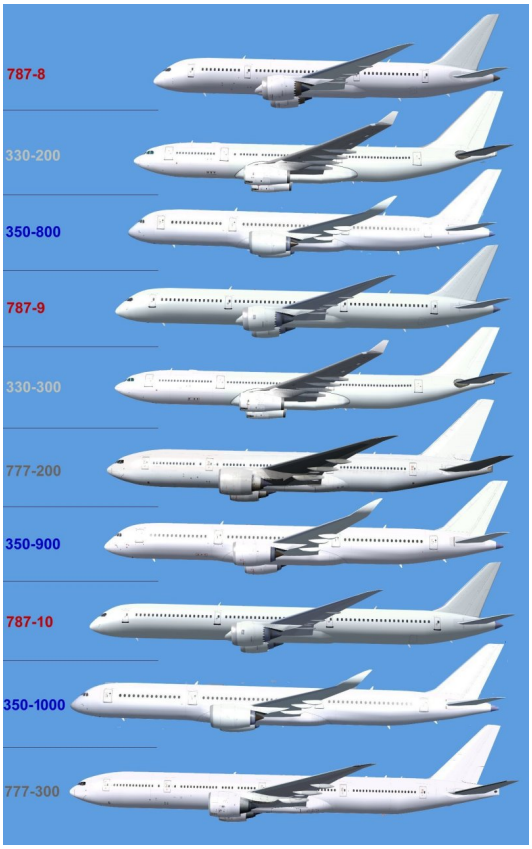
Depending on the available information, different methods exist in certification context in order to handle uncertainties: Monte-Carlo processes [89], Bayesian

inference [9, 88], stochastic expansion methods [5, 39, 56, 57, 87], antioptimization [48], *etc.* However, these methods require assumptions that are often not met in practice and even vary from one system to another. The main goal of this thesis is to develop a rigorous approach of UQ in order to robustly certify the operational efficiency of an aircraft. For instance, concentration-of-measure inequalities [16] are a powerful tool for rewording these needs into rigorous and precise mathematical terms, though they are seldom used in engineering applications at this time. Actually, thanks to the increase of computational power, the use of powerful software to simulate, notably, the flight of an aircraft and design its shape has been on a sharp rise during the last decades. In that respect, a numerical model is built in order to mimic the reality and to account for physical phenomena underlying the system. The mathematical formulation of a numerical model can be represented as

$$\begin{aligned} F\colon \mathcal{X} \subset \mathbb{R}^d &\to \mathcal{Y} \subset \mathbb{R} \\ \mathbf{X} &\mapsto F(\mathbf{X}) = Y, \end{aligned}$$

where F is a quantity of interest given by the numerical model, and $\mathbf{X}$ is a vector of the d input variables living in $\mathcal{X}$, the input set. The numerical model and the performance function share the same notation because the performance function is computed through the numerical model in practical cases. The input variables $\mathbf{X}$ gather the parameters of the system, which can be geometrical parameters, physical parameters, operational conditions, numerical error sources, *etc.* In aerodynamics, the input parameters are classically the angle of attack of the airfoil, the Mach number, the shape of the airfoil, *etc.* In this thesis, the numerical model F is considered to be deterministic. That is, running it twice with exactly the same inputs yields the same output. Assuming that the numerical model F is deterministic does not prevent from studying the effect of the input uncertainties on the output given by F. Of course, the quality of the output will have an impact on the fidelity of the system representation. For high-fidelity numerical model, a standard run can last from a few hours to a few days. In that respect, the costly, high-fidelity numerical model F can be replaced by a low-cost model G to bypass this obstacle. This is what is called surrogate modeling. Such methods allow to simplify the model F in order to considerably reduce the running time without loosing too much fidelity. Ideally, the low-cost numerical model G must be built such that $G \equiv F$. Nevertheless, engineers or regulatory agencies do not totally trust the results obtained through these simulations so far. Thus certification by means of costly full-scale experiments is still vastly used. In terms of security and environmental policy, the rules of certification are very conservative. Feedback from experiments acquired over a long period of time are used. Hence, design of aircraft has seen very little change over the last half-century, as demonstrated by the sketch below: Airbus' and Boeing's aircraft are actually very similar to each other.

In recent years, UQ and robust design methodologies have been addressed by a growing number of researchers and applications. An exponential growth of the number of publications is observed in this emerging area, of major importance for the future of



Boeing's and Airbus' airplanes picture comparison.

virtual prototyping and risk management. The Optimal Uncertainty Quantification (OUQ) framework recently developed by Owhadi *et al.* [2, 64, 72, 75, 112, 144] could fill the gap between numerical modeling and rigorous certification. It provides mathematically rigorous tools that can be important assets for robust certification through numerical modeling.

Outline of the thesis

This thesis work is first focused on the application of the OUQ framework in Chapter 1. In Section 1.1, we remind concentration-of-measure inequalities of the McDiarmid type, which can be used in a robust design certification context. Then we recall the OUQ framework in Section 1.2. In Section 1.2.1, we apply this framework in the case of McDiarmid's inequality, and then in Section 1.2.2 we recall that this framework can be applied in much more general cases. It arises as an optimization problem and can be solved thanks to the reduction Theorem 1 stated on page 12. A specific software to solve this optimization problem as well as several examples are presented in Section 1.3. The `mystic` framework is presented in Section 1.3.1. Then, several cases are considered. Firstly, the bending of a circular clamped beam

is studied in Section 1.3.2 to figure out simply the OUQ framework. Secondly, cases from the aeronautical domain are considered in Section 1.3.3 and Section 1.3.4 in order to rigorously bound the variations of performance functions classically studied in aerodynamics, such as the lift or the drag. Nevertheless, even after reduction of the optimization problem, this approach remains complex and the accuracy of the obtained results is thus impacted.

Then in Chapter 2, an algorithm is formulated that allows to ensure the validity of the numerical results obtained in Chapter 1. This algorithm is analogous to isotonic regression but differs on the assumptions made—see Section 2.1. An overview of the problem considered is presented in Section 2.2. Several useful notations are defined in Section 2.3. The mathematical convergence of the algorithm is demonstrated in Section 2.4. In Section 2.5, it is validated on simple validation test cases. Then, it is tested on a more complex aerodynamic case in Section 2.6. Nevertheless, the application of the OUQ framework requires to evaluate the performance function F of the system several tens of thousands of times. Therefore, the use of a time-consuming model is not feasible and a computationally inexpensive model is required.

Chapter 3 outlines different methods to build surrogate models. The general problem considered in regression is first introduced in Section 3.1. Then several methods are presented in Section 3.2 for the construction of surrogate models that are inexpensive to evaluate in a regression context. In Section 3.3, a first approach based on the Kernel Flow algorithm initially applied to a classification problem is extended to a regression problem for the determination of the kernel (covariance) function in Gaussian process regression. Then two new algorithms are presented in Section 3.4: the sparse spectral kernel ridge regression algorithm detailed in Section 3.4.3.2, and the non-sparse spectral kernel ridge regression algorithm detailed in Section 3.4.3.3. These two algorithms are based on a kernel function built within Mercer's framework, of which a brief summary is first given in Section 3.4.1. They also rely on the equivalence between the Gaussian process regression approach and the Reproducing Kernel Hilbert Space (RKHS) approach to determine an appropriate kernel. In Section 3.5, a remainder about the Polynomial Chaos Expansion (PCE) method is given for completeness. Finally, these different constructions of surrogate models are applied to synthetic test cases in Section 3.6 and more complex objective functions pertaining to aerodynamic applications in Section 3.7.

Chapter 1

Certification with uncertainties

Contents

In this chapter, an approach for the certification of a system under uncertainties is outlined. In Section 1.1 we will first show that certification can be achieved resorting to Concentration-of-Measure inequalities, more particularly of the McDiarmid type. Then we will introduce in Section 1.2 the Optimal Uncertainty Quantification (OUQ) framework developed by Owhadi and his colleagues in the early 2010s. This framework allows us to obtain optimal bounds on the probability of occurrence of a given scenario given a certain class of assumptions on the system by transforming an optimization problem in an a priori infinite-dimensional set to a finite-dimensional one through a reduction theorem. In Section 1.3, the `mystic` *numerical framework is presented, which is used to solve the newly obtained finite-dimensional optimization problem. Then several numerical examples including aerodynamic test cases are carried out using the* OUQ *and* `mystic` *frameworks.*

1.1 Certification using concentration-of-measure inequalities

Concentration-of-measure phenomenon refers to the observation that functions depending on a large number of variable parameters with controlled oscillations

in each variable are almost constant. The fluctuations about that constant mean values are in addition certified by rigorous inequalities, the so-called Concentration-of-Measure (CoM) inequalities [16, 84] (a nice survey on CoM is also given by T. Tao on his blog [148]). In this thesis, we shall mainly work with bounded-differences inequalities of the McDiarmid type [100], which bounds the fluctuations of a real function $\mathbf{X} \mapsto F(\mathbf{X})$ away from its mean $\mathbb{E}\{F(\mathbf{X})\}$ without *a priori* knowledge of the probability measure $\mathbb{P}$ of the random variables $\mathbf{X} = (X_1, \ldots, X_d)$. Assuming that the latter are mutually independent and take their values in the set $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$, where, say, $\mathcal{X}_j \subset \mathbb{R}$, for $j = 1, \ldots, d$, one has for all $a > 0$:

$$\mathbb{P}\left[|F(\mathbf{X}) - \mathbb{E}\{F(\mathbf{X})\}| \geq a\right] \leq \exp\left(-2\frac{a^2}{D_F^2}\right), \tag{1.1}$$

where $D_F = (\sum_{j=1}^d D_{F_j}^2)^{\frac{1}{2}}$ is the verification diameter [92] of the function F, and D_{F_j}, $j = 1, \ldots, d$, are its subdiameters defined by:

$$D_{F_j} = \sup_{\mathbf{x} \in \mathcal{X}} \sup_{x_j' \in \mathcal{X}_j} \left|F(x_1, \ldots x_{j-1}, x_j, x_{j+1}, \ldots x_d) - F(x_1, \ldots x_{j-1}, x_j', x_{j+1}, \ldots x_d)\right|. \tag{1.2}$$

This is the range of fluctuations of F if one freezes all but its j-th variable for some $1 \leq j \leq d$. These subdiameters can be viewed as nonlinear sensitivity indices. They can be used to quantify the range of variation of the performance function F from the ranges of variation of the input parameters $\mathbf{X}$.

To figure out what is meant by this result, one may wish to apply it for the simple case $F(\mathbf{X}) = \frac{1}{d}\sum_{j=1}^d X_j$ and $\mathcal{X}_j = [a_j, b_j]$, for which $D_{Fj} = \frac{1}{d}(b_j - a_j)$ and the following Hoeffding's inequality holds [66]:

$$\mathbb{P}\left[|F(\mathbf{X}) - \mathbb{E}\{F(\mathbf{X})\}| \geq a\right] \leq \exp\left(\frac{-2a^2d^2}{\sum_{j=1}^d (b_j - a_j)^2}\right).$$

So if the random variables X_j roughly have the same ranges of variation $b_j - a_j \simeq \Delta$, the above bound is $\exp(-2d(a/\Delta)^2)$. It says that the probability to deviate from the mean is exponentially decreasing with d, the number of random variables. This fundamental result questions the relevance of the objective of handling many sources of uncertainty simultaneously. In addition, the fact that these inequalities do not require any information on the marginal probability density functions (PDFs) of the random variables $\mathbf{X}$ is clearly an advantage in design processes of complex systems, where experimental data may often be scarce. Indeed, the sole data about the random variables $\mathbf{X}$ that are strictly necessary in the above estimates are their ranges of variation $\mathcal{X}$, independently of their marginal distributions.

Remark 1. *If the marginal distribution of each input parameter is known, Equation (1.1) can be improved by partitioning the input parameter domain. Sullivan et al. [143] have shown that successive well-chosen divisions generate upper bounds of the "probability of failure" (1.1) that converge to the exact probability of failure in the limit of smaller and smaller divisions.*

1.1.1 Application to post-optimization design

Assume now that F is a performance measure of the system under consideration, such as a drag coefficient for a profile which has to be controlled for varying operational conditions (Mach number or wind speed, angle of attack, *etc.*), varying geometries resulting from manufacturing tolerances, impacts or icing, numerical error sources, or uncertain physical model parameters, for example [27, 36, 67]. The performance is formulated as the objective $F(\mathbf{X}) \leq a$ where a is the threshold for the economic operation of the system. The parameters $\mathbf{X}$ are the aforementioned sources of uncertainties. Then a direct application of McDiarmid's inequality (1.1) yields the following certification criterions

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \exp\left(-2\frac{(a - \mathbb{E}\{F(\mathbf{X})\})_+^2}{D_F^2}\right), \tag{1.3}$$

$$\mathbb{P}[F(\mathbf{X}) \leq a] \leq \exp\left(-2\frac{(\mathbb{E}\{F(\mathbf{X})\} - a)_+^2}{D_F^2}\right), \tag{1.4}$$

where $x_+ := \max(0, x)$ (this cutting stems from the fact that if the mean performance is $\mathbb{E}\{F(\mathbf{X})\} \geq a$ then very little chance remains to certify the system). Introducing the design margin $M = (a - \mathbb{E}\{F(\mathbf{X})\})_+$ and the uncertainty measure $U = D_F$ of that system, it follows that the latter is guaranteed to achieve the desirable performance criterion with the tolerance ε provided that the confidence factor $\delta = \frac{M}{U}$ [92, 151] satisfies $\exp(-2\delta^2) \leq \varepsilon$, or $\delta \geq (-\frac{1}{2}\log\varepsilon)^{\frac{1}{2}}$. Thus not only do CoM inequalities provide with rigorous certification criteria, but they also yield rigorous quantitative definitions of design margins, system uncertainties, and confidence factors [92, 151]. The above bound (1.3) ensures that the mean performance $\mathbb{E}\{F(\mathbf{X})\}$ achieved by the design after, say, optimization processes involving various design parameters, remains within an acceptable range once variable operational and/or geometrical conditions are envisaged—hence the terminology post-optimization design.

1.1.2 Post-optimization using simulations

Actually it is expected in the above that the behavior of the system as embodied in the performance function $\mathbf{X} \mapsto F(\mathbf{X})$ is known extensively, including its mean performance $\mathbb{E}\{F(\mathbf{X})\}$, in order to compute its diameter D_F. For example, it could be assessed through intensive experiments or high-fidelity numerical simulations. Both approaches are highly expensive and one may wish to achieve robust certification in a post-optimization process using cheaper tools, such as a lower-fidelity numerical model or a surrogate model. Denoting the latter by $\mathbf{X} \mapsto G(\mathbf{X})$, the "true" performance obtained by experiments or high-fidelity simulations yields different outcomes from this numerical surrogate—in other words, $G(\mathbf{X}) \neq F(\mathbf{X})$ in general. Consequently $E = F - G$ may be regarded as the modeling error function of the system. This function can be used to estimate the verification diameter D_F by simply observing that owing to a standard triangular inequality one has $D_F \leq D_G + D_{F-G}$ [75]. Thus D_F is replaced by $D_G + D_{F-G}$ in Equation (1.3) and Equation (1.4) above to bound

the certification criterion. Indeed, applying once again McDiarmid's inequality (1.1) to the system response F with reference to its surrogate G it is deduced that:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \exp\left(-2\frac{(a-\mathbb{E}\{F(\mathbf{X})\})_+^2}{D_F^2}\right) \leq \exp\left(-2\frac{(a-\mathbb{E}\{F(\mathbf{X})\})_+^2}{(D_G+D_{F-G})^2}\right), \quad (1.5)$$

and

$$\mathbb{P}[F(\mathbf{X}) \leq a] \leq \exp\left(-2\frac{(\mathbb{E}\{F(\mathbf{X})\}-a)_+^2}{D_F^2}\right) \leq \exp\left(-2\frac{(\mathbb{E}\{F(\mathbf{X})\}-a)_+^2}{(D_G+D_{F-G})^2}\right). \quad (1.6)$$

In this setting D_{F-G} stands for the validation diameter [92] that measures the discrepancies between the surrogate predictions and the experimental observations or high-fidelity simulations. Interestingly, one may expect that the error function E is better behaved than F or G separately because it should vary only slightly—see Figure 1.1. Hence the computation of D_{F-G} may be performed by means of rapidly converging iterative schemes with a minimum recourse to evaluations of F (experiments or high-fidelity simulations). Besides, the computation of D_G is based on intensive calls to the surrogate G, which is however presumably inexpensive to evaluate.

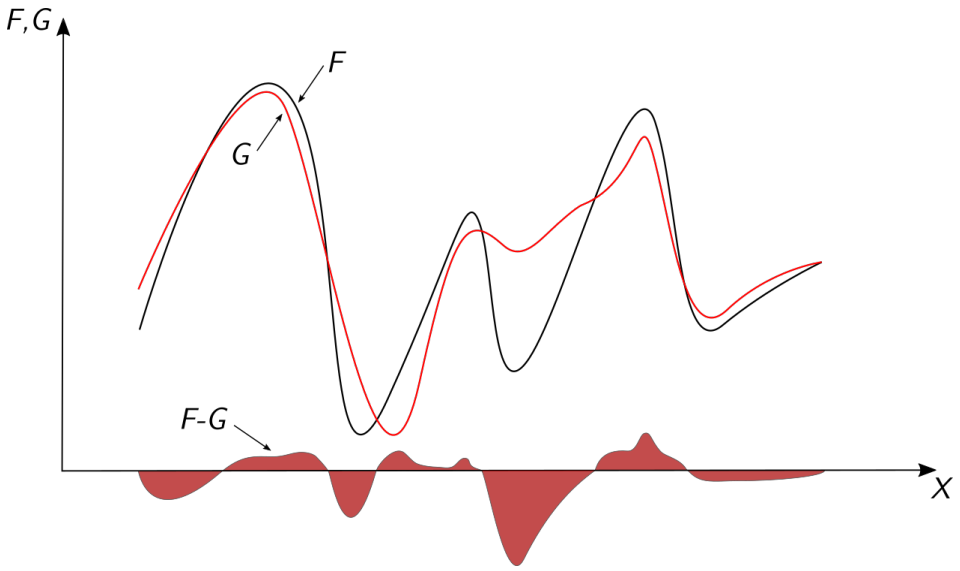


Figure 1.1: Error function $E = F - G$. The response function G is close to the "true" performance F except at some localized areas.

1.2 Optimal concentration inequalities

1.2.1 Optimal McDiarmid's inequality

In pratical problems, we often do not exactly know the performance function F and the probability measure $\mu^\dagger$ of $\mathbf{X}$ (hence $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}$). This absence of information,

usually called epistemic uncertainty, means that we have to consider not an unique pair (f, μ) which could be the reality $(F, \mu^\dagger)$ but an admissible set $\mathcal{A}$ of measures μ and functions f. This admissible set $\mathcal{A}$ does not uniquely determine the pair (f, μ) but yields admissible scenarios (f, μ) which could be $(F, \mu^\dagger)$ given the available information that one has about the problem. Available information could be obtained for example from expert judgement or experimental data. In a robust certification mindset, this leads naturally to compute the best and worst cases of the true probability $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a]$ given the information included in $\mathcal{A}$. That is, one seeks to compute the optimal lower and upper bounds:

$$\inf_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a] \leq \mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a] \leq \sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]. \tag{1.7}$$

Throughout this thesis, we simply denote $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}$ by $\mathbb{P}$ for the true probability measure, and $\mathbb{E}_{\mathbf{X}\sim\mu^\dagger}$ by $\mathbb{E}$ for the true expected value. Also the probability $\mathbb{P}[F(\mathbf{X}) \geq a]$ is called the "probability of failure" for it is the quantity of interest one aims at bounding. First, we define the admissible set $\mathcal{A}_{\text{McD}}$ which corresponds to the assumptions of McDiarmid's inequality [100] as follows:

$$\mathcal{A}_{\text{McD}} = \left\{ (f,\mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_d \to \mathbb{R} \\ \mu = \bigotimes_{j=1}^d \mu_j \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{f(\mathbf{X})\} = \mathbb{E}\{F(\mathbf{X})\} \\ D_{f_j} \leq D_{F_j}\,,\ j = 1, \ldots, d \end{array} \right\}. \tag{1.8}$$

$\mathcal{A}_{\text{McD}}$ is the set of response functions f and probability measures μ that could be $\mathbf{X} \mapsto F(\mathbf{X})$ and $\mu^\dagger$ given the information about the constraints imposed by the verification diameters $\{D_{F_j}\}_{j=1}^d$, the expected value $\mathbb{E}\{F(\mathbf{X})\}$ of the performance function F, and the independence of the input variables $\mathbf{X}$. The admissible set $\mathcal{A}_{\text{McD}}$ represents here epistemic uncertainty because the response function F and the PDFs of the input parameters $\mathbf{X}$ are not exactly known. Recalling Equation (1.1), concentration-of-measure inequalities lead to suboptimal bounds. Indeed, it has been shown in [112] that we can obtain the optimal upper bounds of the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ by resorting to an optimization problem. By optimal, we mean that we cannot tighten anymore the upper bounds of the probability of failure given the information included in $\mathcal{A}_{\text{McD}}$. In that way, we want to determine the least upper bounds of that probability given the information included in $\mathcal{A}_{\text{McD}}$. Thus, we are seeking to calculate

$$\mathcal{U}(\mathcal{A}_{\text{McD}}) = \sup_{(f,\mu)\in\mathcal{A}_{\text{McD}}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]\,, \tag{1.9}$$

where:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \mathcal{U}(\mathcal{A}_{\text{McD}}) \leq \exp\left(-2\frac{(a - \mathbb{E}\{F(\mathbf{X})\})_+^2}{D_F^2}\right). \tag{1.10}$$

The solution of Equation (1.9) is generally not tractable because the admissible set $\mathcal{A}_{\text{McD}}$ is possibly infinite-dimensional. Nevertheless, Owhadi *et al.* have demonstrated

in [112] that the computation of $\mathcal{U}(\mathcal{A}_{\text{McD}})$ can be reduced to a finite-dimensional admissible set of convex combinations of Dirac masses. This set, denoted by $\mathcal{A}_\Delta$, is defined as follows:

$$\mathcal{A}_\Delta = \left\{ (f,\mu) \in \mathcal{A}_{\text{McD}} \;\middle|\; \begin{array}{c} \mu_k = \alpha^k \delta_{x_1^k} + (1-\alpha^k)\delta_{x_2^k}\,,\ k \in [\![1;d]\!] \\ \alpha^k \geq 0 \\ x_1^k, x_2^k \in \mathcal{X}_k \end{array} \right\}, \tag{1.11}$$

with [112]:

$$\mathcal{U}(\mathcal{A}_{\text{McD}}) = \mathcal{U}(\mathcal{A}_\Delta).$$

For an input vector parameter $\mathbf{X}$ of dimension one, two and three, one can exhibit the explicit value of $\mathcal{U}(\mathcal{A}_{\text{McD}})$. For higher dimensions, the solution can be found in [112, Prop. 5.7]. For instance, in the case of dimension two and when $\mathbb{E}\{F(\mathbf{X})\} \leq a$, we have [112, Th. 5.2]:

$$\mathcal{U}(\mathcal{A}_{\text{McD}}) = \left\{ \begin{array}{ccl} 0 & \text{if} & D_{F_1} + D_{F_2} \leq a - \mathbb{E}\{F(\mathbf{X})\}\,, \\ \frac{(D_{F_1}+D_{F_2}-a+\mathbb{E}\{F(\mathbf{X})\})^2}{4D_{F_1}D_{F_2}} & \text{if} & |D_{F_1} - D_{F_2}| \leq a - \mathbb{E}\{F(\mathbf{X})\} \leq D_{F_1} + D_{F_2}\,, \\ 1 - \frac{a-\mathbb{E}\{F(\mathbf{X})\}}{\max(D_{F_1},D_{F_2})} & \text{if} & 0 \leq a - \mathbb{E}\{F(\mathbf{X})\} \leq |D_{F_1} - D_{F_2}|\,. \end{array} \right. \tag{1.12}$$

For this specific case, one can note that if $0 \leq a - \mathbb{E}\{F(\mathbf{X})\} \leq |D_{F_1} - D_{F_2}|$, the optimal bound does not depend on $\min(D_{F_1}, D_{F_2})$. In other words if, for instance, most of the uncertainty comes from the first variable (*i.e.* $D_{F_1} \geq a - \mathbb{E}\{F(\mathbf{X})\} + D_{F_2}$), then the uncertainty associated with the second variable does not contribute to the global uncertainty. In order to reduce the least upper bound, a reduction in D_{F_1} is required. The inequality $D_{f_2} \leq D_{F_2}$ in $\mathcal{A}_{\text{McD}}$ is a non-binding information. We have a "screening effect" [145, Example 14.26].

1.2.2 General case

This framework can be applied to more general types of admissible sets. However, in most of these cases, no analytical expression such as Equation (1.12) can be given. Nevertheless, one can solve the associated optimization problem through the reduction theorem and the `mystic` framework [101, 102]. This framework will be more detailed in Section 1.3.1. The reduction theorem is stated as follows:

Theorem 1 (Reduction theorem [112, 145])**.** *Suppose that $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ is a product of Radon spaces. Let $\mathcal{A}$ be a moment admissible set:*

$$\mathcal{A} = \left\{ (f,\mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_d \to \mathbb{R} \text{ is a measurable function;} \\ \text{some conditions on } f \text{ alone;} \\ \mu = \bigotimes_{k=1}^d \mu_k \in \bigotimes_{k=1}^d \mathcal{M}_1(\mathcal{X}_k) \text{ and for each } f; \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{\phi_j\} \leq 0\,, \text{ for } j \in [\![1;n_0]\!]; \\ \mathbb{E}_{X_k\sim\mu_k}\{\phi_j^{(k)}\} \leq 0\,, \text{ for } j \in [\![1;n_k]\!] \text{ and for } k \in [\![1;d]\!] \end{array} \right\}, \tag{1.13}$$

where $\mathcal{M}_1(\mathcal{X}_k)$ is the set of all probability measures on $\mathcal{X}_k$ [145, Definition 2.3], $\phi_j : \mathcal{X} \to \mathbb{R}$ and $\phi_j^{(k)} : \mathcal{X}_k \to \mathbb{R}$ are prescribed measurable functions. Let $\mathcal{U}(\mathcal{A})$ be

the optimal upper bound of the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ *given the available information in the the admissible set* $\mathcal{A}$*, that is*

$$\mathcal{U}(\mathcal{A}) = \sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]\,, \tag{1.14}$$

then we have $\mathcal{U}(\mathcal{A}) = \mathcal{U}(\mathcal{A}_\Delta)$ *where:*

$$\mathcal{A}_\Delta = \left\{ (f,\mu) \in \mathcal{A} \;\middle|\; \begin{array}{c} \mu_k = \sum_{i_k=1}^{N_k} \alpha_{i_k}^k \delta_{x_{i_k}^k} \text{ where } N_k = 1 + n_0 + n_k \text{ with } k \in [\![1;d]\!]; \\ \alpha_{i_k}^k \geq 0\,,\ \sum_{i_k} \alpha_{i_k}^k = 1; \\ x_{i_k}^k \in \mathcal{X}_k \end{array} \right\}. \tag{1.15}$$

Remark 2. *The more general reduction theorem [145, Theorem 14.16] says, loosely speaking, that if for each* f *defined on* $\mathcal{X} \subset \mathbb{R}^d$*,* $\{\mu \in \mathcal{M}_1(\mathcal{X}) \mid (\mu, f) \in \mathcal{A}\}$ *is specified by a system of* m *equality or inequality constraints on expected values of arbitrary test functions under* μ*, then for the determination of* $\mathcal{U}(\mathcal{A})$ *it is sufficient to consider only distributions* μ *that are convex combinations of at most* $m+1$ *point masses; the optimization variables are then the* m *independent weights and* $m+1$ *locations in* $\mathcal{X}$ *of these point masses. If* μ *factors as a product of distributions (i.e.* $\mathbf{X}$ *is a vector with independent components), then the reduction theorems apply componentwise.*

Remark 3. *From Theorem 1, a probability measure* $\mu \in \mathcal{A}_\Delta$ *has the following form*

$$\mu = \bigotimes_{k=1}^{d} \sum_{i_k=1}^{N_k} \alpha_{i_k}^k \delta_{x_{i_k}^k} = \sum_{\mathbf{i}=(1,\ldots,1)}^{(N_1,\ldots,N_d)} \alpha_{\mathbf{i}} \delta_{\mathbf{x_i}}$$

where

$$\begin{aligned} \alpha_{\mathbf{i}} &= \alpha_{i_1}^1 \alpha_{i_2}^2 \ldots \alpha_{i_d}^d \geq 0, \\ \mathbf{x_i} &= (x_{i_1}^1, x_{i_2}^2, \ldots, x_{i_d}^d) \in \mathcal{X}. \end{aligned}$$

That is, the support of μ *is a rectangular grid in* $\mathcal{X}$*.*

Remark 4. *Theorem 1 has been formulated for the optimal upper bound* $\mathcal{U}(\mathcal{A})$*, but likewise, it is also applicable for the optimal lower bound* $\mathcal{L}(\mathcal{A}) = \inf_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$*.*

Remark 5. *Theorem 1 is not limited to the case where the joint law is a product measure (*i.e. *mutually independent random variables). This assumption was used because this piece of information is vastly encountered in practical problems. One can integrate other kinds of constraints such as the variance of one of the random input variable, the mean of another one, the correlations between inputs, etc. More information can be found in [145, Theorem 14.16].*

Remark 6. *If one is interested in certifying that the probability of the performance function F is below a specified threshold a is below a tolerance ε, that is $\mathbb{P}[F(\mathbf{X}) \leq a] \leq \varepsilon$, instead, Theorem 1 is also applicable. In this case, we are seeking to determine the optimal lower and upper bounds define as*

$$\inf_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq a] \leq \mathbb{P}[F(\mathbf{X}) \leq a] \leq \sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq a], \tag{1.16}$$

and Theorem 1 holds by simply replacing $\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$ by $\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq a]$.

Thanks to this theorem, the infinite-dimensional optimization problem of Equation (1.14) can be reduced to a finite-dimensional problem where each marginal measure μ_k is supported on $N_k = 1 + n_0 + n_k$ points of the parameter space $\mathcal{X}_k$, for n_0 being the number of global generalized moment constraints, and n_k being the number of local generalized moment constraints for each random input variable X_k. That is, each μ_k is the convex combination of N_k Dirac measures. The support of μ is a rectangular grid in $\mathcal{X}$. In that way, the search over the admissible measures μ is now finite-dimensional. Only the values of f on the finite support of μ and nowhere else are relevant. In that respect, we can evaluate the probability of failure $\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$ as follows:

$$\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a] = \sum_{\mathbf{i}=(1,\ldots,1)}^{(N_1,\ldots,N_d)} \alpha_{\mathbf{i}} \mathbb{1}_{[a,+\infty[}[f(\mathbf{x_i})], \tag{1.17}$$

where

$$\mathbb{1}_{[a,+\infty[}[y] = \begin{cases} 1 & \text{if } \ y \in [a,+\infty[, \\ 0 & \text{otherwise.} \end{cases} \tag{1.18}$$

Additionally, the mean performance function can be calculated as follows:

$$\mathbb{E}_{\mathbf{X}\sim\mu}\{f(\mathbf{X})\} = \sum_{\mathbf{i}=(1,\ldots,1)}^{(N_1,\ldots,N_d)} \alpha_{\mathbf{i}} f(\mathbf{x_i}). \tag{1.19}$$

The computation of the OUQ bounds (see for example Equation (1.14) for the maximum) amounts now to finite-dimensional optimization problems in which the optimization variables are with $k \in [\![1; d]\!]$ and $i_k \in [\![1; N_k]\!]$:

- the positions of the support points $x_{i_k}^k \in \mathcal{X}_k$ of the discrete measure μ_k;
- the weights $\alpha_{i_k}^k$ of the support points $x_{i_k}^k$;
- the values of the performance function at the support points $f(x_{i_1}^1, \ldots, x_{i_d}^d)$.

Example 1 ($d = 3$). For instance, if one knows the mean M of the performance function $\mathbf{X} \mapsto F(\mathbf{X})$ and say the mean m_2 of the second input parameter x_2 where $\mathbf{X} = (X_1, X_2, X_3) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, we can define the admissible set $\mathcal{A}$:

$$\mathcal{A} = \left\{ (f,\mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to \mathbb{R} \\ \mu = \bigotimes_{k=1}^3 \mu_k \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{f(\mathbf{X})\} = \mathrm{M} \\ \mathbb{E}_{X_2\sim\mu_2}\{X_2\} = \mathrm{m}_2 \end{array} \right\}. \tag{1.20}$$

Finding the optimum $\mathcal{U}(\mathcal{A})$, which is *a priori* an infinite-dimensional optimization problem, can be reduced to a finite-dimensional problem by using the reduction theorem above. The reduced searching space is given by

$$\mathcal{A}_\Delta = \left\{ (f,\mu) \in \mathcal{A} \;\middle|\; \begin{array}{c} \mu_k \in \Delta_1(\mathcal{X}_k), k = 1,3 \\ \mu_2 \in \Delta_2(\mathcal{X}_2) \end{array} \right\}, \tag{1.21}$$

where:

$$\begin{aligned} \Delta_1(\mathcal{X}_k) &= \{\alpha_1^k \delta_{x_1^k} + \alpha_2^k \delta_{x_2^k} \mid x_1^k, x_2^k \in \mathcal{X}_k \text{ and } \alpha_1^k + \alpha_2^k = 1\} \text{ for } k = \{1,3\}, \\ \Delta_2(\mathcal{X}_2) &= \{\alpha_1^2 \delta_{x_1^2} + \alpha_2^2 \delta_{x_2^2} + \alpha_3^2 \delta_{x_3^2} \mid x_1^2, x_2^2, x_3^2 \in \mathcal{X}_2 \text{ and } \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1\}. \end{aligned}$$

Thus, the probability of failure $\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$ is calculated as follows:

$$\mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a] = \sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{2} \alpha_i^1 \alpha_j^2 \alpha_k^3 \, \mathbb{1}_{[a,+\infty[}\left(f\left(x_i^1, x_j^2, x_k^3\right)\right). \tag{1.22}$$

The optimization problem amounts now to determine with $i \in [\![1;2]\!]$, $j \in [\![1;3]\!]$ and $k \in [\![1;2]\!]$:

- the positions of the support points $\left(x_i^1, x_j^2, x_k^3\right) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$ of the discrete measure μ;
- the weights $\left(\alpha_i^1, \alpha_j^2, \alpha_k^3\right) \in [0,1]^3$ of the support points;
- the values of the performance function at the support points $f\left(x_i^1, x_j^2, x_k^3\right)$.

The support of μ is represented on Figure 1.2.

1.2.3 Canonical moments

Recently, a new method has been developed by J. Stenger *et al.* [138, 139] in order to solve the optimization problem detailed in Theorem 1. The main idea of this method is to transform the optimization of the optimal bounds $\mathcal{U}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$ under generalized moment constraints on the probability measure μ to an unconstrainted optimization. In this way, one does not have to resort to numerical optimization through `mystic`, for instance. The method is derived from the theory of canonical moments [37, 134]. It is summarized here following the same structure and notations as in [138]. For simplicity, we will consider that $\mathcal{X} = \mathcal{X}_1 = [a_1, b_1] \subset \mathbb{R}$ with $a_1 < b_1 \in \mathbb{R}$ and $\mu = \mu_1 \in \mathcal{M}_1(\mathcal{X})$. In addition, we will assume that N generalized moment constraints are imposed on the probability measure μ. From Theorem 1, it reads

$$\mu = \sum_{i=1}^{N+1} \alpha_i \delta_{x_i}, \tag{1.23}$$

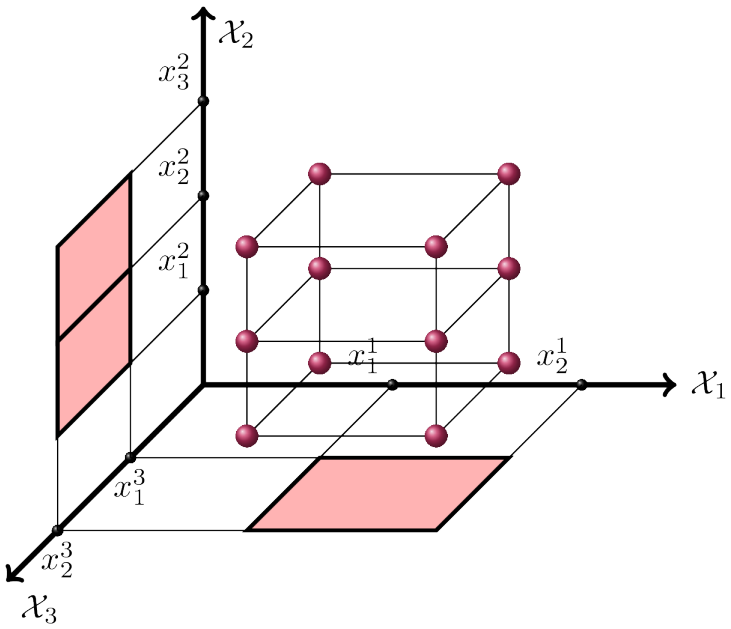


Figure 1.2: Support of μ when $d = 3$ and when we have one global generalized moment constraint and one local generalized moment constraint on the second random input variable X_2 corresponding to Example 1. Marginal measures μ_1 and μ_3 are supported on at most two Dirac points while the marginal measure μ_3 is supported on at most three Dirac points. The support of μ consists in at most twelve Dirac support points.

where $\sum_{i=1}^{N+1} \alpha_i = 1$ with $\alpha_i \geq 0$ are the weights of the Dirac points, and $x_i \in \mathcal{X}$ are their corresponding positions. We start first with some generalities about canonical moments. We denote by M the moment space defined by

$$M = \{\mathbf{c}; \mu \in \mathcal{M}_1(\mathcal{X})\}, \tag{1.24}$$

where $\mathbf{c}$ is the sequence of all moments on the probability measure μ. The m-th moment space M_m is defined by

$$M_m = \{(c_1, \ldots, c_m); \mu \in \mathcal{M}_1(\mathcal{X})\}, \tag{1.25}$$

where $c_j = \int_{\mathcal{X}} x^j \mu(dx)$ is the j-th moment on the probability measure μ. For instance, if $m = 2$, then $c_1 = \mathbb{E}_{X\sim\mu}\{F(X)\}$ and $c_2 = \mathbb{E}_{X\sim\mu}\{F(X)^2\}$ are respectively the expected value and the variance of the performance function F. Then, we define the extremes values of the m-th moment as

$$c_m^+ = \max_{c\in\mathbb{R}}\{(c_1, \ldots, c_{m-1}, c) \in M_m\},$$
$$c_m^- = \min_{c\in\mathbb{R}}\{(c_1, \ldots, c_{m-1}, c) \in M_m\}.$$

That is, they represent the maximum and minimum values of the m-th moment that a probability measure μ can have when its $m-1$ moments are equal to $\mathbf{c}_{m-1}$. In

this way, the m-th canonical moment is defined as

$$p_m(\mathbf{c}) = \frac{c_m - c_m^+}{c_m^+ - c_m^-} \in [0,1]. \tag{1.26}$$

The canonical moments have as nice properties that they are in $[0,1]$ and they are invariant by any affine transformation of the support of the measure μ [37, Theorem 1.3.2]. However, after applying this affine transformation, the values of the classical moments will be affected. Denote by $\tilde{\mathbf{c}} = (\tilde{c_1}, \ldots)$ the values of the classical moments after the affine transformation on the measure μ, their values can be found in [139, Theorem 4.1]. In any case, for practical applications, moments of order two are often considered. The corresponding canonical moments can be calculated as

$$p_1 = c_1 \text{ and } p_2 = \frac{c_2 - c_1^2}{c_1(1-c_1)}. \tag{1.27}$$

For higher orders, the values of canonical moments can be found in [37, p. 23] through a recursive algorithm. From this point, returning to the initial problem, given the values of the canonical moments computed above, one wishes to obtain the support of the discrete measure of Equation (1.23); that is, find the positions of the Dirac points $\{x_i\}_{i=1}^{N+1}$ and their corresponding weights $\{\alpha_i\}_{i=1}^{N+1}$. This can be done through the Stieltjes transform [37]. To summarize, if μ has a finite support, then the Stieltjes transform S has the following expression

$$S(z) = \int_{a_1}^{b_1} \frac{\mu(dx)}{z-x} = \sum_{i=1}^{N+1} \frac{\alpha_i}{z - x_i}, \tag{1.28}$$

where $z \in \mathbb{C} \setminus \text{supp}(\mu)$. Equation (1.28) also reads

$$S(z) = \frac{Q_N(z)}{P_{N+1}(z)}, \tag{1.29}$$

where $P_{N+1}(z) = \prod_{i=1}^{N+1}(z - x_i)$ such that its roots are the positions $\{x_i\}_{i=1}^{N+1}$ of the Dirac measures [37, Theorem 3.6.1], and $Q_N(z) = \sum_{i=1}^{N} \alpha_i \prod_{j \neq i}(z - x_j)$ such that

$$\alpha_i = \frac{Q_N(x_i)}{\frac{dP_{N+1}(x)}{dx}|_{x=x_i}}, \; i = 1, \ldots, N+1, \tag{1.30}$$

are the associated weights. The polynomials $\{P_k\}_{k \geq 0}$ and $\{Q_k\}_{k \geq 1}$ are obtained as functions of the canonical moments, of which expressions can be found in [139, Theorem 4.5]. In fact, the weights $\{\alpha_i\}_{i=1}^{N+1}$ can be found by two different ways. The first one is by solving Equation (1.30). The second one is to notice that N generalized moment constraints are enforced on the probability measure μ, whose support is at most $N+1$ Dirac points from Theorem 1. These N constraints lead to N equations on the measure μ. Adding the fact that the sum of the weights $\{\alpha_i\}_{i=1}^{N+1}$

must be equal to one, we obtain the following set of equations:

$$\begin{cases} \alpha_1 & + & \dots & + & \alpha_{N+1} & = & 1 \\ \alpha_1 x_1 & + & \dots & + & \alpha_{N+1} x_{N+1} & = & c_1 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \alpha_1 (x_1)^{N+1} & + & \dots & + & \alpha_{N+1} (x_{N+1})^{N+1} & = & c_N, \end{cases} \tag{1.31}$$

As the positions of the support points $\{x_i\}_{i=1}^{N+1}$ are distinct, Equation (1.31) is a Vandermonde system. Therefore, the weights $\{\alpha_i\}_{i=1}^{N+1}$ are uniquely determined by inverting it. Nonetheless, a Vandermonde system is ill-conditioned [55] whenever the number of generalized moment constraint N is increasing. Thereby, it is advised to determine the weights $\{\alpha_i\}_{i=1}^{N+1}$ by using Equation (1.30). An algorithm describing the process of determining the positions of the Dirac points $\{x_i\}_{i=1}^{N+1}$ and its associated weights $\{\alpha_i\}_{i=1}^{N+1}$ in order to compute the probability of failure (1.17) is given in [138, 139].

1.3 Numerical examples

In this section, we first describe the `mystic` framework used throughout the thesis for the computation of the optimal upper bounds and the subdiameters. This framework enables us to compute efficiently the upper bound of the probability of failure of Equation (1.17). Then we show several examples. Firstly, the bending of a circular clamped beam is chosen to illustrate simply the OUQ framework. Secondly, we study the lift of a RAE2822 [34] wing profile using Lazereff's results in [83] and a Thin Plate Spline (TPS) interpolation method [44, 45]. Finally, we investigate the lift-to-drag ratio of a RAE2822 wing profile using the software ISES [40, 42] and a Polynomial Chaos Expansion (PCE) surrogate model. The PCE surrogate is derived by using a projection approach.

1.3.1 `mystic` framework

Even after using the reduction theorem, the optimization problem of Equation (1.17) can be high-dimensional and heavily constrained. A specific framework has to be used in order to be able to solve it. McKerns *et al.* [101, 102] have developed a software framework called `mystic` for high-dimensional global constrained optimizations. This framework provides a large number of optimization algorithms, both global and local, including gradient solvers. The package can be downloaded on the following link https://github.com/uqfoundation/mystic. The differential evolution (DE) algorithm of Price and Storn [140] is used to compute the subdiameters $\{D_{F_j}\}_{j=1}^{d}$—see Equation (1.2)—and to solve the finite-dimensional optimization problem—see Equation (1.14) and Equation (1.17). Such algorithm[1] is known as metaheuristic because very few or no assumptions are to be made about the optimization problem. For instance, the DE algorithm does not use the gradient of

[1] See for example https://en.wikipedia.org/wiki/Differential_evolution.

the problem, unlike gradient solvers. It means that the optimization problem does not have to be differentiable. Therefore, the DE algorithm can even be used over optimization problems which are not continuous or are noisy. Moreover, it is very suited to massively-parallel optimization, which could be very attractive as usually the evaluation of the model requires considerable time. A sketch of a DE algorithm is shown on Figure 1.3. DE algorithm is a part of the more general type of genetic algorithms. Assume that one wants to minimize a function $g\colon \mathcal{X} \mapsto \mathcal{Y}$. The steps of a DE algorithm can be described as follows:

- Initialization of the trial population;
- Mutation: For each individual $\mathbf{x} \in \mathcal{X}$ in the trial population, select three different ones $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ in the trial population at random and distinct from the individual $\mathbf{x}$. Compose a new vector by combining them, known as the "donor" vector;
- Crossover: The components of the individual $\mathbf{x}$ are randomly changed by the components of the donor vector. This new vector is known as the "trial" vector denoted by $\mathbf{x}_t$;
- Selection: If $g(\mathbf{x}_t) \leq g(\mathbf{x})$ then $\mathbf{x}$ is replaced by $\mathbf{x}_t$ in the trial solution;
- Do the same for each individual in the trial population;
- The best candidate solution from the trial population is taken if the stopping criterion is reached.

`mystic` provides flexibility about the choice of how to apply constraints. Two main type of methods are implemented. The first methods are penalty-based: one adds a penalty function to the cost function. Then, the constrained optimization problem is replaced by successive unconstrained problems. The available penalty methods in `mystic` are the exterior penalty function method [154], the augmented Lagrange multiplier method [73] and the logarithmic barrier method [70]. The second methods the set-based: the constraints are satisfied through kernel methods. That is, kernel methods apply a transform which reduces the search space so that the optimizer will only search over the set of candidates that satisfy the constraints. We use kernel methods because it allows us to separate the constraints from the optimization problem. Indeed, the evaluations of the constraints are totally decoupled from the evaluation of the cost function. The optimization is divided into two loops. The outer loop is over the cost function. The inner loop is over the constraints. In that way, one may expect that the number of function evaluations is greatly reduced as these evaluations are usually the most expensive step.

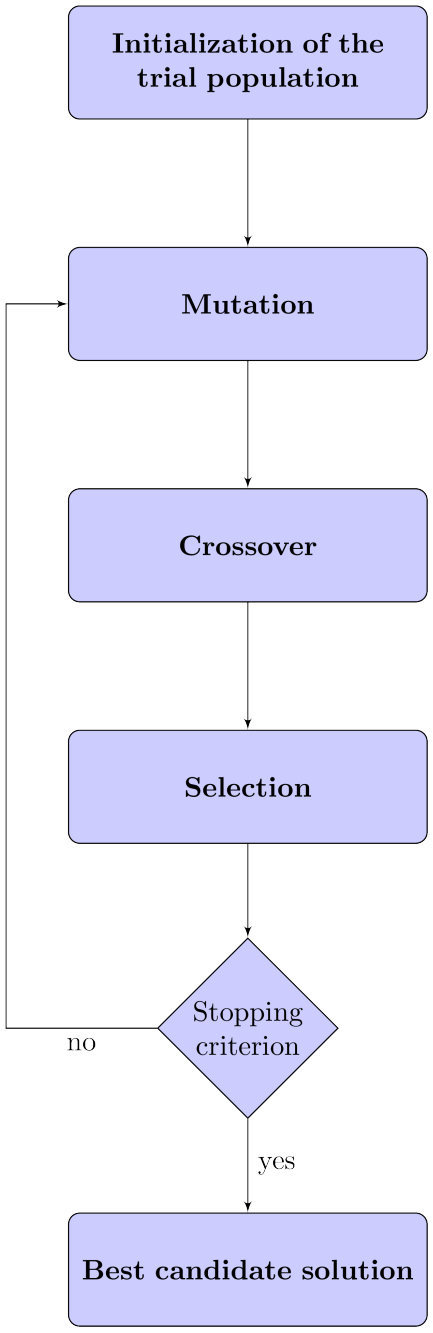


Figure 1.3: Sketch of a differential evolution algorithm.

1.3.2 Case 1: Bending of a beam

1.3.2.1 Setup

As a first step, a basic case is chosen to study simply the Optimal Uncertainty Quantification (OUQ) framework: the bending of a circular clamped beam under Euler-Bernoulli theory—see Figure 1.4.

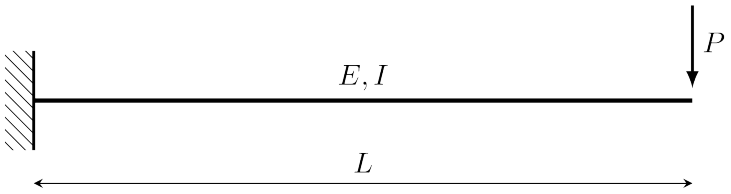


Figure 1.4: Setup of the clamped beam.

The objective is to certify that the probability that the maximum deflection of the beam—this is the performance function F—exceeds a specified threshold a is below a given tolerance ε. We call this probability the "probability of failure" and aim at bounding it when some geometrical and mechanical characteristics of the beam are variable (random inputs) and/or only partial information on F is available. We choose as the beam's random inputs the Young's modulus E and its radius R with their respective range—see Table 1.1. These ranges were obtained by taking $\underline{E} = 75$ GPa and $\underline{R} = 12.5$ mm as mean values for E and R with $E \in [\underline{E} \pm 5\%]$ and $R \in [\underline{R} \pm 5\%]$. Both define the input vector $\mathbf{X} = (E, R)$. Actually for this case, an analytical solution for the function F is available, namely

$$F(\mathbf{X}) = \frac{4PL^3}{3ER^4}. \tag{1.32}$$

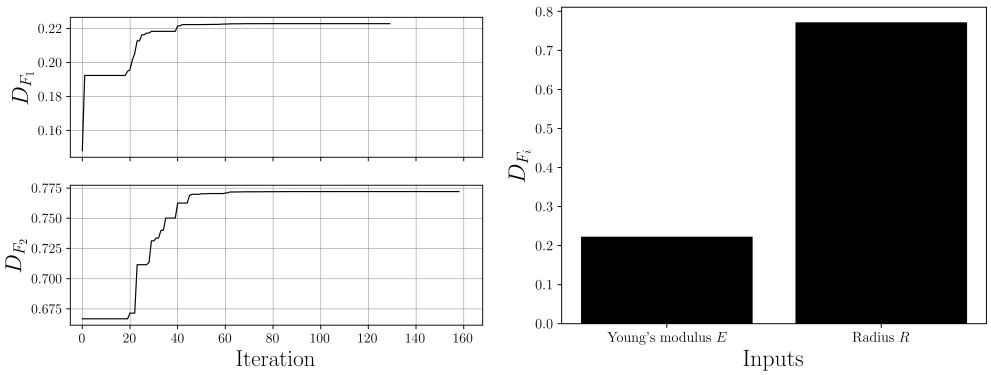
However, it may happen that some characteristics of the beam have not been identified as variable, or random, or that the boundary conditions are not exactly the specified ones, or that the material is not isotropic elastic, *etc.* It may also happen that no analytical solution is available for the performance function F in more complex cases. That is why F and $\mu^\dagger$ may be only partially known because of these epistemic uncertainties. In the following, several scenarios for the available information on the random inputs $\mathbf{X}$ and the performance function F are thus considered to bound the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$.

As an initial guess (scenario 0), it is assumed that the random input variables are mutually independent and follow uniform laws $\mu^\dagger \equiv \mathcal{U}$ over their respective ranges, and that the performance function F is exactly known by (1.32). This information allows us to compute the expected value of the performance function $\mathbb{E}\{F(\mathbf{X})\} = \mathrm{U}$ and the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ using a Monte-Carlo (MC) simulation over 10^6 random samples. In addition, using Equation (1.32) and the DE algorithm of Price and Storn [140], the verification diameter $D_F = (D_{F_1}^2 + D_{F_2}^2)^{1/2}$

and the associated subdiameters D_{F_1} and D_{F_2} can be computed, where D_{F_1} is the subdiameter related to the Young's modulus E and D_{F_2} is the subdiameter related to the radius R of the beam. These diameters are shown on Figure 1.5. Therein, one can note that the variation of the radius R of the beam has much more influence on the performance function F than the variation of Young's modulus E. Indeed, we have $D_{F_2} \gg D_{F_1}$. Thus, the uncertainty on the maximum of deflection of the beam can be mostly attributed to the radius of the beam ($\approx 92\%$ of D_F^2) and not to Young's modulus ($\approx 8\%$ of D_F^2). This result could have been intuited because the maximum deflection of a clamped beam—the function F—as known by Equation (1.32), is proportional to $(\frac{1}{R})^4$ while it is only proportional to $\frac{1}{E}$.

Input	Range
Young's modulus E (in GPa)	[71.25, 78.75]
Radius R (in mm)	[11.875, 13.125]

Table 1.1: Range of the random input parameters $\mathbf{X}$.



(a) Evolution of the subdiameters as a function of the number of iterations of the DE algorithm.

(b) Values of the subdiameters.

Figure 1.5: Subdiameters $D_{F_i}, i = 1, 2$ *for case 1.* $D_{F_1} = 0.223$ *mm is the subdiameter related to the Young's modulus E of the beam.* $D_{F_2} = 0.772$ *mm is the subdiameter related to the radius R of the beam.*

1.3.2.2 Scenario 1: McDiarmid's assumptions

The expected value U of the performance function F and its subdiameters D_{F_1} and D_{F_2} computed in the previous section are used to construct an admissible set corresponding to McDiarmid's inequality (1.3), ignoring however the PDFs of the random inputs which allowed to compute U, and the exact expression (1.32) of F which allowed to compute D_{F_1} and D_{F_2}. This is the scenario 1. For this scenario,

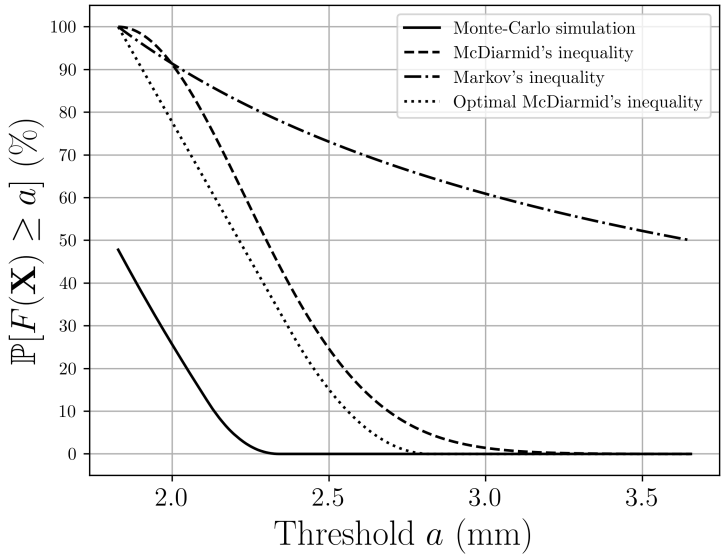


Figure 1.6: Probability of exceeding the threshold a for the deflection of a clamped beam.

the admissible set $\mathcal{A}_{\text{McD}}$ is defined by

$$\mathcal{A}_{\text{McD}} = \left\{ (f, \mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \\ \mu = \mu_1 \otimes \mu_2 \in \mathcal{M}_1(\mathcal{X}_1) \otimes \mathcal{M}_1(\mathcal{X}_2) \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{f(\mathbf{X})\} = \mathrm{U} \\ D_{f_j} \leq D_{F_j}\,,\ j = 1, 2 \end{array} \right\}. \tag{1.33}$$

Using the values of the subdiameters D_{F_1} and D_{F_2} computed in the previous section, the bounds on the probability of failure obtained by a non-optimal McDiarmid's inequality (1.3) and by the optimal McDiarmid's inequality (1.12) are shown on Figure 1.6. The probability of failure computed by a MC simulation with 10^6 samples corresponding to scenario 0 (known PDFs and performance function F) is also shown on Figure 1.6, together with the bound given by Markov's inequality for comparison purposes. Markov's inequality is the optimal upper bound of the probability of failure if only the mean $\mathrm{U} \leq a$ is known [145, Example 14.18]. As a reminder, it states that

$$\mathbb{P}\left[F(\mathbf{X}) \geq a\right] \leq \frac{\mathrm{U}}{a}. \tag{1.34}$$

We can see that the bound given by McDiarmid's inequality (1.3) is conservative and indeed non-optimal—recall Equation (1.10). On the contrary, by using the reduction Theorem 1 and solving the associated optimization problem, the optimal upper bound of the probability of failure can be obtained. This upper bound is the sharpest one that can be achieved given the information in the admissible set $\mathcal{A}_{\text{McD}}$. If one wants to further reduce it, more information has to be given—see the following scenario.

1.3.2.3 Scenario 2: F and the expected value of F are known

If, in addition to its expected value U, the performance function F itself is exactly known, a new admissible set $\mathcal{A}_F$ can be defined as

$$\mathcal{A}_F = \left\{ (F,\mu) \;\middle|\; \begin{array}{c} F : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \text{ is known} \\ \mu = \mu_1 \otimes \mu_2 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{F(\mathbf{X})\} = \mathrm{U} \end{array} \right\}. \tag{1.35}$$

This is scenario 2, where we are seeking to determine the optimal upper bound on the probability of failure given the information contained in $\mathcal{A}_F$. In other words, we want to determine:

$$\mathcal{U}(\mathcal{A}_F) = \sup_{(F,\mu)\in\mathcal{A}_F} \mathbb{P}_{\mathbf{X}\sim\mu}[F(\mathbf{X}) \geq a]\,. \tag{1.36}$$

A priori, this problem is not computationally tractable because finding the optimum requires a search over the space of probability measures on the intervals $[71.25, 78.75]$ GPa and $[11.875, 13.125]$ mm. Nevertheless, by using Theorem 1, the search can be reduced to a one over probability measures that are products of finite convex combinations of Dirac masses where each marginal measure μ_1 and μ_2 is supported on at most two Dirac masses of the parameter ranges $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively—see Figure 1.7. That is,

$$\mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) \tag{1.37}$$

where $\mathcal{A}_\Delta$ is given by

$$\mathcal{A}_\Delta = \left\{ (F,\mu) \in \mathcal{A}_F \;\middle|\; \begin{array}{c} \mu_1 = \alpha_1\delta_{x_1} + (1-\alpha_1)\delta_{y_1} \\ \mu_2 = \alpha_2\delta_{x_2} + (1-\alpha_2)\delta_{y_2} \\ \alpha_1, \alpha_2 \geq 0 \\ x_1, y_1 \in \mathcal{X}_1,\ x_2, y_2 \in \mathcal{X}_2 \end{array} \right\}. \tag{1.38}$$

Although no analytical formula exists for $\mathcal{U}(\mathcal{A}_F)$, contrary to $\mathcal{U}(\mathcal{A}_{\mathrm{McD}})$, see (1.12), its calculation is made possible by using the equality (1.37) and the `mystic` framework [101, 102]. We obtain the following optimal upper bound of the probability of failure with for example a threshold $a = 2.2$ mm:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) = 48.7\%. \tag{1.39}$$

One must not forget that these results are obtained through an optimization process. Thus, they may depend on the efficiency of the optimization algorithm. The issue of guaranteeing that the latter yields a satisfactory estimate of the upper bound (1.39) is addressed in Chapter 2. We observe that the actual number of points forming the support of μ is lower than the one expected at first from Theorem 1, which was four points. Figure 1.8 and Figure 1.9 show this numerical collapse. Young's modulus marginal collapses to a single Dirac mass at the middle of its range while the radius marginal is still having support on two points on the two extreme values of its range. Unfortunately, there is no theoretical result to highlight this hidden

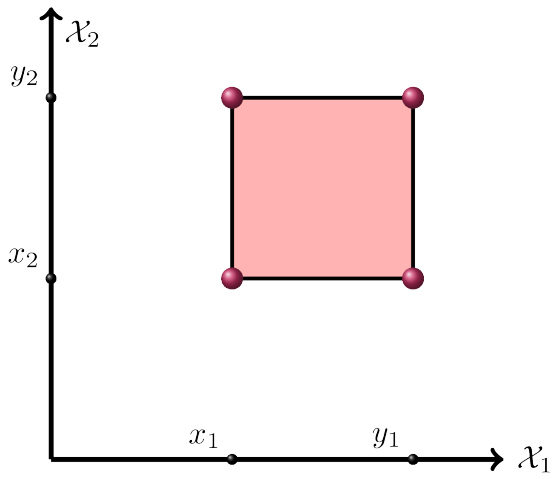
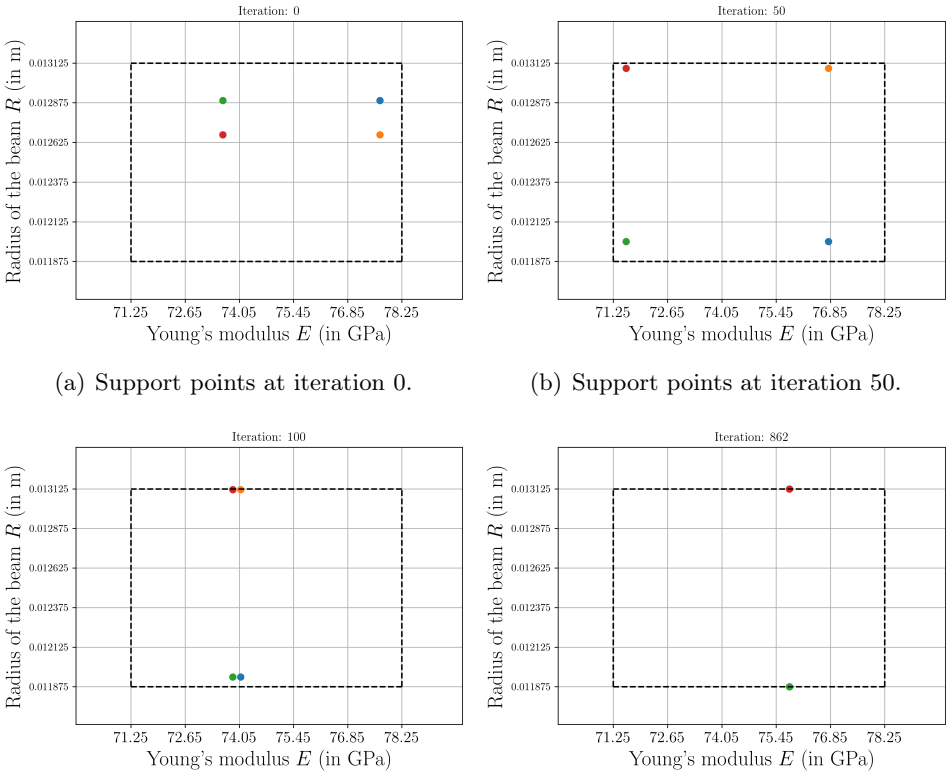


Figure 1.7: Support of μ for scenario 2 of case 1. The marginal measures μ_1 and μ_2 are supported on at most two Dirac points. The support of μ consists in at most four Dirac support points.

reduction property but one can develop an adequate numerical implementation of OUQ to detect it. A sketch of this numerical implementation can be found in [112, Section 6]. Moreover, the support points can help us to identify weak points of the system for the given threshold a. For instance, the location of the support points of the radius shows that reducing the range of the latter will decrease the optimal bound on the probability of failure. On the contrary, the position of the single Young's modulus support point shows that reducing its range will not decrease the optimal bound on the probability of failure. This is a non-binding information to the optimization problem $\mathcal{U}(\mathcal{A}_F)$.

1.3.2.4 Summary of the results

The results obtained for scenario 0 (full knowledge of the PDFs of the input parameters and performance function), scenario 1, and scenario 2 (partial knowledge) are gathered in Table 1.2. One can see that including more information about the system in the admissible set lowers the upper bound $\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$ of the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$. First, we can see the difference between the upper bound given by McDiarmid's inequality, being 65.1%, which is non-optimal, and the least upper bound $\mathcal{U}(\mathcal{A}_{\text{McD}})$ given by the optimal McDiarmid's inequality, being 51.7%. This latter percentage directly stems from the information included in $\mathcal{A}_{\text{McD}}$. We note in passing that for small values of the threshold a, McDiarmid's inequality is even more conservative than Markov's inequality (1.34). When one knows not only the subdiameters D_{F_1} and D_{F_2} but the whole function F, an even sharper bound on the probability of failure can be achieved. This upper bound drops to 48.7%. This shows that including more information about the problem in the



(a) Support points at iteration 0.

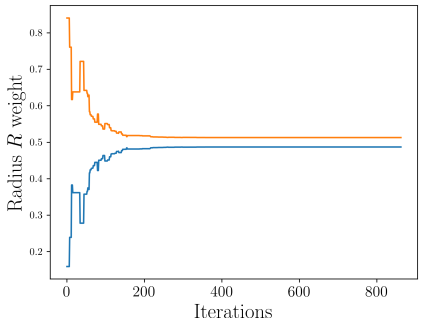
(b) Support points at iteration 50.

(c) Support points at iteration 100.

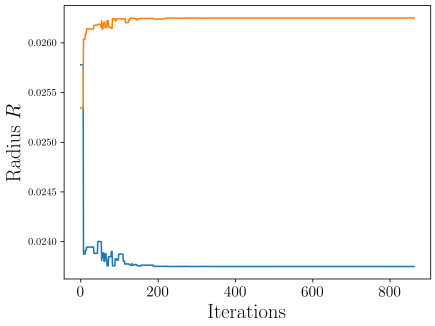
(d) Support points at iteration 862.

Figure 1.8: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.38) with $a = 2.2$ mm for scenario 2 of case 1. The support maximizers of the reduced OUQ problem for the admissible set (1.38) collapses to a two-point support. The support of Young's modulus E marginal collapses to a single Dirac mass while the support of the radius R marginal converges to the extremes of its range.

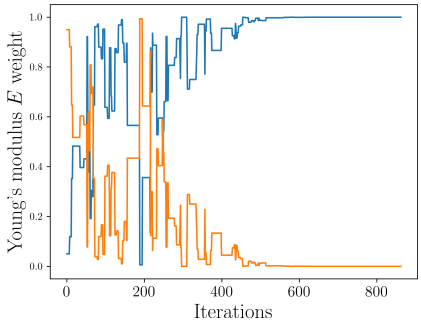
admissible set leads to a sharper least upper bound on the probability of failure. For comparison purpose, the probability of failure at $a = 2.2$ mm using a MC simulation with 10^6 samples is shown in the last row. The differences between the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ given by the MC simulation in scenario 0 and the bounds computed in scenario 1 and 2 are that, for the MC simulation, the performance function F and the probability measure $\mu^\dagger$ of the random input variables $\mathbf{X}$ are exactly known. This is hardly the case in practical applications. Returning to the initial problem, *i. e.* certifying that $\mathbb{P}[F(\mathbf{X}) \geq a] \leq \varepsilon$, the values of the upper bounds computed above can be used. Consider for instance $\varepsilon = 50\%$. That is, we want to certify that the probability of failure remains below 50%. From Table 1.2, we can assure that, if the assumptions of scenario 2 are known, then the probability of failure will stay below ε. The beam can be certified with respect to the considered performance function F. On the contrary, the assumptions of scenario 1 does not guarantee it.



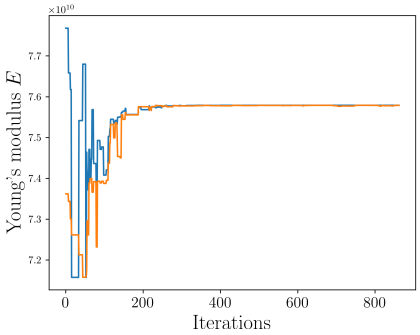
(a) Convergence for radius R weight.



(b) Convergence for radius R.



(c) Convergence for Young's modulus E weight.



(d) Convergence for Young's modulus E.

Figure 1.9: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.38) with $a = 2.2$ mm for scenario 2 of case 1. The support of the Young's modulus E converges to a single support point at 75.8 GPa. The support of the radius R converges to the extremes of its range, with a weight of 0.49 at 0.02375 m and a weight of 0.51 at 0.02625 m.

Threshold		$a = 2.2$ (mm)
Admissible scenario		$\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$
Scenario 1	McDiarmid's inequality	$\leq 65.1\%$
	$\mathcal{U}(\mathcal{A}_{\text{McD}})$	$= 51.7\%$
Scenario 2	$\mathcal{U}(\mathcal{A}_F)$	$= 48.7\%$
Scenario 0	$\mathbb{P}[F(\mathbf{X}) \geq a]$ (MC over 10^6 samples)	$= 4.4\%$

Table 1.2: Summary of the upper bounds of the probability of failure with $a = 2.2$ mm for several different scenarios for case 1.

1.3.3 Case 2: Lift coefficient for the RAE2822 wing profile

1.3.3.1 Setup

The following example is taken from the case studied by Lazareff in [83]. Essentially, a near-stall regime of the RAE2822 wing profile is considered. The near-stall regime of a wing profile is defined as the moment that the lift coefficient is maximal before it is reduced by increasing the angle of attack beyond its critical value. The RAE2822 wing profile is a supercritical airfoil which has become a standard test case for turbulence modeling validation [34]. The geometry of this airfoil can be seen on Figure 1.10. The flow is modeled by the Reynolds-Averaged Navier–Stokes (RANS) equations together with Wilcox k–ω turbulence model closure [160] and Menter Shear Stress Transport (SST) correction for farfield boundary conditions [103]. The nominal flow conditions are the ones described in [82, 83]. The nominal free-stream Mach number and angle of attack are $\underline{M} = 0.734$ and $\underline{\alpha} = 2.79°$, respectively, and the Reynolds number is $\mathrm{Re} = 6.5 \times 10^6$. Mach-induced stall arises from shock-boundary layer interaction slightly above the nominal value $\underline{M}$ (in [82] it is argued that without the SST correction "the stall tendency is widely under-estimated").

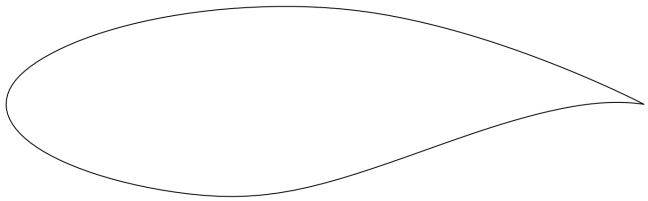


Figure 1.10: Geometry of the RAE2822 airfoil.

The objective of this section is to certify that the probability that the lift coefficient of the RAE2822 wing profile—this is the performance function F—exceeds a specified threshold a is below a given tolerance ε. We call this probability the "probability of failure" and aim at bounding it when some characteristics of the flow and/or the profile are variable (random inputs) and/or only partial information on F is available. We choose as the profile's random inputs the Mach number M and the angle of attack α with their respective range; see Table 1.3. Both define the input vector $\mathbf{X} = (M, \alpha)$. However, as in Section 1.3.2, it may happen that some characteristics of the flow and/or the profile have not been identified as variable, or random, or that the boundary conditions are not exactly the specified ones, *etc.* That is why F and $\mu^\dagger$ may be only partially known because of these epistemic uncertainties. For this case, as no analytical solution is available for the performance function F, a numerical solver has to be used. In that respect, the Computational Fluid Dynamics (CFD) solver *elsA* [20] is used to simulate two-dimensional transonic flows around that airfoil with the numerical parameters described in [82]. *elsA* is developed by ONERA and can deal with the low subsonic to the high supersonic flow regimes. More information can be found at http://elsa.onera.fr. The solver will be considered as a black-box. However, as one numerical evaluation of F through *elsA*

Input	Range
Mach number M	$[0.7216, 0.745]$
Angle of attack α (in °)	$[2.49, 2.99]$

Table 1.3: Range of each input parameter $\mathbf{X}$.

is costly, a surrogate model G has been built. This surrogate model G has been built by using 128 points distributed in the input space $\mathcal{X}$ computed through *elsA*—see the black dots on Figure 1.11—and the Thin Plate Spline (TPS) interpolation method [44, 45]. The surrogate model G will be assumed accurate enough to state that G is equivalent to F, namely $G \equiv F$. Consequently, we will consider that $D_{F-G} = 0$, see Equation (1.5). In the context of the NODESIM-CFD project,[2] it has been shown that this case involves shock-boundary layer interaction, characterizing flows with high Reynolds number. It leads to a canyon-like structure of the surrogate model—see Figure 1.11. In the following, several scenarios for the available information on the random inputs $\mathbf{X}$ and the performance function F are thus considered to bound the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$.

As an initial guess (scenario 0), it is assumed that the random input variables are mutually independent and follow uniform laws $\mu^{\dagger} \equiv \mathcal{U}$ over their respective ranges, and that the performance function F is exactly known by the TPS surrogate model. This information allows us to compute the expected value of the performance function $\mathbb{E}\{F(\mathbf{X})\} = \mathrm{L}$ and the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ using a Monte-Carlo (MC) simulation over 10^5 random samples. In addition, using the TPS surrogate model and the DE algorithm of Price and Storn [140], the verification diameter $D_F = (D_{F_1}^2 + D_{F_2}^2)^{1/2}$ and the associated subdiameters D_{F_1} and D_{F_2} can be computed, where D_{F_1} is the subdiameter related to the Mach number M and D_{F_2} is the subdiameter related to the angle of attack α. These diameters are shown on Figure 1.12. In that regard, one can note that the variation of the angle of attack α has much more influence on the performance function F than the variation of the Mach number M. Indeed, we have $D_{F_2} \gg D_{F_1}$. Thus, the uncertainty on the lift coefficient of the RAE2822 wing profile mainly stems from the variation of the angle of attack α ($\approx 70\%$ of D_F^2) rather than from the variation of the Mach number M ($\approx 30\%$ of D_F^2).

1.3.3.2 Scenario 1: McDiarmid's assumptions

The expected value L of the performance function F and its subdiameters D_{F_1} and D_{F_2} computed in the previous section are used to construct an admissible set corresponding to McDiarmid's inequality (1.5), ignoring however the PDFs of the random inputs which allowed to compute L, and the performance function F which allowed to compute D_{F_1} and D_{F_2}. This is the scenario 1. For this scenario, the

[2] Non-deterministic simulation for CFD-Based Design methodologies. https://cordis.europa.eu/result/rcn/47478_ en.html

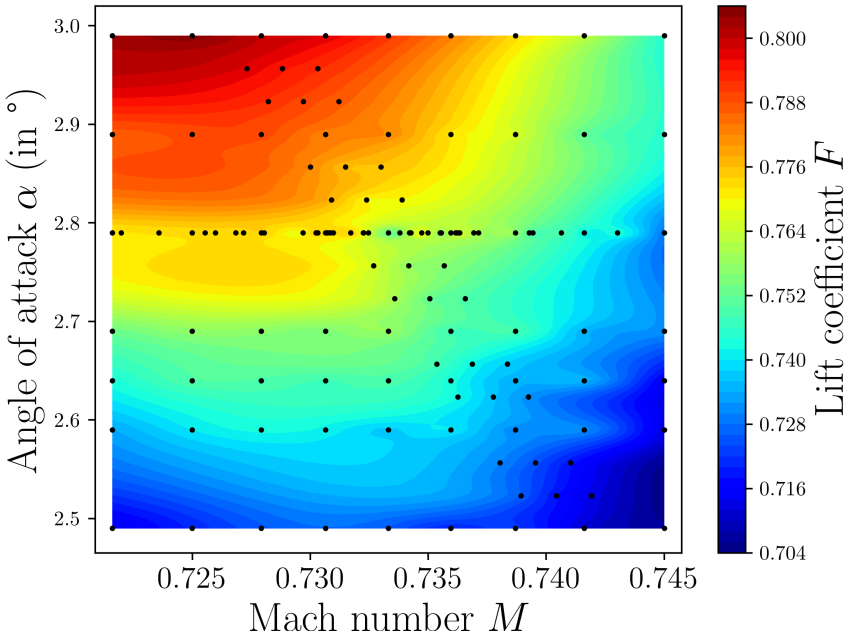
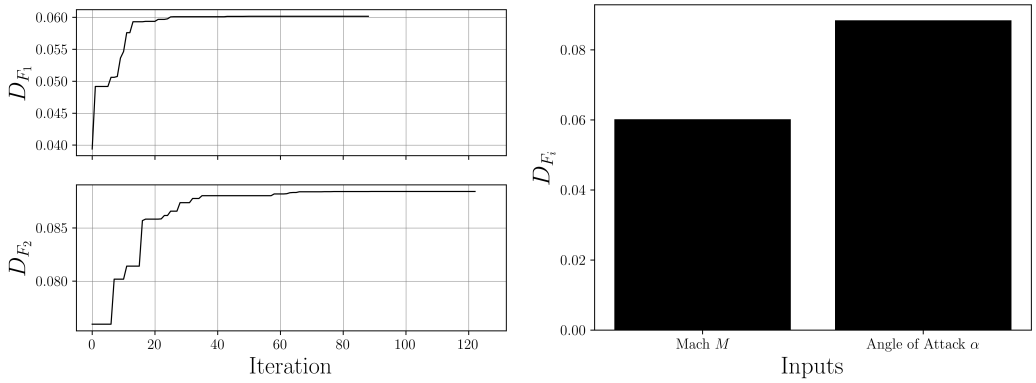


Figure 1.11: Surrogate model obtained by the TPS method in the (M, α) parameter space for the lift coefficient of the RAE2822 wing profile. The black circles are the 128 data points available.



(a) Evolution of the subdiameters as a function of to the number of iterations of the DE algorithm.

(b) Values of the subdiameters.

Figure 1.12: Subdiameters $D_{F_i}, i = 1, 2$ for case 2. $D_{F_1} = 0.06$ is the subdiameter related to the Mach number M. $D_{F_2} = 0.09$ is the subdiameter related to the angle of attack α.

admissible set $\mathcal{A}_{\text{McD}}$ is defined by

$$\mathcal{A}_{\text{McD}} = \left\{ (f, \mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \\ \mu = \mu_1 \otimes \mu_2 \\ \mathbb{E}_{\mathbf{X} \sim \mu}\{f(\mathbf{X})\} = \mathrm{L} \\ D_{f_j} \leq D_{F_j}\,,\ j = 1, 2 \end{array} \right\}. \tag{1.40}$$

Using the values of the subdiameters D_{F_1} and D_{F_2} computed in the previous section, the bounds on the probability of failure obtained by a non-optimal McDiarmid's inequality (1.5) and by the optimal McDiarmid's inequality (1.12) are shown on Figure 1.13. The probability of failure computed by a MC simulation with 10^5 samples corresponding to scenario 0 (known PDFs and performance function F) is also shown on Figure 1.13, together with the bound given by Markov's inequality for comparison purposes. Markov's inequality is the optimal upper bound of the probability of failure if only the mean $\mathrm{L} \leq a$ is known [145, Example 14.18]. As a reminder, it states that

$$\mathbb{P}\left[F(\mathbf{X}) \geq a\right] \leq \frac{\mathrm{L}}{a}. \tag{1.41}$$

We can see that the bound given by McDiarmid's inequality (1.5) is conservative and indeed non-optimal—recall Equation (1.10). On the contrary, by using the reduction Theorem 1 and solving the associated optimization problem, the optimal upper bound of the probability of failure can be obtained. This upper bound is the sharpest one that can be achieved given the information in the admissible set $\mathcal{A}_{\mathrm{McD}}$. If one wants to further reduce it, more information has to be given—see the following scenarios.

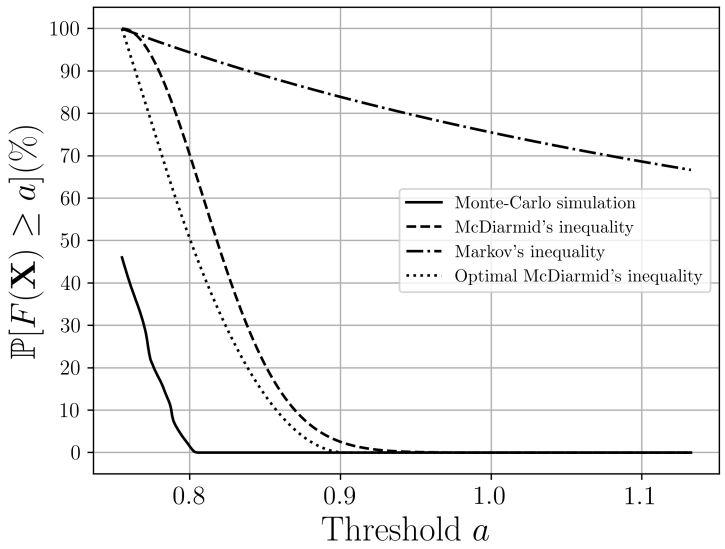


Figure 1.13: Probability of exceeding the threshold a for the lift coefficient of the RAE2822 wing profile.

1.3.3.3 Scenario 2: F and the expected value of F are known

If, in addition to its expected value L, the performance function F itself is exactly known, a new admissible set $\mathcal{A}_F$ can be defined as

$$\mathcal{A}_F = \left\{ (F,\mu) \;\middle|\; \begin{array}{c} F : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \text{ is known} \\ \mu = \mu_1 \otimes \mu_2 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{F(\mathbf{X})\} = \mathrm{L} \end{array} \right\} \tag{1.42}$$

This is scenario 2, where we are seeking to determine the optimal upper bound on the probability of failure given the information contained in $\mathcal{A}_F$. In other words, we want to determine:

$$\mathcal{U}(\mathcal{A}_F) = \sup_{(F,\mu)\in\mathcal{A}_F} \mathbb{P}_{\mathbf{X}\sim\mu}[F(\mathbf{X}) \geq a]\,. \tag{1.43}$$

A priori, this problem is not computationally tractable because finding the optimum requires a search over the space of probability measures on the intervals $[0.7216, 0.745]$ and $[2.49, 2.99]^\circ$. Nevertheless, by using Theorem 1, the search can be reduced to a one over probability measures that are products of finite convex combinations of Dirac masses where each marginal measure μ_1 and μ_2 is supported on at most two Dirac masses of the parameter ranges $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively—see Figure 1.14. In other words,

$$\mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) \tag{1.44}$$

where $\mathcal{A}_\Delta$ is given by

$$\mathcal{A}_\Delta = \left\{ (F,\mu) \in \mathcal{A}_F \;\middle|\; \begin{array}{c} \mu_1 = \alpha_1\delta_{x_1} + (1-\alpha_1)\delta_{y_1} \\ \mu_2 = \alpha_2\delta_{x_2} + (1-\alpha_2)\delta_{y_2} \\ \alpha_1, \alpha_2 \geq 0 \\ x_1, y_1 \in \mathcal{X}_1,\ x_2, y_2 \in \mathcal{X}_2 \end{array} \right\}. \tag{1.45}$$

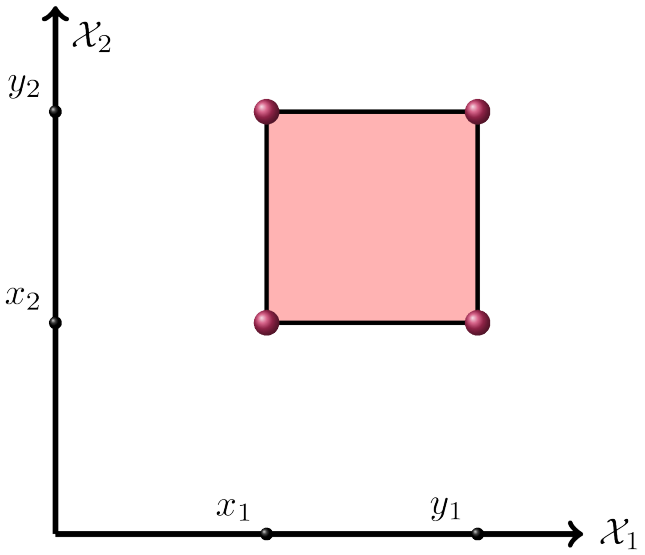


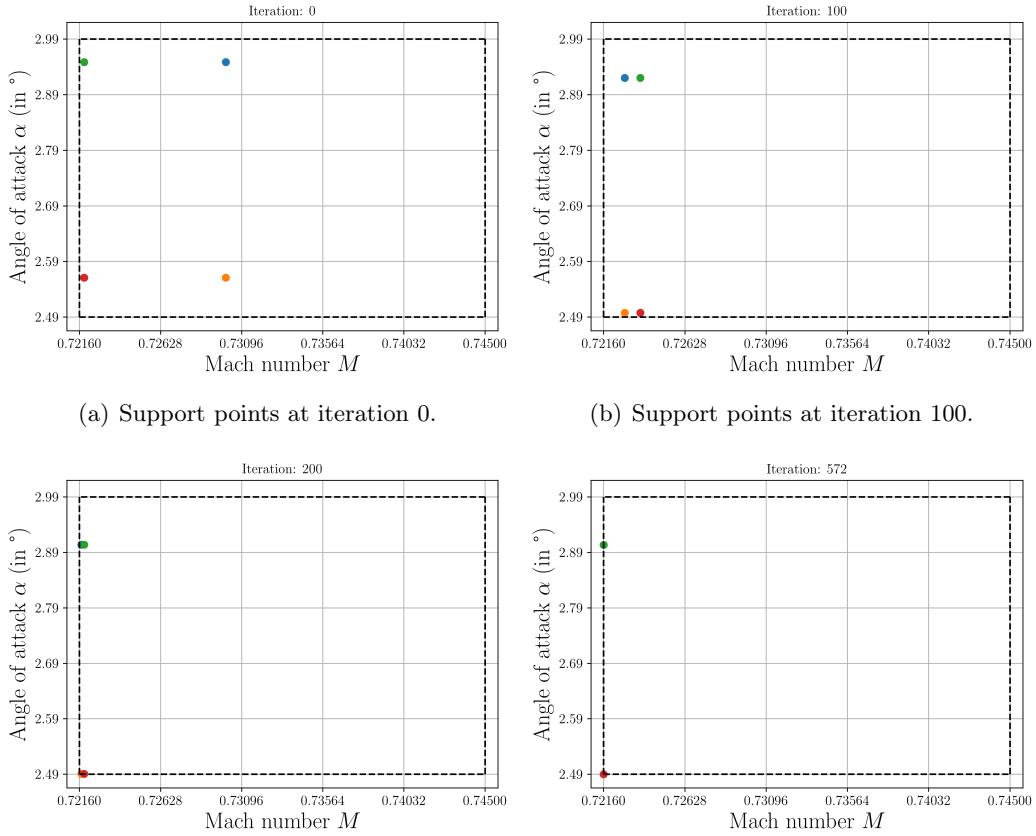
*Figure 1.14: Support of μ for scenario 2 of case 2. The marginal measures μ_1 and μ_2 are supported on at most two Dirac points. The support of μ consists in **at most** four Dirac support points.*

Even though no analytical formula exists for $\mathcal{U}(\mathcal{A}_F)$, contrary to $\mathcal{U}(\mathcal{A}_{\text{McD}})$, see (1.12), its calculation is made possible by using the equality (1.44) and the `mystic` framework [101, 102]. We obtain the following optimal upper bound of the probability of failure with for example a threshold $a = 0.79$:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) = 53.6\%. \tag{1.46}$$

One must not forget that these results are obtained through an optimization process. Thus, they may depend on the efficiency of the optimization algorithm. The issue of guaranteeing that the latter yields a satisfactory estimate of the upper bound (1.46) is addressed in Chapter 2. We observe that the actual number of points forming the support of μ is lower than the one expected at first from Theorem 1, which was four points. Figure 1.15 and Figure 1.16 show this numerical collapse. Mach number marginal collapses to a single Dirac mass at the lowest value of its range while the angle of attack marginal is still having support on two points. Unfortunately, there is no theoretical result to highlight this hidden reduction property but one can develop an adequate numerical implementation of OUQ to detect it. A sketch of this numerical implementation can be found in [112, Section 6]. In addition, the support points can help us to identify weak points of the system for the given threshold a. For instance, the location of the support points of the Mach number shows that increasing the lower value of the range of the latter will decrease the optimal bound on the probability of failure. On the contrary, the position of the angle of attack support points shows that reducing its range will not decrease the optimal bound

on the probability of failure. This is a non-binding information to the optimization problem $\mathcal{U}(\mathcal{A}_F)$.



(a) Support points at iteration 0.

(b) Support points at iteration 100.

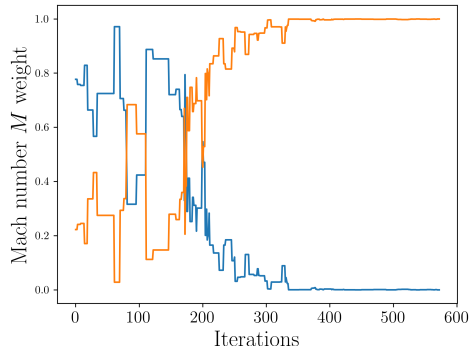
(c) Support points at iteration 200.

(d) Support points at iteration 572.

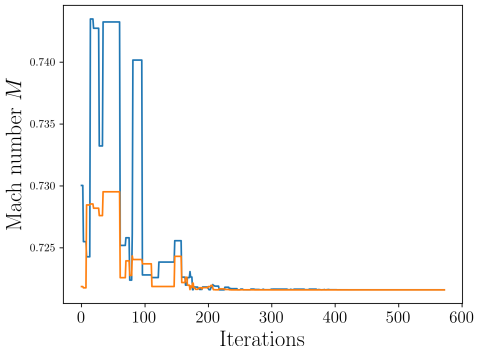
Figure 1.15: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.45) with $a = 0.79$ for scenario 2 of case 2. The support maximizers of the reduced OUQ problem for the admissible set (1.45) collapses to a two-point support. The support of the Mach number M marginal collapses to a single Dirac mass while the support of the angle of attack α marginal converges to two support points.

1.3.3.4 Scenario 3: F, the expected value of F, and the expected value of α are known

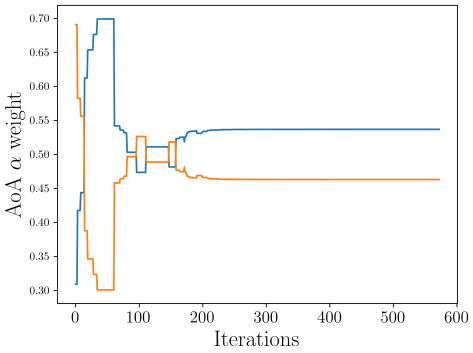
If, in addition to its expected value L and the performance function F itself is exactly known, the expected value of the angle of attack α is known, a new admissible



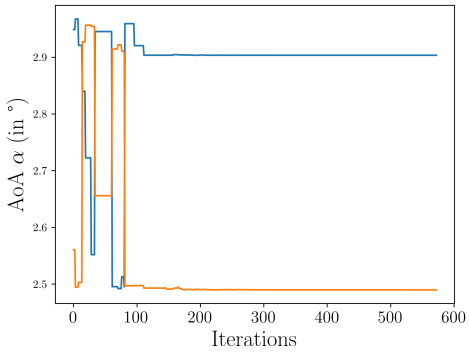
(a) Convergence for Mach number M weight.



(b) Convergence for Mach number M.



(c) Convergence for angle of attack α weight.



(d) Convergence for angle of attack α.

Figure 1.16: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.45) with $a = 0.79$ for scenario 2 of case 2. The support of the Mach number M converges to a single support point at 0.7216*. The support of the angle of attack α converges to two support points, one with a weight of* 0.54 *at* 2.9036° *and the other one with a weight of* 0.46 *at* 2.49°*.*

set $\mathcal{A}_{F_\alpha}$ can be defined as

$$\mathcal{A}_{F_\alpha} = \left\{ (F, \mu) \;\middle|\; \begin{array}{c} F : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \text{ is known} \\ \mu = \mu_1 \otimes \mu_2 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{F(\mathbf{X})\} = \mathrm{L} \\ \mathbb{E}_{X_2\sim\mu_2}\{X_2\} = \underline{\alpha} \end{array} \right\}. \tag{1.47}$$

This is scenario 3, where we are seeking to determine the optimal upper bound on the probability of failure given the information contained in $\mathcal{A}_{F_\alpha}$. In other words, we want to determine:

$$\mathcal{U}(\mathcal{A}_{F_\alpha}) = \sup_{(F,\mu)\in\mathcal{A}_{F_\alpha}} \mathbb{P}_{\mathbf{X}\sim\mu}[F(\mathbf{X}) \geq a]\,. \tag{1.48}$$

We add one moment constraint (the expected value) on the second input variable $X_2 = \alpha$. Therefore, the support of μ_2 is now made of at most three Dirac support

points and μ_1 is still made of at most two Dirac support points. We proceed as in scenario 2 to compute the upper bound of the probability of failure given the information of the admissible set $\mathcal{A}_{F_\alpha}$. That is, we have the following equality

$$\mathcal{U}(\mathcal{A}_{F_\alpha}) = \mathcal{U}(\mathcal{A}_\Delta) \tag{1.49}$$

where $\mathcal{A}_\Delta$ is given by

$$\mathcal{A}_\Delta = \left\{ (F,\mu) \in \mathcal{A}_{F_\alpha} \;\middle|\; \begin{array}{c} \mu_1 = \alpha_1\delta_{x_1} + (1-\alpha_1)\delta_{y_1} \\ \mu_2 = \alpha_2\delta_{x_2} + \alpha_3\delta_{y_2} + (1-\alpha_2-\alpha_3)\delta_{z_2} \\ \alpha_1,\alpha_2,\alpha_3 \geq 0 \\ x_1,y_1 \in \mathcal{X}_1,\ x_2,y_2,z_2 \in \mathcal{X}_2 \end{array} \right\}. \tag{1.50}$$

The calculation of the upper bound $\mathcal{U}(\mathcal{A}_{F_\alpha})$ is made possible by using the equality (1.49) and the `mystic` framework [101, 102]. We obtain the following optimal upper bound of the probability of failure with for example a threshold $a = 0.79$:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) = 45.9\%. \tag{1.51}$$

1.3.3.5 Scenario 4: F, the expected value of F, and the expected value of M are known

If, in addition to its expected value L and the performance function F itself, the expected value of the Mach number M is known, a new admissible set $\mathcal{A}_{F_M}$ can be defined as

$$\mathcal{A}_{F_M} = \left\{ (F,\mu) \;\middle|\; \begin{array}{c} F : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R} \text{ is known} \\ \mu = \mu_1 \otimes \mu_2 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{F(\mathbf{X})\} = \mathrm{L} \\ \mathbb{E}_{X_1\sim\mu_1}\{X_1\} = \underline{M} \end{array} \right\}. \tag{1.52}$$

This is scenario 4, where we are seeking to determine the optimal upper bound on the probability of failure given the information contained in $\mathcal{A}_{F_M}$. In other words, we want to determine:

$$\mathcal{U}(\mathcal{A}_{F_M}) = \sup_{(F,\mu)\in\mathcal{A}_{F_M}} \mathbb{P}_{\mathbf{X}\sim\mu}[F(\mathbf{X}) \geq a]\,. \tag{1.53}$$

We add one moment constraint (the expected value) on the first input variable $X_1 = M$. Therefore, the support of μ_1 is now made of at most three Dirac support points and μ_2 is still made of at most two Dirac support points. We proceed as in scenario 2 and scenario 3 to compute the upper bound of the probability of failure given the information of the admissible set $\mathcal{A}_{F_M}$. That is, we have the following equality

$$\mathcal{U}(\mathcal{A}_{F_M}) = \mathcal{U}(\mathcal{A}_\Delta) \tag{1.54}$$

where $\mathcal{A}_\Delta$ is given by

$$\mathcal{A}_\Delta = \left\{ (F,\mu) \in \mathcal{A}_{F_\alpha} \;\middle|\; \begin{array}{c} \mu_1 = \alpha_1\delta_{x_1} + \alpha_2\delta_{y_1} + (1-\alpha_1-\alpha_2)\delta_{z_1} \\ \mu_2 = \alpha_3\delta_{x_2} + (1-\alpha_3)\delta_{y_2} \\ \alpha_1,\alpha_2,\alpha_3 \geq 0 \\ x_1,y_1 \in \mathcal{X}_1,\ x_2,y_2,z_2 \in \mathcal{X}_2 \end{array} \right\}. \tag{1.55}$$

The calculation of the upper bound $\mathcal{U}(\mathcal{A}_{F_M})$ is made possible by using the equality (1.54) and the `mystic` framework [101, 102]. We obtain the following optimal upper bound of the probability of failure with for example a threshold $a = 0.79$:

$$\mathbb{P}[F(\mathbf{X}) \geq a] \leq \mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) = 46.0\%. \tag{1.56}$$

1.3.3.6 Summary of the results

The results obtained for scenario 0 (full knowledge of the PDFs of the input parameters and performance function), scenario 1, scenario 2, scenario 3, and scenario 4 (partial knowledge) are gathered in Table 1.4. One can see that including more information about the system in the admissible set lowers the upper bound $\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$ of the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$. First, we can see the difference between the upper bound given by McDiarmid's inequality, being 80.8%, which is non–optimal, and the least upper bound $\mathcal{U}(\mathcal{A}_{\mathrm{McD}})$ given by the optimal McDiarmid's inequality, being 60.7%. This latter percentage directly stems from the information included in $\mathcal{A}_{\mathrm{McD}}$. We note in passing that for small values of the threshold a, McDiarmid's inequality is even more conservative than Markov's inequality (1.41). When one knows not only the subdiameters D_{F_1} and D_{F_2} but the whole function F, a sharper bound on the probability of failure can be achieved. This upper bound drops to 53.6%. Finally, adding the knowledge of either the expected value of the angle of attack or the expected value of the Mach number, an even sharper bound on the probability of failure can be achieved, being 45.9% and 46.0% respectively. The information of knowing either the expected value of the angle of attack or the expected value of the Mach number is equivalent for the upper bound of the probability of failure—45.9% versus 46.0%. This shows that including more information about the problem in the admissible set leads to a sharper least upper bound on the probability of failure. For comparison purpose, the probability of failure at $a = 0.79$ using a MC simulation with 10^5 samples is shown in the last row. The differences between the probability of failure $\mathbb{P}[F(\mathbf{X}) \geq a]$ given by the MC simulation in scenario 0 and the bounds computed in scenario 1, 2, 3 and 4 are that, for the MC simulation, the performance function F and the probability measure $\mu^\dagger$ of the random input variables $\mathbf{X}$ are exactly known. This is hardly the case in practical applications. Returning to the initial problem, *i. e.* certifying that $\mathbb{P}[F(\mathbf{X}) \geq a] \leq \varepsilon$, the values of the upper bounds computed above can be used. Consider for instance $\varepsilon = 50\%$. That is, we want to certify that the probability of failure remains below 50%. From Table 1.4, we can assure that, if the assumptions of scenario 3 or 4 are known, then the probability of failure will stay below ε. The RAE2822 wing profile can be certified with respect to the considered performance function F. On the contrary, the assumptions of scenarios 1 and 2 do not guarantee it. To compute the subdiameters D_{F_1} and D_{F_2}, 6,360 function evaluations of F were needed. To compute the upper bounds $\mathcal{U}(\mathcal{A}_F)$, $\mathcal{U}(\mathcal{A}_{F_\alpha})$, and $\mathcal{U}(\mathcal{A}_{F_M})$, 35,580, 50,760, and 73,200 function evaluations of F were needed respectively.

Threshold		$a = 0.79$
Admissible scenario		$\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \geq a]$
Scenario 1	McDiarmid's inequality	$\leq 80.8\%$
	$\mathcal{U}(\mathcal{A}_{\text{McD}})$	$= 60.7\%$
Scenario 2	$\mathcal{U}(\mathcal{A}_F)$	$= 53.6\%$
Scenario 3	$\mathcal{U}(\mathcal{A}_{F_\alpha})$	$= 45.9\%$
Scenario 4	$\mathcal{U}(\mathcal{A}_{F_M})$	$= 46.0\%$
Scenario 0	$\mathbb{P}[F(\mathbf{X}) \geq a]$ (MC over 10^5 samples)	$= 6.8\%$

Table 1.4: Summary of the upper bounds of the probability of failure with $a = 0.79$ for several different scenarios for case 2.

1.3.4 Case 3: Lift-to-drag ratio for the RAE2822 wing profile

1.3.4.1 Setup

We consider a case close to the one of Section 1.3.3. The RAE2822 wing profile is again studied, where its geometry can be seen on Figure 1.10. The objective of this section is to certify that the probability that the lift-to-drag ratio of the profile—this is the performance function F—is below a specified threshold a is below a given tolerance ε. We call this probability the "probability of failure" and aim at bounding it when some characteristics of the flow and/or the profile are variable (random inputs) and/or only partial information on F is available. One notices that contrary to Section 1.3.2 and Section 1.3.3, the probability of failure is now $\mathbb{P}[F(\mathbf{X}) \leq a]$. We choose as the profile's random inputs the Mach number M, the angle of attack α, and the geometrical imperfection t with their respective ranges; see Table 1.5. These three parameters define the input vector $\mathbf{X} = (M, \alpha, t)$. The geometrical imperfection t is a parameter which takes into account the uncertainty on the exact dimensions of the RAE2822 wing profile. It is considered by modifying the thickness of the profile and its influence can be seen on Figure 1.17. $t = 0$ corresponds to the nominal thickness.

Along the same lines as in Section 1.3.2 and Section 1.3.3, it may happen that some characteristics of the flow and/or the profile have not been identified as variable, or random, or that the boundary conditions are not exactly the specified ones, *etc.* That is why F and $\mu^\dagger$ may be only partially known because of these epistemic uncertainties. For this case, as in Section 1.3.3, as no analytical solution is available for the performance function F, a numerical solver has to be used. In that respect, ISES [40, 42] is used to simulate two-dimensional transonic flows around that airfoil. ISES is a 2D aerodynamic code developed by Mark Drela and Michael B. Giles at MIT. In more details, the inviscid flow is represented by the steady Euler equations in integral form. As for the boundary layers and the wakes, a compressible lag-dissipation integral method is used. Instead of iterating between the viscous and inviscid solvers via an approximate interaction law, a global Newton-Raphson method is used to solve simultaneously the coupled viscous-inviscid equations. Nevertheless,

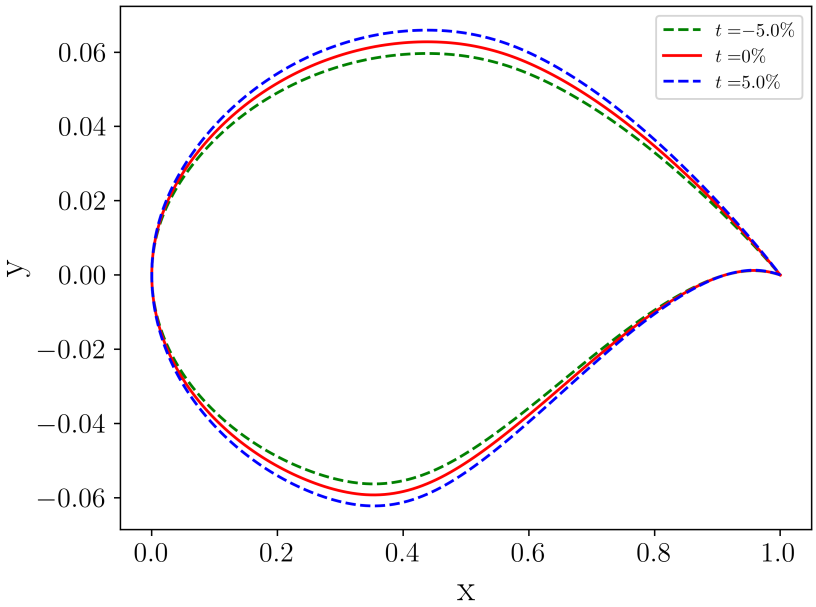


Figure 1.17: Dimensions of the RAE2822 profile for different geometrical imperfection values t.

as one numerical evaluation of F by using ISES can be costly, a surrogate model G of F has been built. The generalized Polynomial Chaos (gPC) expansion method is chosen to build the surrogate model G [49, 56, 87, 135, 162]. A quick reminder of this approach is given here.

Consider an orthogonal polynomial basis $\mathcal{B}$ of $L^2(\mathcal{X}, \mu)$, the Hilbert space of square integrable functions on $\mathcal{X}$ with respect to the probability measure μ. Assuming that the performance function $\mathbf{x} \mapsto G(\mathbf{x})$ is in $L^2(\mathcal{X}, \mu)$, that is

$$\int_{\mathcal{X}} |G(\mathbf{x})|^2 \, \mu(d\mathbf{x}) = \mathbb{E}_{\mathbf{X}\sim\mu}\left\{|G(\mathbf{X})|^2\right\} < +\infty, \tag{1.57}$$

then it can be expanded on the basis $\mathcal{B} = \{\phi_j\}_{j\in\mathbb{N}}$. Thus, one can build a polynomial surrogate model G of F by an L^2 projection on a finite dimensional subspace of $L^2(\mathcal{X}, \mu)$ spanned by a truncated family of d-variate orthonormal polynomials denoted by $\mathcal{B}^p = \{\phi_j\}_{j=0}^P$. The orthonormalization of this basis is done through

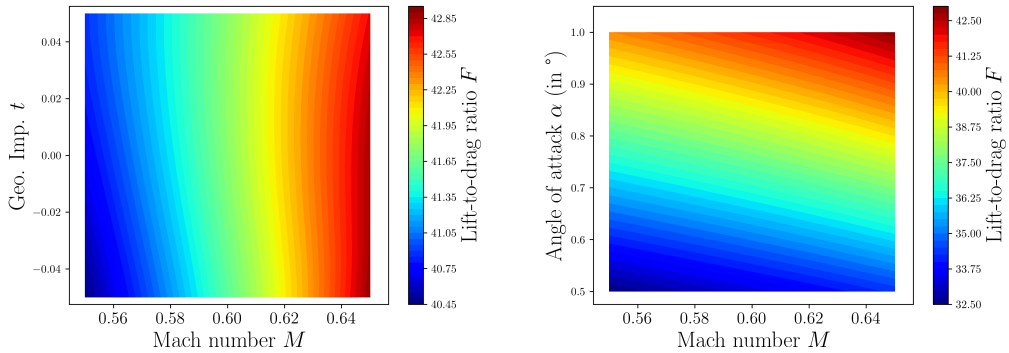
$$\langle \phi_i, \phi_j \rangle_{L^2} = \int_{\mathcal{X}} \phi_i(\mathbf{x})\phi_j(\mathbf{x})\mu(d\mathbf{x}) = \delta_{ij}, \ 0 \le i, j \le P, \tag{1.58}$$

where δ_{ij} is the Kronecker symbol such that $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. Then, the polynomial surrogate model G of F using the truncated basis $\mathcal{B}^p$ is

$$G(\mathbf{x}) = \sum_{i=0}^{P} F_i\phi_i(\mathbf{x}) = \sum_{i=0}^{P} \langle F, \phi_i \rangle_{L^2}\phi_i(\mathbf{x}). \tag{1.59}$$

Here the expansion coefficients $\langle F, \phi_i \rangle_{L^2}$ are computed by numerical quadratures. More details on this technique are given in Section 3.5. The surrogate model G will be assumed accurate enough to state that G is equivalent to F, namely $G \equiv F$. Consequently, we have that $D_{F-G} = 0$—see Equation (1.6). In the following, several scenarios for the available information on the random inputs $\mathbf{X}$ and the performance function F are considered to bound the probability of failure $\mathbb{P}[F(\mathbf{X}) \leq a]$.

As an initial guess (scenario 0), it is assumed that the random input variables are mutually independent and follow uniform laws $\mu^\dagger \equiv \mathcal{U}$ over their respective ranges, and that the performance function F is exactly known by the polynomial surrogate model. First, this information allows us to choose the Legendre polynomials as the basis $\mathcal{B}^p$ in order to compute the expansion coefficients $\{F_j\}_{j=0}^P$. The values of these coefficients can be found on Figure 1.19. An example of the polynomial surrogate model G can be seen on Figure 1.18. Secondly, this information allows us to compute the expected value of the performance function $\mathbb{E}\{F(\mathbf{X})\} = \mathrm{L/D}$ and the probability of failure $\mathbb{P}[F(\mathbf{X}) \leq a]$ using a Monte-Carlo (MC) simulation over 10^5 random samples. We note that $\mathbb{E}\{F(\mathbf{X})\} = \langle F, \phi_0 \rangle_{L^2}$ using the gPC expansion. Finally, using the polynomial surrogate model and the DE algorithm of Price and Storn [140], the verification diameter $D_F = (D_{F_1}^2 + D_{F_2}^2 + D_{F_3}^2)^{1/2}$ and the associated subdiameters D_{F_1}, D_{F_2}, and D_{F_3} can be computed, where D_{F_1} is the subdiameter related to Mach number M, D_{F_2} is the subdiameter related to the angle of attack α, and D_{F_3} is the subdiameter related to the geometrical imperfection t. These diameters are shown on Figure 1.12. One can see that the uncertainty in our model mainly stems from the variation of the angle of attack α ($\approx 92\%$ of D_F^2) rather than from the variation of the Mach number M ($\approx 7.4\%$ of D_F^2) or from the variation of the geometrical imperfection t ($\approx 0.6\%$ of D_F^2). Moreover, the subdiameter D_{F_3} is way below the other ones that is, $D_{F_3} \ll D_{F_1}$ and $D_{F_3} \ll D_{F_2}$.



(a) Slice at $\alpha = 1°$.

(b) Slice at $t = -5\%$.

Figure 1.18: Two slices of the polynomial surrogate model obtained by gPC for the lift-to-drag ratio of the RAE2822 wing profile.

Input	Range
Mach number M	$[0.55, 0.65]$
Angle of Attack α (in °)	$[0.5, 1]$
Geometrical Imp. t (in %)	$[-5, 5]$

Table 1.5: Range of each input parameter $\mathbf{X}$.

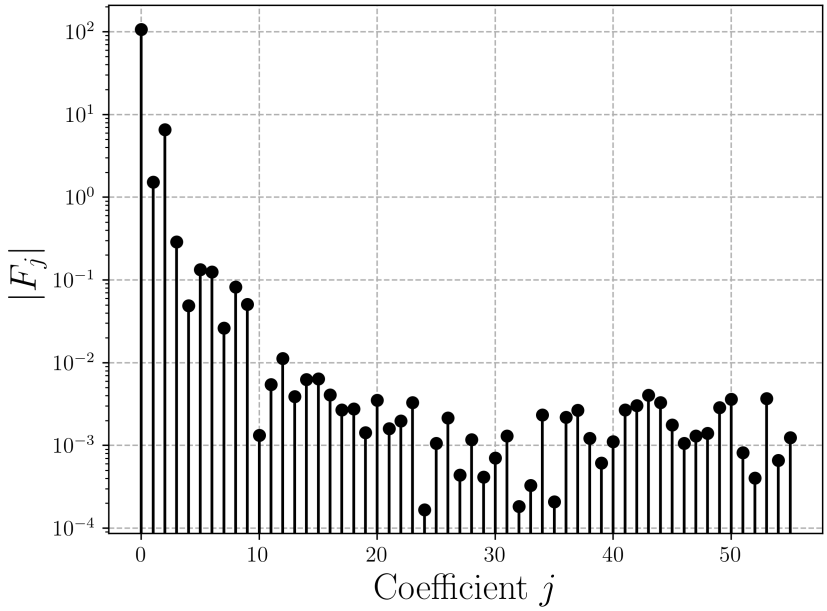
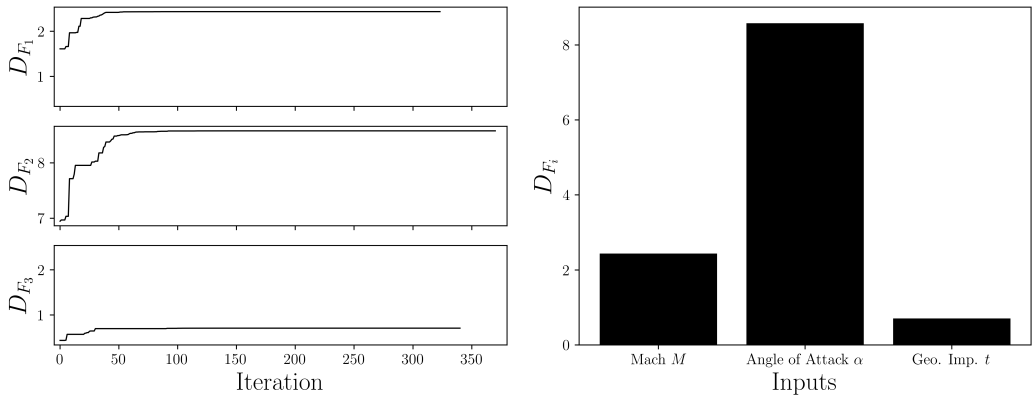


Figure 1.19: Coefficients $\{F_j\}_{j=0}^{P}$ *associated to the Legendre polynomials.*



(a) Evolution of the subdiameters with respect to the number of iterations of the DE algorithm.

(b) Value of the subdiameters.

Figure 1.20: Subdiameters $D_{F_i}, i = 1, 2, 3$ *for case* 3. $D_{F_1} = 2.435$ *is the subdiameter related to the Mach number* M. $D_{F_2} = 8.579$ *is the subdiameter related to the angle of attack* α. $D_{F_3} = 0.708$ *is the subdiameter related to the geometrical imperfection* t.

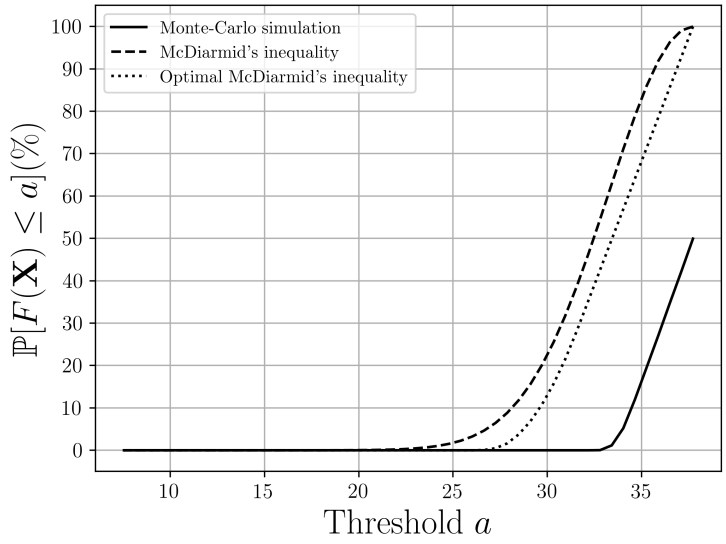


Figure 1.21: Probability of being below the threshold a for the lift-to-drag ratio of the RAE2822 wing profile.

1.3.4.2 Scenario 1: McDiarmid's assumptions

The expected value L/D of the performance function F and its subdiameters D_{F_1}, D_{F_2}, and D_{F_3} computed in the previous section are used to construct an admissible set corresponding to McDiarmid's inequality (1.6), ignoring however the PDFs of the random inputs which allowed to compute L/D, and the performance function F which allowed to compute D_{F_1}, D_{F_2}, and D_{F_3}. This is the scenario 1. For this scenario, the admissible set $\mathcal{A}_{\text{McD}}$ is defined by

$$\mathcal{A}_{\text{McD}} = \left\{ (f, \mu) \;\middle|\; \begin{array}{c} f : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to \mathbb{R} \\ \mu = \mu_1 \otimes \mu_2 \otimes \mu_3 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{f(\mathbf{X})\} = \text{L/D} \\ D_{f_j} \leq D_{F_j}\,,\ j = 1, 2, 3 \end{array} \right\}. \tag{1.60}$$

Using the values of the subdiameters D_{F_1}, D_{F_2}, and D_{F_3} computed in the previous section, the bounds on the probability of failure obtained by a non-optimal McDiarmid's inequality (1.6) and by the optimal McDiarmid's inequality (1.12) are shown on Figure 1.21. The probability of failure computed by a MC simulation with 10^5 samples corresponding to scenario 0 (known PDFs and performance function F) is also shown on Figure 1.21. We can see that the bound given by McDiarmid's inequality (1.6) is conservative and indeed non-optimal—recall Equation (1.10). On the contrary, by using the reduction Theorem 1 and solving the associated optimization problem, the optimal upper bound of the probability of failure can be obtained. This upper bound is the sharpest one that can be achieved given the information in the admissible set $\mathcal{A}_{\text{McD}}$. If one wants to further reduce it, more information has to be given; see the following scenario.

1.3.4.3 Scenario 2: F and the expected value of F are known

If, in addition to its expected value L/D, the performance function F itself is exactly known, a new admissible set $\mathcal{A}_F$ can be defined as

$$\mathcal{A}_F = \left\{ (F,\mu) \;\middle|\; \begin{array}{c} F : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to \mathbb{R} \text{ is known} \\ \mu = \mu_1 \otimes \mu_2 \otimes \mu_3 \\ \mathbb{E}_{\mathbf{X}\sim\mu}\{F(\mathbf{X})\} = \text{L/D} \end{array} \right\}. \tag{1.61}$$

This is scenario 2, where we are seeking to determine the optimal upper bound on the probability of failure given the information contained in $\mathcal{A}_F$. In other words, we want to determine:

$$\mathcal{U}(\mathcal{A}_F) = \sup_{(F,\mu)\in\mathcal{A}_F} \mathbb{P}_{\mathbf{X}\sim\mu}[F(\mathbf{X}) \leq a]\,. \tag{1.62}$$

A priori, this problem is not computationally tractable because finding the optimum requires a search over the space of probability measures on the intervals $[0.55, 0.65]$, $[0.5, 1]^\circ$, and $[-5, 5]\%$. Nevertheless, by using Theorem 1 and Remark 6, the search can be reduced to a one over probability measures that are products of finite convex combinations of Dirac masses where each marginal measure μ_1, μ_2, and μ_3 is supported on at most two Dirac masses of the parameter ranges $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ respectively; see Figure 1.22. In other words,

$$\mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) \tag{1.63}$$

where $\mathcal{A}_\Delta$ is given by

$$\mathcal{A}_\Delta = \left\{ (F,\mu) \in \mathcal{A}_F \;\middle|\; \begin{array}{c} \mu_1 = \alpha_1\delta_{x_1} + (1-\alpha_1)\delta_{y_1} \\ \mu_2 = \alpha_2\delta_{x_2} + (1-\alpha_2)\delta_{y_2} \\ \mu_3 = \alpha_3\delta_{x_3} + (1-\alpha_3)\delta_{y_3} \\ \alpha_1, \alpha_2, \alpha_3 \geq 0 \\ x_1, y_1 \in \mathcal{X}_1,\ x_2, y_2 \in \mathcal{X}_2,\ x_3, y_3 \in \mathcal{X}_3 \end{array} \right\}. \tag{1.64}$$

Even though no analytical formula exists for $\mathcal{U}(\mathcal{A}_F)$, contrary to $\mathcal{U}(\mathcal{A}_{\text{McD}})$, see (1.12), its calculation is made possible by using the equality (1.63) and the `mystic` framework [101, 102]. We obtain the following optimal upper bound of the probability of failure with for example a threshold $a = 35$:

$$\mathbb{P}[F(\mathbf{X}) \leq a] \leq \mathcal{U}(\mathcal{A}_F) = \mathcal{U}(\mathcal{A}_\Delta) = 65.5\%. \tag{1.65}$$

One must not forget that these results are obtained through an optimization process. Thus, they may depend on the efficiency of the optimization algorithm. The issue of guaranteeing that the latter yields a satisfactory estimate of the upper bound (1.65) is addressed in Chapter 2. We observe that the actual number of points forming the support of μ is lower than the one expected at first from Theorem 1, which was eight points. Figure 1.24 and Figure 1.23 show this numerical collapse. Two marginals, Mach number and geometrical imperfection, collapse to a single Dirac mass while the angle of attack marginal is still having support on two points. Unfortunately,

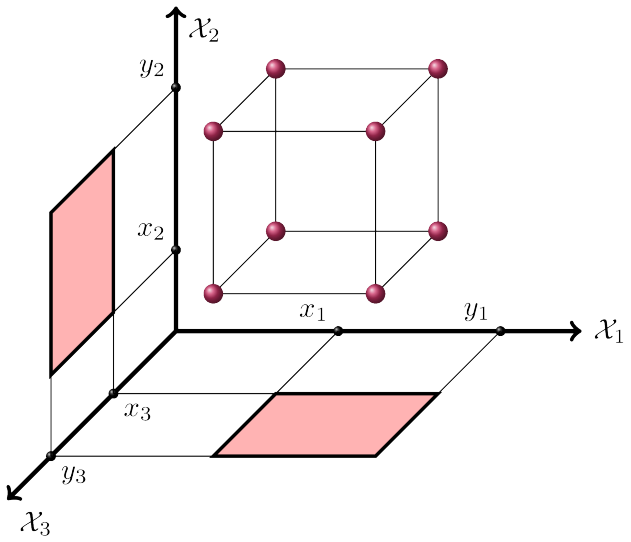


Figure 1.22: Support of μ for scenario 2 of case 3. The marginal measures μ_1, μ_2 and μ_3 are supported on at most two Dirac points. The support of μ consists in at most eight Dirac support points.

there is no theoretical result to highlight this hidden reduction property but one can develop an adequate numerical implementation of OUQ to detect it. A sketch of this numerical implementation can be found in [112, Section 6]. In addition, the support points can help us to identify weak points of the system for the given threshold a. Their locations show that rising the minimum angle of attack value, reducing the maximum geometrical imperfection and rising the range of Mach number will not decrease the optimal bound on the probability. These are non-binding information to the optimization problem $\mathcal{U}(\mathcal{A}_F)$. That is, instead of considering $2 \times 2 \times 2$ support points of μ, we performed the algorithm with $1 \times 2 \times 1$ support points.

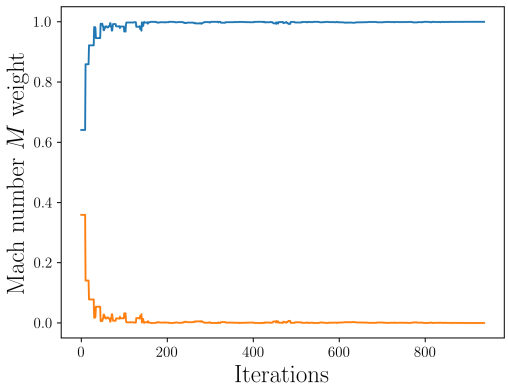
1.3.4.4 Summary of the results

The results obtained for scenario 0 (full knowledge of the PDFs of the input parameters and performance function), scenario 1, and scenario 2 (partial knowledge) are gathered in Table 1.6. One can see that including more information about the system in the admissible set lowers the upper bound $\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq a]$ of the probability of failure $\mathbb{P}[F(\mathbf{X}) \leq a]$. First, we can see the difference between the upper bound given by McDiarmid's inequality, being 83.0%, which is non-optimal, and the least upper bound $\mathcal{U}(\mathcal{A}_{\text{McD}})$ given by the optimal McDiarmid's inequality, being 68.2%. This latter percentage directly stems from the information included in $\mathcal{A}_{\text{McD}}$. When one knows not only the subdiameters D_{F_1}, D_{F_2}, and D_{F_3} but the whole function F, a sharper bound on the probability of failure can be achieved. This upper bound drops to 65.5%. Unlike previous cases, the upper bound of the probability of failure obtained in scenario 2 is very close to the one obtained in scenario 1. It may indicate that in this case, knowning the entire function instead

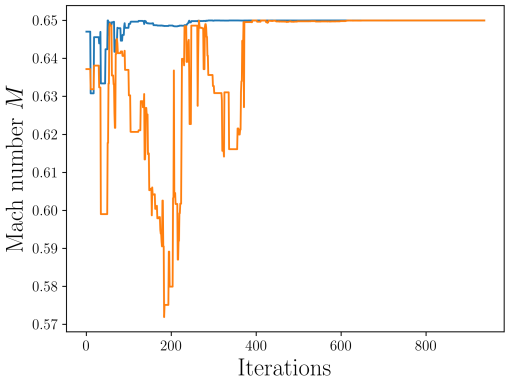
of the subdiameters does not affect much the upper bound of the probability of failure. For comparison purpose, the probability of failure at $a = 35$ using a MC simulation with 10^5 samples is shown in the last row. The differences between the probability of failure $\mathbb{P}[F(\mathbf{X}) \leq a]$ given by the MC simulation in scenario 0 and the bounds computed in scenario 1 and scenario 2 are that for the MC simulation, the performance function F and the probability measure $\mu^\dagger$ of the random input variables $\mathbf{X}$ are exactly known. This is hardly the case in practical applications. Returning to the initial problem, *i. e.* certifying that $\mathbb{P}[F(\mathbf{X}) \leq a] \leq \varepsilon$, the values of the upper bounds computed above can be used. Consider for instance $\varepsilon = 70\%$. That is, we want to certify that the probability of failure remains below 70%. From Table 1.6, we can assure that if the assumptions of scenario 1, using the Optimal McDiarmid's inequality, or scenario 2 are known, then the RAE2822 wing profile can be certified with respect to the considered performance function F. To compute the subdiameters D_{F_1}, D_{F_2}, and D_{F_3}, $36,330$ function evaluations of F were needed. To compute the upper bound $\mathcal{U}(\mathcal{A}_F)$, $30,016$ function evaluations of F were needed.

Threshold		$a = 35$
Admissible scenario		$\sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq a]$
Scenario 1	McDiarmid's inequality	$\leq 83.0\%$
	$\mathcal{U}(\mathcal{A}_{\mathrm{McD}})$	$= 68.2\%$
Scenario 2	$\mathcal{U}(\mathcal{A}_F)$	$= 65.5\%$
Scenario 0	$\mathbb{P}[F(\mathbf{X}) \leq a]$ (MC over 10^5 samples)	$= 16.3\%$

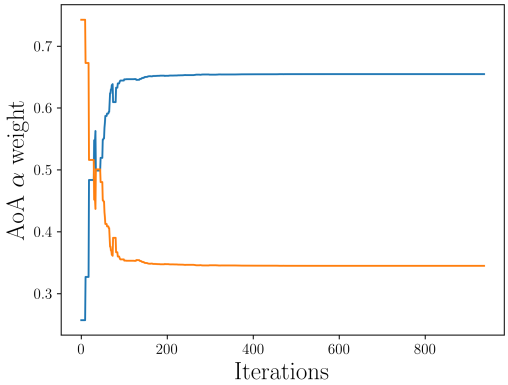
Table 1.6: Summary of the upper bounds of the probability of failure with $a = 0.35$ for several different scenarios for case 3.



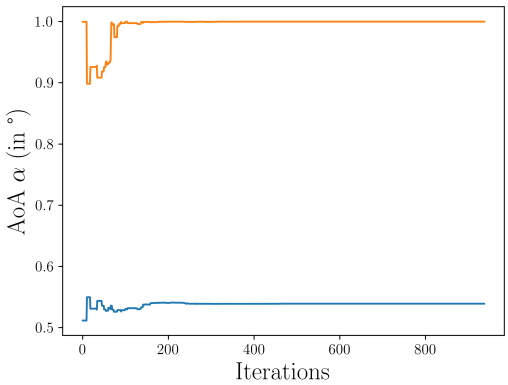
(a) Convergence for Mach number M weight.



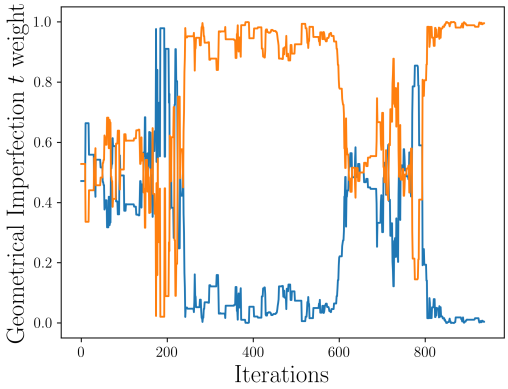
(b) Convergence for Mach number M.



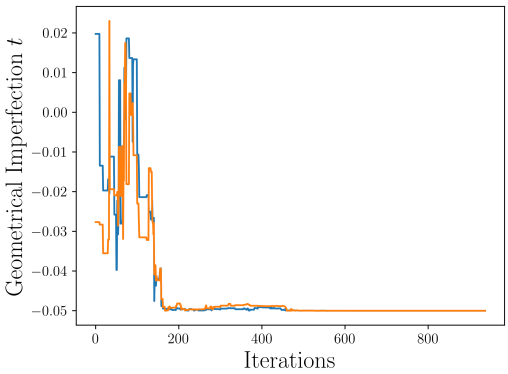
(c) Convergence for angle of attack α weight.



(d) Convergence for angle of attack α.

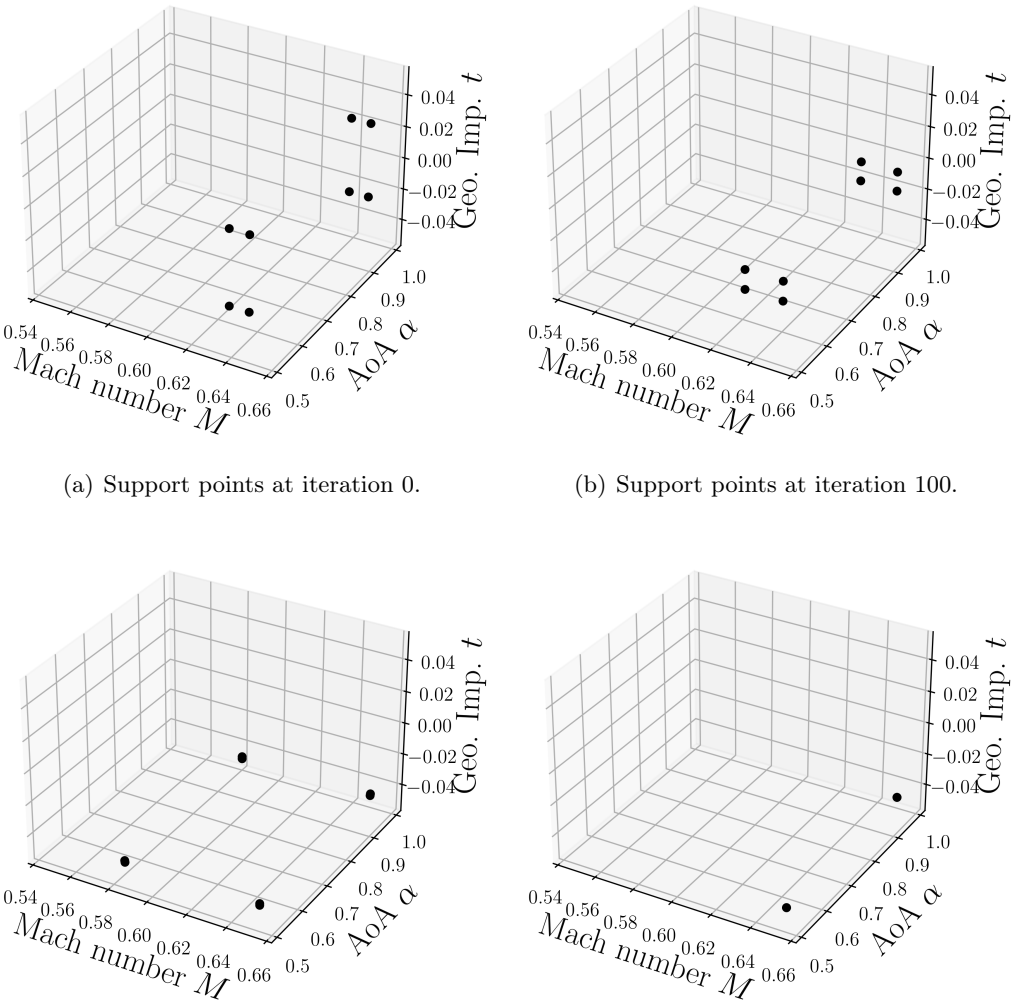


(e) Convergence for geometrical imperfection t weight.



(f) Convergence for geometrical imperfection t.

Figure 1.23: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.64) with $a = 0.35$ for scenario 2 of case 3. The support of the Mach number M converges to a single support point at 0.65. The support of the angle of attack α converges to two support points, one with a weight of 0.65 at $0.54°$ and another one with a weight of 0.35 at $1.0°$. The support of the geometrical imperfection t converges to a single support point at -5%.



(a) Support points at iteration 0.

(b) Support points at iteration 100.

(c) Support points at iteration 200.

(d) Support points at iteration 937.

Figure 1.24: Evolution of the optimal points as a function of number of iterations of the DE algorithm after using the reduction set $\mathcal{A}_\Delta$ (1.64) with $a = 0.35$ for scenario 2 of case 3. The support maximizers of the reduced OUQ problem for the admissible set (1.64) collapses to a two-point support for scenario 2 of case 3. The supports of the Mach number M and the geometrical imperfection t marginals collapse to a single Dirac mass while the support of the angle of attack α marginal converges to two support points.

1.4 Conclusions

In this chapter, we have seen that McDiarmid's inequality given by Equation (1.1) can be used in a certification context considering varying, possibly random inputs to the system to be certified. Here certification is defined as the process of guaranteeing that the probability of failing to satisfy a given mission, target, or scenario is below an acceptable small tolerance ε. Owhadi *et al.* have shown in [112] that McDiarmid's upper bound of the probability of failure is not optimal given the related assumptions, namely knowning the expected value of the performance function F and its associated subdiameters, and can be improved through an optimization problem [112, Prop. 5.7]. These optimal bounds are not exclusive to McDiarmid's inequality but apply to much more general classes of assumptions. That is, they can be extended to problems over admissible classes of distributions μ and functions f defined by constraints on their generalized moments, which defines an admissible set $\mathcal{A}$ containing all the information we have about the system. This is the Optimal Uncertainty Quantifiation (OUQ) framework proposed in [112]. At first, this optimization problem is infinite-dimensional but can be reduced to a finite-dimensional one thanks to the reduction Theorem 1. For that purpose, numerical optimizations have to be carried out. The `mystic` framework [101, 102] has been used in order to solve these optimization problems. Several examples have been studied to showcase the OUQ framework [112]: the bending of a clamped beam, the lift of a RAE2822 wing profile, and the lift-to-drag ratio of a RAE2822 wing profile. We have also seen that the OUQ framework allows us among others to determine weak points of the system. One of the main drawbacks of the OUQ framework and the reduction theorem is that the resulting optimization problem to solve can be non-linear, non-convex, non-continuous and heavily constrained. That is, classical optimization algorithms such as gradient descent are not possible and thus differential evolution algorithms have been implemented. However, such kind of algorithms does not guarantee that an optimal solution is ever found. Therefore, a procedure has to be developed in order to verify that the optimums picked by the differential evolution algorithms are relevant estimates. This is the topic of the next chapter.

Chapter 2

Reconstruction of monotone functions with application to uncertainty quantification

Contents

The work presented in this chapter is based on the article corresponding to [15], and was done in collaboration with Jean-Luc Akian (ONERA), Éric Savin (ONERA) and Tim J. Sullivan (University of Warwick/Zuse Institute Berlin).
This chapter considers the problem of adaptively reconstructing a monotonically increasing function $\mathcal{F}^\dagger$ from imperfect pointwise observations of this function. In the statistical literature, the problem of estimating a monotone function is commonly known as isotonic regression, and it is assumed that the observed data consist of noisy pointwise evaluations of $\mathcal{F}^\dagger$. However, we consider this problem under assumptions that differ from the standard formulation, and these differences motivate our algorithmic approach to the problem. This chapter is structured as follows. In Section 2.1 the motivations for developing such an adaptive reconstruction algorithm are outlined. The links with Chapter 1 are explained in the short Section 2.2. Section 2.3 introduces the problem description and notations, after which the proposed adaptive algorithm for the reconstruction of $\mathcal{F}^\dagger$ is detailed in Section 2.4. The algorithm itself is described in Section 2.4.1, and its convergence properties are proven in Section 2.4.2. Then

we study its performance on several analytically tractable test cases in Section 2.5. Finally Section 2.6 details the application of the algorithm to a challenging problem drawn from aerodynamic design. Some closing remarks are given in Section 2.7.

2.1 Motivations

Our two motivating examples are that

$$\mathcal{F}^\dagger(x) \coloneqq \mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \leq x] \tag{2.1}$$

is the cumulative distribution function (CDF) of a known real-valued function F of a random variable $\mathbf{X}$ with known distribution $\mu^\dagger$ (denoted in short by $\mathbb{P}[F(\mathbf{X}) \leq x]$ as in Chapter 1), or that

$$\mathcal{F}^\dagger(x) \coloneqq \sup_{(f,\mu)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq x] \tag{2.2}$$

is the supremum of a family of such CDFs over some class $\mathcal{A}$. We assume that we have access to a numerical optimization routine that can, for each x and some given numerical parameters q (*e.g.* the number of iterations or other convergence tolerance parameters), produce a numerical estimate or observation $\mathcal{G}(x,q)$ of $\mathcal{F}^\dagger(x)$; furthermore, we assume that $\mathcal{G}(x,q) \leq \mathcal{F}^\dagger(x)$ is always true, *i.e.* the numerical optimization routine always under-estimates the true optimum value, and that the positive error $\mathcal{F}^\dagger(x) - \mathcal{G}(x,q)$ can be controlled to some extent through the choice of the optimization parameters q, but remains essentially influenced by randomness in the optimization algorithm for each x. The assumption $\mathcal{G}(x,q) \leq \mathcal{F}^\dagger(x)$ is for example coherent with either Equation (2.1), which may be approached by increasing the number of iterations (say q) in an iterative algorithm, or Equation (2.2), which is a supremum over a set that may be explored only partially by the algorithm. A single observation $\mathcal{G}(x,q)$ yields some limited information about $\mathcal{F}^\dagger(x)$; a key limitation is that one may not even know *a priori* how accurate $\mathcal{G}(x,q)$ is. Naturally, one may repeatedly evaluate $\mathcal{G}$ at x, perhaps with different values of the optimization parameters q. However, a key observation is that a suite of observations $\mathcal{G}(x_i,q_i)$, $i = 1,\dots,I$, contains much more information than simple estimates of $\mathcal{F}^\dagger(x_i)$, $i = 1,\dots,I$, and this information can and must be used. For example, if the values $(\mathcal{G}(x_i,q_i))_{i=1}^I$ are not increasing, *e.g.* because $\mathcal{G}(x_i,q_i) > \mathcal{G}(x_{i'},q_{i'})$ and $x_i < x_{i'}$ then the observations are inconsistent with the axiomatic requirement that $\mathcal{F}^\dagger$ is an increasing function. It follows that, while the observation at x_i may be good or bad, the observation at $x_{i'}$ is even worse in the sense that it gives no more information about $\mathcal{F}^\dagger(x_{i'})$ than the observation at x_i does; the observation at $x_{i'}$ is thus a good candidate for repetition with more stringent optimization parameters q—and this is not something that could have been known without comparing it to the rest of the data set.

The purpose of this chapter is to leverage this and similar observations to define an algorithm for the reconstruction of the function $\mathcal{F}^\dagger$, repeating old observations of

insufficient quality and introducing new ones as necessary. The principal parameter in the algorithm is an "exchange rate" $\mathcal{E}$ that quantifies the degree to which the algorithm prefers to have a few high-quality evaluations versus many poor-quality evaluations. Our approach is slightly different from classical isotonic (or monotonic) regression, which is understood as the least-squares fitting of an increasing function to a set of points in the plane. The latter problem is uniquely solvable and its solution can be constructed by the pool adjacent violators algorithm (PAVA) extensively studied in [7]. This algorithm consists of exploring the data set from left to right until the monotonicity condition is violated, and replacing the corresponding observations by their average while back-averaging to the left if needed to maintain monotonicity. Extensions to the PAVA have been developed in [85] to consider non least-squares loss functions and repeated observations, in [150] to consider "nearly-isotonic" or "nearly-convex" fits, and in [71] to consider general loss functions and partially ordered data sets. Useful references on isotonic regression also include [124] and [58].

2.2 The Optimal Uncertainty Quantification framework

In Chapter 1 we are interested in the probability that a system, whose output is a function $F\colon \mathcal{X} \to \mathbb{R}$ of inputs $\mathbf{X}$ distributed according to a probability measure $\mu^\dagger$ on an input space $\mathcal{X}$, satisfies $F(\mathbf{X}) \leq x$, where x is a specified performance threshold value. We emphasise that although we focus on a scalar performance measure, the input $\mathbf{X}$ may be a multivariate random variable. In practice, $\mu^\dagger$ and F are not known exactly; rather, it is only known that $(\mu^\dagger, F) \in \mathcal{A}$ for some admissible subset $\mathcal{A}$ of the product space of all probability measures on $\mathcal{X}$ with the set of all real-valued functions on $\mathcal{X}$. Thus, one is interested in

$$\mathcal{L}(\mathcal{A})(x) \coloneqq \inf_{(\mu,f)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq x] \quad \text{and} \quad \mathcal{U}(\mathcal{A})(x) \coloneqq \sup_{(\mu,f)\in\mathcal{A}} \mathbb{P}_{\mathbf{X}\sim\mu}[f(\mathbf{X}) \leq x].$$

The inequality

$$0 \leq \mathcal{L}(\mathcal{A})(x) \leq \mathbb{P}[F(\mathbf{X}) \leq x] \leq \mathcal{U}(\mathcal{A})(x) \leq 1$$

is, by definition, the tightest possible bound on the quantity of interest $\mathbb{P}[F(\mathbf{X}) \leq x]$ (where $\mathbb{P} \equiv \mathbb{P}_{\mathbf{X}\sim\mu^\dagger}$ as reminded above) that is compatible with the information used to specify $\mathcal{A}$. Thus, the Optimal Uncertainty Quantification (OUQ) perspective enriches the principles of worst- and best-case design to account for distributional and functional uncertainty. We concentrate our attention hereafter, without loss of generality, on the least upper bound $\mathcal{U}(\mathcal{A})(x) \equiv \mathcal{F}^\dagger(x)$, where this dependency on x is the main application we have in mind in this chapter. As a function of the performance threshold x, $\mathcal{U}(\mathcal{A})(x)$ is an increasing function, and so it is potentially advantageous to determine $\mathcal{U}(\mathcal{A})(x)$ jointly for a wide range of x values using the algorithm developed above. Indeed, determining $\mathcal{U}(\mathcal{A})(x)$ for many values of x, rather than just one value, is desirable for multiple reasons:

1. Since numerical optimization to determine $\mathcal{U}(\mathcal{A})(x)$ may be affected by errors, computing several values of $\mathcal{U}(\mathcal{A})(x)$ could lead to validate their consistency as the function $x \mapsto \mathcal{U}(\mathcal{A})(x)$ must be increasing;

2. The function $\mathcal{U}(\mathcal{A})(x)$ can be discontinuous. Thus, by computing several values of $\mathcal{U}(\mathcal{A})(x)$, one can highlight potential discontinuities and can identify key threshold values of $x \mapsto \mathcal{U}(\mathcal{A})(x)$.

In practice, an underlying task is, for any individual x, reducing the calculation of $\mathcal{U}(\mathcal{A})(x)$ to a tractable finite-dimensional optimization problem. In the OUQ paradigm presented in Chapter 1, upper and lower bounds on the performance of an incompletely-specified system are calculated via optimization problems. The estimates $\mathcal{G}(x, q)$ are such results of the optimization routine run for different optimization parameters q.

2.3 Notations and problem description

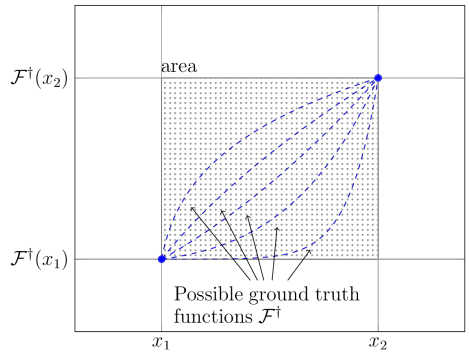
In the following, the "ground truth" performance function that we wish to reconstruct is denoted by $\mathcal{F}^\dagger\colon [a, b] \to \mathbb{R}$ and has inputs $x \in [a, b] \subset \mathbb{R}$. It is assumed that $\mathcal{F}^\dagger$ is monotonically increasing and non constant on $[a, b]$. In contrast, $\mathcal{G}\colon [a, b] \times \mathbb{R}_+ \to \mathbb{R}$ denotes the numerical process used to obtain an imperfect pointwise observation y of $\mathcal{F}^\dagger(x)$ at some point $x \in [a, b]$ for some numerical parameter $q \in \mathbb{R}_+$. Here, on a heuristic level, $q > 0$ stands for the "quality" of the noisy evaluation $\mathcal{G}(x, q)$. The main aim of this chapter is to show the effectiveness of the proposed algorithm for the adaptive reconstruction of $\mathcal{F}^\dagger$, which could be continuous or not, from imperfect pointwise observations $\mathcal{G}(x_i, q_i)$ of $\mathcal{F}^\dagger$, where we are free to choose x_{i+1} and q_{i+1} adaptively based upon x_j, q_j, and $\mathcal{G}(x_j, q_j)$ for $j \leq i$.

First, we associate with I imperfect pointwise observations $\{x_i, y_i = \mathcal{G}(x_i, q_i)\}_{i=1}^I \subset [a, b] \times \mathbb{R}$, positive numbers $\{q_i\}_{i=1}^I \subset \mathbb{R}_+$ which we will call qualities. The quality q_i quantifies the confidence we have in the pointwise observation y_i of $\mathcal{F}^\dagger(x_i)$ using the numerical process $\mathcal{G}(x_i, q_i)$. The higher this value, the greater the confidence. We divide this quality as the product of two different numbers c_i and r_i, $q_i = c_i \times r_i$, with the following definitions:

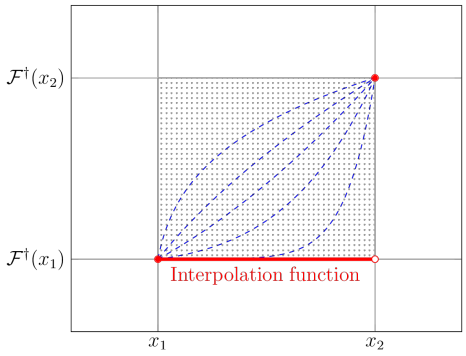
- Consistency $c_i \in \{0, 1\}$: This describes the fact that two successive points must be monotonically consistent with respect to each other. That is, when one takes two input values $x_2 > x_1$, one should have $y_2 \geq y_1$ as y must be monotonically increasing. There is no consistency associated with the very first data point as it does not have any predecessor.

- Reliability $r_i \in \mathbb{R}_+$: This describes how confident we are about the numerical value. Typically, it will be related to some error estimator if one is available, or the choice of optimization parameters. It is expected that the higher the reliability, the closer the pointwise observation is to the true value, in average.

Typically, if the observation $y_{i+1} = \mathcal{G}(x_{i+1}, q_{i+1})$ is consistent with regard to the observation $y_i = \mathcal{G}(x_i, q_i)$ where $x_{i+1} > x_i$, the quality q_{i+1} associated with y_{i+1} will be equal to $q_{i+1} = r_{i+1} \in \mathbb{R}_+^*$ since $c_{i+1} = 1$ in this case. If the value is not consistent, we have $q_{i+1} = r_{i+1} \times c_{i+1} = 0$. Finally, if $x = a$ there is no notion of



(a) Possible ground truth functions between two consecutive points x_1 and x_2. The ground truth function must lie in the area formed by these two points.



(b) Right-continuous piecewise constant interpolation function.

Figure 2.1: Possible ground truth functions between two consecutive points x_1 and x_2, and our choice of piecewise constant interpolant.

consistency as there is no point preceding it. Thereby, the quality associated with this point is only equal to its reliability.

Moreover, we associate to these pointwise observations a notion of area, illustrated in Figure 2.1 and defined as follows. Consider two consecutive points x_i and x_{i+1} with their respective observations y_i and y_{i+1}, the area $\mathfrak{a}_i$ for these two points is

$$\mathfrak{a}_i = (x_{i+1} - x_i) \times (y_{i+1} - y_i)\,. \tag{2.3}$$

Thus, we can define a vector $\mathfrak{a} = \{\mathfrak{a}_i\}_{i=1}^{I-1}$ which contains all the computed areas for the whole dataset. In addition, we can assure that if we take two points x_1 and $x_2 > x_1$ with $y_1 = \mathcal{F}^\dagger(x_1)$ and $y_2 = \mathcal{F}^\dagger(x_2)$—namely, the error at these point is equal to zero, the graph of the ground truth function $\mathcal{F}^\dagger$ must lie in the rectangular area spanned by the two points $(x_1, \mathcal{F}^\dagger(x_1))$ and $(x_2, \mathcal{F}^\dagger(x_2))$.

To adopt a conservative point of view, we choose as the approximating function $\mathcal{F}$ of $\mathcal{F}^\dagger$ a piecewise constant interpolation function, say:

$$\mathcal{F}(x) = \sum_{i=1}^{I-1} y_i \mathbb{1}_{[x_i, x_{i+1})}(x)\,, \tag{2.4}$$

where $\mathbb{1}_{\mathcal{I}}$ denotes the indicator function of the interval $\mathcal{I}$. We do not want this interpolation function to overestimate the true function $\mathcal{F}^\dagger$ as one knows that the numerical estimate in our case always underestimates the ground truth function $\mathcal{F}^\dagger$. See Figure 2.1 for an illustration of this choice, which can be viewed as a worst-case approach. Indeed, this chosen interpolation function is the worst possible function underestimating $\mathcal{F}^\dagger$ given two points x_1 and x_2 and their respective observations y_1 and y_2.

2.4 Reconstruction algorithm

The reconstruction algorithm that we propose, Algorithm 1, is driven to produce a sequences of reconstructions that converges to $\mathcal{F}^\dagger$ by following a principle of area minimisation: we associate to the discrete data set $\{x_i, y_i\}_{i=1}^I \subset [a,b] \times \mathbb{R}$ a natural notion of area (2.3) as explained above, and seek to drive this area towards zero. The motivation behind this objective is in Proposition 2 which states that the area goes to 0 as more points are added in the data set. However, the objective of minimising the area is complicated by the fact that evaluations of $\mathcal{F}^\dagger$ are imperfect. Therefore, a key user-defined parameter in the algorithm is $\mathcal{E} \in (0, \infty)$, which can be thought of as an "exchange rate" that quantifies to what extent the algorithm prefers to redo poor-quality evaluations of the target function versus driving the area measure to zero.

2.4.1 Algorithm

The main algorithm is organized as follows, starting from $I^{(0)} \geq 2$ points and a dataset that is assumed to be consistent at the initial step $n = 0$. It goes through N iterations, where N is either fixed *a priori*, or obtained *a posteriori* once a stopping criterion is met. Note that q_{new} stands for the quality of a newly generated observation y_{new} for any new point x_{new} introduced by the algorithm. The latter is driven by the user-defined "exchange rate" $\mathcal{E}$ as explained just above. At each step n, the algorithm computes the weighted area $\text{WA}^{(n)}$ as the minimum of the quality times the sum of the areas of the data points:

$$\text{WA}^{(n)} = q_-^{(n)} \times \text{A}^{(n)}\,, \tag{2.5}$$

where

$$q_-^{(n)} = \min_{1 \leq i \leq I^{(n)}} \{q_i^{(n)}\}\,, \quad \text{A}^{(n)} = \sum_{i=1}^{I^{(n)}-1} \mathfrak{a}_i^{(n)}\,, \tag{2.6}$$

$\mathfrak{a}_i^{(n)}$ is the area computed by Equation (2.3) at step n—see also Equation (2.9), and $I^{(n)}$ is the number of data points. Then it is divided into two parts according to the value of $\text{WA}^{(n)}$ compared to $\mathcal{E}$.

- If $\text{WA}^{(n)} < \mathcal{E}$, then the algorithm aims at increasing the quality $q_-^{(n)}$ of the worst data point (the one with the lowest quality) with index $i_-^{(n)} = \arg\min_{1 \leq i \leq I^{(n)}} \{q_i^{(n)}\}$ at step n. It stores the corresponding old value y_{old}, searches for a new value y_{new} by improving successively the quality of this very point, and stops when $y_{\text{new}} > y_{\text{old}}$.
- If $\text{WA}^{(n)} \geq \mathcal{E}$, then the algorithm aims at driving the total area $\text{A}^{(n)}$ to zero. In that respect, it identifies the biggest rectangle

$$\mathfrak{a}_+^{(n)} = \max_{1 \leq i \leq I^{(n)}-1} \{\mathfrak{a}_i^{(n)}\} \tag{2.7}$$

and its index

$$i_+^{(n)} = \underset{1\leq i\leq I^{(n)}-1}{\arg\max}\ \{\mathfrak{a}_i^{(n)}\} \tag{2.8}$$

and adds a new point x_{new} at the middle of this biggest rectangle. Then, it computes a new data value $y_{\text{new}} = \mathcal{G}(x_{\text{new}}, q_{\text{new}})$ with a new quality q_{new}.

In both cases, the numerical parameters q_{new} (for example a number of iterations, or the size of a sampling set or a population) are arbitrary and any value can be chosen in practice each time a new point x_{new} is added to the dataset. They can be increased arbitrarily as well each time such a new point has to be improved. Indeed, the numerical parameters q of the optimization routine we have access to can be increased as much as desired, and increasing them will improve the estimates $\mathcal{G}(x,q)$ of the true function $\mathcal{F}^\dagger(x)$ uniformly in x; see Assumption 1. The algorithm then verifies the consistency of the dataset by checking the quality of each point. If there is any inconsistent point, the algorithm computes a new value until obtaining consistency by improving successively the corresponding reliability. This is achieved in a finite number of steps starting from an inconsistent point and exploring the dataset from the left to the right.

Finally, the algorithm updates the quality vector $\{q_i^{(n+1)}\}_{i=1}^{I^{(n+1)}}$, the area vector $\{\mathfrak{a}_i^{(n+1)}\}_{i=1}^{I^{(n+1)}}$, the worst quality $q_-^{(n+1)}$ and the index $i_-^{(n+1)}$ of the corresponding point, the biggest rectangle $\mathfrak{a}_+^{(n+1)}$ and its index $i_+^{(n+1)}$, and then the new weighted area $\text{WA}^{(n+1)}$.

The stopping criterion could be a maximum number of iterations or a maximum time of computation, for example.

2.4.2 Proof of convergence

We denote by $I^{(n)}$ the number of data points, and $\{x_i^{(n)}, y_i^{(n)}, q_i^{(n)}\}_{i=1}^{I^{(n)}}$ the positions of the data points, the observations given by the optimization algorithm at these positions, and the qualities associated with the optimization algorithm at the step n of Algorithm 1. For each $i = 1, \dots, I^{(n)} - 1$, we define $s_i^{(n)} = [x_i^{(n)}, x_{i+1}^{(n)}[\subset [a,b]$ and the vector containing all rectangle areas $\{\mathfrak{a}_i^{(n)}\}_{i=1}^{I^{(n)}-1}$ by:

$$\mathfrak{a}_i^{(n)} = (x_{i+1}^{(n)} - x_i^{(n)}) \times (y_{i+1}^{(n)} - y_i^{(n)})\,. \tag{2.9}$$

The pointwise observation $y_i^{(n)} = \mathcal{G}(x_i^{(n)}, q_i^{(n)})$ is thus associated to the quality $q_i^{(n)} \in \mathbb{R}_+$, which quantifies the confidence we have in this observation as outlined in the problem description in Section 2.3. This number can represent the inverse error achieved by the optimization algorithm, for example, or the number of iterations, or the number of individuals in a population, or any other numerical parameter pertaining to this optimization process. The higher it is, the closer the observation is to the true target value. Therefore we consider the following assumption on the numerical process $\mathcal{G}$.

Algorithm 1: Adaptive algorithm to reconstruct a monotonically increasing function $\mathcal{F}^\dagger$.

Input: $I^{(0)} \geq 2$, $\{x_i^{(0)}, y_i^{(0)}, q_i^{(0)}\}_{i=1}^{I^{(0)}}$ and $\mathcal{E}$.
Output: $\{x_i^{(N)}, y_i^{(N)}, q_i^{(N)}\}_{i=1}^{I^{(N)}}$ with $I^{(N)} \geq I^{(0)}$.

Initialization:

Get the worst quality point and its index:

- $q_-^{(0)} = \min\limits_{1\leq i\leq I^{(0)}} \{q_i^{(0)}\}$;
- $i_-^{(0)} = \arg\min\limits_{1\leq i\leq I^{(0)}} \{q_i^{(0)}\}$.

Compute the area of each pair of data points: $\mathfrak{a}_i^{(0)} = (x_{i+1}^{(0)} - x_i^{(0)}) \times (y_{i+1}^{(0)} - y_i^{(0)})$.
Get the biggest rectangle and its index:

- $\mathfrak{a}_+^{(0)} = \max\limits_{1\leq i\leq I^{(0)}-1} \{\mathfrak{a}_i^{(0)}\}$;
- $i_+^{(0)} = \arg\max\limits_{1\leq i\leq I^{(0)}-1} \{\mathfrak{a}_i^{(0)}\}$.

Define the weighted area at step $n = 0$ as $\mathrm{WA}^{(0)} = q_-^{(0)} \times \sum\limits_{i=1}^{I^{(0)}-1} \mathfrak{a}_i^{(0)}$.

while $n < N$ **do**
  if $\mathrm{WA}^{(n)} < \mathcal{E}$ **then**
    Data points are unchanged: $I^{(n+1)} = I^{(n)}$ and $\{x_i^{(n+1)}\}_{i=1}^{I^{(n+1)}} = \{x_i^{(n)}\}_{i=1}^{I^{(n)}}$;
    Store the old value $y_{\mathrm{old}} = y_{i_-^{(n)}}^{(n)}$;
    while $y_{\mathrm{new}} \leq y_{\mathrm{old}}$ **do**
      Compute a new value $y_{\mathrm{new}} = \mathcal{G}(x_{i_-^{(n)}}^{(n)}, q_{\mathrm{new}})$;
    end
  else
    Introduce a new point at the middle of the biggest rectangle:
    $I^{(n+1)} = I^{(n)} + 1$, $x_{\mathrm{new}} = \frac{1}{2}(x_{i_+^{(n)}}^{(n)} + x_{i_+^{(n)}+1}^{(n)})$, and
    $(x_1^{(n+1)}, \ldots, x_{i_+^{(n)}}^{(n+1)}, x_{i_+^{(n)}+1}^{(n+1)}, x_{i_+^{(n)}+2}^{(n+1)}, \ldots, x_{I^{(n+1)}}^{(n+1)}) =$
    $(x_1^{(n)}, \ldots, x_{i_+^{(n)}}^{(n)}, x_{\mathrm{new}}, x_{i_+^{(n)}+1}^{(n)}, \ldots, x_{I^{(n)}}^{(n)})$;
    Compute the new value $y_{\mathrm{new}} = \mathcal{G}(x_{\mathrm{new}}, q_{\mathrm{new}})$;
  end
  Verify consistency of the pointwise observations $\{y_i^{(n+1)})\}_{i=1}^{I^{(n+1)}}$ by checking their quality. If there are not consistent, recompute them until obtaining consistency and then update the quality vector;
  Compute the new quality vector $\{q_i^{(n+1)}\}_{i=1}^{I^{(n+1)}}$ and area vector $\{\mathfrak{a}_i^{(n+1)}\}_{i=1}^{I^{(n+1)}}$;
  Update $q_-^{(n+1)}, i_-^{(n+1)}, \mathfrak{a}_+^{(n+1)}$ and $i_+^{(n+1)}$;
  Compute $\mathrm{WA}^{(n+1)} = q_-^{(n+1)} \times \sum\limits_{i=1}^{I^{(n+1)}-1} \mathfrak{a}_i^{(n+1)}$;
  $n = n + 1$;
end

Assumption 1. *$\mathcal{G}(x,q)$ converges to $\mathcal{F}^\dagger(x)$ as $q \to +\infty$ uniformly in $x \in [a,b]$, that is:*

$$\forall \varepsilon > 0\,,\ \exists Q > 0 \text{ such that } \forall q \geq Q\,,\ \forall x \in [a,b]\,,\ \left|\mathcal{G}(x,q) - \mathcal{F}^\dagger(x)\right| \leq \varepsilon\,.$$

Moreover, we can guarantee that:

$$\forall x \in [a,b]\,, \quad \forall q \in \mathbb{R}_+\,, \quad \mathcal{G}(x,q) \leq \mathcal{F}^\dagger(x). \tag{2.10}$$

That is, the optimization algorithm will always underestimate the true value $\mathcal{F}^\dagger(x)$. In this way, one can model the relationship between the numerical estimate $\mathcal{G}$ and the true value $\mathcal{F}^\dagger$ as:

$$\forall x \in [a,b]\,, \quad \forall q \in \mathbb{R}_+\,, \quad \mathcal{G}(x,q) = \mathcal{F}^\dagger(x) - \varepsilon(x,q)\,, \tag{2.11}$$

where ε is a positive random variable. These assumptions imply some robustness and stability of the algorithm we use.

In the following, we will assume that $I^{(0)} \geq 2$. That is, we have at least two data points at the beginning of the reconstruction algorithm. Also among these points, we have one point at $x = a$ and another one at $x = b$. Moreover, we will assume that the initial dataset is consistent. Since Algorithm 1 recomputes the inconsistent points at all steps, we can also consider in the following that any new numerical observation is actually consistent. Also, we need to guarantee that the weighted area $\mathrm{WA}^{(n)}$ will permanently oscillate about $\mathcal{E}$ as the iteration step n is increasing; this is the purpose of Assumption 3 below as shown in the subsequent Proposition 1. Then from these properties it will be shown that Algorithm 1 is convergent, as stated in Theorem 2.

Assumption 2. *Any new numerical value obtained by Algorithm 1 is consistent.*

Assumption 3. $q_-^{(n)} \to +\infty$ *as* $n \to \infty$.

Within Assumption 2 all points have a consistency of 1, and therefore $q = r > 0$ the reliability. Besides, one has $\mathcal{G}(x_i^{(n)}, q_i^{(n)}) \leq \mathcal{G}(x_{i+1}^{(n)}, q_{i+1}^{(n)})$, that is, $y_i^{(n)} \leq y_{i+1}^{(n)}$ for all points i and steps n. We finally define the sequence of piecewise constant reconstruction functions $\mathcal{F}^{(n)}$ as follows.

Definition 1. *For each $x \in [a,b]$, we define the reconstructing function $\mathcal{F}^{(n)}$ at step n as:*

$$\mathcal{F}^{(n)}(x) = \sum_{i=1}^{I^{(n)}-1} y_i^{(n)} \mathbb{1}_{s_i^{(n)}}(x)\,,$$

and $\mathcal{F}^{(n)}(x_{I^{(n)}}^{(n)}) = \mathcal{F}^{(n)}(b) = y_{I^{(n)}}^{(n)}$.

Now let:

$$E^+ = \{n \in \mathbb{N}\,;\ \mathrm{WA}^{(n)} \geq \mathcal{E}\}\,, \quad E^- = \{n \in \mathbb{N}\,;\ \mathrm{WA}^{(n)} < \mathcal{E}\}\,, \tag{2.12}$$

which are such that $E^+ \cup E^- = \mathbb{N}$ and $E^+ \cap E^- = \emptyset$. In order to prove the convergence (in a sense to be given) of Algorithm 1, we first need to establish the following intermediate results, Proposition 1, Proposition 2, and Proposition 3. They clarify the behaviour of the sequence $\mathrm{WA}^{(n)}$ when points are added to the dataset and the largest area $\mathfrak{a}_+^{(n)}$ is divided into four parts at each iteration step n; see Figure 2.2.

Proposition 1. *E^+ is infinite.*

Proof. Let us assume that E^+ is finite: $\exists N$ such that $\forall n \geq N$, $n \in E^-$. Therefore we are in the situation $\mathrm{WA}^{(n)} < \mathcal{E}$, the minimum quality $q_-^{(n)}$ of the data goes to infinity, and the total area $\mathrm{A}^{(n)}$ is modified although the evaluation points $\{x_i^{(n)}\}_{i=1}^{I^{(n)}}$ and their number $I^{(n)}$ are unchanged; thus they are independent of n. Repeating this step yields

$$\lim_{n\to\infty} \mathrm{A}^{(n)} = \sum_{i=1}^{I-1} (x_{i+1} - x_i)(\mathcal{F}^\dagger(x_{i+1}) - \mathcal{F}^\dagger(x_i)) = \mathrm{A} > 0$$

since $\mathcal{F}^\dagger$ is monotonically increasing and non constant on $[a, b]$, and Assumption 1 is used. Consequently $\mathrm{WA}^{(n)} \to +\infty$ as $n \to \infty$, that is $\mathrm{WA}^{(n)} \geq \mathcal{E}$ $\forall n \geq N_1$ for some N_1, which is a contradiction. □

The set E^+ is therefore of the form:

$$E^+ = \bigcup_{k\geq 1} [\![m_k, n_k]\!], \quad [\![m_k, n_k]\!] = \{n \in \mathbb{N};\ m_k \leq n \leq n_k\}.$$

Let us introduce the strictly increasing application $\varphi : \mathbb{N} \to \mathbb{N}$ such that $\varphi(p)$ is the p-th element of E^+ (in increasing order), and $[\![m_k, n_k]\!] = \varphi([\![p_k + 1, p_{k+1}]\!])$. p is the counter of the elements of E^+, and n is the corresponding iteration number.

Proposition 2. *Let $I^{(\varphi(p))} = I^{(\varphi(0))} + p$. Then*

$$\mathrm{A}^{(\varphi(p))} = \sum_{i=1}^{I^{(\varphi(p))}-1} \mathfrak{a}_i^{(\varphi(p))} = \mathrm{O}\left(\frac{1}{\sqrt{p}}\right)$$

as $p \to \infty$, and $\mathrm{A}^{(n)} \to 0$ as $n \to \infty$.

Proof. Let $k \geq 1$ and $n = \varphi(p) \in [\![m_k, n_k]\!]$, where $p \in [\![p_k + 1, p_{k+1}]\!]$. Let $\mathrm{A}^{(n)}$ be given by Equation (2.6), $\mathfrak{a}_+^{(n)}$ be given y Equation (2.7), and $i_+^{(n)}$ be given by Equation (2.8). At iteration $n + 1$ one has:

$$x_i^{(n+1)} = \begin{cases} x_i^{(n)} & \text{for } 1 \leq i \leq i_+^{(n)}, \\ \dfrac{1}{2}\left(x_{i_+^{(n)}}^{(n)} + x_{i_+^{(n)}+1}^{(n)}\right) & \text{for } i = i_+^{(n)} + 1, \\ x_{i-1}^{(n)} & \text{for } i_+^{(n)} + 2 \leq i \leq I^{(n+1)}. \end{cases}$$

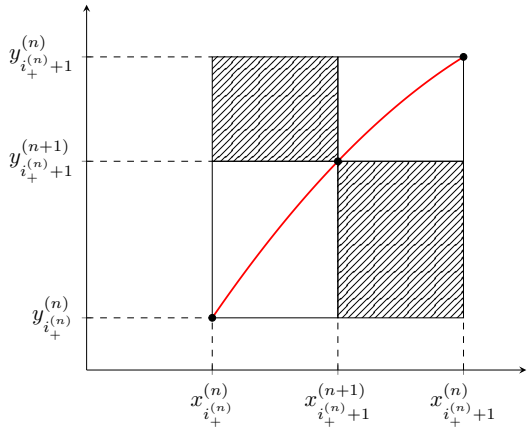


Figure 2.2: New area when one adds a point at the middle of the biggest rectangle.

Also $y_i^{(n+1)} \leq y_{i+1}^{(n+1)}$ for $1 \leq i \leq I^{(n+1)} - 1$. One may check that $\mathfrak{a}_+^{(n)} = 2\mathfrak{a}_{i_+^{(n)}}^{(n+1)} + 2\mathfrak{a}_{i_+^{(n)}+1}^{(n+1)}$ (see Figure 2.2) and therefore:

$$\mathrm{A}^{(n+1)} = \mathrm{A}^{(n)} - \mathfrak{a}_+^{(n)} + \mathfrak{a}_{i_+^{(n)}}^{(n+1)} + \mathfrak{a}_{i_+^{(n)}+1}^{(n+1)} = \mathrm{A}^{(n)} - \frac{1}{2}\mathfrak{a}_+^{(n)}. \tag{2.13}$$

Besides $\mathrm{A}^{(n)} \leq (I^{(n)} - 1)\mathfrak{a}_+^{(n)}$ so that one has:

$$\begin{aligned}\mathrm{A}^{(n+1)} &\leq \mathrm{A}^{(n)} - \frac{\mathrm{A}^{(n)}}{2(I^{(n)} - 1)} \\ &\leq \mathrm{A}^{(n)} \left(\frac{2(I^{(n)} - 1) - 1}{2(I^{(n)} - 1)} \right),\end{aligned}$$

or:

$$\mathrm{A}^{(\varphi(p)+1)} \leq \mathrm{A}^{(\varphi(p))} \left(\frac{2(I^{(\varphi(p))} - 1) - 1}{2(I^{(\varphi(p))} - 1)} \right). \tag{2.14}$$

At this stage two situations arise:

- either $p \in [\![p_k + 1, p_{k+1} - 1]\!]$, in which case $\varphi(p) + 1 = \varphi(p+1)$;
- or $p = p_{k+1}$, in which case by our algorithm $\mathrm{A}^{(n)}$ is kept constant from $n = n_k + 1$ to $n = m_{k+1}$; that is $\mathrm{A}^{(n_k+1)} = \mathrm{A}^{(m_{k+1})}$, or:
$$\mathrm{A}^{(\varphi(p_{k+1})+1)} = \mathrm{A}^{(\varphi(p_{k+1}+1))}.$$

The choice of k being arbitrary, one concludes that Equation (2.14) also reads $\forall p \in \mathbb{N}$:

$$\begin{aligned}\mathrm{A}^{(\varphi(p+1))} &\leq \mathrm{A}^{(\varphi(p))} \left(\frac{2(I^{(\varphi(p))} - 1) - 1}{2(I^{(\varphi(p))} - 1)} \right) \\ &\leq \mathrm{A}^{(\varphi(p))} \left(\frac{2(I^{(\varphi(0))} + p - 1) - 1}{2(I^{(\varphi(0))} + p - 1)} \right).\end{aligned}$$

Thus:

$$\begin{aligned}\mathrm{A}^{(\varphi(p))} &\leq \mathrm{A}^{(\varphi(1))} \prod_{i=1}^{p-1} \left(\frac{2(I^{(\varphi(0))} + i - 1) - 1}{2(I^{(\varphi(0))} + i - 1)} \right) \\ &\leq \mathrm{A}^{(\varphi(1))} \prod_{i=1}^{p-1} \left(\frac{1 + \frac{\alpha}{i}}{1 + \frac{\beta}{i}} \right),\end{aligned}$$

letting $\alpha = I^{(\varphi(0))} - \frac{3}{2}$ and $\beta = I^{(\varphi(0))} - 1$. However,

$$\sum_{i=1}^{p} \log\left(1 + \frac{\alpha}{i}\right) = \alpha \sum_{i=1}^{p} \frac{1}{i} + C''_p$$

where $\lim_{p\to\infty} C''_p = C''$, and

$$\sum_{i=1}^{p} \frac{1}{i} = \log p + \gamma + \varepsilon'_p\,,$$

where γ is the Euler constant and $\lim_{p\to\infty} \varepsilon'_p = 0$. Consequently:

$$\begin{aligned}\sum_{i=1}^{p-1} \log\left(1 + \frac{\alpha}{i}\right) - \sum_{i=1}^{p-1} \log\left(1 + \frac{\beta}{i}\right) &= (\alpha - \beta)\log(p-1) + C'_p \\ &= (\alpha - \beta)\left[\log p + \log\left(1 - \frac{1}{p}\right)\right] + C'_p \\ &= \log\left(\frac{1}{\sqrt{p}}\right) + C_p\,,\end{aligned}$$

since $\alpha - \beta = -\frac{1}{2}$; again C_p and C'_p are sequences with constant limits $\lim_{p\to\infty} C_p = C$ and $\lim_{p\to\infty} C'_p = C'$. Therefore,

$$\prod_{i=1}^{p-1} \left(\frac{1 + \frac{\alpha}{i}}{1 + \frac{\beta}{i}} \right) = \frac{\mathcal{C}}{\sqrt{p}}(1 + \varepsilon_p)$$

where $\mathcal{C}$ is a constant, and $\lim_{p\to\infty} \varepsilon_p = 0$. One also concludes that $\mathrm{A}^{(n)}$, which is either kept constant or equal to $\mathrm{A}^{(\varphi(p))}$, converges to 0 as $n \to \infty$. Hence the claimed results hold. □

Proposition 3. *E^- is infinite.*

Proof. Let us assume that E^- is finite: $\exists N$ such that $\forall n \geq N$, $n \in E^+$. Therefore we are in the situation $\mathrm{WA}^{(n)} \geq \mathcal{E} > 0$, and $\varphi(n)$ has the form $\varphi(n) = n - n_0$, $n \geq N$ for some $n_0 \in \mathbb{N}$. From Proposition 2:

$$\mathrm{A}^{(n-n_0)} = \mathrm{O}\left(\frac{1}{\sqrt{n}}\right),$$

thus $\mathrm{A}^{(n)} \to 0$ and $\mathrm{WA}^{(n)} \to 0$ as $n \to \infty$ since $q_-^{(n)}$ is kept unchanged, which is a contradiction. □

We now provide three results on the convergence of Algorithm 1. As is to be expected, the algorithm can only be shown to converge uniformly when the target response function $\mathcal{F}^\dagger$ is sufficiently smooth; otherwise, the convergence is at best pointwise or in mean.

Theorem 2 (Algorithm convergence)**.** *Assume that $\mathcal{F}^\dagger$ is strictly increasing. Then, for any choice of $\mathcal{E} > 0$, Algorithm 1 is convergent in the following senses:*

- *If $\mathcal{F}^\dagger$ is piecewise continuous on $[a, b]$, then $\lim_{n\to\infty} \mathcal{F}^{(n)}(x) = \mathcal{F}^\dagger(x)$ at all points $x \in [a, b]$ where $\mathcal{F}^\dagger$ is continuous;*
- *If $\mathcal{F}^\dagger$ is continuous on $[a, b]$, then convergence holds uniformly: $\|\mathcal{F}^{(n)} - \mathcal{F}^\dagger\|_\infty \xrightarrow[n\to\infty]{} 0$.*

Proof. Let $\mathcal{E} > 0$. We know from Propositions 1 and 3 that $\mathrm{WA}^{(n)}$ will oscillate about $\mathcal{E}$ in the iterating process as $n \to \infty$, while $\lim_{n\to\infty} q_-^{(n)} = +\infty$ from Assumption 3. Besides let:

$$\Delta^{(n)} = \sup_{1\le i\le I^{(n)}-1} \left| x_{i+1}^{(n)} - x_i^{(n)} \right| .$$

Assuming for example that for some j, $s_j^{(n)} = [x_j^{(n)}, x_{j+1}^{(n)})$ is never divided in two in the iteration process and is thus independent of n, it turns out that $\mathfrak{a}_j^{(n)} \to (x_{j+1} - x_j)(\mathcal{F}^\dagger(x_{j+1}) - \mathcal{F}^\dagger(x_j)) > 0$ as $n \to \infty$, which is impossible because $\mathrm{A}^{(n)}$ goes to 0 as $n \to \infty$ from Proposition 2. Therefore there exists some $m \in \mathbb{N}^*$ (depending on n) such that $\Delta^{(n+m)} \le \frac{1}{2}\Delta^{(n)}$; also the sequence $\Delta^{(n)}$ is decreasing, hence $\Delta^{(n)} \to 0$ as $n \to \infty$.

Now let $x \in [x_i^{(n)}, x_{i+1}^{(n)})$, then:

$$\begin{aligned} \left| \mathcal{F}^{(n)}(x) - \mathcal{F}^\dagger(x) \right| &= \left| \mathcal{G}(x_i^{(n)}, q_i^{(n)}) - \mathcal{F}^\dagger(x) \right| \\ &\le \left| \mathcal{G}(x_i^{(n)}, q_i^{(n)}) - \mathcal{F}^\dagger(x_i^{(n)}) \right| + \left| \mathcal{F}^\dagger(x_i^{(n)}) - \mathcal{F}^\dagger(x) \right| . \end{aligned}$$

But $x_i^{(n)} \to x$ as $n \to \infty$ because $\Delta^{(n)} \to 0$; thus if $\mathcal{F}^\dagger$ is continuous at x, the second term on the right hand side above goes to 0 as $n \to \infty$. However, if $\mathcal{F}^\dagger$ is continuous everywhere on $[a, b]$, it is in addition uniformly continuous on $[a, b]$ by Heine's theorem, and the second term goes to 0 as $n \to \infty$ uniformly on $[a, b]$. Finally, invoking Assumption 1, the first term on the right hand side above also tends to 0 as $n \to \infty$. This completes the proof. □

Proposition 4 (Convergence in mean)**.** *Let $\mathcal{F}^\dagger \colon [a, b] \to \mathbb{R}$ be piecewise continuous. Then Algorithm 1 is convergent in mean in the sense that*

$$\left\| \mathcal{F}^{(n)} - \mathcal{F}^\dagger \right\|_1 \xrightarrow[n\to\infty]{} 0.$$

Proof. We can check that the sequence $\mathcal{F}^{(n)}$ is monotone. Indeed, if $\mathrm{WA}^{(n)} < \mathcal{E}$, then by construction we have

$$\mathcal{F}^{(n+1)}(x) - \mathcal{F}^{(n)}(x) \ge \left(y_{i_-^{(n)}}^{(n+1)} - y_{i_-^{(n)}}^{(n)} \right) \mathbb{1}_{s_-^{(n)}}(x) \ge 0$$

where $s_-^{(n)} = \left[x_{i_-^{(n)}}^{(n)}, x_{i_-^{(n)}+1}^{(n)}\right)$. However, if $\mathrm{WA}^{(n)} > \mathcal{E}$, then consistency implies that

$$\mathcal{F}^{(n+1)}(x) - \mathcal{F}^{(n)}(x) \geq \left(y_{i_+^{(n)}+1}^{(n+1)} - y_{i_+^{(n)}}^{(n)}\right) \mathbb{1}_{s_+^{(n+1)}}(x) \geq 0$$

where $s_+^{(n+1)} = \left[x_{i_+^{(n)}+1}^{(n+1)}, x_{i_+^{(n)}+2}^{(n+1)}\right)$. The claim now follows from the monotone convergence theorem and the fact that $\mathcal{F}^{(0)}$ is integrable. □

2.5 Analytical test cases

To show the effectiveness of Algorithm 1, we try it on two cases, in which $\mathcal{F}^\dagger$ is a continuous function and a discontinuous function respectively. For both cases, the error between the numerical estimate and the ground truth function is modelled as a random variable following a Log-normal distribution. That is,

$$\forall x \in [a,b],\ \varepsilon(x) \sim \mathrm{Log}\mathcal{N}(\mu(x), \sigma^2), \tag{2.15}$$

with $\sigma^2 = 1$ and $\mu(x)$ is chosen as $\mathbb{P}[0 \leq \varepsilon(x) \leq 0.1 \cdot \mathcal{F}^\dagger(x)] = 0.9$. Thus, the mean μ is different for each $x \in [a,b]$. As we have access to the ground truth function and for validation purpose, the quality value associated to a numerical point is the inverse of the relative error. Moreover, we assume that the initial points are consistent. For illustrative purposes, we set the parameter $\mathcal{E} = 15$ for the examples considered below.

2.5.1 $\mathcal{F}^\dagger$ is a continuous function

First, consider the function $\mathcal{F}^\dagger \in C^0([1,2],[1,2])$ defined as follows:

$$\mathcal{F}^\dagger(x) = \begin{cases} \mathcal{F}_1^\dagger(x) & \text{if } x \in [1, \frac{3}{2}]\,, \\ \mathcal{F}_2^\dagger(x) & \text{if } x \in [\frac{3}{2}, 2]\,, \end{cases}$$

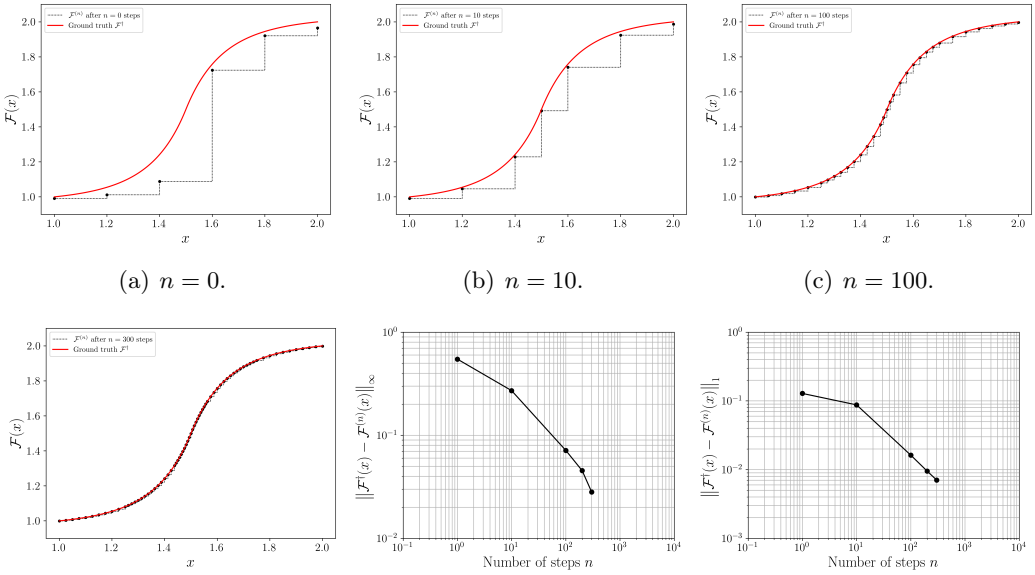
with

$$\begin{aligned} \mathcal{F}_1^\dagger(x) &= a_1 \exp(x^3) + b_1\,, \\ \mathcal{F}_2^\dagger(x) &= a_2 \exp((3-x)^3) + b_2\,, \end{aligned} \tag{2.16}$$

where:

$$\begin{aligned} a_1 &= -\frac{1}{2(\exp(1) - \exp(27/8))}\,, & b_1 &= \frac{3 - 2\exp(19/8)}{2(1 - \exp(19/8))}\,, \\ a_2 &= -a_1\,, & b_2 &= 2a_1 \exp(27/8) + b_1\,. \end{aligned}$$

The target function $\mathcal{F}^\dagger$ and the reconstructions $\mathcal{F}^{(n)}$ obtained through the algorithm for several values of the step n are shown on Figure 2.3. For each n, the reconstruction $\mathcal{F}^{(n)}$ is increasing and the initial points are consistent. The ∞-norm and 1-norm of the error appear to converge to zero with approximate rates -0.512 and -0.534 respectively.



(a) $n = 0$. (b) $n = 10$. (c) $n = 100$.

(d) $n = 300$. (e) ∞-norm error. (f) 1-norm error.

Figure 2.3: Evolution of $\mathcal{F}^{(n)}$ and the ∞- and 1-norms of the error $\mathcal{F}^\dagger - \mathcal{F}^{(n)}$ as functions of the iteration count n for a smooth ground truth $\mathcal{F}^\dagger$.

2.5.2 $\mathcal{F}^\dagger$ is a discontinuous function

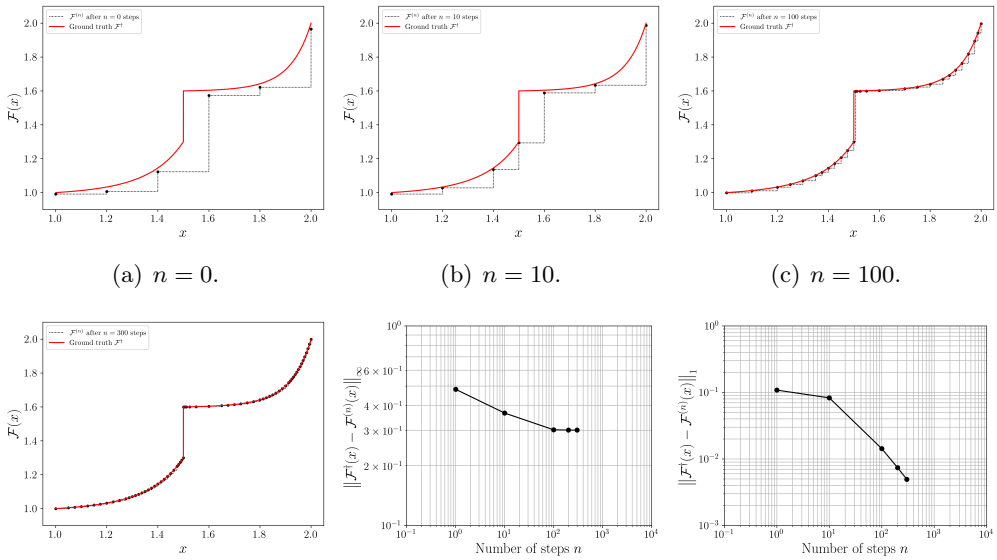
Now, consider the discontinuous function $\mathcal{F}^\dagger$ defined as follows:

$$\mathcal{F}^\dagger(x) = \begin{cases} \mathcal{F}_1^\dagger & \text{if } x \in \left[1, \frac{3}{2}\right] , \\ \mathcal{F}_2^\dagger & \text{if } x \in \left(\frac{3}{2}, 2\right] , \end{cases}$$

where $\mathcal{F}_1^\dagger$ and $\mathcal{F}_2^\dagger$ are given by (2.16), and:

$$a_1 = -\frac{1}{2(\exp(1) - \exp(27/8))} , \qquad b_1 = \frac{3 - 2\exp(19/8)}{2(1 - \exp(19/8))} ,$$
$$a_2 = \frac{2}{5(\exp(8) - \exp(27/8))} , \qquad b_2 = \frac{10 - 8\exp(37/8)}{5(1 - \exp(37/8))} .$$

Here, $\mathcal{F}^\dagger$ is piecewise continuous on $[1, \frac{3}{2}]$ and $(\frac{3}{2}, 2]$. In this case, one can apply Proposition 4. The target function $\mathcal{F}^\dagger$ and the reconstructions $\mathcal{F}^{(n)}$ obtained through the algorithm for several values of the step n are shown on Figure 2.4. Observe that the approximation quality, as measured by the ∞-norm of the error $\mathcal{F}^\dagger - \mathcal{F}^{(n)}$, quite rapidly saturates and does not converge to zero. This is to be expected for this discontinuous target $\mathcal{F}^\dagger$, since closeness of two functions in the supremum norm mandates that they have approximately the same discontinuities in exactly the same places. The 1-norm error, in contrast, appears to converge at the rate -0.561. Regarding computational costs, the number of calls to the numerical model is lower when $\mathcal{F}^\dagger$ is continuous than when it is discontinuous. For both examples above and for the same number of data points, the number of evaluations of the numerical



(a) $n = 0$. (b) $n = 10$. (c) $n = 100$.

(d) $n = 300$. (e) ∞-norm error. (f) 1-norm error.

Figure 2.4: Evolution of $\mathcal{F}^{(n)}$ and the ∞- and 1-norms of the error $\mathcal{F}^\dagger - \mathcal{F}^{(n)}$ as functions of the iteration count n for a discontinuous ground truth $\mathcal{F}^\dagger$.

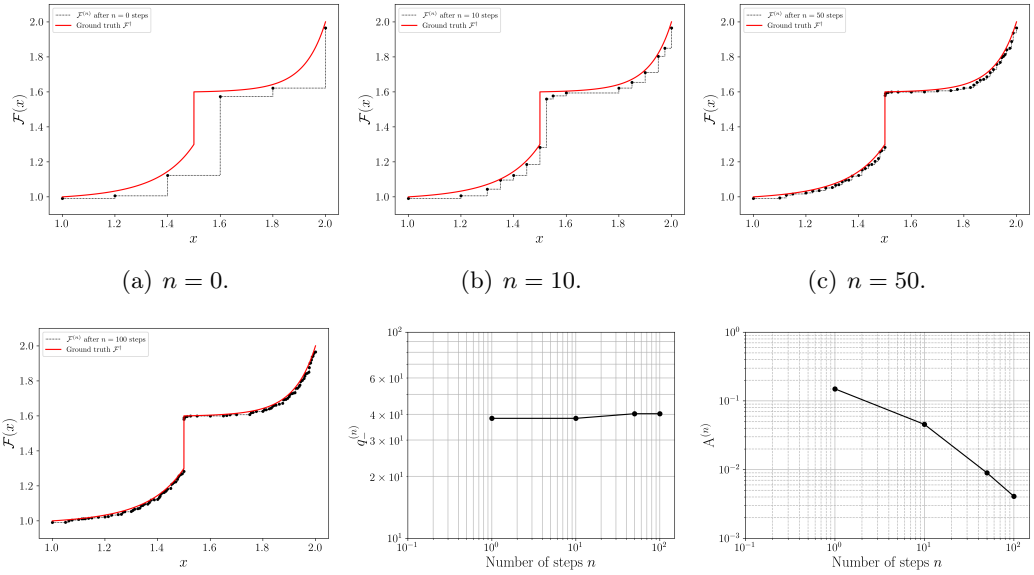
model (analytical formula in the present case) in the discontinuous case is about six times higher than the number of evaluations in the continuous case. This is because the algorithm typically adds more points near discontinuities and making them consistent increases the number of calls to the model.

2.5.3 Influence of the user-defined parameter $\mathcal{E}$

We consider the case in which $\mathcal{F}^\dagger$ is discontinuous, as in Section 2.5.2. We will show the influence of the choice of the parameter $\mathcal{E}$ on the reconstruction function $\mathcal{F}^{(n)}$.

2.5.3.1 Case $\mathcal{E} \ll 1$

Let us consider the case $\mathcal{E} = 10^{-4} \ll 1$. This choice corresponds to the case where one wishes to split over redo the worst quality point. This can be seen on Figure 2.5 where the worst quality is almost constant over 100 steps while the sum of areas strongly decreases; see Figure 2.5(e) and Figure 2.5(f) respectively. At each step, the algorithm is adding a new point by splitting the biggest rectangle. One can note on Figure 2.5(f) that the minimum of the quality is not constant. It means that when the algorithm added a new data point, the point with the worst quality was not consistent any more and had to be recomputed. In summary, in this case, we obtain more points but with lower quality values.



(a) $n = 0$. (b) $n = 10$. (c) $n = 50$.

(d) $n = 100$. (e) Minimum of the quality. (f) Total area.

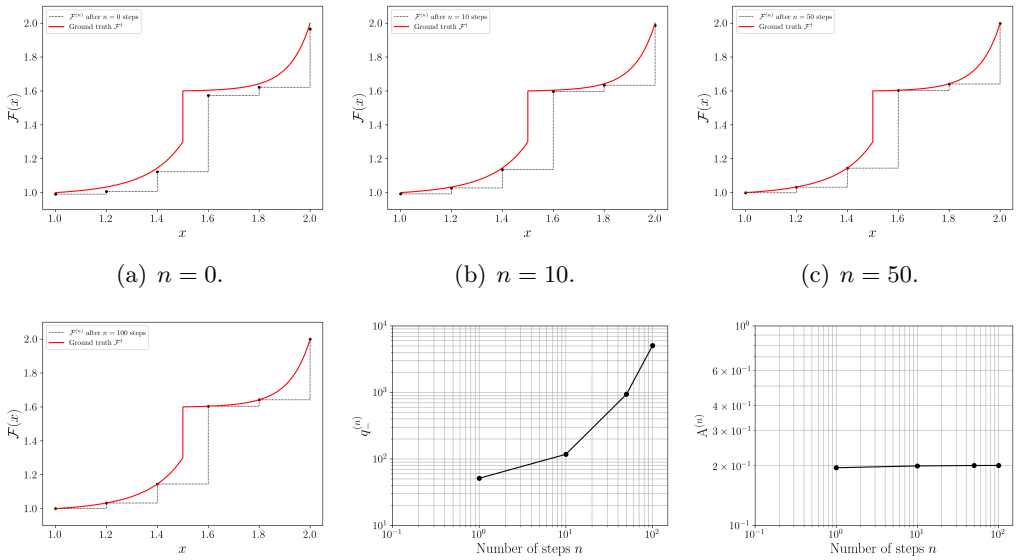
Figure 2.5: Evolution of $\mathcal{F}^{(n)}$ and the minimum of the quality and the total area as functions of the iteration count n for a discontinuous ground truth $\mathcal{F}^\dagger$ with $\mathcal{E} = 10^{-4}$.

2.5.3.2 Case $\mathcal{E} \gg 1$

We now consider the case $\mathcal{E} = 10^4 \gg 1$. This choice corresponds to the case where one wishes to redo the worst quality point over split. This can be seen on Figure 2.6 where the sum of areas stays more or less the same over 100 steps while the minimum of the quality surges; see Figure 2.6(f) and Figure 2.6(e) respectively. There is no new point. The algorithm is only redoing the worst quality point to improve it. To sum up, we obtain fewer points with higher quality values.

2.6 Application to RAE2822 airfoil performance in transonic flows

For the application of Algorithm 1 to OUQ, we study the influence of geometrical imperfections of the two-dimensional RAE2822 airfoil [34, Appendix A6] on its aerodynamic performance using ONERA's CFD software *elsA* [20]. This example is taken from [46] where all numerical parameters for *elsA* runs are detailed. More particularly, the transonic flow is modeled by the steady-state Reynolds-Averaged Navier-Stokes (RANS) equations together with a Spalart-Allmaras turbulence model closure [136]. The baseline conditions of the flow are those described in [34] for the test case #6 together with the wall interference correction formulas derived in [53, pp. 386–387] and their slight modifications suggested in [59, pp. 130]. The operational parameters considered here are thus $\underline{M} = 0.729$ for the free-stream Mach number, $\underline{\alpha} = 2.31°$ for the angle of attack, and $\underline{\mathrm{Re}} = 6.50 \times 10^6$ for the Reynolds number



(a) $n = 0$. (b) $n = 10$. (c) $n = 50$.

(d) $n = 100$. (e) Minimum of the quality. (f) Total area

Figure 2.6: Evolution of $\mathcal{F}^{(n)}$ and the minimum of the quality and the total area as functions of the iteration count n for a discontinuous ground truth $\mathcal{F}^\dagger$ with $\mathcal{E} = 10^4$.

based on the chord length c, fluid velocity, temperature, and molecular viscosity at infinity. They arise from the corrections $\Delta M = 0.004$ and $\Delta\alpha = -0.61°$ given in [59, pp. 130] for the test case #6 outlined in [34], for which $M = 0.725$, $\alpha = 2.92°$, and $\mathrm{Re} = 6.50 \times 10^6$. The shape of the original RAE2822 is subsequently altered using four bumps located at four different locations: 5%, 20%, 40%, and 60% of the way along the chord c—see Figure 2.7. These bumps are characterized by B-splines functions.

The lift-to-drag ratio $\mathrm{L/D} = C_L/C_D$ of the RAE2822 wing profile (for C_L and C_D being its lift and drag coefficients, respectively; see Figure 2.8) at the baseline Reynolds Number $\underline{\mathrm{Re}}$, Mach number $\underline{M}$, and angle of attack $\underline{\alpha}$ is chosen as the performance function F with inputs $\mathbf{X} = (X_1, X_2, X_3, X_4)$, where $\{X_i\}_{i=1}^4$ are the amplitudes of the bumps. They will be considered as random variables over their respective range given in Table 2.1. Moreover, we will assume that $\{X_i\}_{i=1}^4$ are mutually independent.

Remark 7. *The ground truth law $\mu^\dagger$ of each input variable given in Table 2.1 is only used to compute the expected value $\mathbb{E}\{F(\mathbf{X})\} = \underline{\mathrm{L/D}}$. This expected value is computed through a MC simulation with 10^4 random samples. It is analogous to scenario 0 defined in the numerical examples in Section 1.3.*

An ordinary Kriging procedure has been chosen to build a surrogate model G of F. More details about ordinary Kriging will be given in Section 3.2.5. A tensorized grid of 9 evenly distributed abscissas for each parameter is used. The surrogate model is then based on $N = 9^4 = 6561$ observations. In that respect, a Gaussian

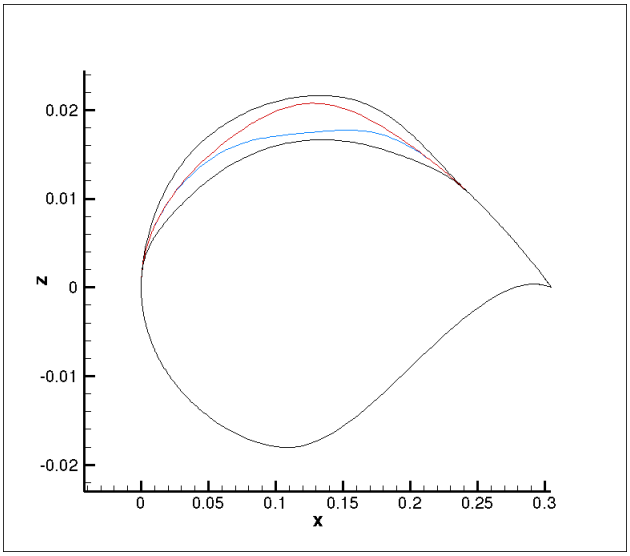


Figure 2.7: Black lines: Maximum and minimum deformation of the RAE2822 profile. Red: Maximum deformation of the third bump alone. Blue: Minimum deformation of the third bump alone. This image is taken from [46].

Input	Range	Law
Bump 1: X_1	[-0.0025c; +0.0025c]	$\mu_1^\dagger$: Beta law with $\alpha = 6, \beta = 6$
Bump 2: X_2	[-0.0025c; +0.0025c]	$\mu_2^\dagger$: Beta law with $\alpha = 2, \beta = 2$
Bump 3: X_3	[-0.0025c; +0.0025c]	$\mu_3^\dagger$: Beta law with $\alpha = 2, \beta = 2$
Bump 4: X_4	[-0.0025c; +0.0025c]	$\mu_4^\dagger$: Beta law with $\alpha = 2, \beta = 2$

Table 2.1: Range of each input parameter $\mathbf{X}$.

kernel

$$K(\mathbf{X}, \mathbf{X}') = \exp\left(-\frac{1}{2}\sum_{i=1}^{4}\frac{(X_i - X_i')^2}{\gamma_i^2}\right)$$

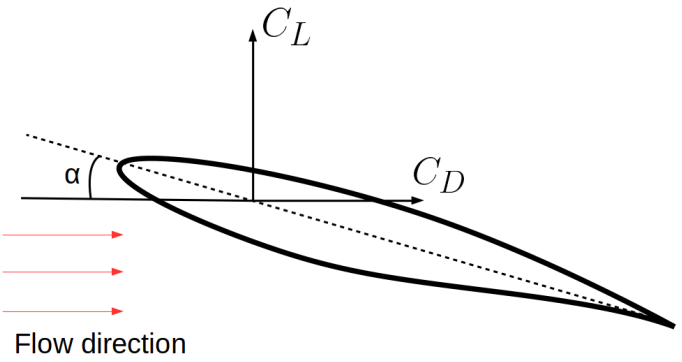
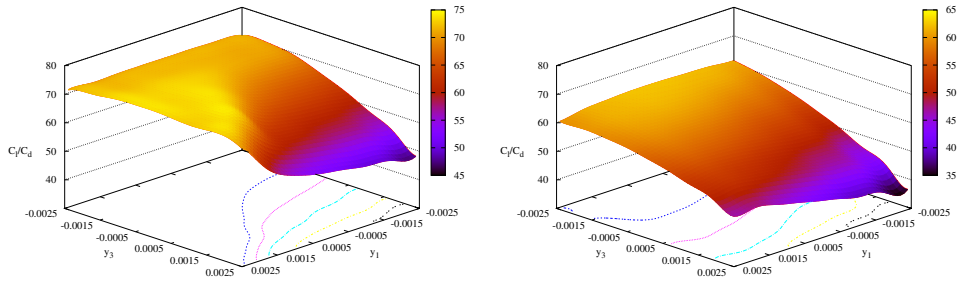


Figure 2.8: Picture depicting the lift C_L *and the drag* C_D *of an airfoil.*

has been chosen, where $\mathbf{X} = (X_1, X_2, X_3, X_4)$ and $\mathbf{X}' = (X_1', X_2', X_3', X_4')$ are inputs of the function G, and where $\gamma = (\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ are the parameters of the kernel. These parameters are chosen to minimize the variance between the ground truth data defined by the N observations and their Kriging metamodel G. The response surfaces in the (X_1, X_3) plan for two values of (X_2, X_4) are shown on Figure 2.9. The surrogate model G will be assumed accurate enough to state that G is equivalent to F, namely $G \equiv F$. Consequently, we will assume that $D_{F-G} = 0$—see Equation (1.6).



(a) $X_2 = -0.0025,\ X_4 = 0.$ (b) $X_2 = 0.0025,\ X_4 = 0.$

Figure 2.9: Response surface in the (X_1, X_3) plane with $(X_2 = -0.0025, X_4 = 0)$ (a) and $(X_2 = 0.0025, X_4 = 0)$ (b). These images are taken from [46].

One seeks to determine $\mathcal{U}(\mathcal{A})(x) \coloneqq \sup_{\mu \in \mathcal{A}} \mathbb{P}_{\mathbf{X} \sim \mu}[F(\mathbf{X}) \leq x]$, where the admissible set $\mathcal{A}$ is defined as follows:

$$\mathcal{A} = \left\{ (f, \mu) \;\middle|\; \begin{array}{c} \mathbf{X} \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4 \\ f : \mathcal{X} \mapsto \mathcal{Y} \text{ is known equal to } F \\ \mu = \mu_1 \otimes \mu_2 \otimes \mu_3 \otimes \mu_4 \\ \mathbb{E}_{\mathbf{X} \sim \mu}[f(\mathbf{X})] = \underline{\text{L/D}} \end{array} \right\}. \tag{2.17}$$

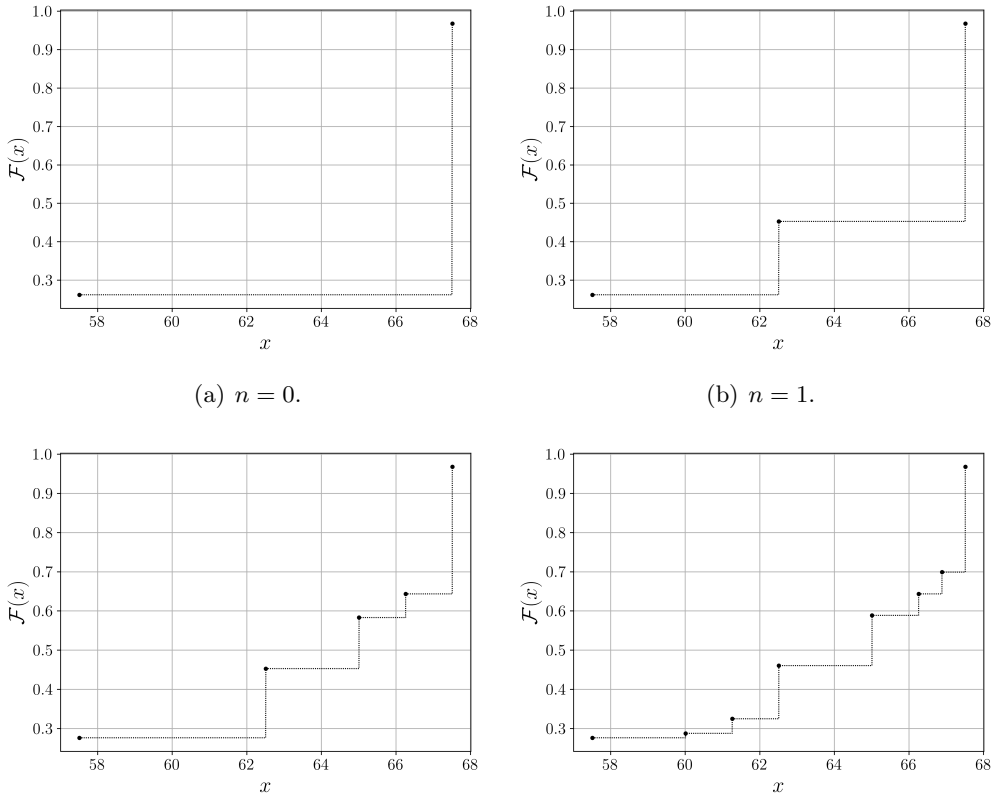
A priori, finding $\mathcal{U}(\mathcal{A})(x)$ is not computationally tractable because it requires a search over a infinite-dimensional space of probability measures defined by $\mathcal{A}$. However, as outlined in Chapter 1, this optimization problem can be reduced to a finite-dimensional one, where now the probability measures are products of finite convex combinations of Dirac masses.

Remark 8. *The admissible set $\mathcal{A}$ from (2.17) can be understood as follows:*

- *One knows the range of each input parameter $\{X_i\}_{i=1}^4$;*
- *f is exactly known as $f = F$;*
- *$\{X_i\}_{i=1}^4$ are mutually independent;*
- *Only the expected value of f is known: $\mathbb{E}_{\mathbf{X} \sim \mu}[f(\mathbf{X})]$.*

The optimization problem of determining $\mathcal{U}(\mathcal{A})(x)$ for each chosen x was solved using the DE algorithm of [140] within the `mystic` optimization framework [102] (see

Section 1.3.1). Ten iterations of Algorithm 1 have been performed using $\mathcal{E} = 1 \times 10^4$. The evolution of $\mathcal{U}(\mathcal{A})(x)$ as function of the iteration count n is shown on Figure 2.10. At $n = 0$ two consistent points are present at $x = 57.51$ and $x = 67.51$; see Figure 2.10(a). At this step, $\mathrm{WA}^{(0)} = 35289$. As $\mathrm{WA}^{(0)} \geq \mathcal{E}$, at next step $n = 1$, the algorithm adds a new point at the middle of the biggest rectangle; see Figure 2.10(b) and Figure 2.11(b). After $n = 10$ steps, eight points are now present in total with a minimum quality increasing from 5000 to 11667 and with a total area decreasing from 7.05 to 0.84; see Figure 2.11(a) and Figure 2.11(b), respectively.



(a) $n = 0$.

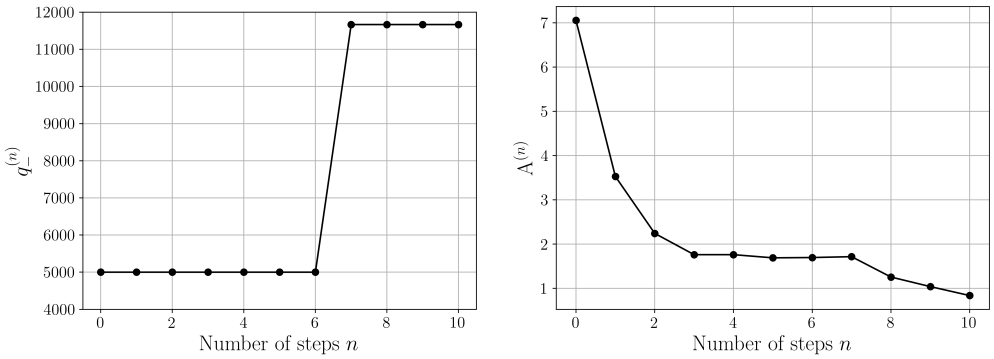
(b) $n = 1$.

(c) $n = 5$.

(d) $n = 10$.

Figure 2.10: Evolution of $\mathcal{F}(x) = \mathcal{U}(\mathcal{A})(x)$ as function of the iteration count n.

The number of iterations in this complex numerical experiment has been limited to 10 because obtaining new or improved data points consistent throughout the optimization algorithm may take up to two days (wall-clock time on a personal computer equipped with an Intel Core i5-6300HQ processor with 4 cores and 6 MB cache memory) for one single point. This running time is increased further for data points of higher quality. However this experiment shows that the proposed algorithm can be used for real-world examples in an industrial context.



(a) Evolution of the minimum of the quality.

(b) Evolution of the total area.

Figure 2.11: Evolution of the minimum of the quality and the total area as function of the iteration count n.

2.7 Conclusions

In this chapter we have developed an algorithm to reconstruct a monotonically increasing function such as the cumulative distribution function of a real-valued random variable, or the least upper bound of the performance criterion of a system as a function of its performance threshold. In particular, this latter setting has relevance to the Optimal Uncertainty Quantification (OUQ) framework of Chapter 1 we have in mind for applications to real-world incompletely specified systems. The algorithm uses imperfect pointwise evaluations of the target function, subject to partially controllable one-sided errors, to direct further evaluations either at new sites in the function's domain or to improve the quality of evaluations at already-evaluated sites. It allows for some flexibility at targeting either strategy through a user-defined "exchange rate" parameter, yielding an approximation of the target function with few high-quality points or alternatively more lower-quality points. We have studied its convergence properties and have applied it to several examples: known target functions that are either continuous and discontinuous, and a performance function for aerodynamic design of a well-documented standard profile in the OUQ setting.

Algorithm 1 is reminiscent of the classical PAVA approach to isotonic regression that applies to statistical inference with order restrictions. Examples of its use can be found in shape constrained or parametric density problems as illustrated in *e. g.* [58]. In any case, in order to obain one data point using `mystic` and Price and Storn's DE algorithm, for instance $\mathcal{U}(\mathcal{A})(x)$, several tens of thousands of evaluations of the performance function are required. Therefore, the use of a surrogate model is highly desirable, as in the aerodynamic test case of Section 2.6. Consequently, the value of the upper bound of the performance function obtained through the optimization process will strongly depend on the quality of this surrogate model. Building an accurate surrogate model is the topic of the next chapter.

Chapter 3

Building a surrogate model

Contents

In this chapter, we will present different kernel regression methods to build a surrogate model from limited knowledge of a performance function F. In Section 3.1, the motivations of building surrogate models are exposed and the problem considered is defined. Section 3.2 presents classical kernel regression methods together with their connections. These methods raise the issue of finding a "best" kernel function, which is addressed here in two ways. First, the Kernel Flow algorithm initially implemented in a classification context [31, 114] is extented to the regression context and applied to an aerodynamic case in Section 3.3. Second, a spectral version of the classical Kernel Ridge Regression algorithm within the framework of Mercer's theorem is developed in Section 3.4. Mercer's framework is reminded in Section 3.4.1, and subsequently used in Section 3.4.3.2 and Section 3.4.3.3 to introduce two algorithms based on regression and projection approaches. The Polynomial Chaos Expansion (PCE) method is briefly reviewed in Section 3.5 for completeness. The proposed algorithms are finally tested on synthetic examples in Section 3.6 and on a more complex aerodynamic example in Section 3.7.

3.1 Motivations

In the OUQ framework of Chapter 1 several tens of thousands function evaluations are required to obtain one optimal bound, by using `mystic` [101, 102] for instance. In aerodynamic applications, one function evaluation can take up to several hours. It is not conceivable to use this type of complex models for every single function evaluation. Thus, a middle ground has to be found. One way to solve this issue is to accept to evaluate the complex model at some sample points but using an approximation of the model at the remaining points to mimic the behavior of the complex model. That is called a metamodel or a surrogate model. Obviously, the quality of the model will strongly influence the trustworthiness of the obtained optimum bound. In that respect, the surrogate model has to verify the following two properties:

- It has to be low-cost to evaluate in order to be able to obtain tens of thousands function evaluations in a reasonable time. Typically, an evaluation should be less than 1 second;
- It has to be the most accurate possible given a metric to have confidence in the computations.

The problem can be summed up as follows:

Problem 1. *Let d be the dimension of the input space $\mathcal{X}$. Let $F : \mathcal{X} \subset \mathbb{R}^d \to \mathcal{Y} = \mathbb{R}$ be a smooth function. Given I observations of the function F, denoted by $(\mathbf{X}, \mathbf{Y}) = (\mathbf{X}_i, Y_i)_{i=1,\dots,I}$, approximate F.*

Here, F can be for instance a very complex model requiring several hours to obtain one output data. There exists many different methods to construct a surrogate model. These methods depend on the available information and to cite a few among others: Polynomial Chaos Expansion (PCE) and polynomial regression [49, 56, 87, 111, 120, 135, 162], Proper Orthogonal Decomposition (POD) [11, 19, 28, 78, 97], Kriging [51, 77, 79, 80, 98, 99, 127, 128], Artificial Neural Networks [10, 14, 110, 133, 146, 149, 155, 166] (ANN), more recently Physics Informed Neural Networks (PINN) for PDEs [54, 74, 147], *etc.* In the following chapter, we will focus on one specific method to construct a surrogate model: Gaussian Process Regression (GPR) [122]. GPR has been chosen because this method is quite flexible, it has a solid theoretical background, and it has an interesting connection with Reproducing Kernel Hilbert Spaces [118] and Kriging; see for example [164]. Besides, the link between ANN and Reproducing Kernel Hilbert Spaces is outlined in [115].

3.2 The regression setting

3.2.1 Reproducing Kernel Hilbert Space

This section follows [118, Chapter 1 and 2] and is a short reminder about Reproducing Kernel Hilbert Spaces (RKHS). We denote by $\mathfrak{F}(\mathcal{X}, \mathbb{R})$ the set of functions from $\mathcal{X}$ to $\mathbb{R}$. $\mathfrak{F}(\mathcal{X}, \mathbb{R})$ is a vector space over $\mathbb{R}$ with the operations of addition and scalar multiplication.

Definition 2 (RKHS)**.** *Let $\mathcal{X}$ be a non-empty set. We will call a subset $\mathcal{H} \subseteq \mathfrak{F}(\mathcal{X}, \mathbb{R})$ a Reproducing Kernel Hilbert Space (RKHS) on $\mathcal{X}$ if*

- *$\mathcal{H}$ is a vector subspace of $\mathfrak{F}(\mathcal{X}, \mathbb{R})$;*
- *$\mathcal{H}$ is endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, with respect to which $\mathcal{H}$ is a Hilbert space;*
- *for every $\mathbf{x} \in \mathcal{X}$, the linear evaluation functional $E_{\mathbf{x}} : \mathcal{H} \to \mathbb{R}$ defined by $E_{\mathbf{x}}(f) = f(\mathbf{x})$ is bounded: $\exists C_{\mathbf{x}} > 0,\ \forall f \in \mathcal{H},\ |f(\mathbf{x})| = |E_{\mathbf{x}}(f)| \leq C_{\mathbf{x}} \|f\|_{\mathcal{H}}$, where $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$.*

If $\mathcal{H}$ is a RKHS, then the Riesz representation theorem shows that the linear evaluation functional $E_{\mathbf{x}}$ is given by the inner product with a unique vector in $\mathcal{H}$.

Definition 3 (Reproducing kernel)**.** *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if*

- *$\forall \mathbf{x} \in \mathcal{X},\ K(\mathbf{x}, \cdot) \in \mathcal{H}$;*
- *$\forall \mathbf{x} \in \mathcal{X},\ \forall f \in \mathcal{H},\ \langle f, K(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} = f(\mathbf{x})$ (reproducing property).*

More particularly, for any $\mathbf{x},\ \mathbf{y} \in \mathcal{X}$,

$$K(\mathbf{x}, \mathbf{y}) = \langle K(\mathbf{x}, \cdot), K(\mathbf{y}, \cdot) \rangle_{\mathcal{H}}\,.$$

Following the previous definition, we introduce a kernel function as follows:

Definition 4 (Kernel function [118]). *Let $\mathcal{X}$ be a non-empty set and let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function. K is called a kernel function if it is positive semi-definite that is, for any $m \geq 1$, for any $(a_1, \ldots, a_m) \in \mathbb{R}^m$, for any distinct $(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m$,*

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \,. \tag{3.1}$$

Proposition 5. *Let $\mathcal{X}$ be a non-empty set and let $\mathcal{H}$ be an RKHS on $\mathcal{X}$ with reproducing kernel K. Then K is a kernel function.*

Proof. Let $(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m$, $(a_1, \ldots, a_m) \in \mathbb{R}^m$, and $m \geq 1$. One has

$$\begin{aligned} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) &= \left\langle \sum_{i=1}^{m} a_i K(\mathbf{x}_i, \cdot), \sum_{j=1}^{m} a_j K(\mathbf{x}_j, \cdot) \right\rangle_{\mathcal{H}} \\ &= \left\| \sum_{i=1}^{m} a_i K(\mathbf{x}_i, \cdot) \right\|_{\mathcal{H}}^2 \geq 0 \,. \end{aligned}$$

□

In addition, let $\mathbf{x}, \mathbf{y} \in \mathcal{X}$; one has

$$K(\mathbf{x}, \mathbf{y}) = \langle K(\mathbf{x}, \cdot), K(\mathbf{y}, \cdot) \rangle_{\mathcal{H}} = \langle K(\mathbf{y}, \cdot), K(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} = K(\mathbf{y}, \mathbf{x}),$$

which shows that K is symmetric. From Proposition 5, a RKHS $\mathcal{H}$ defines a reproducing kernel K which is a symmetric kernel function. Although Proposition 5 is quite elementary, it has a deep converse known as the Moore-Aronszajn theorem.

Theorem 3 (Moore-Aronszajn [4, 106]). *Let $\mathcal{X}$ be a non-empty set and let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function. If K is a kernel function, then there exists a unique RKHS $\mathcal{H}$ of functions on $\mathcal{X}$ such that K is the reproducing kernel of $\mathcal{H}$.*

Therefore, Proposition 5 and Theorem 3 shows there is an one-to-one correspondence between RKHS on a set and kernel functions on this set. In the following, we will denote by $\mathcal{H}_K$ the unique RKHS associated with the reproducing kernel K, and $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ its inner product with its associated norm $\| \cdot \|_{\mathcal{H}_K}$.

Definition 5. *Given a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\mathcal{H}_K$ denotes the unique RKHS with reproducing kernel K.*

For the remainder of this chapter, we will make one more assumption about the kernel function K which will be useful:

Assumption 4 (Positive definite). *Let $\mathcal{X}$ be a non-empty set and let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function. K is assumed positive definite, or non-degenerate, that is, for any $m \geq 1$, for any $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{R}^m$, $\mathbf{a} \neq \mathbf{0}$, for any distinct $(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m$,*

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) > 0 \,.$$

3.2.2 The Optimal Recovery Solution

Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel. Using $\|\cdot\|_{\mathcal{H}_K}$ as the loss, the Optimal Recovery Solution (ORS) of Problem 1 is the minimizer [113, Theorem 12.4 and 12.5] of

$$\begin{cases} \min\limits_{F \in \mathcal{H}_K} \|F\|^2_{\mathcal{H}_K}, \\ \text{subjected to } F(\mathbf{X}_i) = Y_i, i = 1, \dots, I, \end{cases} \tag{3.2}$$

where the functional space $\mathcal{H}_K$ is the RKHS associated with the kernel K. This regression provides a minimax optimal approximation of F in $\mathcal{H}_K$ [104]. From the representer theorem [105], the solution of Equation (3.2) is

$$G(\mathbf{x}) = \sum_{i=1}^{I} \alpha_i \mathbf{K}(\mathbf{x}, \mathbf{X}_i). \tag{3.3}$$

The coefficients $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^I$ are obtained by solving the following equation

$$\mathbf{K}(\mathbf{X}, \mathbf{X})\boldsymbol{\alpha} = \mathbf{Y}, \tag{3.4}$$

where $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the block matrix defined as

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} K(\mathbf{X}_1, \mathbf{X}_1) & \cdots & K(\mathbf{X}_1, \mathbf{X}_I) \\ \vdots & \ddots & \vdots \\ K(\mathbf{X}_I, \mathbf{X}_1) & \cdots & K(\mathbf{X}_I, \mathbf{X}_I) \end{bmatrix}, \tag{3.5}$$

and $\mathbf{Y} = (Y_1, \dots, Y_I)^T$. Thus, one has:

$$F(\mathbf{x}) \approx G(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{Y} \tag{3.6}$$

where

$$\mathbf{K}(\mathbf{x}, \mathbf{X}) = \begin{pmatrix} K(\mathbf{x}, \mathbf{X}_1) & \dots & K(\mathbf{x}, \mathbf{X}_I) \end{pmatrix}. \tag{3.7}$$

Equation (3.6) implies that the ORS is interpolant. That is,

$$\forall i = 1, \dots, I,\ G(\mathbf{X}_i) = Y_i. \tag{3.8}$$

In some cases, depending on the position of the data points $\mathbf{X}$ and their number, the kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ might be ill-conditioned and thus numerically non-invertible. Therefore, Kernel Rigde Regression is often preferred because it ensures that the kernel matrix is indeed invertible by adding a smoothing term.

3.2.3 The Kernel Ridge Regression solution

Let $\lambda > 0$. The Kernel Ridge Regression (KRR) solution of Problem 1 is [115]

$$\min_{F \in \mathcal{H}_K} \sum_{i=1}^{I} (Y_i - F(\mathbf{X}_i))^2 + \lambda \|F\|^2_{\mathcal{H}_K}, \tag{3.9}$$

where the functional space $\mathcal{H}_K$ is the RKHS associated with the kernel K. The parameter λ controls the smoothness of the KRR solution. This parameter is useful to avoid overfitting and is often called nugget. From the representer theorem [105], the solution of Equation (3.9) is

$$G_\lambda(\mathbf{x}) = \sum_{i=1}^{I} \alpha_i \mathbf{K}(\mathbf{x}, \mathbf{X}_i). \tag{3.10}$$

The coefficients $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^I$ are given by

$$(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda \mathbf{I}_I)\, \boldsymbol{\alpha} = \mathbf{Y}, \tag{3.11}$$

where $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the block matrix defined by Equation (3.5) and $\mathbf{I}_I$ is the $I \times I$ identity matrix. The matrix $\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda \mathbf{I}_I$ is invertible if $\lambda > 0$. Thus, the prediction at an unobserved point $\mathbf{x}$ has the following expression:

$$F(\mathbf{x}) \approx G_\lambda(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1} \mathbf{Y}. \tag{3.12}$$

The main difference with Equation (3.2) is that the KRR solution is not interpolant because of the addition of the parameter λ, which controls the overfitting of the solution. In practical cases, this parameter is usually chosen as $\lambda \ll 1$. For $\lambda = 0$, $G_0 \equiv G$.

3.2.4 Deterministic error estimation of the KRR solution

Let $F \in \mathcal{H}_K$ be the ground truth function and let G_λ be its KRR approximation (3.12) with $\lambda \geq 0$. From [115, Theorem 8.4], one has for any $\mathbf{x} \in \mathcal{X}$,

$$|F(\mathbf{x}) - G_\lambda(\mathbf{x})| \leq \sigma(\mathbf{x}) \, \|F\|_{\mathcal{H}_K} \quad \text{with} \quad \|F\|_{\mathcal{H}_K} < +\infty, \tag{3.13}$$

and

$$|F(\mathbf{x}) - G_\lambda(\mathbf{x})| \leq \sqrt{\sigma^2(\mathbf{x}) + \lambda} \, \|F\|_{\mathcal{H}_{K+\lambda}} \quad \text{with} \quad \|F\|_{\mathcal{H}_{K+\lambda}} < +\infty, \tag{3.14}$$

where $\mathcal{H}_K$ is the RKHS associated with the kernel K, $\mathcal{H}_{K+\lambda}$ is the RKHS associated with the kernel $K + \lambda$, and

$$\sigma^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - \mathbf{K}(\mathbf{x}, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1} \mathbf{K}(\mathbf{X}, \mathbf{x}), \tag{3.15}$$

with $\mathbf{K}(\mathbf{X}, \mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X})^T$. Thus, Equation (3.13) and Equation (3.14) provide with bounds on the deterministic error $|F(\mathbf{x}) - G_\lambda(\mathbf{x})|$, which depend on the norms $\|F\|_{\mathcal{H}_K}$, $\|F\|_{\mathcal{H}_{K+\lambda}}$, and $\sigma^2(\mathbf{x})$. Therefore, reducing the discrepancy between the ground truth function F and its approximation G_λ at some point $\mathbf{x} \in \mathcal{X}$ amounts to finding the "best" kernel K in a sense that will be elaborated further on in Section 3.3 and in Section 3.4.

3.2.5 Equivalence with Kriging

Kriging was first developed by Krige [79, 80] in mining exploration in a geostatistical approach. In his first studies, he was searching for estimating the distribution of gold at the Witwatersrand reef complex in South Africa. Then Matheron [98, 99] studied Krige's work and generalized it. Since then, many people have been working on this approach to expand it to further fields, among others: engineering design optimization [12], topology optimization [165] or uncertainty quantification [47, 108, 159]. The essential idea of Kriging is that the function F is not deterministic but can be viewed as a stochastic process. That is, the stochastic process F over $\mathcal{X}$ is a collection of random variables $\{F(\mathbf{x});\ \mathbf{x} \in \mathcal{X}\}$. In this way, we decompose F as follows:

$$\forall \mathbf{x} \in \mathcal{X},\ F(\mathbf{x}) = m(\mathbf{x}) + Z(\mathbf{x}), \tag{3.16}$$

where $m(\mathbf{x})$ is the deterministic contribution or the mean function, and $Z(\mathbf{x})$ is the fluctuation contribution or residual. The random process associated with Z is zero-mean second order stationary. That is to say, the mean of the process is constant equal to zero $\mathbb{E}[Z(\mathbf{x})] = 0$ and the variance is constant $\mathbb{E}\left[Z(\mathbf{x})^2\right] = \sigma^2$, for all $\mathbf{x} \in \mathcal{X}$. Also the covariance function between two points $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^2$, $\mathrm{Cov}[Z(\mathbf{x}), Z(\mathbf{y})] = \mathbb{E}[(Z(\mathbf{x}) - \mathbb{E}[Z(\mathbf{x})])(Z(\mathbf{y}) - \mathbb{E}[Z(\mathbf{y})])] = \mathbb{E}[Z(\mathbf{x})Z(\mathbf{y})]$, depends only on $\mathbf{x} - \mathbf{y}$. In summary,

$$\forall \mathbf{x} \in \mathcal{X},\ \mathbb{E}[F(\mathbf{x})] = m(\mathbf{x}),\ \mathrm{Var}[F(\mathbf{x})] = \sigma^2,$$

where $\mathrm{Var}[F(\mathbf{x})] = \mathbb{E}\left[(F(\mathbf{x}) - m(\mathbf{x}))^2\right] = \mathrm{Cov}[Z(\mathbf{x}), Z(\mathbf{x})] = \mathbb{E}\left[Z(\mathbf{x})^2\right]$. Moreover, we define the covariance matrix of a vector $\mathbf{X} \in \mathcal{X}^I$ the $I \times I$ matrix denoted by $\mathbf{K}(\mathbf{X}, \mathbf{X})$ such that

$$\forall \mathbf{X} \in \mathcal{X}^I,\ (\mathbf{K}(\mathbf{X}, \mathbf{X}))_{i,j} = K(\mathbf{X}_i, \mathbf{X}_j) = \mathrm{Cov}[F(\mathbf{X}_i), F(\mathbf{X}_j)]. \tag{3.17}$$

The matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ needs to be positive definite and it is symmetric by definition. That is,

$$\begin{aligned} \forall m \in \mathbb{N}^*,\ \forall \mathbf{a} \in \mathbb{R}^m,\ \mathbf{a} \neq \mathbf{0},\ \mathbf{a}^T \mathbf{K}(\mathbf{X}, \mathbf{X})\mathbf{a} > 0, \text{ and} \quad & \text{(positive definite)} \\ \forall i, j,\ K(\mathbf{X}_i, \mathbf{X}_j) = K(\mathbf{X}_j, \mathbf{X}_i). \quad & \text{(symmetric)} \end{aligned}$$

We keep the same notations as in Section 3.2.1 but here $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is built from a covariance function and not from a kernel. Then one can compute the Best Linear Unbiased Predictor (BLUP).

Definition 6. *(BLUP) A predictor $G(\mathbf{x})$ of a random variable $F(\mathbf{x})$ is said to be the Best Linear Unbiased Predictor (BLUP) if the following properties are verified:*

- *The predictor $G(\mathbf{x})$ is a linear combination of the observations $\mathbf{Y}$: $G(\mathbf{x}) = \sum_{i=1}^{I} \omega_i(\mathbf{x}) Y_i$;*
- *The predictor $G(\mathbf{x})$ is unbiased: $\mathbb{E}[G(\mathbf{x}) - F(\mathbf{x})] = 0$;*

- *The predictor $G(\mathbf{x})$ minimizes the mean squared error (or variance in this case):* $\mathbb{E}\left[(G(\mathbf{x}) - F(\mathbf{x}))^2\right] = \mathrm{Var}[G(\mathbf{x}) - F(\mathbf{x})]$.

Two different types of Kriging will be studied:

- Simple Kriging: The mean function m is assumed to be constant and known for the whole domain $\mathcal{X}$. That is, for any $\mathbf{x} \in \mathcal{X}$, $m(\mathbf{x}) = \mathbb{E}[F(\mathbf{x})] = \underline{m}$ with $\underline{m}$ known;
- Ordinary Kriging: The mean function m is assumed to be constant but unknown. That is, for any $\mathbf{x} \in \mathcal{X}$, $m(\mathbf{x}) = \mathbb{E}[F(\mathbf{x})] = \underline{m}$ with $\underline{m}$ unknown. This case is the most commonly used.

A third type is also often used: co-Kriging. One also knows secondary or auxiliary information at the observation points, or at other locations, such as the partial gradients of the function of interest F. In this case, co-Kriging is also known as gradient-enhanced Kriging. Gradient-enhanced Kriging has been successfuly applied in a robust design optimization context in [132] for example. Co-Kriging will not be further developed here but additional information can be found in *e.g.* [33, 81, 86, 90, 94, 167] and references therein.

3.2.5.1 Simple Kriging

Assumption 5. *In the simple Kriging framework, the assumptions are the following:*

- *The mean $\underline{m} \in \mathbb{R}$ is constant and known;*
- *The covariance function K associated with the random process Z is known.*

One seeks the BLUP of F at a point $\mathbf{x}$. The predictor, denoted by $G(\mathbf{x})$, is supposed to be a linear combination of the observed values $\{Y_i\}_{i=1}^I$

$$G(\mathbf{x}) = \alpha(\mathbf{x}) + \sum_{i=1}^{I} \omega_i(\mathbf{x}) Y_i, \tag{3.18}$$

where $\boldsymbol{\omega}(\mathbf{x}) = \begin{pmatrix}\omega_1(\mathbf{x}) & \dots & \omega_I(\mathbf{x})\end{pmatrix}^T \in \mathbb{R}^I$ are the weights to be determined, and $\alpha(\mathbf{x}) \in \mathbb{R}$ is a function to be determined. In an equivalent way, the predictor G can be written as:

$$G(\mathbf{x}) = \alpha(\mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T \mathbf{Y}. \tag{3.19}$$

Determination of the weights $\boldsymbol{\omega}(\mathbf{x})$ First, the predictor G must be unbiased:

$$\begin{aligned}
\mathbb{E}[G(\mathbf{x}) - F(\mathbf{x})] &= \alpha(\mathbf{x}) + \sum_{i=1}^{I} \omega_i(\mathbf{x}) \underbrace{\mathbb{E}[Y_i]}_{=\underline{m}} - \underbrace{\mathbb{E}[F(\mathbf{x})]}_{=\underline{m}} \\
&= \alpha(\mathbf{x}) + \sum_{i=1}^{I} \omega_i(\mathbf{x}) \underline{m} - \underline{m} \\
&= 0.
\end{aligned} \tag{3.20}$$

Thus, Equation (3.20) yields

$$\alpha(\mathbf{x}) = \underline{m}\left(1 - \sum_{i=1}^{I} \omega_i(\mathbf{x})\right). \tag{3.21}$$

In the Kriging framework, "best" has to be understood as the predictor which minimizes the variance (the BLUP):

$$\boldsymbol{\omega}(\mathbf{x}) = \underset{\boldsymbol{\omega}\in\mathbb{R}^I}{\arg\min}\, \mathrm{Var}[F(\mathbf{x}) - G(\mathbf{x})], \tag{3.22}$$

under the assumption that the predictor G is unbiased. One has

$$\begin{aligned}
\mathbb{E}\left[(G(\mathbf{x}) - F(\mathbf{x}))^2\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{I} \omega_i(\mathbf{x}) Y_i + \underline{m}\left(1 - \sum_{i=1}^{I} \omega_i(\mathbf{x})\right) - F(\mathbf{x})\right)^2\right] \\
&= \sum_{i=1}^{I}\sum_{j=1}^{I} \omega_i(\mathbf{x})\omega_j(\mathbf{x})\mathbb{E}\left[(\underline{m} - Y_i)(\underline{m} - Y_j)\right] \\
&\quad -2\sum_{i=1}^{I} \omega_i(\mathbf{x})\mathbb{E}\left[(\underline{m} - Y_i)(\underline{m} - F(\mathbf{x})\right] + \mathbb{E}\left[(\underline{m} - F(\mathbf{x}))^2\right],
\end{aligned}$$

or

$$\mathbb{E}\left[(\underline{m} - Y_i)(\underline{m} - Y_j)\right] = \mathrm{Cov}[Y_i, Y_j] = K(\mathbf{X}_i, \mathbf{X}_j).$$

Thus

$$\mathbb{E}\left[(G(\mathbf{x}) - F(\mathbf{x}))^2\right] = \sum_{i=1}^{I}\sum_{j=1}^{I} \omega_i(\mathbf{x})\omega_j(\mathbf{x}) K(\mathbf{X}_i, \mathbf{X}_j) - 2\sum_{i=1}^{I} \omega_i(\mathbf{x}) K(\mathbf{X}_i, \mathbf{x}) + K(\mathbf{x}, \mathbf{x}).$$

Finally,

$$\begin{aligned}
\sigma_{SK}^2(\mathbf{x}) &= \mathbb{E}\left[(G(\mathbf{x}) - F(\mathbf{x}))^2\right] \\
&= K(\mathbf{x}, \mathbf{x}) - 2\boldsymbol{\omega}(\mathbf{x})^T \mathbf{K}(\mathbf{X}, \mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T \mathbf{K}(\mathbf{X}, \mathbf{X})\boldsymbol{\omega}(\mathbf{x}),
\end{aligned} \tag{3.23}$$

where $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is given by Equation (3.5) (or Equation (3.17)) and $\mathbf{K}(\mathbf{X}, \mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X})^T$ is given by Equation (3.7). Now, we want to minimize the prediction variance $\sigma_{SK}^2(\mathbf{x})$ with respect to the weights $\boldsymbol{\omega}(\mathbf{x})$. The minimum can be found by setting to zero the first derivate of $\sigma_{SK}^2(\mathbf{x})$ with respect to $\boldsymbol{\omega}(\mathbf{x})$ under the condition that the Hessian matrix is positive definite. One has first:

$$\boldsymbol{\nabla}_{\boldsymbol{\omega}} \sigma_{SK}^2 = -2\mathbf{K}(\mathbf{X}, \mathbf{x}) + 2\mathbf{K}(\mathbf{X}, \mathbf{X})\boldsymbol{\omega}(\mathbf{x}). \tag{3.24}$$

Setting Equation (3.24) to zero yields:

$$\mathbf{K}(\mathbf{X}, \mathbf{X})\boldsymbol{\omega}(\mathbf{x}) = \mathbf{K}(\mathbf{X}, \mathbf{x}). \tag{3.25}$$

It is solvable because $\mathbf{K}$ is a positive definite matrix from Assumption 4, thus it is invertible. Let us verify that it is indeed a minimum by determining the Hessian matrix:

$$\frac{1}{2}\boldsymbol{\nabla}_{\boldsymbol{\omega}} \otimes \boldsymbol{\nabla}_{\boldsymbol{\omega}} \sigma_{SK}^2 = \mathbf{K}(\mathbf{X}, \mathbf{X}),$$

which is positive definite for every $\boldsymbol{\omega}$ because $\mathbf{K}(\mathbf{X},\mathbf{X})$ is. Thereby, if $\boldsymbol{\omega}(\mathbf{x})$ is solution of Equation (3.25), then $\boldsymbol{\omega}(\mathbf{x})$ is a global minimum of $\sigma^2_{SK}(\mathbf{x})$. Therefore,

$$\boldsymbol{\omega}(\mathbf{x}) = \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}). \tag{3.26}$$

Determination of Kriging variance $\sigma^2_{SK}(\mathbf{x})$ Following Equation (3.23), the expression of the prediction variance $\sigma^2_{SK}(\mathbf{x})$ is

$$\sigma^2_{SK}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - 2\boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}), \tag{3.27}$$

with $\boldsymbol{\omega}(\mathbf{x}) = \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x})$ from Equation (3.26). Hence,

$$\begin{aligned}\sigma^2_{SK}(\mathbf{x}) &= K(\mathbf{x},\mathbf{x}) - 2\mathbf{K}(\mathbf{X},\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-T}\mathbf{K}(\mathbf{X},\mathbf{x}) \\ &\quad + \mathbf{K}(\mathbf{X},\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-T}\mathbf{K}(\mathbf{X},\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}) \\ &= K(\mathbf{x},\mathbf{x}) - \mathbf{K}(\mathbf{X},\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-T}\mathbf{K}(\mathbf{X},\mathbf{x}) \\ &= K(\mathbf{x},\mathbf{x}) - \mathbf{K}(\mathbf{x},\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}),\end{aligned} \tag{3.28}$$

because $\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}$ is symmetric.

Simple Kriging predictor and variance For every $\mathbf{x} \in \mathbb{R}^d$, the simple Kriging predictor G and the prediction variance σ^2_{SK} are equal to

$$G(\mathbf{x}) = \boldsymbol{\omega}(\mathbf{x})^T(\mathbf{Y} - \mathbf{1}\underline{m}) + \underline{m}, \tag{3.29}$$

$$\sigma^2_{SK}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - \mathbf{K}(\mathbf{x},\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}), \tag{3.30}$$

with $\boldsymbol{\omega}(\mathbf{x})$ given by Equation (3.26) and $\mathbf{1} = (1,\dots,1)^T$ with I instances. One can notice that $\sigma^2_{SK}(\mathbf{x})$ does not depend on the value of the observations $\mathbf{Y}$ but only on the covariance function K and the positions $\mathbf{X}$ of the observations. If $\underline{m} = 0$, one finds the classical expression of the predictor

$$G(\mathbf{x}) = \boldsymbol{\omega}(\mathbf{x})^T\mathbf{Y}. \tag{3.31}$$

Remark 9. *Another way to directly find the expression of α—see Equation (3.21)—is to consider for each $\mathbf{x}$, $\tilde{F}(\mathbf{x}) = F(\mathbf{x}) - \underline{m}$ instead of $F(\mathbf{x})$. In that way, the process $\tilde{F}$ is a zero-mean process. The predictor $\tilde{G}$ is searched as a linear combination of the observations: $\tilde{G}(\mathbf{x}) = \sum_{i=1}^{I}\omega_i(\mathbf{x})\tilde{F}(\mathbf{X}_i)$. It is unbiased because it has zero-mean. By taking advantage of $\tilde{G}(\mathbf{x}) = G(\mathbf{x}) - \underline{m}$ and $\tilde{F}(\mathbf{X}_i) = Y_i - \underline{m}$, for all $i = 1,\dots,I$, Equation (3.21) is obtained.*

Numerical simplification To simplify Equation (3.29), we will assume that $\underline{m} = 0$. When the data set is large and when one wishes to obtain multiple output values, the weights have to be computed for each output value because $\mathbf{K}(\mathbf{X},\mathbf{x})$ depends on $\mathbf{x}$. To overcome this numerical complication, one can express the weights into a new basis:

$$G(\mathbf{x}) = \begin{pmatrix}\omega_1(\mathbf{x}) & \dots & \omega_I(\mathbf{x})\end{pmatrix}\mathbf{Y},$$

with $\begin{pmatrix}\omega_1(\mathbf{x}) & \dots & \omega_I(\mathbf{x})\end{pmatrix}^T = \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x})$. Hence,

$$\begin{aligned} G(\mathbf{x}) &= \left(\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x})\right)^T \mathbf{Y} \\ &= \mathbf{K}(\mathbf{x},\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{Y}. \end{aligned}$$

Introducing $\tilde{\boldsymbol{\omega}} = \begin{pmatrix}\tilde{\omega_1} & \dots & \tilde{\omega_I}\end{pmatrix}^T$ such that:

$$\mathbf{K}(\mathbf{X},\mathbf{X})\tilde{\boldsymbol{\omega}} = \mathbf{Y} \tag{3.32}$$

which does not depend on $\mathbf{x}$, the predictor G can be expressed for every $\mathbf{x} \in \mathcal{X}$:

$$G(\mathbf{x}) = \mathbf{K}(\mathbf{x},\mathbf{X})\tilde{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}}^T\mathbf{K}(\mathbf{X},\mathbf{x}), \tag{3.33}$$

where only one inversion of a linear system is needed to obtain $\tilde{\boldsymbol{\omega}}$ from Equation (3.32).

3.2.5.2 Ordinary Kriging

In most practical cases, the mean $\underline{m}$ is not known. Thus, simple Kriging is inapplicable. We introduce an alternative prediction method, called ordinary Kriging, which unlike simple Kriging does not assume the knowledge of $\underline{m}$.

Assumption 6. *In the ordinary Kriging framework, the assumptions are the following:*

- *The mean $\underline{m} \in \mathbb{R}$ is constant but unknown;*
- *The covariance function K associated with the random process Z is known.*

As earlier, One seeks the BLUP of F at a point $\mathbf{x}$. The predictor, denoted by $G(\mathbf{x})$, is supposed to be a linear combination of the observed values $\{Y_i\}_{i=1}^I$:

$$G(\mathbf{x}) = \sum_{i=1}^{I} \omega_i(\mathbf{x})Y_i, \tag{3.34}$$

or in an equivalent way,

$$G(\mathbf{x}) = \boldsymbol{\omega}(\mathbf{x})^T\mathbf{Y}. \tag{3.35}$$

Determination of the weights $\boldsymbol{\omega}(\mathbf{x})$ The prediction G must be unbiased,

$$\mathbb{E}\left[G(\mathbf{x}) - F(\mathbf{x})\right] = \mathbb{E}\left[G(\mathbf{x})\right] - \mathbb{E}\left[F(\mathbf{x})\right] = \sum_{i=1}^{I} \omega_i(\mathbf{x})\underline{m} - \underline{m} = 0. \tag{3.36}$$

Equation (3.36) must be true for every $\underline{m}$. Thus, it implies that

$$\sum_{i=1}^{I} \omega_i(\mathbf{x}) = 1. \tag{3.37}$$

One also seeks to minimize the following variance

$$\boldsymbol{\omega}(\mathbf{x}) = \underset{\boldsymbol{\omega}\in\mathbb{R}^I}{\arg\min}\, \mathrm{Var}[F(\mathbf{x}) - G(\mathbf{x})] \text{ subjected to } \sum_{i=1}^{I} \omega_i(\mathbf{x}) = 1. \tag{3.38}$$

Contrary to simple Kriging, the optimization problem (3.38) is henceforth constrained to satisfy the unbiased-ness requirement. One can solve it by the method of Lagrange multipliers. Denoting by $\Lambda(\mathbf{x}) \in \mathbb{R}$ this Lagrange multiplier, one obtains the equivalent unconstrained problem

$$(\boldsymbol{\omega}(\mathbf{x}), \Lambda(\mathbf{x})) = \underset{\boldsymbol{\omega}\in\mathbb{R}^I,\, \Lambda\in\mathbb{R}}{\arg\min}\, \mathrm{Var}[F(\mathbf{x}) - G(\mathbf{x})] + 2\Lambda(\mathbf{x}) \left(\sum_{i=1}^{I} \omega_i(\mathbf{x}) - 1 \right). \tag{3.39}$$

Using Equation (3.23), the optimization problem amounts to solve for

$$\begin{aligned} &(\boldsymbol{\omega}(\mathbf{x}), \Lambda(\mathbf{x})) = \\ &\underset{\boldsymbol{\omega}\in\mathbb{R}^I,\, \Lambda\in\mathbb{R}}{\arg\min}\; K(\mathbf{x},\mathbf{x}) - 2\boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}) + 2\Lambda(\mathbf{x})(\boldsymbol{\omega}(\mathbf{x})^T\mathbf{1} - 1). \end{aligned} \tag{3.40}$$

Setting to zero the first derivates with respect to $\boldsymbol{\omega}(\mathbf{x})$ and $\Lambda(\mathbf{x})$ of the hereinabove expression gives

$$\begin{aligned} -2\mathbf{K}(\mathbf{X},\mathbf{x}) + 2\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}) + 2\Lambda(\mathbf{x})\mathbf{1} &= 0 \\ \boldsymbol{\omega}(\mathbf{x})^T\mathbf{1} - 1 &= 0, \end{aligned} \tag{3.41}$$

hence

$$\begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{\omega}(\mathbf{x}) \\ \Lambda(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{K}(\mathbf{X},\mathbf{x}) \\ 1 \end{pmatrix}.$$

Finally, the weights $\boldsymbol{\omega}(\mathbf{x})$ and the Lagrange multiplier $\Lambda(\mathbf{x})$ can be found by solving the following system:

$$\begin{pmatrix} \boldsymbol{\omega}(\mathbf{x}) \\ \Lambda(\mathbf{x}) \end{pmatrix} = \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{K}(\mathbf{X},\mathbf{x}) \\ 1 \end{pmatrix}. \tag{3.42}$$

By block matrix inversion, one finds

$$\begin{aligned} \boldsymbol{\omega}(\mathbf{x}) &= \mathbf{K}(\mathbf{X},\mathbf{X})^{-1} \left[\mathbf{K}(\mathbf{X},\mathbf{x}) + \left(\frac{1 - \mathbf{1}^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x})}{\mathbf{1}^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{1}} \right) \mathbf{1} \right], \\ \Lambda(\mathbf{x}) &= \frac{\mathbf{1}^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}) - 1}{\mathbf{1}^T\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{1}}. \end{aligned} \tag{3.43}$$

Determination of Kriging variance $\sigma^2_{OK}(\mathbf{x})$ Now, as for simple Kriging, one seeks to determine the prediction variance $\sigma^2_{OK}(\mathbf{x}) = \mathbb{E}\left[(G(\mathbf{x}) - F(\mathbf{x}))^2\right]$ for ordinary Kriging. By following the same path as for simple Kriging, one has

$$\begin{aligned} \sigma^2_{OK}(\mathbf{x}) &= K(\mathbf{x},\mathbf{x}) - 2\boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}) \\ &= K(\mathbf{x},\mathbf{x}) + \boldsymbol{\omega}(\mathbf{x})^T \left(\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}) - 2\mathbf{K}(\mathbf{X},\mathbf{x}) \right), \end{aligned}$$

where $\mathbf{K}(\mathbf{X},\mathbf{X})\boldsymbol{\omega}(\mathbf{x}) = \mathbf{K}(\mathbf{X},\mathbf{x}) - \Lambda(\mathbf{x})\mathbf{1}$ by using the first line of Equation (3.41). Hence,

$$\sigma^2_{OK}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) - \Lambda(\mathbf{x})\boldsymbol{\omega}(\mathbf{x})^T\mathbf{1},$$

where $\boldsymbol{\omega}(\mathbf{x})^T\mathbf{1} = 1$ because the predictor is unbiased. Finally,

$$\sigma^2_{OK}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) - \Lambda(\mathbf{x}),$$

where $\boldsymbol{\omega}(\mathbf{x})$ and $\Lambda(\mathbf{x})$ are given by Equation (3.43).

Ordinary Kriging predictor and variance For every $\mathbf{x} \in \mathbb{R}^d$, the ordinary Kriging predictor G and the prediction variance σ^2_{OK} are equal to

$$G(\mathbf{x}) = \boldsymbol{\omega}(\mathbf{x})^T\mathbf{Y} \tag{3.44}$$
$$\sigma^2_{OK}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - \boldsymbol{\omega}(\mathbf{x})^T\mathbf{K}(\mathbf{X},\mathbf{x}) - \Lambda(\mathbf{x}), \tag{3.45}$$

where $\boldsymbol{\omega}(\mathbf{x})$ and $\Lambda(\mathbf{x})$ are given by Equation (3.43). As for simple Kriging, σ^2_{OK} does not depend on the value of the observations $\mathbf{Y}$ but only on the covariance function K and the positions $\mathbf{X}$ of the observations.

Numerical simplification When the data set is large and when one wishes to obtain multiple output values, the weights have to be computed for each output value because $\mathbf{K}(\mathbf{X},\mathbf{x})$ depends on $\mathbf{x}$. To overcome this numerical complication, one can express the weights into a new basis. From the previous section, we have

$$G(\mathbf{x}) = \begin{pmatrix}\omega_1(\mathbf{x}) & \dots & \omega_I(\mathbf{x}) & \Lambda(\mathbf{x})\end{pmatrix}\begin{pmatrix}\mathbf{Y}\\ 0\end{pmatrix}.$$

From Equation (3.42), we have

$$\begin{aligned}\begin{pmatrix}\boldsymbol{\omega}(\mathbf{x})\\ \Lambda(\mathbf{x})\end{pmatrix} &= \begin{bmatrix}\mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}^{-1}\begin{pmatrix}\mathbf{K}(\mathbf{X},\mathbf{x})\\ 1\end{pmatrix}\\ &= \tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})^{-1}\tilde{\mathbf{K}}(\mathbf{X},\mathbf{x}),\end{aligned} \tag{3.46}$$

where

$$\tilde{\mathbf{K}}(\mathbf{X},\mathbf{X}) = \begin{bmatrix}\mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}$$

and

$$\tilde{\mathbf{K}}(\mathbf{X},\mathbf{x}) = \begin{pmatrix}\mathbf{K}(\mathbf{X},\mathbf{x})\\ 1\end{pmatrix}.$$

Hence

$$\begin{aligned}G(\mathbf{x}) &= (\tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})^{-1}\tilde{\mathbf{K}}(\mathbf{X},\mathbf{x}))^T\begin{pmatrix}\mathbf{Y}\\ 0\end{pmatrix}\\ &= \tilde{\mathbf{K}}(\mathbf{x},\mathbf{X})\tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})^{-1}\begin{pmatrix}\mathbf{Y}\\ 0\end{pmatrix}.\end{aligned}$$

Introducing $\tilde{\boldsymbol{\omega}} = \begin{pmatrix} \tilde{\omega}_1 & \dots & \tilde{\omega}_I & \tilde{\omega}_{I+1} \end{pmatrix}^T$ such that:

$$\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})\tilde{\boldsymbol{\omega}} = \begin{pmatrix} \mathbf{Y} \\ 0 \end{pmatrix} \tag{3.47}$$

which does not depend on $\mathbf{x}$, the predictor G can be expressed for every $\mathbf{x} \in \mathcal{X}$

$$G(\mathbf{x}) = \tilde{\mathbf{K}}(\mathbf{x}, \mathbf{X})\tilde{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}}^T \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{x}), \tag{3.48}$$

where only one inversion of a linear system is needed to obtain $\tilde{\boldsymbol{\omega}}$ from Equation (3.47).

3.2.5.3 Kriging with noisy data

So far, we have considered noise-free data. In reality, one almost never observes the function F itself but only a noisy version of it. Now we consider that

$$\forall \mathbf{x} \in \mathcal{X},\ F(\mathbf{x}) = m(\mathbf{x}) + Z(\mathbf{x}) + \varepsilon(\mathbf{x}), \tag{3.49}$$

where ε is the noise which is such that $\forall \mathbf{x},\ \varepsilon(\mathbf{x})$ is a Gaussian random variable with zero-mean and variance σ_ε^2, $\varepsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, independent from $Z(\mathbf{x})$. Here $\mathcal{N}(m, \sigma^2)$ stands for the univariate Gaussian distribution with mean m and variance σ^2. By assuming so, one only needs to add σ_ε^2 to the diagonal of the covariance matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ in Equation (3.32) or Equation (3.47).

3.2.6 Equivalence with Gaussian process regression

The result obtained by Kriging can be derived in an alternative way by Gaussian Process Regression (GPR). In that respect, we consider that the function F is not deterministic but can be viewed as a stochastic process by introducing a prior on it. In this way, we define a probability distribution on the possible functions F. This prior assumption is that F is a Gaussian process.

Definition 7. *A Gaussian process is a stochastic process such that any finite subcollection of random variables has a multivariate Gaussian distribution.*

A Gaussian process is entirely specified by two functions:

- The mean function $m : \mathcal{X} \mapsto \mathbb{R}$ with $m(\mathbf{x}) = \mathbb{E}[F(\mathbf{x})]$ for any $\mathbf{x} \in \mathcal{X}$;
- The covariance function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ with $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}[(F(\mathbf{x}) - m(\mathbf{x}))(F(\mathbf{y}) - m(\mathbf{y}))]$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

That is, one can write the Gaussian prior process of the process F as follows for some $\mathbf{x} \in \mathcal{X}$:

$$F(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \cdot)), \tag{3.50}$$

or equivalently,

$$F(\mathbf{x}) = m(\mathbf{x}) + Z(\mathbf{x}), \tag{3.51}$$

where $Z(\mathbf{x}) \sim \mathcal{N}(0, K(\mathbf{x}, \cdot))$. In that respect, the joint distribution of the observations and the unobserved point $\mathbf{x}$ is

$$\begin{pmatrix} \mathbf{Y} \\ g(\mathbf{x}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m}(\mathbf{X}) \\ m(\mathbf{x}) \end{pmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{x}) \\ \mathbf{K}(\mathbf{x}, \mathbf{X}) & K(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right), \tag{3.52}$$

where $\mathbf{m}(\mathbf{X}) = (m(\mathbf{X}_1), \ldots, m(\mathbf{X}_I))^T$, and $\mathcal{N}(\mathbf{a}, \mathbf{A})$ stands for the multivariate Gaussian distribution with mean vector $\mathbf{a}$ and covariance matrix $\mathbf{A}$. Then, computing the posterior distribution given the Gaussian prior and conditioned on the observations yields

$$g(\mathbf{x})|\mathbf{x}, \mathbf{X}, \mathbf{Y} \sim \mathcal{N}(G(\mathbf{x}), \mathrm{Var}[g(\mathbf{x})]), \tag{3.53}$$

with

$$G(\mathbf{x}) = \mathbb{E}[F(\mathbf{x})|\mathbf{x}, \mathbf{X}, \mathbf{Y}] = m(\mathbf{x}) + \mathbf{K}(\mathbf{x}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - \mathbf{m}(\mathbf{X}))$$

and

$$\mathrm{Var}[g(\mathbf{x})] = K(\mathbf{x}, \mathbf{x}) - \mathbf{K}(\mathbf{x}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{x}).$$

One can notice that if the mean function m is known and equal to $\underline{m}$ for every $\mathbf{x}$, then

$$G(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - \mathbf{1}\underline{m}) + \underline{m}, \tag{3.54}$$

which is the expression obtained by simple Kriging, see Equation (3.29). Now, assume that $\underline{m} = 0$, Equation (3.54) is the same as Equation (3.29) and the same as Equation (3.6). What is interesting to notice is that the covariance function introduced in Section 3.2.6 and in Section 3.2.5 is actually the reproducing kernel defined in Section 3.2.1. While the Gaussian process is not in the RKHS $\mathcal{H}_K$ associated with the kernel K, the posterior mean G conditioned on the observations does lie in that RKHS $\mathcal{H}_K$. Moreover, the parameter $\sigma^2(\mathbf{x})$ given by Equation (3.15) has an interesting equivalence with the GPR/Kriging context: it is the variance of the prediction conditioned on the observations, that is $\mathrm{Var}[g(\mathbf{x})]$ given in Equation (3.53). Additional information can be found in [122, Chapter 2].

3.2.7 Examples of kernel

We denote by $\|\mathbf{x}\|_p = \left(\sum_{j=1}^d |x_j|^p\right)^{\frac{1}{p}}$, $p > 0$ with $\|\mathbf{x}\|_0 = \#\{j \ ; x_j \neq 0\}$ the p-norm of the vector $\mathbf{x}$. For practical cases, many different kernels K are available and for citing the most encountered ones [122, 137]:

- Polynomial kernel defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \left(b + \mathbf{x}_1^T \mathbf{x}_2\right)^p, \tag{3.55}$$

  where $b \geq 0$ and $p > 0$ are parameters;

- Gaussian kernel (also known as squared exponential) defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{\gamma^2}\right), \tag{3.56}$$

  where $\gamma > 0$ is called the length scale. Alternatively, different length scales can be chosen for the input dimensions:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\sum_{i=1}^{d} \frac{|x_{1,i} - x_{2,i}|^2}{\gamma_i^2}\right); \tag{3.57}$$

- Matérn-like kernels defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\,\|\mathbf{x}_1 - \mathbf{x}_2\|_2}{\gamma}\right)^{\nu} B_\nu \left(\frac{\sqrt{2\nu}\,\|\mathbf{x}_1 - \mathbf{x}_2\|_2}{\gamma}\right), \tag{3.58}$$

  where $\gamma > 0$ is called the length scale, ν is a positive parameter, and B_ν is a modified Bessel function [1]. The most used ones are for $\nu = 3/2$ and $\nu = 5/2$. Note that for $\nu \to +\infty$, the Gaussian kernel (3.56) is recovered. Stein in [137] named this type of kernels after the work of Matérn [95];

- Rational Quadratic (RQ) kernel defined as

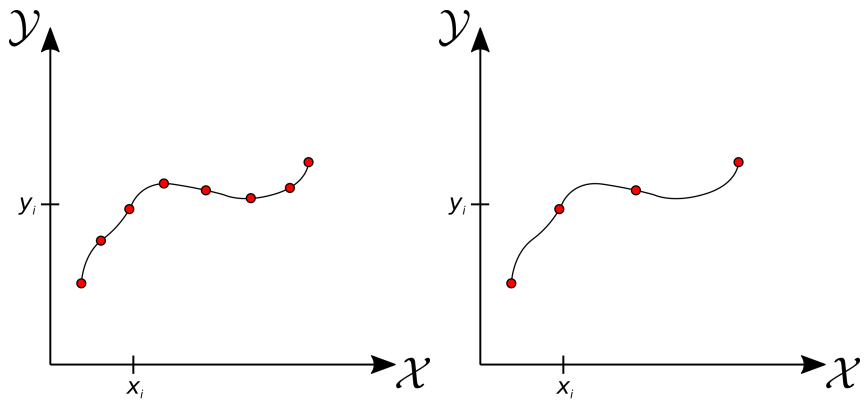
$$K(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + \frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2\alpha\gamma^2}\right)^{-\alpha}, \tag{3.59}$$

  with $a > 0$ and $\gamma > 0$. The case $\alpha \to +\infty$ corresponds to the Gaussian kernel (3.56).

Stein in [137] argues that the Gaussian kernel is too smooth for modelling many physical systems and recommands to use Matérn-like kernels.

3.3 The Kernel Flow algorithm

A certain number of parameters have to be determined for each type of kernel, for instance the kernels presented in Section 3.2.7. They are often called the hyperparameters of the kernel. One of the most challenging aspect of Kriging is to determine these free parameters, or even more generally which kernel K to select, in order to approximate the function $F : \mathcal{X} \mapsto \mathcal{Y}$ given I observations. In that regard, multiple methods exist including maximum likelihood [161] (choosing the parameter which maximizes the probability of observing the data), Bayesian approach [131] (placing a prior on the kernel and conditioning with respect to the data), *etc.* Here, we follow another approach to find the "best" kernel K: the Kernel Flow (KF) algorithm [114]. It was first used in a machine learning context (see [114]) for classification and more recently in geophysical forecasting [63] and in dynamical systems [62].



(a) (b)

Figure 3.1: (a) G_{f}: Interpolating F with kernel K and I_{f} observations. (b) G_{c}: Interpolating F with kernel K and $I_{\mathrm{c}} = \mathrm{round}(\frac{I_{\mathrm{f}}}{2})$ observations.

The main aim of the Kernel Flow algorithm is, for instance, to learn kernels of the following form, by denoting by n the iteration number of this algorithm

$$K_n(\mathbf{X}_1, \mathbf{X}_2) = K(f_n(\mathbf{X}_1), f_n(\mathbf{X}_2)), \tag{3.60}$$

where K is a kernel, for instance the Gaussian kernel defined by Equation (3.56), and $f_n : \mathcal{X} \mapsto \mathcal{X}$ is called the flow in the input space at iteration n. This can be understood as a non-parametric approach to find the "best" kernel K. Instead of searching for a single parameter, for instance γ, we search for a whole function f_n. The Kernel Flow algorithm can also be used in a parametric approach where one seeks to learn hyperparameters $\boldsymbol{\theta}$ of the kernel K, for example the parameter γ of a Gaussian kernel. That is,

$$K_n(\mathbf{X}_1, \mathbf{X}_2) = K(\mathbf{X}_1, \mathbf{X}_2, \boldsymbol{\theta}_n), \tag{3.61}$$

where $\boldsymbol{\theta}_n$ is the value of the parameters $\boldsymbol{\theta}$ at iteration n of the parametric KF algorithm.

3.3.1 What is the "best" kernel?

First, we have to define when a kernel is considered as the "best". Here, the kernel K is defined as "best" if the number of regression points can be halved without losing too much accuracy, where the latter is measured with the RKHS norm defined by the kernel.

We introduce the following function ρ_n to quantify the accuracy of the surrogate model at iteration n as

$$\rho_n(\mathbf{X}_{\pi_{\mathrm{f}}^n}, \mathbf{X}_{\pi_{\mathrm{c}}^n}) = \frac{\|G_{\mathrm{f}} - G_{\mathrm{c}}\|^2_{\mathcal{H}_n}}{\|G_{\mathrm{f}}\|^2_{\mathcal{H}_n}} \in [0, 1], \tag{3.62}$$

where the functional space $\mathcal{H}_n$ is the unique RKHS associated with the kernel K_n, G_f is the approximation of F with kernel K_n and the $I_\mathrm{f} \leq I$ observations, and G_c is the approximation of F with kernel K_n and the $I_\mathrm{c} = \mathrm{round}(\frac{I_\mathrm{f}}{2})$ observations. At last, π_f^n and π_c^n are the indices corresponding to the I_f and I_c observations respectively at iteration n, and $\mathbf{X}_{\pi_\mathrm{f}^n}$ and $\mathbf{X}_{\pi_\mathrm{c}^n}$ are the corresponding inputs. In that way, the smaller ρ_n is, the better the kernel K_n. That is, if ρ_n is close to zero, the kernel K_n is the "best" according to the definition stated above. In [114], it has been shown that ρ_n in Equation (3.62) reads

$$\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, f_n) = 1 - \frac{\mathbf{Y}_{\pi_\mathrm{c}^n}^T \mathbf{K}(f_n(\mathbf{X}_{\pi_\mathrm{c}^n}), f_n(\mathbf{X}_{\pi_\mathrm{c}^n}))^{-1} \mathbf{Y}_{\pi_\mathrm{c}^n}}{\mathbf{Y}_{\pi_\mathrm{f}^n}^T \mathbf{K}(f_n(\mathbf{X}_{\pi_\mathrm{f}^n}), f_n(\mathbf{X}_{\pi_\mathrm{f}^n}))^{-1} \mathbf{Y}_{\pi_\mathrm{f}^n}} \tag{3.63}$$

for the non-parametric case, and

$$\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, \boldsymbol{\theta}_n) = 1 - \frac{\mathbf{Y}_{\pi_\mathrm{c}^n}^T \mathbf{K}(\mathbf{X}_{\pi_\mathrm{c}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, \boldsymbol{\theta}_n)^{-1} \mathbf{Y}_{\pi_\mathrm{c}^n}}{\mathbf{Y}_{\pi_\mathrm{f}^n}^T \mathbf{K}(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{f}^n}, \boldsymbol{\theta}_n)^{-1} \mathbf{Y}_{\pi_\mathrm{f}^n}} \tag{3.64}$$

for the parametric case, where $\mathbf{Y}_{\pi_\mathrm{f}^n}$ and $\mathbf{Y}_{\pi_\mathrm{c}^n}$ are the observations corresponding to the indices π_f^n and π_c^n, respectively. Also $\mathbf{K}(f_n(\mathbf{X}_{\pi_\mathrm{f}^n}), f_n(\mathbf{X}_{\pi_\mathrm{f}^n}))$ and $\mathbf{K}(f_n(\mathbf{X}_{\pi_\mathrm{c}^n}), f_n(\mathbf{X}_{\pi_\mathrm{c}^n}))$ for the non-parametric case, and $\mathbf{K}(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{f}^n}, \boldsymbol{\theta}_n)$ and $\mathbf{K}(\mathbf{X}_{\pi_\mathrm{c}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, \boldsymbol{\theta}_n)$ for the parametric case, are the matrices (3.5) constructed with the kernel K_n and the inputs $\mathbf{X}_{\pi_\mathrm{f}^n}$ and $\mathbf{X}_{\pi_\mathrm{c}^n}$, respectively. These formulas enable the numerical computation of the accuracy ρ_n in both cases, and stem from the identity $\|G\|_{\mathcal{H}_K}^2 = \mathbf{Y}^T \mathbf{K}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{Y}$ whenever $G(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X}) \mathbf{K}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{Y}$.

3.3.2 The non-parametric Kernel Flow algorithm

The learning process for the non-parametric algorithm can be seen on Figure 3.2 from iteration n to iteration $n+1$. f_n is known at iteration n. To describe it simply:

1. Select $I_\mathrm{f} \leq I$ observations at random among the I observations;
2. Among these I_f observations, we select now $I_\mathrm{c} = \mathrm{round}(\frac{I_\mathrm{f}}{2})$ observations;
3. The I_f and I_c observations allow us to compute the accuracy $\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, f_n)$ given by Equation (3.62). The I_c observations are only used to compute $\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, f_n)$. Then we form the gradient of ρ_n with respect to the positions of the I_f observations: $\boldsymbol{\nabla}_{\mathbf{X}_{\pi_\mathrm{f}^n}} \rho_n$;
4. Move the positions of the I_f observations in the gradient descent direction of ρ_n computed in the previous step;
5. Move the remaining positions of the $I - I_\mathrm{f}$ observations by interpolation with kernel K_n and the moved I_f observations. In this way, one gets f_{n+1} and thus K_{n+1}. For any observation $\mathbf{X}$ outside of the I_f observations, move it by interpolating the I observations using the kernel K_{n+1}.
6. Return to step 1.

In fact, the accuracy given by Equation (3.62) is doubly randomized: by choosing at random the I_f points among the I points and then by choosing at random the I_c points among the I_f points. To sum up, the non-parametric KF algorithm will deform the input space $\mathcal{X}$ through the flow function f_n in order to minimize the accuracy ρ_n.

3.3.3 The parametric Kernel Flow algorithm

The parametric KF algorithm is similar to its non-parametric counterpart. Indeed, we are not seeking to determine an entire function f_n but one or more parameter of the kernel. Consider a family of kernels $K_{\boldsymbol{\theta}}(\mathbf{X}_1, \mathbf{X}_2) = K(\mathbf{X}_1, \mathbf{X}_2, \boldsymbol{\theta})$ parameterized by the parameters $\boldsymbol{\theta}$. The parametric version of the KF algorithm can be described as follows from iteration n to $n+1$:

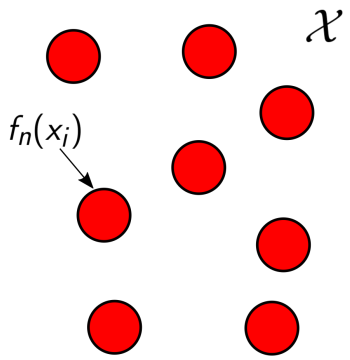
1. Select $I_\mathrm{f} \leq I$ observations at random among the I observations;
2. Among these I_f observations, we select now $I_\mathrm{c} = \mathrm{round}(\frac{I_\mathrm{f}}{2})$ observations;
3. The I_f and I_c observations allow us to compute the accuracy $\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, \boldsymbol{\theta}_n)$ given by Equation (3.62). The I_c observations are only used to compute $\rho_n(\mathbf{X}_{\pi_\mathrm{f}^n}, \mathbf{X}_{\pi_\mathrm{c}^n}, \boldsymbol{\theta}_n)$. Then we form the gradient of ρ_n with respect to the parameters $\boldsymbol{\theta}$: $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \rho_n$;
4. Move $\boldsymbol{\theta}$ in the gradient descent direction $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \rho_n$ of ρ_n computed in the previous step;
5. Return to step 1.

3.3.4 A numerical example

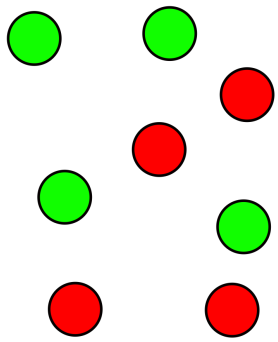
3.3.4.1 Case setting

The objective of this section is to build a surrogate model of the lift-to-drag ratio of a RAE2822 wing profile—this is the function F of Problem 1—when some characteristics of the flow and/or the profile are variable and only a finite number of observations of F is available. We choose as the profile's inputs the Mach number M and the angle of attack α with their respective ranges—see Table 3.1. These two parameters define the input vector $\mathbf{X} = (M, \alpha) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$. For this case, as no analytical solution is available for the performance function F, a numerical solver has to be used. In that respect, the Computational Fluid Dynamics (CFD) solver MSES [41] is used to simulate two-dimensional transonic flows around that airfoil. MSES is an updated version of the CFD code ISES presented in Section 1.3.4. It is a 2D coupled viscous/inviscid fluids CFD code developed by Mark Drela at MIT. The inviscid flow is solved using the steady Euler equations and a finite volume method. The viscous flows in the boundary layer and wake are solved using 2D integral boundary layer equations in a very similar way to ISES. A global Newton-Raphson method is used to solve simultaneously the coupled viscous-inviscid equations.

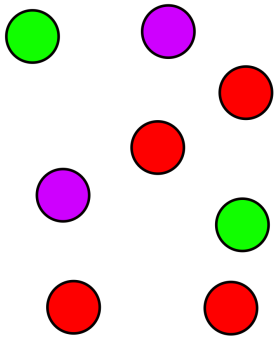
I observation points.



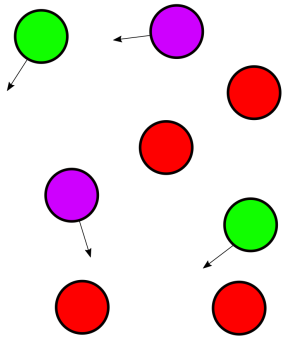
Step 1: Select I_f observation points at random out of I with $I_\mathrm{f} \leq I$.



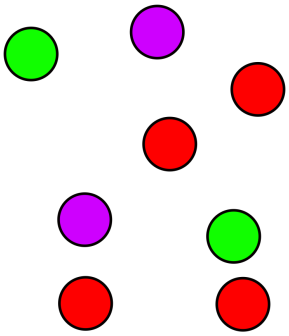
Step 2: Select $I_\mathrm{c} = \mathrm{round}(\frac{I_\mathrm{f}}{2})$ observation points at random out of I.



Step 3: Compute $\rho_n = \frac{||G_\mathrm{f} - G_\mathrm{c}||^2_{\mathcal{H}_n}}{||G_\mathrm{f}||^2_{\mathcal{H}_n}}$ and its gradient w.r.t. the positions of the I_f points.



Step 4: Move the I_f points in the gradient descent direction of ρ_n.



Step 5: Move the remaining $I - I_\mathrm{f}$ points using interpolation with kernel K_n. This is f_{n+1}.

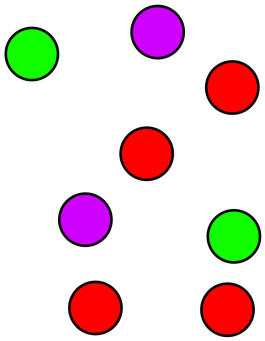


Figure 3.2: The non-parametric Kernel Flow algorithm.

Input	Range
Mach M	$[0.69255, 0.76545]$
Angle of attack α (in °)	$[2.2638, 2.3562]$

Table 3.1: Range of each input.

Nevertheless, one numerical evaluation of F by using MSES can be costly. Therefore, a surrogate model G_λ of F has been built, using $I = 81$ observations denoted by $\{F(\mathbf{X}_i) = Y_i\}_{i=1}^I$ and computed through MSES with $\{\mathbf{X}_i\}_{i=1}^I$ evenly chosen in $\mathcal{X}$, and the KRR method outlined in Section 3.2.3 with the nugget $\lambda > 0$. A Gaussian kernel (3.57) is used

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\sum_{i=1}^{2} \frac{|x_i - y_i|^2}{\gamma_i^2}\right),$$

where $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2) \in \mathcal{X}^2$, and γ_i the length scale with respect to the dimension i of the inputs. Due to the limited number I of available data, we use a three-fold cross-validation procedure in order to tune the hyperparameters of the parametric and non-parametric KF algorithms. In other words, the data set is successively split into three different sets of same size, where two of them constitute the learning set while the third one is used as the validation set. That is, for every fold:

- The learning set gathers $I_\mathrm{L} = 54$ observations out of the $I = 81$ observations of F;
- The validation set gathers $I_\mathrm{V} = 27$ observations out of the of the $I = 81$ observations of F.

This splitting is done three times so that each split set is used once as validation set. As the KF algorithm can be seen as an equivalent of cross-validation, what we are doing is actually a double cross-validation or nested cross-validation [8, 91]. In the case of the parametric KF algorithm, we seek to minimize the average of the accuracies ρ on the folds with respect to the length scales (γ_1, γ_2) of the Gaussian kernel and to the nugget λ: $\boldsymbol{\theta} = (\lambda, \gamma_1, \gamma_2)$. In the case of the non-parametric KF algorithm, we seek now to minimize the average of the accuracies ρ on the folds with respect to f_n, which is the flow such that $K_n(\mathbf{x}, \mathbf{y}) = K(f_n(\mathbf{x}), f_n(\mathbf{y}))$. For the parametric case and the non-parametric case, we initialize the length scales (γ_1, γ_2) as

$$\gamma_1 = \gamma_2 = \frac{2}{I(I-1)} \sum_{i=1}^{I} \sum_{j=i+1}^{I} \|X_i - X_j\|_2 , \tag{3.65}$$

and the nugget as $\lambda = 1 \times 10^{-4}$. We also choose $I_\mathrm{f} = I_\mathrm{L}$ and $I_\mathrm{c} = I_\mathrm{f}/2$ in both cases. For comparison purpose, we carry out a classical Marginal Log-Likelihood (MLL) maximization on $\boldsymbol{\theta}$ using the same cross-validation setup to construct a third

surrogate model. The KRR surrogate models are numerically built by using the framework `GPyTorch` [52]. The plateform `PyTorch` [117] is used to compute the gradients by automatic differentation [116]. The optimization of the parameters in the parametric case is done by using the optimizer `Adam` [76] implemented in `PyTorch`. To assess the performance of the surrogate models obtained by the KF algorithms and the MLL maximization, we draw $I_{\mathrm{T}} = 25$ observations of F at random. These $I_{\mathrm{T}} = 25$ observations define the test set. We use the empirical Mean Squared Error (MSE) e_{MSE} defined by

$$e_{\mathrm{MSE}} = \frac{1}{I_{\mathrm{T}}} \sum_{i=1}^{I_{\mathrm{T}}} \left(Y_i - G_\lambda(\mathbf{X}_i)\right)^2, \tag{3.66}$$

the Maximum Relative Error (MRE) e_{MRE} (in %) defined by

$$e_{\mathrm{MRE}} = \max_{i=1,\dots,I_{\mathrm{T}}} \left(\frac{|Y_i - G_\lambda(\mathbf{X}_i)|}{|Y_i|} \right) \times 100, \tag{3.67}$$

and the Normalized Mean Squared Error (NMSE) e_{NMSE} defined by

$$e_{\mathrm{NMSE}} = \frac{\sum_{i=1}^{I_{\mathrm{T}}} \left(Y_i - G_\lambda(\mathbf{X}_i)\right)^2}{\sum_{i=1}^{I_{\mathrm{T}}} \left(Y_i - \frac{1}{I_{\mathrm{T}}} \sum_{i=1}^{I_{\mathrm{T}}} Y_i \right)^2}. \tag{3.68}$$

Here $G_\lambda(\mathbf{X})$ is the KRR surrogate model optimized by either the parametric KF algorithm or the non-parametric KF algorithm, or the MLL maximization. The only difference between the error e_{MSE} and the error e_{NMSE} is that e_{NMSE} is divided by the variance of the test set to avoid any scaling effect of the performance function F.

3.3.4.2 Parametric KF algorithm

First, the evolution of the accuracy ρ for the parametric KF algorithm of each fold together with its average with respect to the number of iterations n every 100 iterations is given on Figure 3.3. One can see that the values of ρ are quickly decreasing as the algorithm goes on, reaching a stabilized value at iteration $n \approx 1 \times 10^4$.

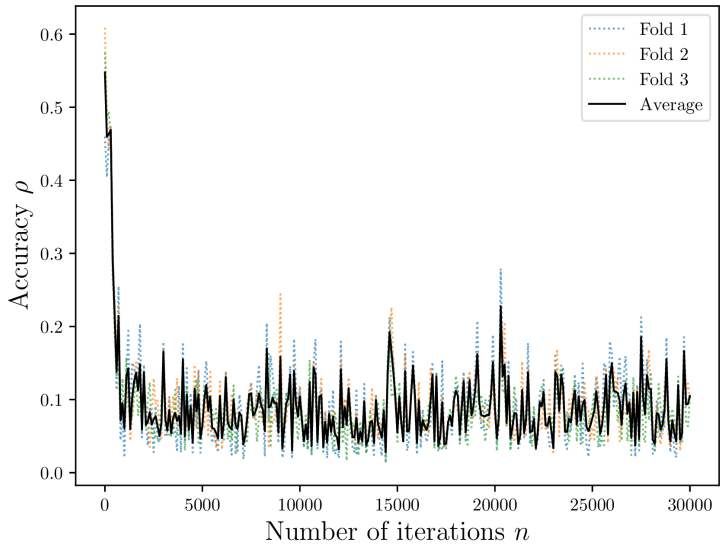
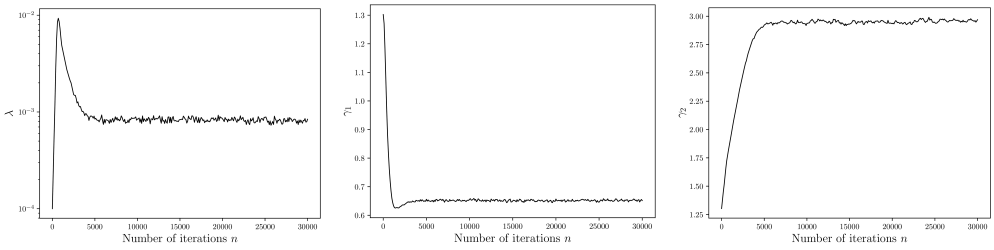


Figure 3.3: Evolution of ρ on each fold and its average with respect to the number n of iterations for the parametric KF algorithm.

The evolutions of the nugget λ and length scales (γ_1, γ_2) with respect to the number of iterations n of the parametric KF algorithm are shown on Figure 3.4(a), Figure 3.4(b), and Figure 3.4(c) respectively. One can notice that the length scale γ_2, which is the one corresponding to the angle of attack α, is higher than the length scale γ_1, which is the one corresponding to the Mach number M. It may mean that the surrogate shows lower amplitude oscillations in the direction of α than in the direction of M. A summary of the results on the test set is shown in Table 3.2. The KF algorithm yields slightly better results than the MLL optimization, while being both very close to each other. The main discrepancies between KF and MLL results are the second length scale γ_2 and the nugget λ. What is interesting to notice is that both methods do not yield the exact same result. First, one can notice the difference between the length scales γ_2 for KF and MLL. The length scale given by the MLL maximization $\gamma_2 = 42.5350$ is much bigger than the length scale given by the parametric KF algorithm $\gamma_2 = 2.9695$. Second, the nugget is larger in the KF case $\lambda = 1.000 \times 10^{-4}$ than in the MLL case $\lambda = 1.1995 \times 10^{-6}$. In any case, thanks to the parametric KF algorithm, we manage to divide e_{NMSE} by a factor of about 7, namely from 3.572×10^{-4} to 5.187×10^{-5}. Nevertheless, e_{MRE} is slightly worse than at the beginning, from 0.922% to 1.156%. It means that the parametric KF surrogate model approximates better F for the majority of points except for a few.

3.3.4.3 Non-parametric KF algorithm

Along the same lines as in Section 3.3.4.2, we learn now the flow function $f_n : \mathcal{X} \mapsto \mathcal{X}$ instead of parameters. Between the iterations n and $n+1$, the locations of the I_{f} observations are translated so that the absolute translation is less than a



(a) Nugget λ. (b) Length scale γ_1. (c) Length scale γ_2.

Figure 3.4: Evolutions of the nugget λ (a) and the length scales (γ_1, γ_2) (b)–(c) with respect to the number n of iterations for the parametric KF algorithm.

	Iteration $n = 0$	Iteration $n = 3 \times 10^4$	
		Kernel Flow	Marginal Likelihood
(γ_1, γ_2)	$(1.3024, 1.3024)$	$(0.6513, 2.9695)$	$(0.5912, 42.5350)$
λ	1.000×10^{-4}	8.000×10^{-4}	1.1995×10^{-6}
e_{MSE}	0.104	0.0151	0.0167
e_{MRE} (in %)	0.922	1.156	1.134
e_{NMSE}	3.572×10^{-4}	5.187×10^{-5}	5.721×10^{-5}

Table 3.2: Summary of the results on the test set for the parametric KF algorithm.

specified constant δ:

$$\max_{i \in \pi_{\text{f}}^n} \|f_{n+1}(\mathbf{X}_i) - f_n(\mathbf{X}_i)\|_2 \leq \delta, \tag{3.69}$$

with $\delta = 1 \times 10^{-6}$. The evolution of the accuracy ρ for each fold and its average with respect to the number n of iterations of the non-parametric KF algorithm is shown on Figure 3.5 for 1×10^5 iterations. The accuracy ρ is slowly decreasing until it stabilizes at $n \approx 8 \times 10^4$. The translation of the I observations corresponding to the flow function f_n between the initial and final iterations is shown on Figure 3.6. One can notice that the translations are mainly with respect to the Mach number M.

A summary of the results on the test set is given in Table 3.3. The performance of the surrogate model is worsen between the initial iteration $n = 0$ and the last one $n = 1 \times 10^5$. The non-parametric KF algorithm may not be suitable for regression and when the available data are scarce. It confirms the results found in [35] where a similar conclusion was reached.

3.3.5 Concluding remarks

In this section, we have seen two methods to determine the "best" kernel K with respect to the available observations of the ground truth function F: the parametric KF algorithm and the non-parametric KF algorithm, where the latter has previously been validated on a classification example [114]. We have seen that the parametric KF algorithm yields promising results and it could be an interesting equivalent of

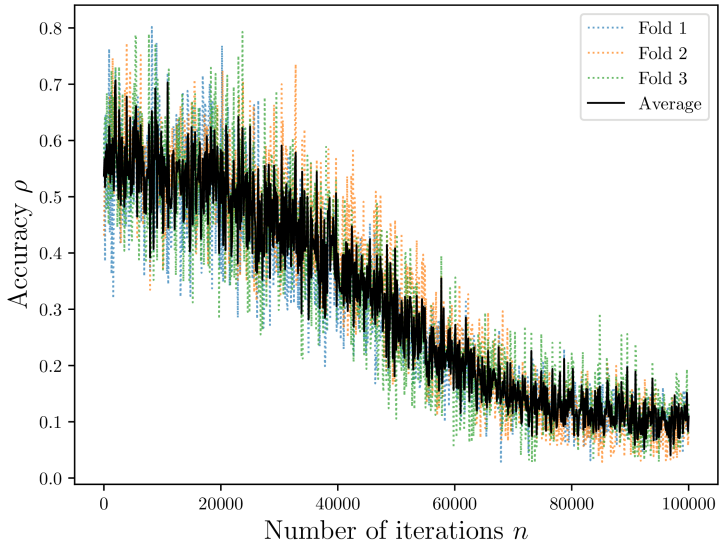
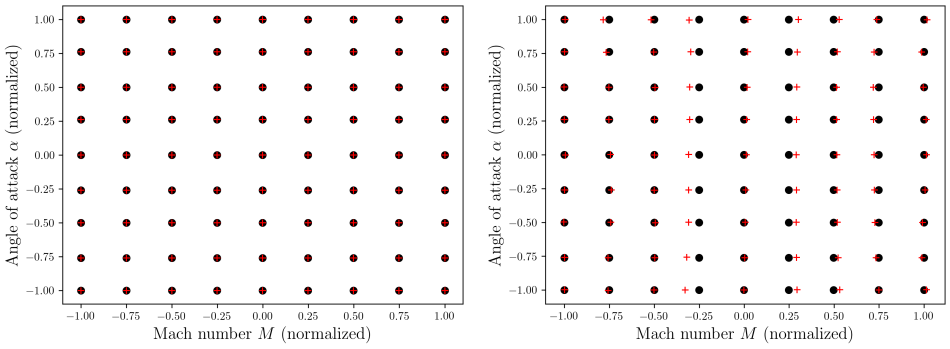


Figure 3.5: Evolution of ρ on each fold and its average with respect to the number n of iterations for the non-parametric KF algorithm.



(a) Iteration $n = 0$. (b) Iteration $n = 1 \times 10^5$.

Figure 3.6: The translation of the I observations corresponding to the flow function f_n between the initial iteration $n = 0$ and the final iteration $n = 1 \times 10^5$. The black dots $\bullet$ are for the initial positions of the observations and the red crosses $+$ are for the moved positions.

	Iteration $n = 0$	Iteration $n = 1 \times 10^5$
		Kernel Flow
(γ_1, γ_2)	$(1.3024, 1.3024)$	$(1.3024, 1.3024)$
λ	1×10^{-4}	1×10^{-4}
e_{MSE}	0.104	0.789
e_{MRE} (in %)	0.922	4.274
e_{NMSE}	3.572×10^{-4}	2.703×10^{-3}

Table 3.3: Summary of the results on the test set for the non-parametric KF algorithm. The nugget and the length scales are unchanged by the non-parametric KF algorithm.

cross-validation in regression context. However, the non-parametric version of the KF algorithm yields worse results and it may be not suitable for regression when data is scarce. This confirms the results obtained in [35]. Another approach to determine the "best" kernel is outlined in the next section.

3.4 Spectral Kernel Ridge Regression algorithms

In this section, new algorithms that we will call Spectral Kernel Ridge Regression (SKRR) are introduced to determine the "best" kernel. Recalling Section 3.2.4, the deterministic errors in (3.13) and (3.14) are bounded if the norms $\|F\|_{\mathcal{H}_K}$ and $\|F\|_{\mathcal{H}_{K+\lambda}}$ are bounded as well. Therefore, we suggest to study the effect of the kernel K on the norm $\|F\|_{\mathcal{H}_K}$ and in this way to find the "best" kernel by minimizing the latter with respect to K. That is, we aim at finding the kernel $K^\star$ such that:

$$K^\star = \arg\min_K \|F\|_{\mathcal{H}_K} . \tag{3.70}$$

3.4.1 Mercer's theorem

This section is a brief summary about Mercer's framework, which allows us to express a Mercer kernel as a function of eigenvalues and eigenvectors of its associated integral operator.

Definition 8. *Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$. A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a Mercer kernel if it is continuous, symmetric, and positive semi-definite in the sense of Equation (3.1).*

This definition allows us to state the Mercer's theorem [118, Chapter 11]:

Theorem 4 (Mercer's theorem)**.** *Let μ be a finite Borel measure with support $\mathcal{X}$ and let $L^2(\mathcal{X}, \mu)$ be the set of square integrable functions on $\mathcal{X}$ with respect to μ. Let K be a Mercer kernel on $\mathcal{X}$ and let $T_K : L^2(\mathcal{X}, \mu) \to L^2(\mathcal{X}, \mu)$ be the associated integral operator defined by $\forall f \in L^2(\mathcal{X}, \mu), \forall \mathbf{x} \in \mathcal{X}, T_K f(\mathbf{x}) = \int_{\mathcal{X}} K(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \mu(d\mathbf{y})$. Then there exists a countable collection of orthonormal functions $\{e_i\}_{i \in \mathbb{N}}$ of $L^2(\mathcal{X}, \mu)$ which are eigenvectors of T_K with associated non-negative eigenvalues $\{\sigma_i \geq 0\}_{i \in \mathbb{N}}$. Moreover, taking the eigenvectors corresponding to the non-zero eigenvalues, they are continuous functions on $\mathcal{X}$ and $K(\mathbf{x}, \mathbf{y})$ has the following representation*

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{+\infty} \sigma_i e_i(\mathbf{x}) \otimes e_i(\mathbf{y}),$$

where the series converges absolutely and uniformly:

$$\lim_{n \to +\infty} \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \left| K(\mathbf{x}, \mathbf{y}) - \sum_{i=0}^{n} \sigma_i e_i(\mathbf{x}) \otimes e_i(\mathbf{y}) \right| = 0.$$

From Mercer's theorem, T_K is a trace class operator with

$$\mathrm{Tr}(T_K) = \int_{\mathcal{X}} K(\mathbf{x},\mathbf{x})\mu(d\mathbf{x}) = \sum_{i=0}^{+\infty} \sigma_i < +\infty. \tag{3.71}$$

Remark 10. *Since T_K is self-adjoint and compact, $\{e_i\}_{i\in\mathbb{N}}$ is a basis of $L^2(\mathcal{X},\mu)$. We remind that a family of functions $\{e_i\}_{i\in\mathbb{N}}$ is a basis of a Banach space H if $\forall f \in H$, $\exists!\{\alpha_i\}_{i\in\mathbb{N}}$ such that $\forall \varepsilon > 0$, $\exists N$ such that $\forall n \geq N$*

$$\left\| f - \sum_{i=0}^{n} \alpha_i e_i \right\|_H \leq \varepsilon.$$

Mercer's theorem also allows us to define explicitly the RKHS functional space $\mathcal{H}_K$ associated with the kernel K, on the condition that K is a Mercer kernel. Indeed, one has [118, Theorem 11.18]

$$\mathcal{H}_K = \left\{ f \in L^2(\mathcal{X},\mu), f = \sum_{i=0}^{+\infty} \langle f, e_i\rangle_{L^2} e_i \text{ with } \sum_{i=0}^{+\infty} \frac{\langle f, e_i\rangle_{L^2}^2}{\sigma_i} < +\infty \right\},$$

where $\langle\cdot,\cdot\rangle_{L^2}$ is the inner product of $L^2(\mathcal{X},\mu)$. The inner product of $\mathcal{H}_K$ is given as $\forall (f,g) \in \mathcal{H}_K \times \mathcal{H}_K$,

$$\langle f, g\rangle_{\mathcal{H}_K} = \sum_{i=0}^{+\infty} \frac{\langle f, e_i\rangle_{L^2} \langle g, e_i\rangle_{L^2}}{\sigma_i}.$$

One can notice that $\langle f, e_i\rangle_{L^2} = \sigma_i \langle f, e_i\rangle_{\mathcal{H}_K}$, so one also has

$$\langle f, g\rangle_{\mathcal{H}_K} = \sum_{i=0}^{+\infty} \sigma_i \langle f, e_i\rangle_{\mathcal{H}_K} \langle g, e_i\rangle_{\mathcal{H}_K}.$$

In this way, the norm $\|f\|^2_{\mathcal{H}_K}$ reads

$$\|f\|^2_{\mathcal{H}_K} = \sum_{i=0}^{+\infty} \frac{\langle f, e_i\rangle_{L^2}^2}{\sigma_i}, \tag{3.72}$$

which gives us its expression in terms of the eigenvalues and eigenvectors of K.

Remark 11. *The functional space $\mathcal{H}_K$ does not depend on the measure μ, actually. Only the eigenvalues $\{e_i\}_{i\in\mathbb{N}}$ and the eigenvectors $\{\sigma_i \geq 0\}_{i\in\mathbb{N}}$ do.*

Remark 12. *It can be shown that $\{\sqrt{\sigma_i} e_i\}_{i\in\mathbb{N}}$ is an orthonormal basis of $\mathcal{H}_K$; see [118, Theorem 11.18].*

Remark 13. *In the sequel μ will be a Borel probability measure, that is $\mu(\mathcal{X}) = 1$.*

3.4.2 Minimizing the norm $\|F\|_{\mathcal{H}_K}$: the SKRR core

We assume that a basis $\{e_i\}_{i\in\mathbb{N}}$ of $L^2(\mathcal{X},\mu)$ is given (these are the "features" in machine learning techniques, for example), and we aim at constructing a Mercer kernel $K=\sum_i \sigma_i e_i\otimes e_i$ where the $\{\sigma_i>0\}_{i\in\mathbb{N}}$ are to be chosen such that, if $F\in\mathcal{H}_K$, its norm $\|F\|_{\mathcal{H}_K}$ is minimal. The latter reads:

$$\|F\|^2_{\mathcal{H}_K}=\sum_{i=0}^{+\infty}\frac{|F_i|^2}{\sigma_i}, \tag{3.73}$$

where $F_i=\langle F,e_i\rangle_{L^2}$, $i\in\mathbb{N}$, such that $\sum_i |F_i|^2<+\infty$ because $F\in L^2(\mathcal{X},\mu)$. The minimization problem thus reads:

$$\min_{\{\sigma_i\}_i}\sum_i\frac{|F_i|^2}{\sigma_i}\text{ subjected to }\sum_i\sigma_i=\kappa, \tag{3.74}$$

where $0<\kappa<+\infty$; this condition arises from Equation (3.71). This minimization problem can be solved by the method of Lagrange multipliers. Let ν be a Lagrange multiplier and $\{\sigma_i^\star\}_{i\in\mathbb{N}}$ be the solution of Equation (3.74), one has

$$\begin{aligned}\frac{\partial}{\partial\sigma_k}\left[\sum_i\frac{|F_i|^2}{\sigma_i^\star}+\nu\left(\sum_i\sigma_i^\star-\kappa\right)\right]&=0\quad\text{for } k=0,1,2,\dots,\\ \frac{\partial}{\partial\nu}\left[\sum_i\frac{|F_i|^2}{\sigma_i^\star}+\nu\left(\sum_i\sigma_i^\star-\kappa\right)\right]&=0.\end{aligned}$$

Then one finds that

$$\nu=\frac{1}{\kappa^2}\left(\sum_i|F_i|\right)^2,\quad \sigma_i^\star=\frac{|F_i|}{\sqrt{\nu}},\quad i=0,1,2,\dots, \tag{3.75}$$

or

$$\sigma_i^\star=\frac{\kappa|F_i|}{\sum_j|F_j|},\quad i=0,1,2,\dots \tag{3.76}$$

In this way we built a Mercer kernel $K^\star=\sum_i\sigma_i^\star e_i\otimes e_i$ which minimizes the norm $\|F\|_{\mathcal{H}_K}$. Practically, the sum extends up to a finite rank R. It remains to compute the expansion coefficients $\{F_i\}_{i\in\mathbb{N}}$: two approaches based on regression and projection are outlined in the next section.

3.4.3 Computing the expansion coefficients

Again, we assume that an orthornomal basis $\{e_i\}_{i\in\mathbb{N}}$ of $L^2(\mathcal{X},\mu)$ is available and that $F\in L^2(\mathcal{X},\mu)$. One has thus the following expansions:

$$F(\mathbf{X}_j)=\sum_{i=0}^{+\infty}\langle F,e_i\rangle_{L^2}\,e_i(\mathbf{X}_j)=Y_j,\text{ for } j=1,\dots,I.$$

Let $\mathcal{K}$ be a set of indices such that $\#\mathcal{K} = R$ and let the corresponding orthonormal vectors $\{e_k\}_{k\in\mathcal{K}} \subset \{e_i\}_{i\in\mathbb{N}}$ in $L^2(\mathcal{X}, \mu)$ be the basis $\mathcal{B}^R$. Let $\boldsymbol{\Theta}$ be the measurement matrix and $\mathbf{c}$ be the expansion coefficients vector such that

$$\boldsymbol{\Theta} = \begin{bmatrix} e_{k_1}(\mathbf{X}_1) & \cdots & e_{k_R}(\mathbf{X}_1) \\ \vdots & \ddots & \vdots \\ e_{k_1}(\mathbf{X}_I) & \cdots & e_{k_R}(\mathbf{X}_I) \end{bmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_{k_1} \\ \vdots \\ c_{k_R} \end{pmatrix} = \begin{pmatrix} \langle F, e_{k_1} \rangle_{L^2} \\ \vdots \\ \langle F, e_{k_R} \rangle_{L^2} \end{pmatrix}, \tag{3.77}$$

one arrives at the following $I \times R$ system

$$\mathbf{Y} = \boldsymbol{\Theta}\mathbf{c} + \boldsymbol{\eta}, \tag{3.78}$$

where $\boldsymbol{\eta} = \begin{pmatrix} \eta_1 & \dots & \eta_I \end{pmatrix}^T$ is the error vector with $\|\boldsymbol{\eta}\|_2 \leq \eta$ accounting for the truncation of the exact function F on the set of R orthonormal vectors $\{e_k\}_{k\in\mathcal{K}}$, and possible noise.

3.4.3.1 Regression approach

We first assume that $I \geq R$. Finding the expansion coefficients $\mathbf{c}$ can be done through a regression approach formulated as a least-squares minimization problem that is, solving the following problem:

$$\mathbf{c}^\star = \underset{\mathbf{h}\in\mathbb{R}^R}{\arg\min} \left(\mathbf{Y} - \boldsymbol{\Theta}\mathbf{h}\right)^T \left(\mathbf{Y} - \boldsymbol{\Theta}\mathbf{h}\right). \tag{3.79}$$

This approach is detailed in [60] and references therein. Here, we are more interested in the case where $I < R$ or even $I \ll R$, namely when the number of "features" R is way more larger than the number of observations I of F. This is the topic of the following section.

3.4.3.2 Sparse regression by ℓ_1-minimization

We now assume that $I \ll R$. Actually, the "features" are chosen such that the ground truth function F is expected to be sparse or nearly sparse on this basis. That is, many components of the vector $\mathbf{c}$ of the expansion coefficients are negligible. Such expansion is known as compressible in the terminology of the theory of compressed sensing, or compressive sampling (CS) [22, 23, 24, 38]; see for instance [129] in an aerodynamics context or [24] in a data acquisition context. Thus one introduces the sparsity S defined by

$$S = \#\left\{i; |c_i| > \delta\right\}, \tag{3.80}$$

where $\delta > 0$ is some tolerance, and assume that $S \ll R$. In other words, only a small number of vectors within the basis $\mathcal{B}^R$ is relevant to reconstruct the ground truth function F without much loss, and this number is that sparsity S. We outline below how to solve Equation (3.78) in this context.

Noiseless observations We first consider that $\boldsymbol{\eta} = \mathbf{0}$, such that Equation (3.78) reads

$$\mathbf{Y} = \boldsymbol{\Theta}\mathbf{c}. \tag{3.81}$$

This system is ill-posed as $R \gg I$ if no additional constraint on the solution is enforced. Regularized versions of Equation (3.81) exist in order to ensure its well-posedness by imposing a sparsity constraint. For instance, Equation (3.81) can be regularized as the following l_0-minimization problem where solutions are found by solving

$$\min_{\mathbf{h}\in\mathbb{R}^R} \|\mathbf{h}\|_0 \text{ subjected to } \boldsymbol{\Theta}\mathbf{h} = \mathbf{Y}, \tag{3.82}$$

where $\|\mathbf{h}\|_0$ is the number of non-zeros entries of the vector $\mathbf{h}$. The main disadvantages of Equation (3.82) are that this problem is non-convex and NP-hard combinatorial [38]. It can be relaxed by considering the following convex ℓ_1-minimization known as Basis Pursuit (BP) [29]:

$$\min_{\mathbf{h}\in\mathbb{R}^R} \|\mathbf{h}\|_1 \text{ subjected to } \boldsymbol{\Theta}\mathbf{h} = \mathbf{Y}. \tag{3.83}$$

Equation (3.83) is now a convex optimization that can be solved numerically. In [24], it has been shown that Equation (3.83) shares the same solution as Equation (3.82) if the measurement matrix $\boldsymbol{\Theta}$ obeys the condition known as the Restricted Isometry Property (RIP) [6, 21, 24].

Definition 9 (Restricted Isometry Property (RIP))**.** *For each integer $m \in \mathbb{N}^*$, the retricted isometry constant (RIC) δ_m of the measurement matrix $\boldsymbol{\Theta}$ is defined as the smallest number such that*

$$(1-\delta_m)\|\mathbf{h}_m\|_2^2 \leq \|\boldsymbol{\Theta}\mathbf{h}_m\|_2^2 \leq (1-\delta_m)\|\mathbf{h}_m\|_2^2$$

holds for all m-sparse vectors $\mathbf{h}_m \in \{\mathbf{h} \in \mathbb{R}^R; \|\mathbf{h}\|_0 \leq m\}$.

The matrix $\boldsymbol{A}$ is said to obey the RIP of order m if δ_m is not too close to one [25]. In fact, it has been proved that $\delta_{2m} < 1$ is a necessary and sufficient condition for Equation (3.82) to yield an unique solution [21]. This definition allows us to state the following theorem that bounds the recovery error:

Theorem 5 (Noiseless recovery [24])**.** *Assume that $\delta_{2S} \leq \sqrt{2} - 1$ and that the measurement matrix $\boldsymbol{\Theta}$ obeys the RIP. Then the solution $\mathbf{c}^\star$ to Equation (3.83) obeys*

$$\|\mathbf{c}^\star - \mathbf{c}\|_1 \leq C_0 \|\mathbf{c}_S - \mathbf{c}\|_1$$

and

$$\|\mathbf{c}^\star - \mathbf{c}\|_2 \leq C_0 s^{-1/2} \|\mathbf{c}_S - \mathbf{c}\|_1$$

for a positive constant $C_0 > 0$ depending only on δ_{2S}. The vector $\mathbf{c}_S$ corresponds to the vector $\mathbf{c}$ with all but the S largest components set to 0. Moreover, the constant C_0 is equal to

$$C_0 = 2\frac{1-(1-\sqrt{2})\delta_{2S}}{1-(1+\sqrt{2})\delta_{2S}}.$$

In particular, if the vector $\mathbf{c}$ is exactly S-sparse the recovery is exact, that is the solution $\mathbf{c}^\star$ to Equation (3.83) is equal to $\mathbf{c}$. But we can go even further. Even if the vector $\mathbf{c}$ is not exactly S-sparse, we know from this theorem that we are guaranteed to recover the S-largest coefficients of $\mathbf{c}$. Given a measurement matrix $\boldsymbol{\Theta}$, computing the RIC may be difficult in practical cases, so that verifying the RIP condition is difficult as well.

If the vector $\mathbf{c}$ is S-sparse in the basis $\mathcal{B}^R$, then a lower bound exists on the minimum number I of observations of F required to exactly obtain $\mathbf{c}$ with high probability. First, we define the coherence parameter [26].

Definition 10 (Coherence parameter)**.** *Let $\boldsymbol{\Theta}$ be a measurement matrix. Then, the coherence parameter is defined as*

$$\mathrm{C}(\boldsymbol{\Theta}) = \max_{i,j} \left(|\boldsymbol{\Theta}_{ij}|^2 \right).$$

Namely, the coherence parameter $\mathrm{C}(\boldsymbol{\Theta})$ is the largest squared component of the measurement matrix $\boldsymbol{\Theta}$. The following theorem holds which gives us the minimum number of observations to recover exactly the expansion coefficients $\mathbf{c}$ in the noiseless case.

Theorem 6 (Noiseless Incoherent Sampling [26])**.** *Let $\mathbf{c}$ be an arbitrary S-sparse vector and pick any scalar $\beta > 0$. Then with probability at least $1 - 5/R - \exp(-\beta)$, the solution of Equation (3.83) is unique and equal to $\mathbf{c}$ provided that*

$$I \geq C(1+\beta)\mathrm{C}(\boldsymbol{\Theta}) S \log(R)$$

for some positive constant C, where the I observations have been drawn randomly following the probability measure μ.

Contrary to Theorem 5, the required minimal number of observations given by Theorem 6 does not yield deterministic results. That is, we are not sure to exactly recover the vector $\mathbf{c}$ even if it is exactly S-sparse.

Noisy observations We now consider that $\eta > 0$ so that $\|\boldsymbol{\eta}\|_2 \leq \eta$ in order to take into account the truncation error. Equation (3.78) can be regularized as the following l_0-minimization problem where solutions can be found by solving:

$$\min_{\mathbf{h} \in \mathbb{R}^R} \|\mathbf{h}\|_0 \text{ subjected to } \|\boldsymbol{\Theta}\mathbf{h} - \mathbf{Y}\|_2 \leq \eta \text{ with } 0 \leq \eta \ll 1. \tag{3.84}$$

As in the noiseless case, Equation (3.84) can be relaxed by considering the following convex ℓ_1-minimization known as Basis Pursuit Denoising (BPDN) [30]:

$$\min_{\mathbf{h} \in \mathbb{R}^R} \|\mathbf{h}\|_1 \text{ subjected to } \|\boldsymbol{\Theta}\mathbf{h} - \mathbf{Y}\|_2 \leq \eta. \tag{3.85}$$

Under the same hypotheses as in Theorem 5, one has the following theorem:

Theorem 7 (Noisy recovery [24]). *Assume that* $\delta_{2S} \leq \sqrt{2}-1$. *Then the solution* $\mathbf{c}^\star$ *to Equation* (3.85) *obeys*

$$\|\mathbf{c}^\star - \mathbf{c}\|_2 \leq C_0 S^{-1/2} \|\mathbf{c_S} - \mathbf{c}\|_1 + C_1 \eta,$$

for some positive constants $C_0, C_1 > 0$ *depending only on* δ_{2S} *and where the vector* $\mathbf{c}_S$ *corresponds to the vector* $\mathbf{c}$ *with all but the* S *largest components set to 0. Moreover, one has the following expressions for the constants* C_0 *and* C_1

$$C_0 = 2\frac{1-(1-\sqrt{2})\delta_{2S}}{1-(1+\sqrt{2})\delta_{2S}} \text{ and } C_1 = 4\frac{\sqrt{1+\delta_{2S}}}{1-(1+\sqrt{2})\delta_{2S}}. \tag{3.86}$$

The constants C_0 and C_1 are typically small. For instance, if $\delta_{2S} = 0.2$, we have $C_0 \approx 4.2$ and $C_1 \approx 8.5$ [24]. The bound $\sqrt{2}-1$ on δ_{2S}, initially proposed in [25], can be improved. Such improvements can be found in [107], where $\delta_{2S} \leq 0.4931$ and even $\delta_{2S} \leq 0.6569$ for some specific cases. This topic is currently an active field of research. Notice that the sparsity of a quantity of interest is seen *a posteriori* and not *a priori*. If the set of orthonormal vectors $\{e_k\}_{k\in\mathcal{K}}$ and η are well chosen, Equation (3.85) will yield a sparse solution which approximates well the ground truth function F, *i.e.* a solution where only a few terms are non vanishing. In this research, the selection of the value η is done arbitrarily but this selection can be done through cross-validation [17, 39, 157].

Sparse SKRR algorithm We propose the following algorithm named Sparse Spectral Kernel Ridge Regression (SSKRR) which couples the sparse reconstruction by ℓ_1-minimization presented in the foregoing section, and the KRR approximation detailed in Section 3.2.3. The main idea of this algorithm is to minimize the RKHS norm of the ground truth function F with respect to the eigenvalues of a Mercer kernel, which are obtained by Equation (3.76) where the expansion coefficients $\{F_i\}_{i\in\mathcal{K}}$ in a finite basis $\mathcal{B}^R$ are computed by ℓ_1-minimization. The sketch of this algorithm can be seen in Algorithm 2 on page 104. It is organized as follows. First, starting from a set of R orthonormal vectors $\mathcal{B}^R \equiv \{e_k\}_{k\in\mathcal{K}}$ in $L^2(\mathcal{X},\mu)$, I observations of F, and the parameter η, the BPDN minimization of Equation (3.85) is solved to find a solution denoted by $\mathbf{c}^\star$. If the set $\mathcal{B}^R$ of "features" is well chosen, only a limited number of terms of the vector $\mathbf{c}^\star$ is not close to zero. Then one solves the KRR equation (3.9), which allows us to get a prediction of the ground truth function F at an unobserved location $\mathbf{x}$ with the parameter λ and using the kernel determined at the previous step. One of the main advantages of the proposed algorithm is that it provides the prediction variance $\sigma^2(\mathbf{x})$ at the unobserved point $\mathbf{x}$, such as in the Gaussian Process Regression framework.

Obviously, the algorithm will strongly depend on the performance of the ℓ_1-minimization to approximate the expansion coefficients $\{F_i\}_{i\in\mathcal{K}}$. Also, several remarks can be made about the parameter κ in Equation (3.76). The SSKRR approximation at an unobserved point $\mathbf{x}$ is

$$G^\star_\lambda(\mathbf{x}) = \mathbf{K}^\star(\mathbf{x},\mathbf{X})\left(\mathbf{K}^\star(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1}\mathbf{Y}. \tag{3.87}$$

Then, by using Equation (3.76),

$$G^{\star}_{\lambda}(\mathbf{x}) = \sum_{k\in\mathcal{K}} \frac{\kappa|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{x}) \otimes e_k(\mathbf{X}) \left(\sum_{k\in\mathcal{K}} \frac{\kappa|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{X}) \otimes e_k(\mathbf{X}) + \lambda \mathbf{I}_I \right)^{-1} \mathbf{Y}$$
$$= \sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{x}) \otimes e_k(\mathbf{X}) \left(\sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{X}) \otimes e_k(\mathbf{X}) + \frac{\lambda}{\kappa} \mathbf{I}_I \right)^{-1} \mathbf{Y}.$$

So, one can see that the SSKRR approximation only depends on the ratio between the parameter λ and the parameter κ. Likewise for the prediction variance of Equation (3.15), one has

$$\sigma^2(\mathbf{x}) = K^{\star}(\mathbf{x},\mathbf{x}) - \mathbf{K}^{\star}(\mathbf{x},\mathbf{X}) \left(\mathbf{K}^{\star}(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1} \mathbf{K}^{\star}(\mathbf{X},\mathbf{x})$$
$$= \kappa \left[\sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{x}) \otimes e_k(\mathbf{x}) - \sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{x}) \otimes e_k(\mathbf{X}) \right.$$
$$\left. \times \left(\sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{X}) \otimes e_k(\mathbf{X}) + \frac{\lambda}{\kappa} \mathbf{I}_I \right)^{-1} \sum_{k\in\mathcal{K}} \frac{|F_k|}{\sum\limits_{j\in\mathcal{K}}|F_j|} e_k(\mathbf{X}) \otimes e_k(\mathbf{x}) \right]$$

The parameter κ fixing the trace of the integral operator with kernel $K^{\star}$ can be understood as a scaling factor on the prediction variance.

3.4.3.3 Non-sparse spectral kernel ridge regression algorithm

If the ground truth function F is not sparse on the basis $\mathcal{B}^R$, one can compute the expansion coefficients $\mathbf{c}$ by projection since $\mathcal{B}^R$ is orthonormal. We propose the following Algorithm 3 sketched on page 105 where projections are carried out iteratively using the iterated surrogate approximations to mimic the ground truth function F. We call this algorithm Non-sparse Spectral Kernel Ridge Regression (NSKRR).

3.5 Polynomial Chaos Expansion

The remaining question is: how to choose a basis $\{e_i\}_{i\in\mathbb{N}}$ of $L^2(\mathcal{X},\mu)$? In this section, we focus on orthonormal polynomial bases. Such kind of representations are referred to as Polynomial Chaos Expansion (PCE) in the case where μ is a Gaussian probability measure [56, 158]. For more general probability measures, they are called generalized Polynomial Chaos (gPC) expansion [49, 87, 135, 162]. Besides, the use of polynomial bases in the context of Gaussian process regression is illustrated in *e.g.* [164].

Algorithm 2: Sparse Spectral Kernel Ridge Regression (SSKRR).

Input: Basis $\{e_k\}_{k\in\mathcal{K}}$, I observations of F, parameters η, λ and κ.
Output: SSKRR approximation $G_\lambda^\star$ and prediction variance $\sigma^2(\mathbf{x})$.

1 Build the observation matrix $\boldsymbol{\Theta}$ from the basis $\{e_k\}_{k\in\mathcal{K}}$;
2 Solve the BPDN minimization to find the projection coefficients $\mathbf{c}^\star$:
$\mathbf{c}^\star = \underset{\mathbf{h}\in\mathbb{R}^R}{\arg\min} \|\mathbf{h}\|_1$ subjected to $\|\boldsymbol{\Theta}\mathbf{h} - \mathbf{Y}\|_2 \leq \eta$;
3 Solve $\boldsymbol{\sigma}^\star = \underset{\{\sigma_k\}_{k\in\mathcal{K}}}{\arg\min} \|g\|^2_{\mathcal{H}_K} = \underset{\{\sigma_k\}_{k\in\mathcal{K}}}{\arg\min} \sum_{k\in\mathcal{K}} \frac{c_k^{\star 2}}{\sigma_k}$ subjected to $\sum_{k\in\mathcal{K}} \sigma_k = \kappa$ following Equation (3.76);
4 Form the new kernel as $K^\star(\mathbf{x},\mathbf{y}) = \sum_{k\in\mathcal{K}} \sigma_k^\star e_k(\mathbf{x}) e_k(\mathbf{y})$;
5 Obtain the SSKRR approximation

$$G_\lambda^\star(\mathbf{x}) = \mathbf{K}^\star(\mathbf{x},\mathbf{X}) \left(\mathbf{K}^\star(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1} \mathbf{Y} \tag{3.88}$$

and the prediction variance

$$\sigma^2(\mathbf{x}) = K^\star(\mathbf{x},\mathbf{x}) - \mathbf{K}^\star(\mathbf{x},\mathbf{X}) \left(\mathbf{K}^\star(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1} \mathbf{K}^\star(\mathbf{X},\mathbf{x})$$

at an unobserved point $\mathbf{x}$.

3.5.1 Polynomial basis

Assume that an orthonormal polynomial basis of $L^2(\mathcal{X},\mu)$ is available, where the random input variables $\mathbf{X} \sim \mu$ are assumed to be mutually independent with values in $\mathcal{X} \subset \mathbb{R}^d$. One can build a polynomial surrogate G_p of F by a standard L^2 projection on a finite dimensional subspace of $L^2(\mathcal{X},\mu)$ spanned by the truncated family of d-variate orthonormal polynomials up to total order p denoted by $\mathcal{B}^p = \{\phi_i\}_{i=0}^P$, where P is given by:

$$P + 1 = \frac{(p+d)!}{p!d!}. \tag{3.89}$$

The orthonormalization of this basis is done through:

$$\langle \phi_i, \phi_j \rangle_{L^2} = \int_{\mathcal{X}} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) \mu(d\mathbf{x}) = \delta_{ij}, \tag{3.90}$$

where δ_{ij} is the Kronecker symbol such that $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. Then, one can write the surrogate model G_p using the truncated basis $\mathcal{B}^p$ as follows:

$$G_p(\mathbf{x}) = \sum_{i=0}^{P} F_i \phi_i(\mathbf{x}) = \sum_{i=0}^{P} \langle F, \phi_i \rangle_{L^2} \phi_i(\mathbf{x}). \tag{3.91}$$

Algorithm 3: Non-sparse Spectral Kernel Ridge Regression (NSKRR).

Input: Basis $\{e_k\}_{k\in\mathcal{K}}$, eigenvalues $\{\sigma_k^{(0)}\}_{k\in\mathcal{K}}$, I observations of F, parameters λ and κ, and the number of iterations N.

Output: NSKRR approximation $G_\lambda^N(\mathbf{x})$ and prediction variance $\sigma^2(\mathbf{x})$.

Initialization: At step $n = 0$, define the initial kernel as $K^{(0)}(\mathbf{x},\mathbf{y}) = \sum_{k\in\mathcal{K}} \sigma_k^{(0)} e_k(\mathbf{x}) e_k(\mathbf{y})$;

for $n \leftarrow 1$ **to** N **do**

  Approximate F by its NSKRR approximation: $G_\lambda^{(n-1)}(\mathbf{x}) = \mathbf{K}^{(n-1)}(\mathbf{x},\mathbf{X})\left(\mathbf{K}^{(n-1)}(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1}\mathbf{Y}$;

  for $k \in \mathcal{K}$ **do**

    Compute $c_k^{(n-1)} = \left\langle G_\lambda^{(n-1)}, e_k \right\rangle_{L^2}$;

  end

  Solve $\boldsymbol{\sigma}^{(n)} = \underset{\{\sigma_k\}_{k\in\mathcal{K}}}{\arg\min} \left\| G_\lambda^{(n-1)} \right\|_{\mathcal{H}_K}^2 = \underset{\{\sigma_k\}_{k\in\mathcal{K}}}{\arg\min} \sum_{k\in\mathcal{K}} \frac{\left(c_k^{(n-1)}\right)^2}{\sigma_k}$ subjected to $\sum_{k\in\mathcal{K}} \sigma_k = \kappa$;

  Form the new kernel as $K^{(n)}(\mathbf{x},\mathbf{y}) = \sum_{k\in\mathcal{K}} \sigma_k^{(n)} e_k(\mathbf{x}) e_k(\mathbf{y})$;

end

Obtain the NSKRR approximation

$$G_\lambda^{(N)}(\mathbf{x}) = \mathbf{K}^{(N)}(\mathbf{x},\mathbf{X})\left(\mathbf{K}^{(N)}(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1}\mathbf{Y}$$

and the prediction variance

$$\left(\sigma^{(N)}(\mathbf{x})\right)^2 = K^{(N)}(\mathbf{x},\mathbf{x}) - \mathbf{K}^{(N)}(\mathbf{x},\mathbf{X})\left(\mathbf{K}^{(N)}(\mathbf{X},\mathbf{X}) + \lambda \mathbf{I}_I\right)^{-1}\mathbf{K}^{(N)}(\mathbf{X},\mathbf{x})$$

at an unobserved point $\mathbf{x}$.

In this framework, finding the coefficients $F_i = \langle F, \phi_i \rangle_{L^2}$ is usually done by a numerical quadrature rule such that

$$\int_{\mathcal{X}} f(\mathbf{x})\mu(d\mathbf{x}) \approx \sum_{l=1}^{q} \omega_l f(\boldsymbol{\vartheta}_l), \tag{3.92}$$

where $\{\omega_l\}_{l=1}^q$ are positive weights and $\{\boldsymbol{\vartheta}_l\}_{l=1}^q$ are nodes in $\mathcal{X}$. Using this quadrature rule, the coefficients are approximated by

$$F_i \approx F_{i,q} = \sum_{l=1}^{q} \omega_l F(\boldsymbol{\vartheta}_l)\phi_i(\boldsymbol{\vartheta}_l). \tag{3.93}$$

The number of nodes q that is needed is given by the selected quadrature rule. The classical Gauss-Jacobi (GJ) quadrature rule requires q nodes to exactly integrate

univariate polynomials up to order $2q-1$ with respect to the Beta law on an interval $\mathcal{I}$. If one of the end point of $\mathcal{I}$ is required to be a fixed quadrature node, the Gauss-Jacobi-Radau (GJR) rule is obtained, which exactly integrates univariate polynomials up to order $2q-2$. If both end points of $\mathcal{I}$ are required to be fixed quadrature nodes, the Gauss-Jacobi-Lobatto (GJL) rule is obtained, which exactly integrates univariate polynomials up to order $2q-3$. The polynomial basis $\mathcal{B}^p$ is constituted by d-variate polynomials $\{\phi_{\mathbf{i}}\}_{\mathbf{i}=(i_1,i_2,\cdots,i_d)\in\mathbb{N}^d}$ such that $\|\mathbf{i}\|_1 = \sum_{j=1}^d i_j \leq p$. The number of such indices is given by Equation (3.89). Thus, these d-variate polynomials read

$$\phi_{\mathbf{i}}(\mathbf{x}) = \prod_{j=1}^{d} \phi_{i_j}^{(j)}(x_j), \tag{3.94}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_j, \ldots, x_d)$, $\mathbf{i} = (i_1, i_2, \ldots, i_j, \ldots, i_d)$ with $\|\mathbf{i}\|_1 \leq p$, and $\{\phi_i^{(j)}\}_{i\in\mathbb{N}}$ are the univariate orthonormal polynomials with respect to the law of the j-th input variable. In this setting, one needs about $\lfloor \frac{p}{2} \rfloor^d$ sampling points to exactly integrate d-variate polynomials of total order p. For complex models with high dimensional input spaces, the expansion (3.91) can be unaffordable; this is the so-called curse of dimensionality. Sparse quadrature rules based on Smolyak's algorithm can be used to circumvent this limitation [129, 163]. In practical examples though, the ground truth function F is often sparse or nearly sparse owing to a "sparsity-of-effects" principle whereby the vector $\mathbf{c} = (c_0, \ldots, c_P) \equiv (F_0, \ldots, F_P)$ of the expansion coefficients of the polynomial surrogate G_p has many negligible components [32, 121]. In these situations, they can be evaluated within the framework of compressed sensing outlined in Section 3.4.3.2. This is the approach retained in *e.g.* [39, 96, 129]; see also [61] and references therein.

3.5.2 Application to Uncertainty Quantification

Once the projection coefficients $\mathbf{c}$ have been computed and the surrogate model G_p has been chosen, several functions of interest can be derived. Indeed, the following quantity of interest $\mathcal{Q}(y)$ can be estimated as:

$$\mathbb{E}\{\mathcal{Q}(y)\} = \int_{\mathcal{X}} \mathcal{Q}(y(\mathbf{x}))\mu(d\mathbf{x}) \simeq \sum_{l=1}^{q} \omega_l \mathcal{Q}(y(\boldsymbol{\vartheta}_l)), \tag{3.95}$$

where $y \mapsto \mathcal{Q}(y)$ is a regular function. For instance, the expected value $\underline{m}$ is obtained with $\mathcal{Q}(y) = y$, the variance Var is obtained with $\mathcal{Q}(y) = (y - \underline{m})^2$, the skewness γ_1 with $\mathcal{Q}(y) = (\frac{y-\underline{m}}{\mathrm{Var}})^3$, *etc.* Using the orthonormality of the polynomial basis, Equation (3.90), the expected value $\underline{m}$ is $\underline{m} = c_0$ and the variance is $\mathrm{Var} \simeq \sum_{i=1}^P c_i^2$. Higher moments can be computed with the formulas and codes given in [130] for orthogonal polynomials of the Askey scheme. Sensitivity indices can be computed in a similar way [141]. Denoting by $\mathcal{J}_j$ the set of indices corresponding to the polynomials of the basis $\mathcal{B}^p$ which only depend on the j-th input variable of the

ground truth function F, Sobol' main-effect sensitivity indices are given by:

$$S_j = \frac{1}{\text{Var}} \sum_{i \in \mathcal{J}_j} c_i^2. \tag{3.96}$$

This can be extended to the m-fold joint sensitivity indices. For instance, denoting by $\mathcal{J}_{j_1 j_2}$ the set of indices corresponding to the polynomials of the basis $\mathcal{B}^p$ which only depend on the first and second variables of F, the two-fold joint sensitivity indices are given by:

$$S_{j_1 j_2} = \frac{1}{\text{Var}} \sum_{i \in \mathcal{J}_{j_1 j_2}} c_i^2. \tag{3.97}$$

3.6 Numerical examples

We will first study the SKRR algorithm on two different synthetic test functions:

- The three-dimensional ($d = 3$) Ishigami function in Section 3.6.1;
- The ten-dimensional ($d = 10$) Rosenbrock function in Section 3.6.2.

For these two examples, a surrogate G is built using four different methods: a fully tensorized gPC surrogate model (3.91) where the expansion coefficients are obtained by tensorized GJL quadrature nodes in (3.93), a sparse gPC surrogate model (3.91) where the expansion coefficients are obtained by solving the problem (3.85), a classical KRR surrogate model (3.12) using a Gaussian kernel (3.57), and finally a SSKRR surrogate model obtained by Algorithm 2.

The I observations of the ground truth function F (Ishigami or Rosenbrock) are obtained by a Latin Hypercube Sampling (LHS) with a minimax criterion using the package `smt` [18]. For the fully tensorized gPC surrogate model, the number of observations is chosen in order to exactly integrate the orthonormality property given by Equation (3.90) for polynomials of total order up to p. We recall that given q nodes, the GJL quadrature rule exactly integrates uni-variate polynomials of order $2q - 3$. The surrogate models are subsequently tested on a test set consisting of I_{T} observations. Therefore, we have two sets:

- The learning set which consists of I observations: $(\mathbf{X}, \mathbf{Y} = F(\mathbf{X}))$;
- The test set which consists of I_{T} observations: $(\mathbf{X}_{I_{\text{T}}}, \mathbf{Y}_{I_{\text{T}}} = F(\mathbf{X}_{I_{\text{T}}}))$.

The test set can be understood as an unseen set and it is used to validate the surrogate models. For both synthetic test functions, we replicate the four surrogate models through ten independent runs. As we are considering synthetic test functions, we are not limited in the choice of the size of the test set. Therefore, we choose $I_{\text{T}} = 1 \times 10^5$ observations taken at random. One can notice that we do not use a validation set, because we know the ground truth functions F and we assume that the test set is large enough to ensure generalization.

The performance of each surrogate model is quantified by computing the empirical Normalized Root Mean Square Error e_{NRMSE} defined by:

$$e_{\text{NRMSE}} = \sqrt{\frac{\sum_{i=1}^{I_{\text{T}}} (Y_i - G(\mathbf{X}_i))^2}{\sum_{i=1}^{I_{\text{T}}} (Y_i)^2}}, \tag{3.98}$$

and the empirical Root Mean Square Error e_{RMSE} defined by:

$$e_{\text{RMSE}} = \sqrt{\frac{1}{I_{\text{T}}} \sum_{i=1}^{I_{\text{T}}} (Y_i - G(\mathbf{X}_i))^2}. \tag{3.99}$$

The only difference between e_{NRMSE} and e_{RMSE} is that e_{RMSE} is divided by $\sum_{i=1}^{I_{\text{T}}} (Y_i)^2$ in order to remove any scaling factor of the ground truth function F. In addition, the prediction coefficient Q^2, also known as the Nash-Sutcliffe coefficient in hydrological models [109], or more generally the coefficient of determination [142], defined by:

$$\text{Q}^2 = 1 - \frac{\sum_{i=1}^{I_{\text{T}}} (Y_i - G(\mathbf{X}_i))^2}{\sum_{i=1}^{I_{\text{T}}} \left(Y_i - \frac{1}{I_{\text{T}}} \sum_{i=1}^{I_{\text{T}}} Y_i \right)^2} \tag{3.100}$$

is computed. As a reminder, a prediction coefficient Q^2 close to one indicates that the surrogate model is accurate over the I_{T} test samples. In other words, the closer Q^2 is to one, the more accurate the surrogate model is. All the three metrics above are computed on the test set. The results concerning the errors e_{RMSE} and e_{NRMSE} are presented using box plots. In more details, the central horizontal line is the median value over the ten independent runs, and the edges of the boxes correspond to the 25th q_{25} and 75th q_{75} percentiles. The circles are the outliers defined as being either smaller than $q_{25} - 1.5(q_{75} - q_{25})$, or larger than $q_{75} + 1.5(q_{75} - q_{25})$.

Here in this research, the Spectral Projected Gradient Algorithm (SPGL1) developed by van den Berg and Friedlander in `python` [152, 153] is considered in order to compute the solution of (3.85). This algorithm is based on primal-dual interior point methods. In order to tune the nugget λ of the kernel of the SSKRR surrogate (3.88) in Algorithm 2, we use a grid search algorithm on the I_{T} data points with the error e_{RMSE} as the metric. That is, we compute e_{RMSE} with respect to λ and then select the parameter $\lambda = \lambda_{\min}$ corresponding to the minimum of e_{RMSE}. This can be done because the ground truth function F is known and inexpensive to evaluate. In addition, the parameter κ of the kernel of the SSKRR surrogate is set to $\kappa = \text{Var}(\mathbf{Y})$. Finally, the parameters $\boldsymbol{\theta} = (\lambda_{\text{KRR}}, \{\gamma_i\}_{i=1}^d)$ of the Gaussian kernel of the KRR surrogate are determined using the parametric KF algorithm presented in Section 3.3.4.2.

3.6.1 Ishigami function

The Ishigami function [68] is commonly used for benchmarking global sensitivity analyses and uncertainty quantification. The analytic expression of this three-dimensional function is:

$$F(\mathbf{X}) = \sin(X_1) + a\sin^2(X_2) + bX_3^4\sin(X_1), \tag{3.101}$$

with $a = 7$, $b = 0.1$ [93], and $\mathbf{X} = (X_1, X_2, X_3) \in [-\pi, \pi]^3$. The input variables $\mathbf{X}$ are assumed to be mutually independent and to follow an uniform law:

$$X_i \sim \mathcal{U}(-\pi, \pi) \text{ for } i = 1, 2, 3. \tag{3.102}$$

The expected value and variance of F are given by:

$$\mathbb{E}(F) = \frac{a}{2}, \tag{3.103}$$

and

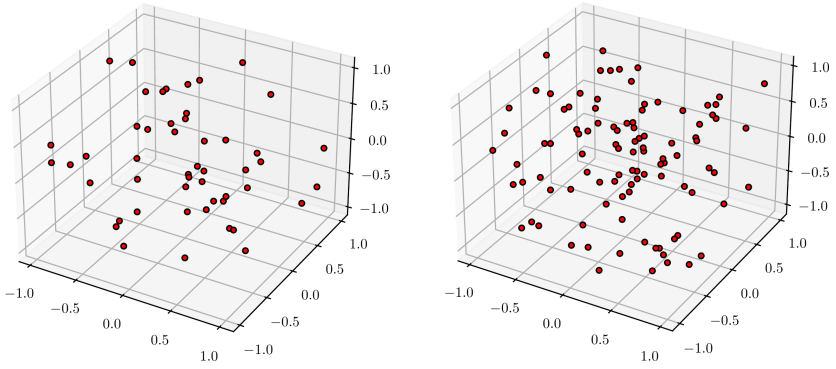
$$\mathrm{Var}(F) = \frac{1}{2} + \frac{a^2}{8} + \frac{b^2\pi^8}{18} + \frac{b\pi^4}{2}. \tag{3.104}$$

We select polynomials of total order up to $p = 10$ to form the polynomial basis $\mathcal{B}^p$, which corresponds to $\binom{p+d}{d} = \binom{10+3}{3} = 286$ multi-dimensional Legendre polynomials. $\mathcal{B}^p$ is considered for the construction of the fully tensorized gPC, the sparse gPC, and the SSKRR surrogates. For the fully tensorized gPC surrogate model, $N = 12^3 = 1728$ GJL quadrature nodes are needed to exactly recover the orthonormality property given by Equation (3.90) since we have chosen a total order up to $p = 10$. For the sparse gPC and the SSKRR surrogates, two learning sets with $I = 50$ and $I = 100$ observations of the ground truth function F are considered to test the influence of I on the recovery of the expansion coefficients by the BPDN minimization (3.85). The values $I = 50$ and $I = 100$ were chosen because they are significantly lower than the size of the polynomial basis $\mathcal{B}^p$. Also $\eta = 1 \times 10^{-6}$ has been chosen in (3.85). Finally, the KRR surrogate is built using the learning set with $I = 100$ observations.

3.6.1.1 Sparsity on Legendre polynomials

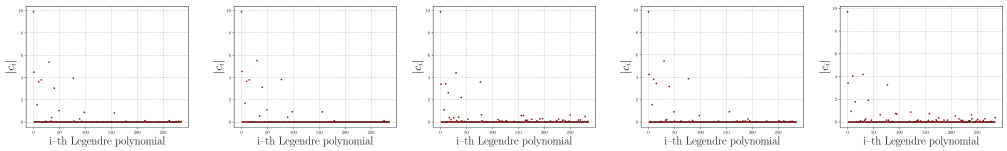
The first step is to determine the sparsity as it is observed *a posteriori*. In that respect, we increase the number of observations in the learning set from $I = 50$ to $I = 100$ and keep the expansion coefficients $\mathbf{c}^\star$ solution of Equation (3.85) that do not significantly change over ten independent runs of the positions of the observations in either set. One run of the positions of the observations can be seen on Figure 3.7 for $I = 50$ and $I = 100$ observations of F.

The evolution of the expansion coefficients $\mathbf{c}^\star$ with respect to the random sampling of the positions of the observations are shown on Figure 3.8 and Figure 3.9, with $I = 50$ and $I = 100$ observations of F respectively. One can notice that for $I = 50$, the coefficients $\mathbf{c}^\star$ are fluctuating greatly from one sampling to another; see for instance

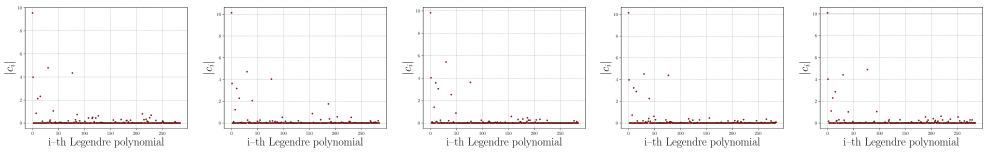


(a) $I = 50$ observations. (b) $I = 100$ observations.

Figure 3.7: One run of the LHS positions with $I = 50$ and $I = 100$ observations of F.



(a) Seed 1. (b) Seed 2. (c) Seed 3. (d) Seed 4. (e) Seed 5.



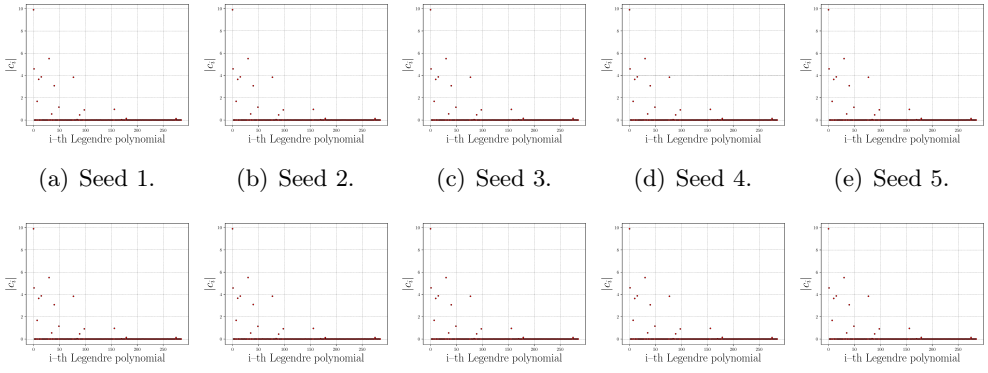
(f) Seed 6. (g) Seed 7. (h) Seed 8. (i) Seed 9. (j) Seed 10.

Figure 3.8: Evolution of the expansion coefficients $\mathbf{c}^\star$ with respect to the random samplings of the positions of the observations with $I = 50$ observations of F.

the difference between Figure 3.8(a) and Figure 3.8(c). Now, look at Figure 3.9, where $I = 100$ observations. From one sampling to another, one can notice that the coefficients $\mathbf{c}^\star$ are similar, and that the sparsity is here $S \approx 15$. The average of the coherence parameter $\mathrm{C}(\boldsymbol{\Theta})$ over the ten samplings is then $\mathrm{C}(\boldsymbol{\Theta}) \approx 5.89$. Thus, Theorem 6 yields $I \gtrsim I_{\mathrm{s}} \approx 500$ observations up to a constant C. Note that such a sparsity was expected as the sine function can be well approximated by polynomials. From now on we choose the learning set with $I = 100$ observations of F to construct its sparse gPC, KRR, and SSKRR surrogate models.

3.6.1.2 Comparisons between the surrogate models

In this section, we compare the performance of each surrogate model using the ten independent runs with $I_{\mathrm{T}} = 1 \times 10^5$ samples in the test set. A comparison of the RMSE error e_{RMSE} and the NRMSE error e_{NRMSE} over ten independent runs



(a) Seed 1. (b) Seed 2. (c) Seed 3. (d) Seed 4. (e) Seed 5.

(f) Seed 6. (g) Seed 7. (h) Seed 8. (i) Seed 9. (j) Seed 10.

Figure 3.9: Evolution of the expansion coefficients $\mathbf{c}^\star$ with respect to the random samplings of the positions of the observations with $I = 100$ observations of F.

is shown on Figure 3.10 and Figure 3.11. The prediction coefficient Q^2 for each surrogate model is given on Table 3.4. One can see that the surrogate model obtained by the SSKRR Algorithm 2 performs slightly better than the sparse gPC surrogate model while the KRR surrogate model performs way worse than the others. The fully tensorized gPC surrogate model performs slightly better than the SSKRR surrogate model but at a much higher computational cost. Indeed, we only used $I = 100$ observations of F to obtain the expansion coefficients $\mathbf{c}^\star$ by ℓ_1-minimization while $I = 1728$ observations of F were needed to obtain the expansion coefficients for the fully tensorized gPC surrogate model. Notice the circles on Figure 3.10 and Figure 3.11: they are outliers. It shows that even with $I = 100$ observations of F, one can still deviate from recovering the true expansion coefficients. Moreover, the low values of the errors e_{RMSE} and e_{NRMSE} of both approaches can be explained by the fact that the Ishigami function is a smooth function consisting of sine functions, which can be well approximated by polynomials over a bounded domain.

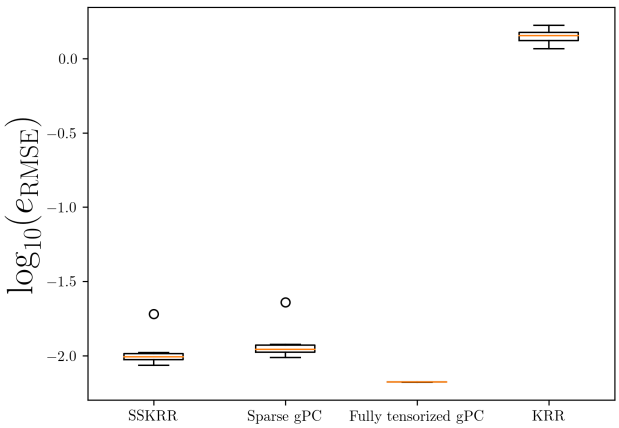


Figure 3.10: The empirical root mean square error e_{RMSE} over the ten independent runs.

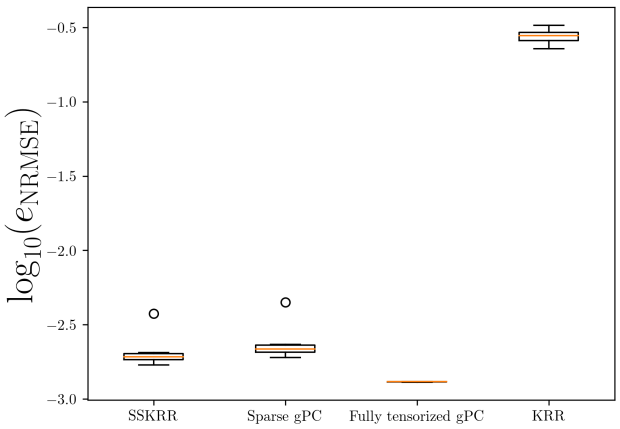


Figure 3.11: The empirical normalized root mean square error e_{NRMSE} over the ten independent runs.

Ishigami function			
	Median	Minimum	Maximum
Fully tensorized gPC	0.999997	0.999997	0.999997
Sparse gPC	0.999991	0.999962	0.999993
KRR	0.853325	0.797717	0.902461
SSKRR	0.999993	0.999973	0.999995

Table 3.4: Prediction coefficient Q^2 over the ten independent runs.

Now, we compare several quantities of interest of the surrogate models using the test set over the ten independent runs; namely the expected value, the variance, and the Kullback-Leibler (KL) divergence. The expected values and the variances of the fully tensorized and sparse gPC surrogate models are obtained using the expansion coefficients directly. The KL divergence D_{KL} is computed between each surrogate model and the ground truth function F by first estimating the PDF from the $I_{\text{T}} = 1 \times 10^5$ observations and then smoothing out the resulting histograms by a normal kernel density function [156]. An example of such a PDF obtained for one run is shown on Figure 3.12. The comparison between the surrogate models can be seen on Table 3.5. One can notice that we obtain roughly the same expected values, variances, and KL divergences with the SSKRR, the sparse gPC, and the fully tensorized gPC surrogate models. These values are close to the exact values; see the last column. Nevertheless, for probabilistic quantities of interest such as the expected value or the variance, the gPC surrogate models perform better because these quantities of interest can be directly computed from the expansion coefficients, see Section 3.5.2; while they were estimated from a Monte-Carlo simulation for the KRR and SSKRR surrogates. The KRR surrogate with Gaussian kernel gives the worst results as hinted by the errors e_{RMSE} and e_{NRMSE} computed previously.

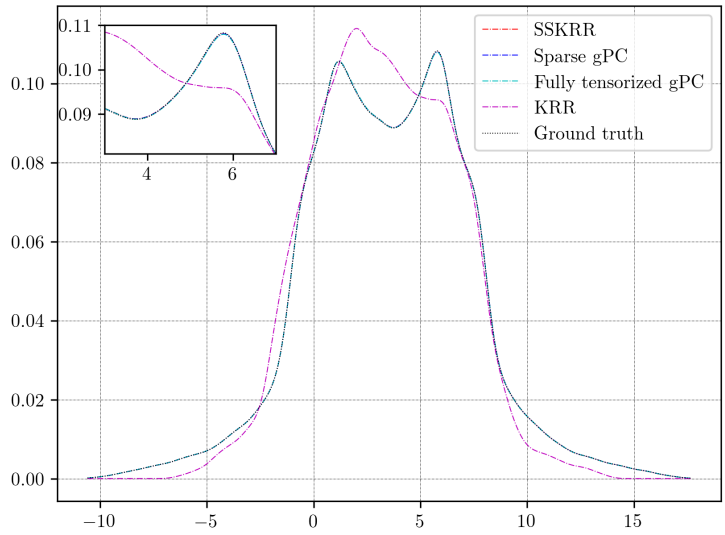


Figure 3.12: PDFs obtained for one run with $I = 100$ observations of F. The PDFs were obtained from the $I_T = 1 \times 10^5$ data points and then smoothing out by a normal kernel density function.

Ishigami function				
Expected values				
	Mean	Minimum	Maximum	Exact
Fully tensorized gPC	3.500	3.500	3.500	3.500
Sparse gPC	3.500	3.498	3.501	3.500
KRR	3.451	3.292	3.760	3.500
SSKRR	3.501	3.490	3.516	3.500
Variance				
Fully tensorized gPC	13.844	13.844	13.844	13.845
Sparse gPC	13.819	13.804	13.834	13.845
KRR	11.211	10.198	12.165	13.845
SSKRR	13.775	13.590	13.902	13.845
Kullback-Leibler divergence				
Fully tensorized gPC	2.126×10^{-6}	2.037×10^{-6}	2.176×10^{-6}	–
Sparse gPC	1.512×10^{-6}	9.276×10^{-7}	2.752×10^{-6}	–
KRR	0.0186	0.0134	0.0314	–
SSKRR	1.735×10^{-6}	1.033×10^{-6}	2.562×10^{-6}	–

Table 3.5: Expected value, variance, and KL divergence over the ten independent runs for the different surrogate models.

3.6.2 Rosenbrock function

The Rosenbrock function is a analytical function widely used in benchmarks for optimization [125]. It is non-convex and reads

$$F(\mathbf{X}) = \sum_{i=1}^{d-1} \left[100 \left(X_{i+1} - X_i^2 \right)^2 + (1 - X_i)^2 \right] \tag{3.105}$$

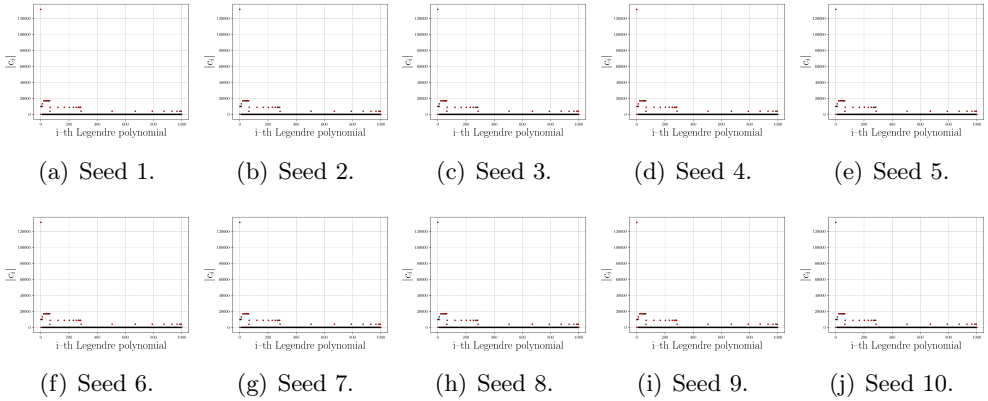
with $d = 10$ and $\mathbf{X} \in [-2, 2]^{10}$. The input variables $\mathbf{X}$ are assumed to be mutually independent random variables following an uniform law:

$$X_i \sim \mathcal{U}(-2, 2) \text{ for } i = 1, \ldots, 10. \tag{3.106}$$

We proceed as in Section 3.6.1. We select polynomials of total order up to $p = 4$ to form the polynomial basis $\mathcal{B}^p$, which corresponds to $\binom{p+d}{d} = \binom{4+10}{10} = 1001$ multi-dimensional Legendre polynomials. $\mathcal{B}^p$ is considered for the construction of the fully tensorized gPC, the sparse gPC, and the SSKRR surrogates. For the fully tensorized gPC surrogate model, $N = 6^{10} = 60,466,176$ GJL quadrature nodes are needed to exactly recover the orthonormality property given by Equation (3.90) since we have chosen a total order up to $p = 4$. The fully tensorized gPC surrogate model is not douable due to the numbers of points needed: this is the curse of dimensionality invoked in Section 3.5.1. Therefore this surrogate model will not be considered for this example. For the sparse gPC and the SSKRR surrogates, one learning set with $I = 400$ observations of the ground truth function F is considered. The value $I = 400$ was chosen because it is significantly lower than the size of the polynomial basis $\mathcal{B}^p$. Also $\eta = 1 \times 10^{-6}$ has been chosen in (3.85). Finally, the KRR surrogate is built using the same learning set.

3.6.2.1 Sparsity on Legendre Polynomials

To determine the sparsity S needed to obtain a lower bound on the number of observations required to have a successful recovery of the expansion coefficients, we gradually increase the number of observations and stop when the solution $\mathbf{c}^\star$ of Equation (3.85) do not change significantly over ten independent runs of the positions of the observations. The foregoing study is carried out and we find that $I = 400$ observations are enough to observe sparsity over these ten runs. This justifies our choice of the size of the learning set picked in the previous section. The evolution of the expansion coefficients $\mathbf{c}^\star$ with respect to the random samplings of the positions of the observations are shown on Figure 3.13. Here one can see that the sparsity is $S \approx 38$. One can notice this $S = 38$ is exactly the number of terms in the expression of the Rosenbrock function. Then we find that the average of the coherence parameter $\mathrm{C}(\mathbf{\Theta})$ over the ten runs is $\mathrm{C}(\mathbf{\Theta}) \approx 0.06$. Theorem 6 thus yields $I \gtrsim I_{\mathrm{s}} \approx 16$ observations up to a constant C.



(a) Seed 1. (b) Seed 2. (c) Seed 3. (d) Seed 4. (e) Seed 5.

(f) Seed 6. (g) Seed 7. (h) Seed 8. (i) Seed 9. (j) Seed 10.

Figure 3.13: Evolution of the expansion coefficients $\mathbf{c}^\star$ *with respect to the random samplings of the positions of the observations with* $I = 400$ *observations of* F*.*

3.6.2.2 Comparison between the surrogate models

In this section, we compare the performance of the sparse gPC, KRR, and SSKRR surrogate models with the $I_{\mathrm{T}} = 1 \times 10^5$ observations in the test set over ten independent runs. Here the fully tensorized gPC surrogate model is not feasible due to the considerable number of quadrature nodes needed. A comparison of the RMSE error e_{RMSE} and the NRMSE error e_{NRMSE} over the ten independent runs is shown on Figure 3.14 and Figure 3.15. The prediction coefficient Q^2 for each surrogate model is given in Table 3.6. Notice that the prediction coefficient Q^2 is sometimes negative for the KRR surrogate model with Gaussian kernel. It means in this case that the mean of the data provides a better approximation than the KRR surrogate. One can see that the SSKRR surrogate obtained by our algorithm performs better than the sparse gPC surrogate while the KRR surrogate performs way worse than the others. One can notice that only 400 observations of F are enough to obtain the nearly exact expansion coefficients $\mathbf{c}^\star$ by ℓ_1-minimization while $N = 6^{10}$ observations of F would have been needed to obtain them through the fully tensorized method. Moreover, the very low values of e_{RMSE} and e_{NRMSE} of the sparse gPC and SSKRR approaches can be explained by the fact that the Rosenbrock function is a polynomial expansion. These values are close to machine precision.

Rosenbrock function			
	Median	Minimum	Maximum
Fully tensorized gPC	–	–	–
Sparse gPC	≈ 1.00000	≈ 1.00000	≈ 1.00000
KRR	-0.17561	-0.61829	0.27424
SSKRR	≈ 1.00000	≈ 1.00000	≈ 1.00000

Table 3.6: Prediction coefficient Q^2 *over the ten independent runs.*

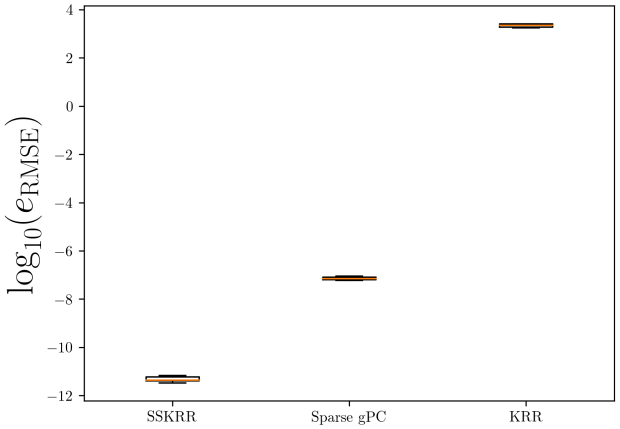


Figure 3.14: The empirical root mean square error e_{RMSE} over the ten independent runs.

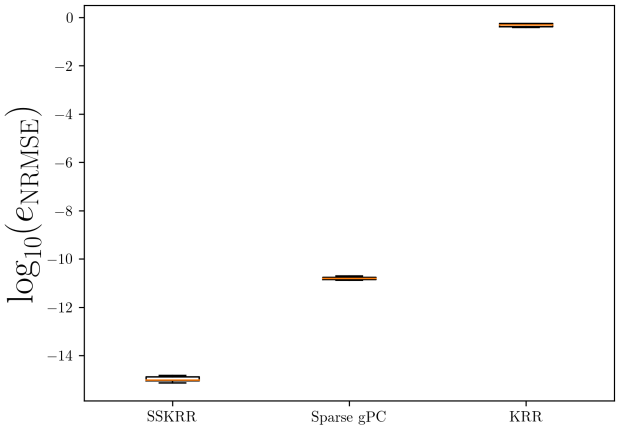


Figure 3.15: The empirical normalized root mean square error e_{NRMSE} over the ten independent runs.

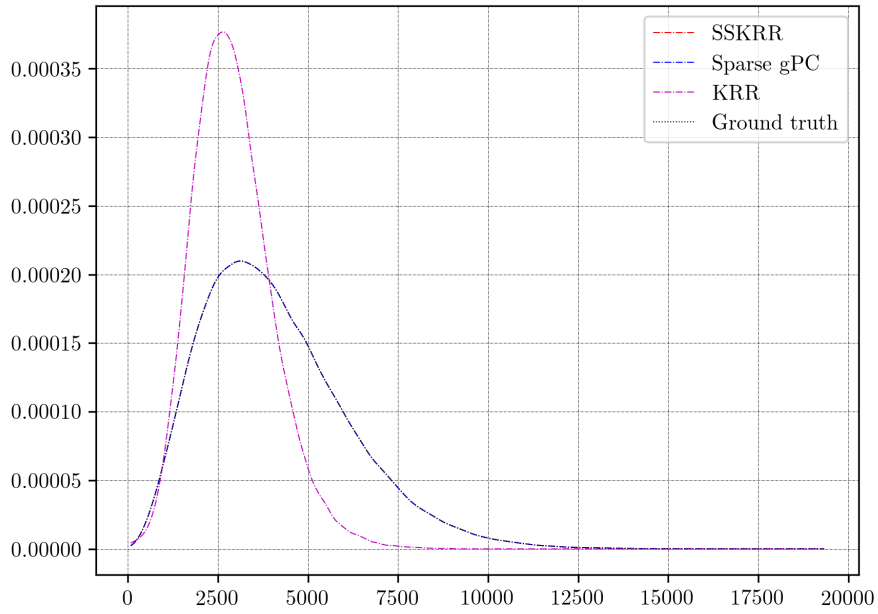


Figure 3.16: PDFs obtained for one run with $I = 400$ observations of F. The PDFs were obtained from the $I_{\mathrm{T}} = 1 \times 10^5$ data points and then smoothing out by a normal kernel density function.

Now, we compare several quantities of interest of the surrogate models using the test set over the ten independent runs; namely the expected value, the variance, and the KL divergence. The expected values and the variances of the sparse gPC surrogate models are obtained using the expansion coefficients directly. The KL divergence D_{KL} is computed between each surrogate model and the ground truth function F by first estimating the PDF from the $I_{\mathrm{T}} = 1 \times 10^5$ observations and then smoothing out the resulting histograms by a normal kernel density function [156]. An example of such a PDF obtained for one run is shown on Figure 3.16. The comparison between the surrogate models can be seen on Table 3.7. The exact results are computed by taking the mean of the ground truth function F on the test set for the ten independent runs. One can notice that we obtain similar expected values, variances, and KL divergences with the SSKRR surrogate compared to the sparse gPC surrogate. It has to be noted that we only needed 400 observations to obtain the nearly exact expansion coefficients $\mathbf{c}$. The KRR surrogate with Gaussian kernel gives the worst results as hinted by the errors e_{RMSE} and e_{NRMSE} computed previously.

3.7 Application to the RAE2822 transonic airfoil

We apply the same framework as in Section 3.6 on a complex aerodynamic test case: the two-dimensional RAE2822 airfoil of Figure 1.10 [34]. The quantities of interest are the lift coefficient C_L, the drag coefficient C_D, and the pitching moment

Rosenbrock function				
Expected values				
	Mean	Minimum	Maximum	Exact
Fully tensorized gPC	–	–	–	4101.391
Sparse gPC	4101.000	4101.000	4101.000	4101.391
KRR	3162.772	2847.550	3518.393	4101.391
SSKRR	4101.391	4089.939	4122.100	4101.391
Variance				
Fully tensorized gPC	–	–	–	4092724.179
Sparse gPC	4105081.752	4105081.752	4105081.752	4092724.179
KRR	1484933.564	1128976.833	1811692.945	4092724.179
SSKRR	4092724.179	4046013.350	4133221.629	4092724.179
Kullback-Leibler divergence				
Fully tensorized gPC	–	–	–	–
Sparse gPC	≈ 0	≈ 0	≈ 0	–
KRR	0.198	0.142	0.323	–
SSKRR	≈ 0	≈ 0	≈ 0	–

Table 3.7: Expected value, variance and KL divergence over the ten independent runs for the different surrogate models.

coefficient C_M as functions of three random input variables: the free-stream Mach number M, the angle of attack α of the airfoil, and the thickness-to-chord ratio r. We compare the performances of three surrogate modeling methods: fully tensorized gPC, sparse gPC, and SSKRR as sketched in Algorithm 2.

3.7.1 Problem setup

The quantities of interest are obtained by solving the steady-state Reynolds-Averaged Navier-Stokes (RANS) equations together with a Spalart-Allmaras turbulence model closure [136]. The CFD solver *elsA* [20] is used to simulate two-dimensional transonic flows around that airfoil and construct the learning set. The nominal flow conditions correspond to the ones described in [34, Test case #6] together with the correction formulas for the wall interference derived in [53, pp. 386–387], and their slight modifications proposed in [59]. The operational parameters considered here are thus $\underline{M} = 0.729$ for the free-stream Mach number, $\underline{\alpha} = 2.31°$ for the angle of attack, and $\underline{\mathrm{Re}} = 6.50 \times 10^6$ for the Reynolds number based on the chord length c, fluid velocity, temperature, and molecular viscosity at infinity. They arise from the corrections $\Delta M = 0.004$ and $\Delta\alpha = -0.61°$ given in [59, pp. 130] for the test case #6 outlined in [34], for which $M = 0.725$, $\alpha = 2.92°$, and $\mathrm{Re} = 6.50 \times 10^6$. More details about this example and the numerical parameters used for *elsA* runs can be found in [129]. Denoting by $(X_1, X_2, X_3) = (r, M, \alpha)$ the random input variables,

they are mutually independent and follow Beta laws of the first kind $\beta_I(a,b)$:

$$\beta_I(x,a,b) = \mathbb{1}_{[X_{\text{lb}},X_{\text{ub}}]}(x)\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\frac{(x-X_{\text{lb}})^{a-1}(X_{\text{ub}}-x)^{b-1}}{(X_{\text{lb}}-X_{\text{ub}})^{a+b-1}},$$

where $a=(4,4,4)$, $b=(4,4,4)$, and $[X_{\text{lb}},X_{\text{ub}}]$ is the compact support of the random parameter $X\sim\beta_I$. Table 3.8 gathers the range of each random input variables and their associated parameters (a,b).

	X_{lb}	X_{ub}	(a,b)
$X_1 = r$	$0.97\times\underline{r}$	$1.03\times\underline{r}$	$(4,4)$
$X_2 = M$	$0.95\times\underline{M}$	$1.05\times\underline{M}$	$(4,4)$
$X_3 = \alpha$	$0.98\times\underline{\alpha}$	$1.02\times\underline{\alpha}$	$(4,4)$

Table 3.8: Range and probability law of each input parameter, with $\underline{r}=1$.

For this example, surrogates G are built for the quantities of interest C_L, C_D, and C_M using three different methods: a fully tensorized gPC surrogate model (3.91) where the expansion coefficients are obtained by tensorized GJL quadrature nodes in (3.93), a sparse gPC surrogate model (3.91) where the expansion coefficients are obtained by solving the problem (3.85), and a SSKRR surrogate model obtained by Algorithm 2.

The I observations of the quantities of interest used to construct the sparse gPC and SSKRR surrogates are obtained by random trials following the Beta laws of the random input variables. For the fully tensorized gPC surrogate, the number of observations is chosen in order to exactly integrate the orthonormality property given by Equation (3.90) for polynomials of total order up to p. We recall that given q nodes, the GJL quadrature rule exactly integrates uni-variate polynomials of order $2q-3$. The surrogates are subsequently validated on a validation set consisting of I_{V} observations, and tested on a test set consisting of I_{T} observations where the input variables are again drawn randomly following the Beta laws with parameters as in Table 3.8. These $I_{\text{Tot}} = I + I_{\text{V}} + I_{\text{T}}$ observations are thus split as follows:

- The learning set which consists of $I = 80$ observations that is 67% of the I_{Tot} observations: $(\mathbf{X}, C_L(\mathbf{X}), C_D(\mathbf{X}), C_M(\mathbf{X}))$;
- The validation set which consists of $I_{\text{V}} = 15$ observations that is 12% of the I_{Tot} observations: $(\mathbf{X}_{I_{\text{V}}}, C_L(\mathbf{X}_{I_{\text{V}}}), C_D(\mathbf{X}_{I_{\text{V}}}), C_M(\mathbf{X}_{I_{\text{V}}}))$;
- The test set which consists of $I_{\text{T}} = 25$ observations that is 21% of the I_{Tot} observations: $(\mathbf{X}_{I_{\text{T}}}, C_L(\mathbf{X}_{I_{\text{T}}}), C_D(\mathbf{X}_{I_{\text{T}}}), C_M(\mathbf{X}_{I_{\text{T}}}))$.

These different sets are shown on Figure 3.17. Other splitting choices could have been made, for instance the classical 60/20/20 splitting (60% for the learning set, 20% for the validation set, 20% for the test set) as in [65].

The performance of each surrogate model is quantified by computing the empirical Normalized Root Mean Square Error e_{NRMSE} of Equation (3.98) and the empirical Root Mean Square Error e_{RMSE} of Equation (3.99) using the validation and test sets. However the knowledge of e_{RMSE} and e_{NRMSE} might not be enough to assess the performance of a surrogate model. Indeed, e_{RMSE} only gives the global error over the whole domain but does not give any information about the distribution. For instance, two similar values of e_{RMSE} for two different surrogate models can be obtained: in one case the surrogate provides a reliable approximation of the ground truth function for the majority of the domain but a poor one for a few points, while in another case the other surrogate provides a less reliable approximation of the ground truth function in the entire domain. In that respect, we compute an additional metric, the maximum relative error e_{MRE} defined by

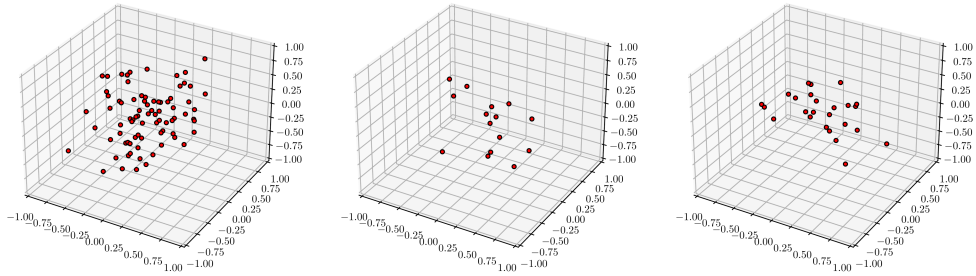
$$e_{\text{MRE}} = \max_{i=1,\dots,I_{\text{T}}} \left(\frac{|Y_i - G(\mathbf{X}_i)|}{|Y_i|} \right) \times 100. \tag{3.107}$$

Following [129], we choose a total order up to $p = 8$ which corresponds to $\binom{p+d}{d} = \binom{8+3}{3} = 165$ multi-dimensional Jacobi polynomials. They constitute the basis $\mathcal{B}^p$ considered for the construction of the fully tensorized gPC, the sparse gPC, and the SSKRR surrogates. Since $p = 8$, $N = 1000$ GJL quadature nodes are needed to exactly recover the orthonormality property given by Equation (3.90) and are selected to compute the expansion coefficients of the fully tensorized gPC surrogate by Equation (3.93). SPGL1 in `python` [152, 153] is again considered in order to compute the solution of (3.85) for the expansion coefficients in the sparse gPC surrogate (3.91) and the SSKRR surrogate (3.88) obtained by Algorithm 2. Also $\eta = 1 \times 10^{-5}$ has been chosen in (3.85). The coherence parameter of Definition 10 is $\mathrm{C}(\boldsymbol{\Theta}) \approx 270$ on the learning set which yields a large theoretical lower bound in Theorem 6. We observe however that the $I = 80$ observations of the learning set yield satisfactory results below. In order to tune the nugget λ of the kernel of the SSKRR surrogate (3.88), we use the parametric KF algorithm of Section 3.3.3 [31, 114] on the I_{V} data points with e_{RMSE} as the metric, $I_{\text{f}} = I$ and $I_{\text{c}} = I_{\text{f}}/2$, and the accuracy ρ defined by Equation (3.64). In addition, the trace parameter κ is chosen as $\kappa = \mathrm{Var}(\mathbf{Y})$.

A pick-freeze estimator [69, 119] is used to compute Sobol' main-effect sensitivity indices obtained from the SSKRR surrogate. A matrix size of 1×10^6 samples is selected, corresponding to a total number of $(d+1) \times 10^6 = 4 \times 10^6$ evaluations of the surrogate model. This method is a Monte-Carlo based one and thus it may be difficult to obtain accurate estimates of small sensitivity indices. On the other hand, these indices are directly obtained from the expansion coefficients of the fully tensorized and sparse gPC surrogates; see Section 3.5.2.

3.7.2 Lift coefficient C_L

We first focus on the lift coefficient C_L. The expansion coefficients $\mathbf{c}^\star$ given by ℓ_1-minimization are shown on Figure 3.18 using the learning set of Figure 3.17(a).



(a) (b) (c)

Figure 3.17: Random sampling points used for the learning set (a), the validation test (b), and the test set (c) with $I = 80$, $I_{\mathrm{V}} = 15$, and $I_{\mathrm{T}} = 25$ points to compute the expansion coefficients of C_D, C_L and C_M by ℓ_1-minimization.

We notice that only low order polynomials are relevant. Indeed, one can see that:

$$G_p(\mathbf{x}) \approx c_1^\star \phi_{(0,0,0)}(\mathbf{x}) + c_2^\star \phi_{(1,0,0)}(\mathbf{x}) + c_3^\star \phi_{(0,1,0)}(\mathbf{x}) + c_4^\star \phi_{(0,0,1)}(\mathbf{x}) + c_6^\star \phi_{(1,1,0)}(\mathbf{x}) + c_8^\star \phi_{(0,2,0)}(\mathbf{x}) + c_{17}^\star \phi_{(0,3,0)}(\mathbf{x}), \quad (3.108)$$

where $\phi_{(i_1,i_2,i_3)}$ is defined as in Equation (3.94) and with $|c_1^\star| \gg |c_3^\star| \gg |c_4^\star| \approx |c_6^\star| \approx |c_8^\star| \approx |c_{17}^\star| \gg |c_2^\star|$. The highest order polynomial of $G_p(\mathbf{x})$ has order 3 and its expansion coefficient is small compared to the others. From Figure 3.18, the sparsity is observed to be $S \approx 7$. Therefore Theorem 6 yields $N \gtrsim I_{\mathrm{s}} \approx 10000$ observations up to a constant C, but we observe in practice that the $I = 80$ observations are enough for an accurate recovery.

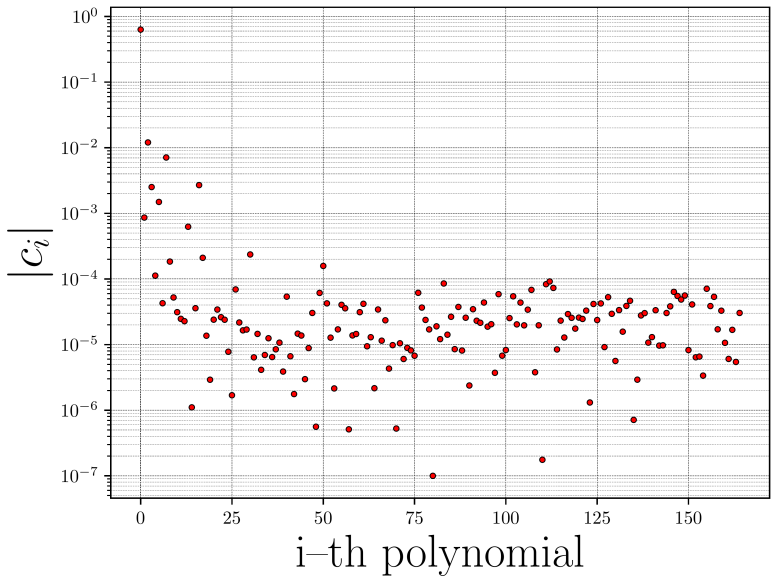
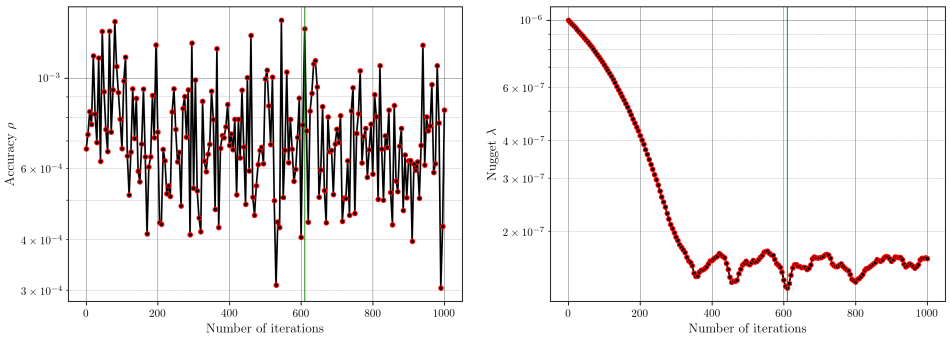
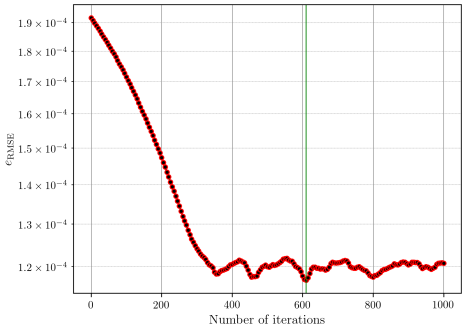


Figure 3.18: Expansion coefficients $\mathbf{c}^\star$ with $I = 80$ observations of the lift coefficient C_L.



(a) Accuracy ρ.

(b) Nugget λ.



(c) Error e_{RMSE} on the validation set.

Figure 3.19: Evolutions of the accuracy ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as functions of the number of iterations of the parametric KF algorithm for the lift coefficient C_L. The green vertical line corresponds to the iteration where the error e_{RMSE} is minimal on the validation set.

Lift coefficient C_L			
	SSKRR	Sparse gPC	Fully tensorized gPC
e_{RMSE}	7.574×10^{-5}	3.715×10^{-4}	1.159×10^{-4}
e_{NRMSE}	1.040×10^{-4}	5.103×10^{-4}	8.437×10^{-5}
e_{MRE}	0.0319%	0.232%	0.0368%
Q^2	0.99996	0.99911	0.99995

Table 3.9: Comparison of the errors between the surrogate models for the lift coefficient C_L with $I = 80$ and $I_{\mathrm{T}} = 25$ sampling points.

The evolutions of the accurary ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as functions of the number of iterations of the parametric KF algorithm, are shown on Figure 3.19. We find $\lambda_{\min} = 1.30 \times 10^{-7}$, which corresponds to the green vertical line on Figure 3.19. One can notice that for the initial choice of λ, the accuracy ρ is already very small (about 1×10^{-3}).

Therefore, the decrease in error e_{RMSE} on the validation set is marginal and the changes in λ are not substantial. The comparison of the errors e_{RMSE}, e_{NRMSE} and e_{MRE} between the surrogate models on the test set are given in Table 3.9 where $\lambda = \lambda_{\min}$. Figure 3.20 shows the values of the lift coefficient C_L with respect to the input parameters on the learning set, the verification set, the test set, and the predictions of the SSKRR surrogate. A strong non-linear dependence between C_L and the Mach number M can be seen on Figure 3.20. The PDFs of C_L using the three surrogate models are estimated from $I_s = 1 \times 10^6$ random data points taken at random following the Beta laws of Table 3.8 and then smoothing out the resulting histrograms by a normal kernel density function [156]. They are shown on Figure 3.21, together with their corresponding expected value and variance. The expected values of the PDFs obtained by each surrogate model are shown on Figure 3.21 with vertical lines. Notice that we obtain comparable results except at the tails of the distributions and at their peaks. Finally, Sobol' main-effect sensitivity indices are gathered in Table 3.10. As expected from the previous results, the variable $X_2 = M$ is more influential than the variables $X_1 = r$ and $X_3 = \alpha$ where $X_1 = r$ has almost no influence on C_L. The SSKRR surrogate slightly outperforms the fully tensorized one. Both have a much better performance than the sparse gPC surrogate.

Lift coefficient C_L			
	$X_1 = h/c$	$X_2 = M$	$X_3 = \alpha$
Fully tensorized gPC	0.00345	0.955	0.0286
Sparse gPC	0.00358	0.953	0.0298
SSKRR	0.00425	0.956	0.0296

Table 3.10: Sobol' main-effect sensitivity indices of the lift coefficient C_L with $I = 80$ samplings points. For our algorithm, they were estimated from a pick-freeze estimator with a matrix size of 1×10^6 samples. For the sparse and the fully tensorized gPC surrogate models, they were estimated from the values of the expansion coefficients using Section 3.5.2.

3.7.3 Drag coefficient C_D

We now focus on the drag coefficient C_D. The expansion coefficients $\mathbf{c}^\star$ given by the ℓ_1-minimization are shown on Figure 3.22 using the learning set on Figure 3.17(a). In a same way as for the lift coefficient C_L, we notice that only a few polynomials are relevant in the expansion. One has:

$$\begin{aligned} G_p(\mathbf{x}) \approx\ & c_1^\star \phi_{(0,0,0)}(\mathbf{x}) + c_2^\star \phi_{(1,0,0)}(\mathbf{x}) + c_3^\star \phi_{(0,1,0)}(\mathbf{x}) + c_4^\star \phi_{(0,0,1)}(\mathbf{x}) + c_6^\star \phi_{(1,1,0)}(\mathbf{x}) \\ & + c_8^\star \phi_{(0,2,0)}(\mathbf{x}) + c_9^\star \phi_{(0,1,1)}(\mathbf{x}) + c_{14}^\star \phi_{(1,2,0)}(\mathbf{x}) + c_{17}^\star \phi_{(0,3,0)}(\mathbf{x}) \\ & + c_{27}^\star \phi_{(1,3,0)}(\mathbf{x}) + c_{31}^\star \phi_{(0,4,0)}(\mathbf{x}) + c_{78}^\star \phi_{(0,6,0)}(\mathbf{x}), \end{aligned} \tag{3.109}$$

where $\phi_{(i_1,i_2,i_3)}$ is defined as in Equation (3.94) and with $|c_1^\star| \gg |c_3^\star| \gg |c_2^\star| \approx |c_6^\star| \approx |c_8^\star| \approx |c_{31}^\star| \gg |c_4^\star| \approx |c_9^\star| \approx |c_{14}^\star| \approx |c_{17}^\star| \approx |c_{27}^\star| \approx |c_{78}^\star|$. The highest order polynomial

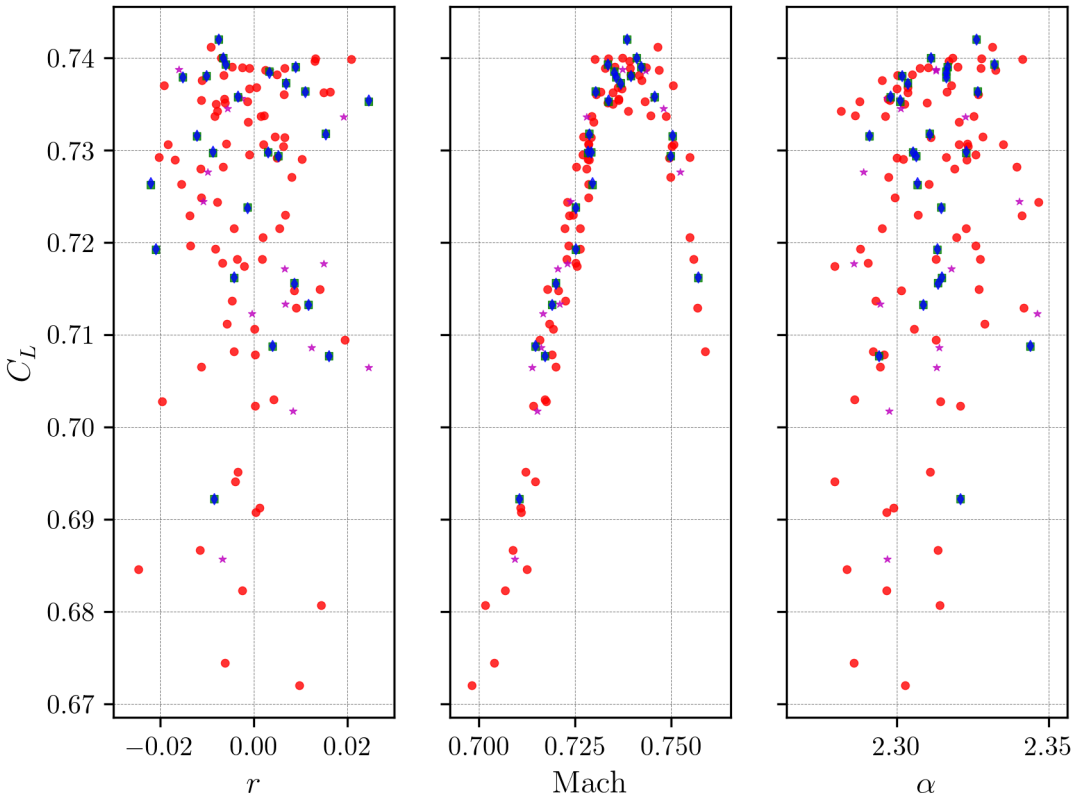
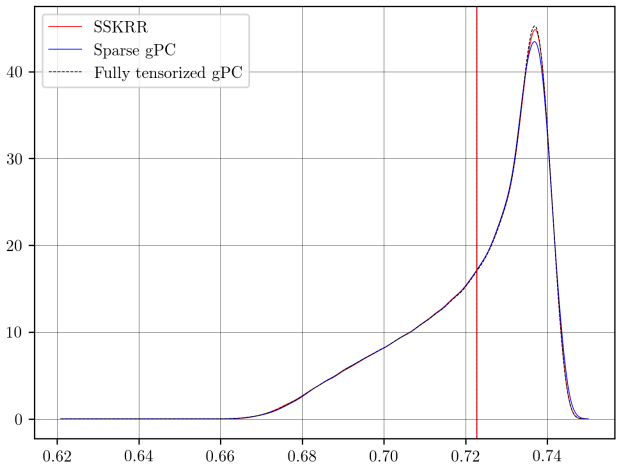


Figure 3.20: Distribution of the difference between the prediction given by SSKRR surrogate and the observations on the test set for the lift coefficient C_L with $I = 80$ and $I_T = 25$ samplings points. The red circles are the observations defining the learning set. The violet stars are the observations defining the validation set. The green squares are the observations defining the test set with their corresponding predictions given by the SSKRR surrogate, depicted by the blue diamonds.

of $G_p(\mathbf{x})$ has order 6 and its expansion coefficient is very small compared to the others. From Figure 3.22, the sparsity is observed to be $S \approx 12$. Therefore Theorem 6 yields $N \gtrsim I \approx 17000$ observations up to a constant C, but we observe in practice that the $I = 80$ observations are enough for an accurate recovery.

The evolutions of the accuracy ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as functions of the number of iterations of the parametric KF algorithm are shown on Figure 3.23. We find $\lambda_{\min} = 3.75 \times 10^{-8}$, which corresponds to the green vertical line on Figure 3.23. Contrary to the lift coefficient C_L, one can notice that the accuracy ρ was greatly reduced (divided by about 10) between the initial and convergence iterations. In this way, e_{RMSE} was divided by about 7 on the validation set. The comparison of the errors e_{RMSE}, e_{NRMSE}, and e_{MRE} between the surrogate models are given in Table 3.11 where $\lambda = \lambda_{\min}$. Figure 3.24 shows the values of the drag coefficient C_D with respect to the input parameters on the learning set, the verification set, the test set, and the



(a) PDFs of the lift coefficient C_L. The vertical lines correspond to their respective expected value.

Lift coefficient C_L		
	Expected value	Variance
Fully tensorized gPC	72.273×10^{-2}	2.787×10^{-4}
Sparse gPC	72.278×10^{-2}	2.782×10^{-4}
SSKRR	72.274×10^{-2}	2.777×10^{-4}

(b) Expected value and variance.

Figure 3.21: The PDFs of the lift coefficient C_L (top) using the three surrogate models and their corresponding expected value and variance (bottom) estimated from the $I_s = 1 \times 10^6$ data points.

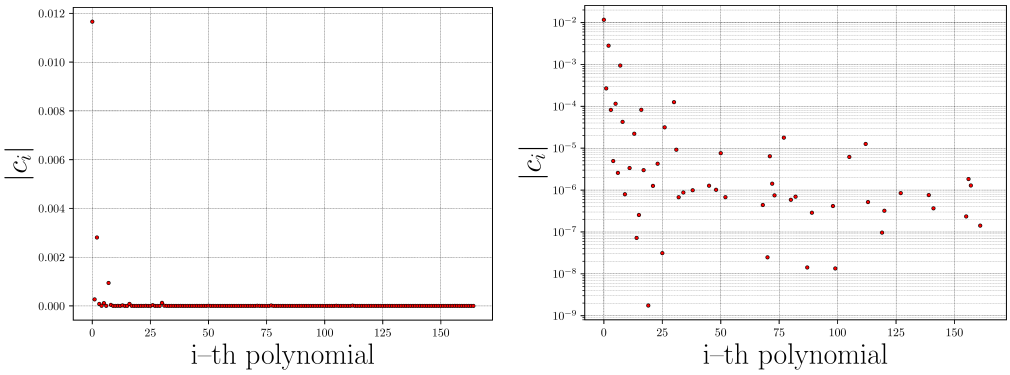
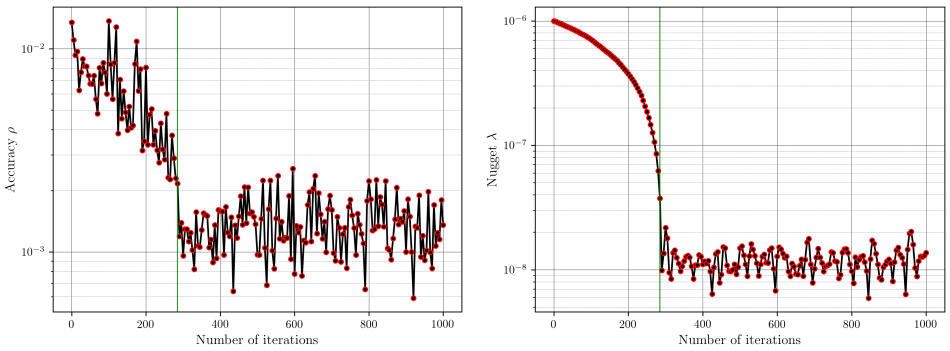


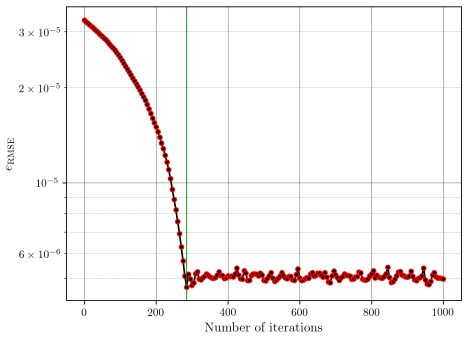
Figure 3.22: Expansion coefficients $\mathbf{c}^\star$ with $I = 80$ observations of F for the drag coefficient C_D.

predictions of the SSKRR surrogate. A strong dependece—nearly linear—between the drag coefficient C_D and the Mach number M can be denoted on Figure 3.23. The PDFs of the drag coefficient C_D using the three surrogate models are estimated



(a) Accuracy ρ.

(b) Nugget λ.



(c) Error e_{RMSE} on the validation set.

Figure 3.23: Evolutions the accuracy ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as a function of the number of iterations of the parametric KF algorithm for the drag coefficient C_D. The green vertical line corresponds to the iteration where the error e_{RMSE} is minimal on the validation set.

Drag coefficient C_D			
	SSKRR	Sparse gPC	Fully tensorized gPC
e_{RMSE}	3.722×10^{-6}	3.323×10^{-6}	5.171×10^{-6}
e_{NRMSE}	2.535×10^{-4}	2.264×10^{-4}	3.523×10^{-4}
e_{MRE}	0.0628%	0.0470%	0.0866%
Q^2	0.9999989	0.9999991	0.9999979

Table 3.11: Comparison of the errors between the surrogate models with $I = 80$ and $I_{\text{T}} = 25$ observations for the drag coefficient C_D.

from $I_{\text{s}} = 1 \times 10^6$ random data points taken at random following the Beta laws of Table 3.8 and then smoothing out the resulting histograms by a normal kernel density function [156]. They are shown on Figure 3.25, together with their corresponding expected value and variance. The expected values of the PDFs obtained by each surrogate model are shown on Figure 3.25 with vertical lines. Notice that we obtain

comparable results expect at their peaks where the fully tensorized is a little above. Finally, Sobol' main-effect sensitivity indices are gathered in Table 3.12. As expected from the previous results, the variable $X_2 = M$ is much more influential than the variables $X_1 = r$ and $X_3 = \alpha$ on the drag coefficient C_D, where the latter have the same influences, as expected from the values of the expansion coefficients $\mathbf{c}^\star$ in Equation (3.109). The SSKRR surrogate gives similar performance as the sparse gPC one. Both slightly outperform the fully tensorized surrogate model.

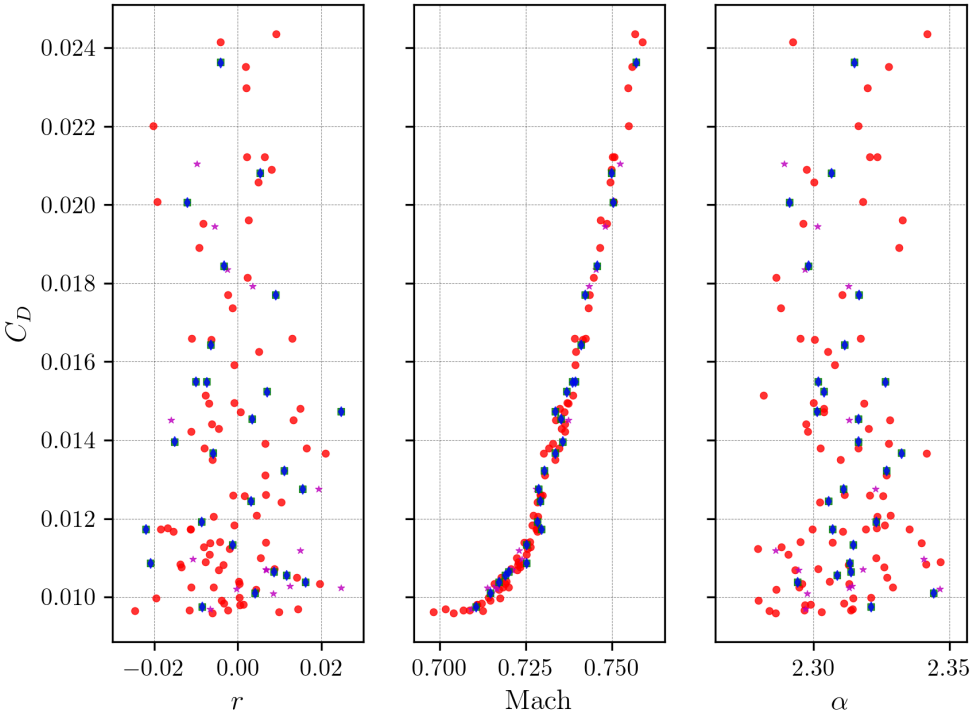
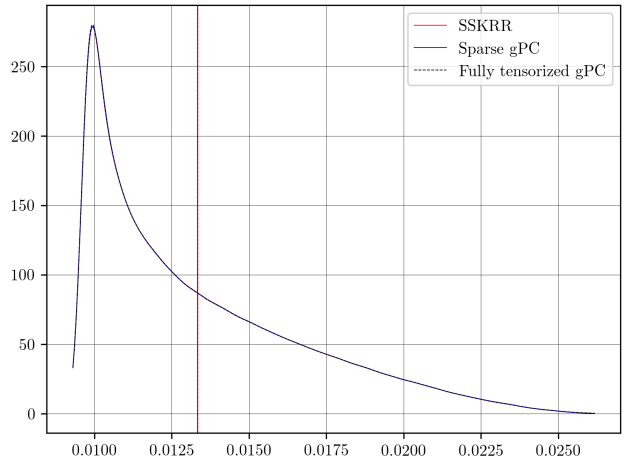


Figure 3.24: Distribution of the difference between the prediction given by our algorithm and the observations on the test set for the drag coefficient C_D with $I = 80$ and $I_{\mathrm{T}} = 25$ samplings points. The red circles are the observations defining the learning set. The violet stars are the observations defining the validation set. The green squares are the observations defining the test set with their corresponding predictions given by the SSKRR surrogate, depicted by the blue diamonds.



(a) PDFs of the drag coefficient C_D. The vertical lines correspond to their respective expected value.

drag coefficient C_D		
	Expected value	Variance
Fully tensorized gPC	133.371×10^{-4}	1.165×10^{-5}
Sparse gPC	133.365×10^{-4}	1.162×10^{-5}
SSKRR	133.422×10^{-4}	1.166×10^{-5}

(b) Expected value and variance.

Figure 3.25: The PDFs of the drag coefficient C_D (top) using the three surrogate models and their corresponding expected value and variance (bottom) estimated from the $I_s = 1 \times 10^6$ data points.

drag coefficient C_D			
	$X_1 = h/c$	$X_2 = M$	$X_3 = \alpha$
Fully tensorized gPC	0.00812	0.989	0.000765
Sparse gPC	0.00806	0.989	0.000761
SSKRR	0.00863	0.989	0.00130

Table 3.12: Sobol' main-effect sensitivity indices of the drag coefficient C_D with $I = 80$ samplings points. For our algorithm, they were estimated from a pick-freeze estimator with a matrix size of 1×10^6 samples. For the sparse and the fully tensorized gPC surrogate models, they were estimated from the values of the expansion coefficients using Section 3.5.2.

3.7.4 Pitching moment coefficient C_M

We finally focus on the pitching moment coefficient C_M. The expansion coefficients $\mathbf{c}^\star$ given by the ℓ_1-minimization are shown on Figure 3.26 using the learning set of Figure 3.17. From the values of the expansion coefficients $\mathbf{c}^\star$, we notice that only low order polynomials are relevant in the expansion coefficients. Indeed, one has:

$$G_p(\mathbf{x}) \approx c_1^\star \phi_{(0,0,0)}(\mathbf{x}) + c_2^\star \phi_{(1,0,0)}(\mathbf{x}) + c_3^\star \phi_{(0,1,0)}(\mathbf{x}) + c_8^\star \phi_{(0,2,0)}(\mathbf{x}) + c_{14}^\star \phi_{(1,2,0)}(\mathbf{x}) + c_{17}^\star \phi_{(0,3,0)}(\mathbf{x}) + c_{31}^\star \phi_{(0,4,0)}(\mathbf{x}), \quad (3.110)$$

where $\phi_{(i_1,i_2,i_3)}$ is defined as in Equation (3.94) and with $|c_1^\star| \gg |c_3^\star| \gg |c_2^\star| \approx |c_8^\star| \approx |c_{17}^\star| \approx |c_{31}^\star| \gg |c_{14}^\star|$. The highest order polynomial of $G_p(\mathbf{x})$ has order 4 and its expansion coefficient is small compared to the others. From Figure 3.26, the sparsity is observed to be $S \approx 7$. Therefore Theorem 6 yields $N \gtrsim I \approx 10000$ observations up to a constant C, but we observe in practice that the $I = 80$ observations are enough for an accurate recovery.

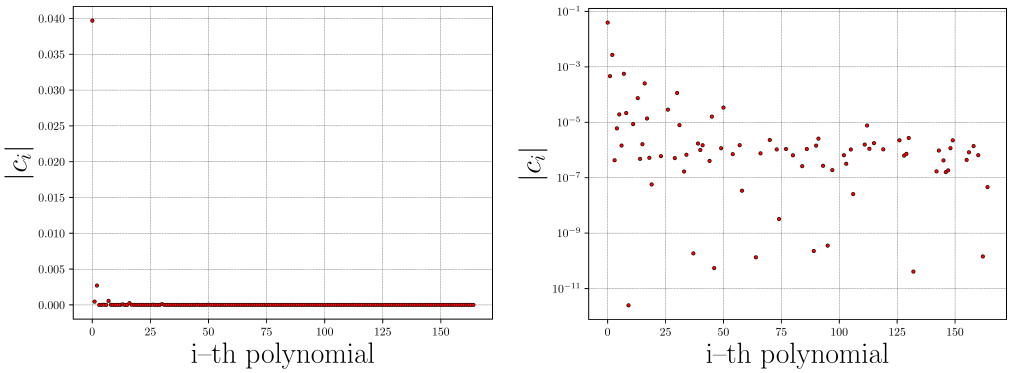
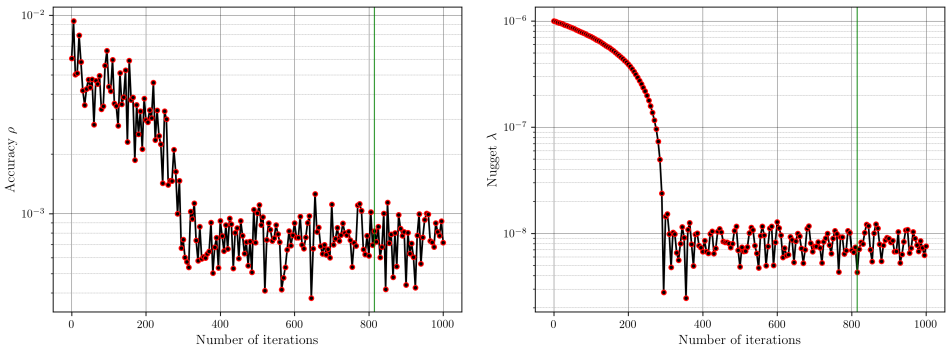


Figure 3.26: Expansion coefficients $\mathbf{c}^\star$ with $I = 80$ observations of F for the pitching moment coefficient C_M.

Pitching moment coefficient C_M			
	SSKRR	Sparse gPC	Fully tensorized gPC
e_{RMSE}	7.071×10^{-6}	8.409×10^{-6}	8.838×10^{-6}
e_{NRMSE}	1.525×10^{-4}	1.813×10^{-4}	1.906×10^{-4}
e_{MRE}	0.0436%	0.0473%	0.0602
Q^2	0.999996	0.999993	0.999992

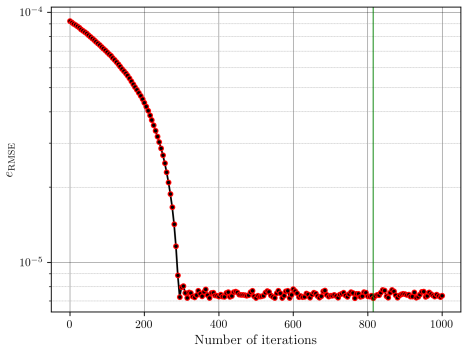
Table 3.13: Comparison of the errors between the surrogate models with with $I = 80$ and $I_T = 25$ observations for the pitching moment coefficient C_M.

The evolutions of the accuracy ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as functions of the number of iterations of



(a) Accuracy ρ.

(b) Nugget λ.



(c) Error e_{RMSE} on the validation set.

Figure 3.27: Evolutions the accuracy ρ of the parametric KF algorithm, the nugget λ, and the error e_{RMSE} on the validation set as functions of the number of iterations of the KF algorithm for the pitching moment coefficient C_M. The green vertical line corresponds to the iteration where the error e_{RMSE} is minimal on the validation set.

the parametric KF algorithm are shown on Figure 3.27. We find $\lambda_{\min} = 4.30 \times 10^{-9}$, which corresponds to the green vertical line on Figure 3.27. One can notice that the accuracy ρ was greatly reduced (divided by ≈ 10) between the initial iteration and convergence on the validation set. The error e_{RMSE} was divided by slightly more than about 12 on the validation set. The comparison of the errors e_{RMSE}, e_{NRMSE} and e_{MRE} between the surrogate models are given in Table 3.13 where $\lambda = \lambda_{\min}$. Figure 3.28 shows the values of the pitching moment coefficient C_M with respect to the input parameters on the learning set, the verification set, the test set, and the predictions of the SSKRR surrogate. A dependence—nearly linear—between the pitching moment coefficient C_M and the Mach number M can be denoted for the highest Mach numbers. The PDFs of the pitching moment coefficient C_M using the three surrogate models are estimated from $I_{\mathrm{s}} = 1 \times 10^6$ random data points taken at random following the Beta laws of Table 3.8 and then smoothing out the resulting histograms by a normal kernel density function [156]. They are shown on Figure 3.29, together with their corresponding expected value and variance. The expected values

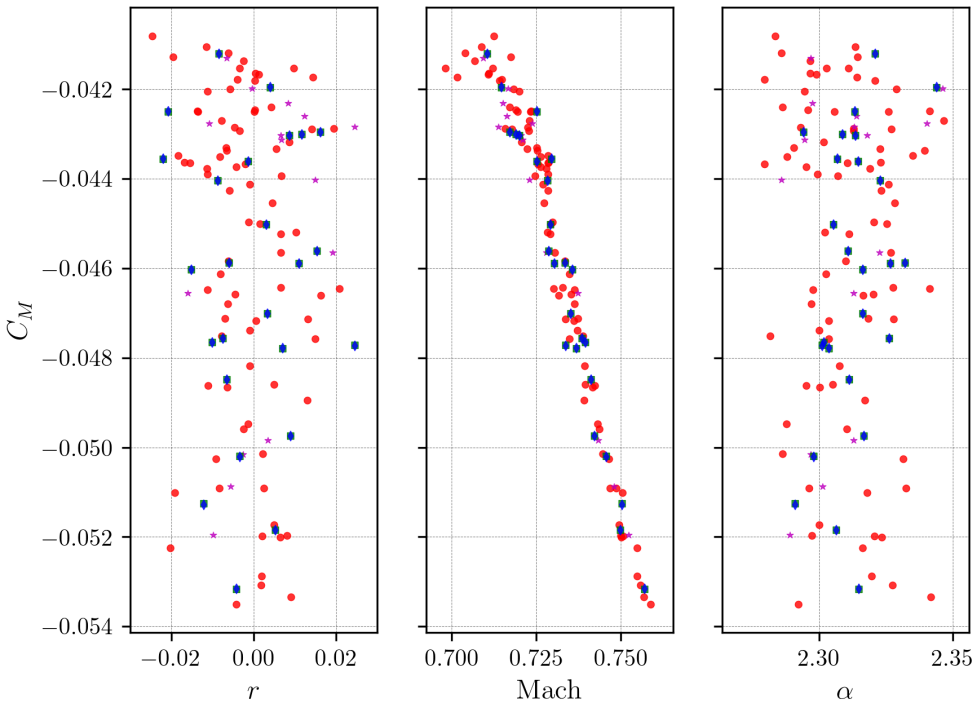
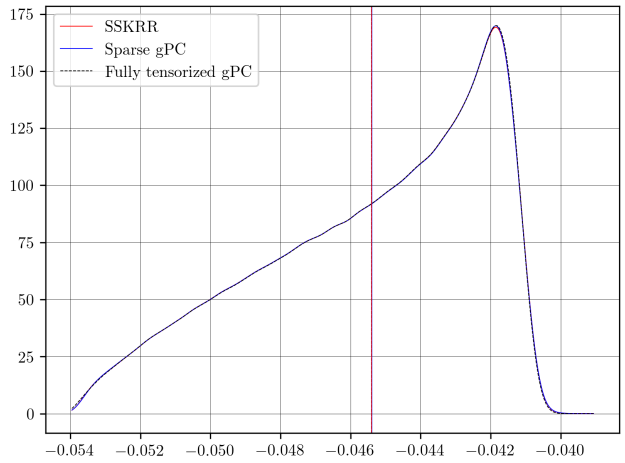


Figure 3.28: Distribution of the difference between the prediction given by our algorithm and the observations on the test set for the pitching moment coefficient C_M with $I = 80$ and $I_T = 25$ samplings points. The red circles are the observations defining the learning set. The violet stars are the observations defining the validation set. The green squares are the observations defining the test set with their corresponding predictions given by the SSKRR surrogate, depicted by the blue diamonds.

of the PDFs obtained by each surrogate model are shown on Figure 3.29 with vertical lines. Notice that we obtain comparable results except at the tails of the distributions. Finally, Sobol' main-effect sensitivity indices are gathered in Table 3.14. As expected from the previous results, the variable $X_2 = M$ is much more influential than the variables $X_1 = r$ and $X_3 = \alpha$ on the pitching moment coefficient C_M, where the parameter $X_3 = \alpha$ has almost no influence at all, as predicted from the values of the expansion coefficients $\mathbf{c}^\star$ in Equation (3.110). The SSKRR surrogate gives similar performance as the sparse gPC one. Both slightly outperform the fully tensorized gPC surrogate model.

3.7.5 Estimation of prediction variance

The prediction given by the SSKRR surrogate allows us to obtain the prediction variance $\sigma^2(\mathbf{x})$ (3.15) at an unobserved point $\mathbf{x}$. Figure 3.30 shows this prediction variance for the lift coefficient C_L, the drag coefficient C_D, and the pitching moment coefficient C_M at different points. The top row is for the lift coefficient C_L, the middle row is for the drag coefficient C_D, and the bottom row is for the pitching



(a) PDFs of the pitching moment coefficient C_M.

pitching moment coefficient C_M		
	Expected value	Variance
Fully tensorized gPC	-0.0454	1.039×10^{-5}
Sparse gPC	-0.0454	1.039×10^{-5}
SSKRR	-0.0454	1.040×10^{-5}

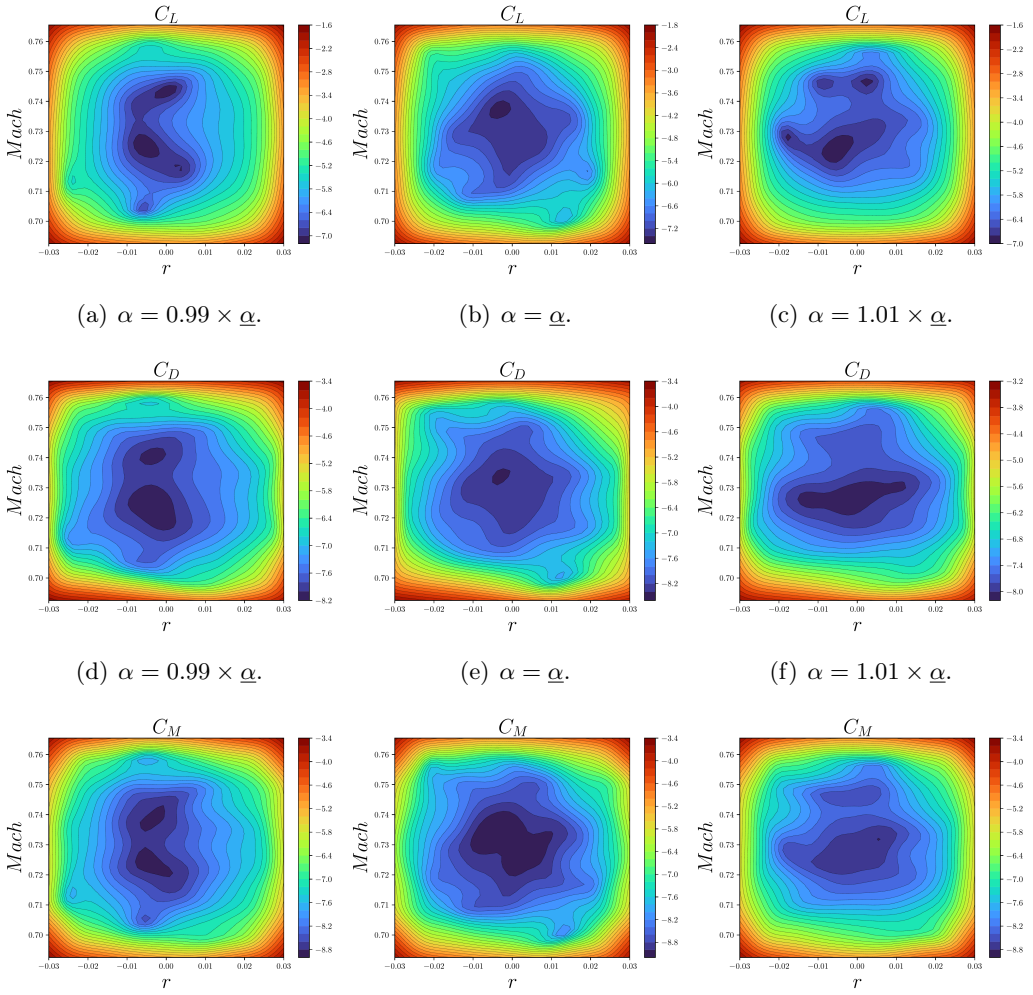
(b) Expected value and variance.

Figure 3.29: The PDFs of the pitching moment coefficient C_M (top) using the three surrogate models and their expected value and variance (bottom) estimated from the $I_s = 1 \times 10^6$ data points.

pitching moment coefficient C_M			
	$\xi_1 = h/c$	$\xi_2 = M$	$\xi_3 = \alpha$
Fully tensorized gPC	0.0269	0.972	3.220×10^{-7}
Sparse gPC	0.0270	0.972	2.210×10^{-7}
SSKRR	0.0272	0.972	0.000469

Table 3.14: Sobol' main-effect sensitivity indices of the pitching moment coefficient C_M with $I = 80$ samplings points. For our algorithm, they were estimated from a pick-freeze estimator with a matrix size of 1×10^6 samples. For the sparse and the fully tensorized gPC surrogate models, they were estimated from the values of the expansion coefficients using Section 3.5.2.

moment coefficient C_M. The variance is maximal at the edges of the input domain because it corresponds to the area where the observations are scarce. Contrary to the sparse gPC or the fully tensorized surrogate model, Algorithm 3 allows us to obtain such information about the function F. For instance, it can guide us to decide where a new observation of F could be effective in order to improve the accuracy of the surrogate model.



(a) $\alpha = 0.99 \times \underline{\alpha}$. (b) $\alpha = \underline{\alpha}$. (c) $\alpha = 1.01 \times \underline{\alpha}$.

(d) $\alpha = 0.99 \times \underline{\alpha}$. (e) $\alpha = \underline{\alpha}$. (f) $\alpha = 1.01 \times \underline{\alpha}$.

(g) $\alpha = 0.99 \times \underline{\alpha}$. (h) $\alpha = \underline{\alpha}$. (i) $\alpha = 1.01 \times \underline{\alpha}$.

Figure 3.30: $\log_{10}$ *of the variance* σ^2 *of the prediction with respect to the input variables* (r, M) *at* $\alpha = 0.99 \times \underline{\alpha}$ *(left column),* $\alpha = \underline{\alpha}$ *(middle column) and* $\alpha = 1.01 \times \underline{\alpha}$ *(right column) for the lift coefficient* C_L *(top row), the drag coefficient* C_D *(middle row), and the pitching moment coefficient* C_M *(bottom row).*

3.8 Conclusions

In the OUQ framework presented in Chapter 1 and the algorithm detailed in Chapter 2, computing an optimal bound demands intensive calculations and an almost free use of the performance function. This can be done by resorting to a surrogate model which must be inexpensive to evaluate. We have presented several methods to build a surrogate model from limited knowledge of a target function F based on a regression approach. First, we have shown the equivalence of three approaches in Section 3.2: the functional one based on RKHS for which a brief summary has been presented in Section 3.2.1, the Kriging one presented in Section 3.2.5, and the Gaussian process one presented in Section 3.2.6. All these approaches depend on the selection of a class of functions known as kernels. Following [115, Theorem 8.4], the bound on the deterministic difference between the ground truth function F and its surrogate model has been reminded in Section 3.2.4. This bound depends on two elements: the prediction variance σ^2 and the RKHS norm of F. In return they both depend on the choice of the kernel. The main focus of this chapter has been to determine the "best" kernel, where "best" has been defined in two ways. First, the parametric and non-parametric KF algorithms have been presented in Section 3.3, where "best" has been here defined with respect to a specified accuracy introduced in Section 3.3.1. These algorithms, which were first developed in a classification context [31, 114], have been extended to regression. It has been seen that the parametric KF algorithm could be suitable to build a surrogate model but not the non-parametric one, confirming in that respect the results obtained in [35]. Then another method to determine the best kernel has been introduced within Mercer's framework, of which an overview has been presented in Section 3.4.1. This framework yields an explicit expression of the RKHS norm of the target function F through the spectral decomposition of the integral operator associated with the kernel and the expansion coefficients of F onto the L^2 space. The SSKRR and the NSKRR algorithms have been presented in Section 3.4.3.2 and in Section 3.4.3.3, respectively. They seek to minimize the RKHS norm of F with respect to the eigenvalues of the integral operator associated with the kernel. The performances of the SSKRR algorithm and its comparison with several other classical surrogate methods have been investigated on two tractable test cases usually considered in sensitivity analysis in Section 3.6. An application of the SSKRR algorithm to an aerodynamic case has finally been considered in Section 3.7.

Conclusions and perspectives

In this thesis, we have presented an approach to achieve certification of the performance of an aircraft, or a system more generally, under uncertainties. Here certification is understood as the process of guaranteeing that the probability of occurence of a certain scenario or target, known as the probability of failure, is below an acceptable (small) tolerance. In Chapter 1, starting from McDiarmid's inequality, we have seen in Section 1.2 that the Optimal Uncertainty Quantification (OUQ) framework [112] is a promising candidate to achieve such certification. This framework can be applied to much more general classes of hypotheses on the system than the ones considered in McDiarmid's inequality. It is based on computing the optimal bounds of the probability of failure with respect to the information known about the system. These bounds can be understood as best- and worst-case approaches. Computing them can be formulated as an optimization problem in an *a priori* infinite-dimensional set, but it can be transformed to a finite-dimensional one through the reduction Theorem 1. In Section 1.3 this finite-dimensional optimization problem has then been solved by using the `mystic` numerical framework presented in Section 1.3.1. The OUQ and `mystic` frameworks have been successfully applied on several examples including aerodynamic test cases. However, even after reduction, the optimization problem is still *a priori* non-convex and highly constrained. In that respect, it is not guaranteed that an optimal solution is ever found.

Therefore, a procedure has to be developed to corroborate the numerical optimums obtained by the computations. This is the topic of Chapter 2, where the problem of adaptively reconstructing a monotonically increasing function from imperfect pointwise observations of this function has been considered. The proposed algorithm is reminiscent of the classical Pool Adjacent Violators Algorithm (PAVA) to isotonic regression but it differs from the assumptions made as detailed in Section 2.1. This reconstruction problem is closely linked to certification as shown in Section 2.2. The algorithm itself has been presented in Section 2.4.1 and its convergence properties have been proved in Section 2.4.2, which ensure its mathematical relevance. In Section 2.5, it has been tested on two synthetic cases, that is a continuous and a discontinuous function, in order to quantify its performance. The algorithm has also been carried out on a challenging problem taken from aerodynamic design in section 2.6. Nevertheless, the application of the OUQ framework requires to evaluate the performance function several tens of thousands of times. Thus the use of a time-consuming model is not feasible and a computationally inexpensive model is required. This is the topic of the next chapter.

Several methods have been successively presented in Chapter 3 for the construction of surrogate models. A short reminder about the theory of Reproducing Kernel Hilbert Spaces (RKHS) has first been given in Section 3.2.1. It has allowed us to present the optimal recovery solution and kernel ridge regression methods in Section 3.2.2 together with mathematical bounds (3.2.4) on the error between the ground truth

function and its surrogate approximation in Section 3.2.4. Kriging methods and the Gaussian process regression method have been presented in Section 3.2.5 and Section 3.2.6, respectively. We have seen that all these methods are equivalent and depend on the choice of the kernel function. Three different algorithms have been presented to determine the appropriate kernel. The kernel flow algorithms have first been detailed in Section 3.3. The definition of what an appropriate kernel is has been given in Section 3.3.1. Two versions of this algorithm exist: the non-parametric version and the parametric one, described in Section 3.3.2 and Section 3.3.3 respectively. The latter was originally developed in a classification context [114]. In this work, we have tested both versions on an aerodynamic case in a regression context in Section 3.3.4. We have seen that the parametric version yields promising results while the non-parametric one does not seem to be suitable for regression, confirming the conclusions reached elsewhere [35]. Then sparse and non-sparse spectral kernel ridge regression (SKRR) algorithms have been introduced in Section 3.4. These two algorithms are based on approximating the RKHS norm of the ground truth function F through its expansion coefficients, but they differ on the assumptions done about F. The sparse SKRR algorithm assumes that F is sparse on a given basis while the non-sparse SKRR algorithm does not. The sparse SKRR algorithm together with several other classical surrogate modeling methods have been applied to synthetic test cases used in uncertainty quantification, and to more complex cases pertaining to aerodynamic applications.

The algorithms developed in this thesis are likely to be improved in several ways. Possible improvements and extensions of Algorithm 1 presented in Chapter 2 are to weight the areas $\mathfrak{a}_i^{(n)}$ as they are summed up to form the total weighted area $\mathrm{WA}^{(n)}$ driving the iterative process, in order to optimally enforce both the addition of "steps" $s_i^{(n)}$ in the reconstruction function $\mathcal{F}^{(n)}$ of Definition 1, and the improvement of their "heights" $y_i^{(n)}$. This could be achieved considering the following alternative definition $i_+^{(n)} = \arg\max_i\{(I^{(n)} - i - 1)\mathfrak{a}_i^{(n)}\}$ in Algorithm 1, for example, which results in both adding a step to the $i_+^{(n)}$-th current one and possibly improving all subsequent evaluations $y_i^{(n+1)}$, $i > i_+^{(n)}$. We may also further envisage to adapt the ideas elaborated in this chapter to the reconstruction of convex functions by extending the notion of consistency, for instance.

The non-sparse SKRR algorithm presented in Chapter 3 for the approximation of non-sparse ground truth function has not yet been tested numerically. Future works should consider its implementation and analyses of its performance on synthetic and complex cases. Moreover, only polynomial bases have been considered so far. Thus it may be interesting to test the sparse SKRR and non-sparse SKRR algorithms using other types of bases, such as the basis corresponding to the eigenvectors of the Gaussian kernel for which analytical expressions are available. Actually the choice of a basis and surrogate model also depends on the available sampling set of the ground truth function: it may be constituted by either structured nodes on a (possibly subsampled) quadrature rule, for example, or by (partially) unstructured nodes as in coherence-optimal, asymptotic, Latin hypercube, or Monte-Carlo and quasi Monte-Carlo sampling strategies, among others; see *e.g.* [60]. The KRR surrogates and their

spectral byproducts considered in this research are quasi interpolant depending on the strength of the smoothing by the nugget, whereas polynomial chaos expansions are not. This may impact the choice of a surrogate model as well. One finally expects that the deterministic bounds on the difference between the ground truth function F and its surrogate G given in Chapter 3 could be useful to obtain a bound or an estimate of the validation diameter D_{F-G} used for post-optimization with simulations or experiments in Chapter 1. This strategy of certification by the data-on-demand protocol elaborated in [2, 75] has not been explored so far. It should be considered in future works in connection with the techniques of data assimilation which have recently emerged in engineering science [123], with noticeable applications in fluid dynamics [19, 126].

Appendices

Résumé étendu

La performance attendue d'un système peut généralement différer de sa performance opérationnelle en raison de la variabilité de certains paramètres. Habituellement, la phase de conception est divisée en deux phases différentes. La première phase consiste à déterminer la conception pré-optimale. Grâce à l'utilisation de logiciels numériques, la meilleure conception possible est choisie. Il s'agit de tenir compte de certaines performances idéales à atteindre en formulant des hypothèses—qui peuvent être subjectives—traduisant la variabilité de certains paramètres. La deuxième phase consiste, quant à elle, à certifier par des expériences à taille réelle que la conception déterminée précédemment est valide. De cette façon, la conception post-optimale est spécifiée. Cette deuxième phase est la plus coûteuse ; c'est pourquoi les industriels, notamment dans le domaine aéronautique, cherchent à réduire le recours aux expériences à taille réelle. L'*Optimal Uncertainty Quantification* (OUQ) [112] est un outil mathématique puissant qui peut être utilisé pour borner rigoureusement la probabilité de dépasser un seuil de performance donné pour des conditions opérationnelles ou des caractéristiques de système incertaines. Autrement dit, nous cherchons à certifier que $\mathbb{P}_{\mathbf{X}\sim\mu^\dagger}[F(\mathbf{X}) \geq a] \leq \varepsilon$, où F est la fonction performance et $\mathbf{X}$ sont les paramètres incertains suivant la mesure de probabilité $\mu^\dagger$ qui influent sur la performance F. La fonction performance F et/ou la mesure de probabilité $\mu^\dagger$ peuvent ne pas être connues, ou seulement partiellement. Ainsi, l'outil OUQ conduit à la résolution d'un problème d'optimisation sur l'ensemble des mesures de probabilités admissibles, permettant ainsi de ne pas à avoir à formuler des hypothèses subjectives qui peuvent être fortes. Ce problème d'optimisation est *a priori* un problème non-convexe et fortement contraint, dans un espace de dimension infinie. Ainsi, il est généralement difficile à résoudre sur le plan numérique. Néanmoins, il peut être réduit à un problème d'optimisation équivalent en dimension finie par l'intermédiaire d'un théorème de réduction.

Ce travail de thèse porte, dans un premier temps, sur l'application de l'outil OUQ sur un cas simple : la déformée maximale d'une poutre en flexion sous les hypothèses d'Euler-Bernoulli. Ensuite, des cas issus du domaine de l'aéronautique permettant de borner rigoureusement la variation de fonctions de performance classiquement étudiées en aérodynamique, telles que la portance ou la trainée, sont abordés. Pour chacun des cas, différents scénarios traduisant différentes hypothèses sur le système sont considérés. Néanmoins, même après réduction du problème d'optimisation, cette approche demeure complexe et l'exactitude des résultats obtenus en est ainsi impactée.

Afin de pallier cette difficulté, un algorithme permettant d'assurer la validité de ces résultats numériques a été par la suite formulé [15]. Cet algorithme est analogue à la régression isotonique mais diffère sur les hypothèses établies. La convergence de cet algorithme a été démontrée. L'algorithme a été ensuite validé sur des cas tests numériques ainsi que sur un cas aérodynamique. Quoi qu'il en soit, l'application de la méthode OUQ nécessite d'évaluer la fonction de performance du système plusieurs dizaines de milliers de fois. De fait, l'usage d'un modèle peu coûteux en termes de

temps est requis.

Dans cette optique, plusieurs méthodes qui permettent de construire des modèles de substitution rapides à évaluer sont finalement présentées. Une première méthode fondée sur l'algorithme Kernel Flow, initialement appliquée à un problème de classification [114], est étendue à un problème de régression permettant la détermination du noyau d'un modèle de substitution de type processus gaussien. Deux versions de cet algorithme sont présentées : la version paramétrique et la version non-paramétrique. Puis, deux nouveaux algorithmes sont présentés. Ces algorithmes utilisent l'équivalence qui existe entre l'approche processus gaussien et l'approche par espace de Hilbert à noyau reproduisant (Reproducing Kernel Hilbert Space, RKHS) afin de déterminer le noyau adéquat. Ces différentes méthodes de modèles de substitution sont appliquées au travers de différents cas tests dont des cas du domaine aérodynamique.

Bibliography

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York NY, 1965 (Cited on page 86.)

[2] M. Adams, A. Lashgari, B. Li, M. M. McKerns, J. Mihaly, M. Ortiz, H. Owhadi, A.J. Rosakis, M. Stalzer, and T. J. Sullivan, Rigorous model-based uncertainty quantification with application to terminal ballistics–Part II. Systems with uncontrollable inputs and large scatter, *J. Mech. Phys. Solids*, **60**(5):1002–1019, 2012. (Cited on pages 4 and 137.)

[3] Airbus, *A Statistical Analysis of Commercial Aviation Accidents 1958–2020*, Technical Report, Airbus, Blagnac, France, March 2021. (Cited on page 2.)

[4] N. Aronszajn, Theory of reproducing kernels, *Trans. Amer. Math. Soc.*, **68**(3):337–404, 1950. (Cited on page 74.)

[5] I. Babuška, F. Nobile and R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM Rev.*, **52**(2):317–355, 2010. (Cited on page 3.)

[6] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, A simple proof of the Restricted Isometry Property for random matrices, *Constr. Approx.*, **28**(3):253–263, 2008. (Cited on page 100.)

[7] R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk, *Statistical Interference under Order Restrictions. The Theory and Application of Isotonic Regression*, John Wiley & Sons, London, 1972. (Cited on page 51.)

[8] D. Baumann and K. Baumann, Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation, *J. Cheminform.*, **47**(6):47, 2014. (Cited on page 91.)

[9] J. L. Beck and L. S. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, *ASCE J. Eng. Mech.*, **124**(4):455–461, 1998. (Cited on page 3.)

[10] L. Berke, S. N. Patnaik, and P. L. N. Murthy, Optimum design of aerospace structural components using neural networks, *Comput. Struct.*, **48**(6):1001–1010, 1993 (Cited on page 73.)

[11] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.*, **25**:539–575, 1993. (Cited on page 73.)

[12] E. Bernardini, S. M. J. Spence, D. Wei, and A. Kareem, Aerodynamic shape optimization of civil structures: A CFD-enabled Kriging-based approach, *J. Wind. Eng. Ind. Aerodyn.*, **144**:154–164, 2015. (Cited on page 77.)

[13] Boeing, *Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations 1959–2020*, Technical Report, Aviation Safety Boeing Commercial Airplanes, Seattle WA, September 2021. (Cited on page 2.)

[14] M. Bompard, J. Peter, and J. Desideri, Surrogate models based on function and derivative values for aerodynamic global optimization, In *Proceedings of the V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010 (J. C. F. Pereira, A. Sequeira, and J. M. C. Pereira, eds.), 14-17 June 2010, Lisbon*, 17 pp., 2010. (Cited on page 73.)

[15] L. Bonnet, J.-L. Akian, É. Savin, and T. J. Sullivan, Adaptive reconstruction of imperfectly-observed monotone functions, with applications to uncertainty quantification, *Algorithms*, **13**(8):196, 2020. (Cited on pages 49 and 141.)

[16] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, Oxford University Press, Oxford, 2013. (Cited on pages 3 and 8.)

[17] P. Boufounos, M. F. Duarte, and R. G. Baraniuk, Sparse signal reconstruction from noisy compressive measurements using cross validation, In *SSP'07: Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing, 26-29 August 2007, Madison WI*, pp 299–303, 2007. (Cited on page 102.)

[18] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins, A Python surrogate modeling framework with derivatives, *Adv. Eng. Softw*, **135**:102662, 2019. (Cited on page 107.)

[19] T. Bui-Thanh, M. Damodaran, and K. Willcox, Aerodynamic data reconstruction and inverse design using Proper Orthogonal Decomposition, *AIAA J.*, **42**(8):1505–1516, 2004. (Cited on pages 73 and 137.)

[20] L. Cambier, S. Heib, and S. Plot, The Onera *elsA* CFD software: Input from research and feedback from industry, *Mechanics & Industry*, **14**(3):159–174, 2013. (Cited on pages 28, 65 and 118.)

[21] E. J. Candès and T. Tao, Decoding by linear programming, *IEEE Trans. Inf. Theory*, **51**(12):4203–4215, 2005 (Cited on page 100.)

[22] E. J. Candès, J. K. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Comm. Pure Appl. Math.*, **59**(8):1207–1223, 2006. (Cited on page 99.)

[23] E. J. Candès and J. K. Romberg, Sparsity and incoherence in compressive sampling, *Inverse Probl.*, **23**(3):969–985, 2007. (Cited on page 99.)

[24] E. J. Candès, The restricted isometry property and its implications for compressed sensing, *C. R. Math.*, **346**(9-10):589–592, 2008. (Cited on pages 99, 100 and 102.)

[25] E. J. Candès and M. B.Wakin, An introduction to compressive sampling, *IEEE Signal Process. Mag.*, **25**(2):21–30, 2008. (Cited on pages 100 and 102.)

[26] E. J. Candès and Y. Plan, A probabilistic and RIPless theory of compressed sensing, *IEEE Trans. Inf. Theory*, **57**(11):7235–7254, 2011. (Cited on page 101.)

[27] J.-C. Chassaing and D. Lucor, Stochastic investigation of flows about airfoils at transonic speeds, *AIAA J.*, **48**(5):938–950, 2010. (Cited on page 9.)

[28] A. Chatterjee, An introduction to the proper orthogonal decomposition, *Current Science*, **78**(7):808–817, 2000. (Cited on page 73.)

[29] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.*, **43**(1):129–159, 2001. (Cited on page 100.)

[30] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.*, **20**(1):33–61, 2006. (Cited on page 101.)

[31] Y. Chen, H. Owhadi, and A. M. Stuart, Consistency of empirical Bayes and kernel flow for hierarchical parameter estimation, *Math. Comp.*, **90**:2527–2578, 2021. (Cited on pages 72, 120 and 134.)

[32] A. Chkifa, A. Cohen, and C. Schwab, Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs, *J. Math. Pures Appl.*, **103**(2):400–428, 2015. (Cited on page 106.)

[33] H. S. Chung and J. Alonso, Using gradients to construct cokriging approximation models for high-dimensional design optimization problems, In *40th AIAA Aerospace Sciences Meeting & Exhibit, 14-17 January 2002, Reno NV*, AIAA paper #2002-0317, 2002. (Cited on page 78.)

[34] P. H. Cook, M. A. McDonald, and M. C. P. Firmin, Aerofoil RAE 2822—Pressure distributions, and boundary layer and wake measurements, In *Experimental Data Base for Computer Program Assessment. AGARD Advisory Report No. 138*, NATO, May 1979. (Cited on pages 18, 28, 65, 66, 117 and 118.)

[35] M. Darcy, *Kernel Flows Demystified: Applications to Regression*, Master's thesis, Imperial College London, September 2020. (Cited on pages 94, 96, 134 and 136.)

[36] A.M. DeGennaro, C.W. Rowley, and L. Martinelli, Uncertainty quantification for airfoil icing using polynomial chaos expansions, *AIAA J. Aircraft*, **52**(5):1404–1411, 2015. (Cited on page 9.)

[37] H. Dette and W. J. Studden, *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*, John Wiley & Sons, New York NY, 1997. (Cited on pages 15 and 17.)

[38] D. L. Donoho, Compressed sensing, *IEEE Trans. Inf. Theory*, **52**(4):1289–1306, 2006. (Cited on pages 99 and 100.)

[39] A. Doostan and H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, *J. Comput. Phys.*, **230**(8):3015–3034, 2011. (Cited on pages 3, 102 and 106.)

[40] M. Drela and M. B. Giles, ISES: A two-dimensional viscous aerodynamic design and analysis code, *25th AIAA Aerospace Sciences Meeting, 24-26 March 1987, Reno NV*, AIAA paper #1987-424, 1987. (Cited on pages 18 and 38.)

[41] M. Drela, A user's guide to MSES 3.05, Massachusetts Inst. of Technology, Dept. of Aeronautics and Astronautics, Cambridge, MA, July 2007. (Cited on page 89.)

[42] M. Drela and H. Youngren, A user's guide to MISES 2.63, Massachusetts Inst. of Technology, Dept. of Aeronautics and Astronautics, Cambridge, MA, February 2009. (Cited on pages 18 and 38.)

[43] R.F. Drenick, P. C. Wang, C. B. Yun, and A. J. Philappacopoulos, Critical seismic response of nuclear reactors, *Nucl. Eng. Des.*, **58**(3):425–435, 1980. (Cited on page 2.)

[44] J. Duchon, Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces, *ESAIM: M2AN*, **10**(R3):5–12 (1976). (Cited on pages 18 and 29.)

[45] J. Duchon, Splines minimizing rotation-invariant semi-norms in Sobolev spaces, In *Constructive Theory of Functions of Several Variables (W. Schempp and K. Zeller, eds.)*, Lecture Notes in Mathematics **571**, pp. 85–100, Springer, Berlin, 1977. (Cited on pages 18 and 29.)

[46] A. Dumont, J.-L. Hantrais-Gervois, P.-Y. Passaggia, J. Peter, I. Salah el Din, and É. Savin, Ordinary kriging surrogates in aerodynamics, In *Uncertainty Management for Robust Industrial Design in Aeronautics (C. Hirsch, D. Wunsch, J. Szumbarski, Ł. Łaniewski-Wołłk, and J. Pons-Prats, eds.)*, pp. 229–245, Springer, Cham, 2019. (Cited on pages 65, 67 and 68.)

[47] R. Dwight and Z-.H. Han, Efficient uncertainty quantification using gradient-enhanced Kriging, In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 4-7 May 2009, Palm Springs CA*, AIAA paper #2009-2276, 2009. (Cited on page 77.)

[48] I. Elishakoff and M. Ohsaki, *Optimization and Anti-Optimization of Structures Under Uncertainty*, World Scientific, London, 2010. (Cited on page 3.)

[49] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, On the convergence of generalized polynomial chaos expansions, *ESAIM: M2AN*, **46**(2):317–339, 2012. (Cited on pages 39, 73 and 103.)

[50] L. Esteva, Seismic risk and seismic design, In *Seismic Design for Nuclear Power Plants (R. J. Hansen, ed.)*, pp. 142–182, MIT Press, Cambridge MA, 1970. (Cited on page 2.)

[51] A. I. J. Forrester and A. J. Keane, Recent advances in surrogate-based optimization, *Prog. Aerosp. Sci.*, **45**(1):50–79, 2009. (Cited on page 73.)

[52] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration, In *Advances in Neural Information Processing Systems 31*, NeurIPS 2018, Montreal, 2018. (Cited on page 92.)

[53] H. C. Garner, E. W. E. Rogers, W. E. A. Acum, and E. C. Maskell, Subsonic wind tunnel wall corrections, AGARDo-graph 109, NATO, 1966. (Cited on pages 65 and 118.)

[54] F. Gatti and D. Clouteau, Towards blending physics-based numerical simulations and seismic databases using Generative Adversarial Network, *Comput. Methods Appl. Mech. Eng.*, **372**: 113421, 2020. (Cited on page 73.)

[55] W. Gautschi, How (un)stable are Vandermonde systems, In *Asymptotic and Computational Analysis (R. Wong, ed.)*, pp. 193–210, CRC Press, Boca Raton FI, 1990. (Cited on page 18.)

[56] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer, New York NY, 1991. (Cited on pages 3, 39, 73 and 103.)

[57] R. Ghanem, Ingredients for a general purpose stochastic finite elements implementation, *Comput. Methods Appl. Mech. Engrg.*, **168**(1–4):19–34, 1999. (Cited on page 3.)

[58] P. Groeneboom and G. Jongbloed, *Nonparametric Estimation under Shape Constraints: Estimators. Algorithms and Asymptotics*, Cambridge University Press, Cambridge, 2014. (Cited on pages 51 and 70.)

[59] W. Haase, F. Bradsma, E. Elsholz, M. Leschziner, and D. Schwamborn, *EUROVAL—An European Initiative on Validation of CFD Codes*, Vieweg Verlag, Wiesbaden, 1993. (Cited on pages 65, 66 and 118.)

[60] M. Hadigol and A. Doostan, Least squares polynomial chaos expansion: A review of sampling strategies, *Comput. Methods Appl. Mech. Eng.*, **332**: 382–407, 2018. (Cited on pages 99 and 136.)

[61] J. Hampton and A. Doostan, Compressive sampling methods for sparse polynomial chaos expansions, In *Handbook of Uncertainty Quantification (R. Ghanem, D. Higdon, and H. Owhadi, eds.)*, pp. 827–855, Springer, Cham, 2016. (Cited on page 106.)

[62] B. Hamzi and H. Owhadi, Learning dynamical systems from data: A simple cross-validation perspective, part I: Parametric kernel flows, *Physica D*, **421**:132817, 2021. (Cited on page 86.)

[63] B. Hamzi, R. Maulik, and H. Owhadi, Simple, low-cost and accurate data-driven geophysical forecasting with learned kernels, *Proc. R. Soc. A*, **477**(2252):20210326, 2021. (Cited on page 86.)

[64] S. Han, M. Tao, U. Topcu, H. Owhadi, and R. M. Murray, Convex Optimal Uncertainty Quantification, *SIAM J. Optim.*, **25**(3):1368–1387, 2015. (Cited on page 4.)

[65] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York NY, 2009. (Cited on page 119.)

[66] W. Hoeffding, Probability inequalities for sums of bounded random variables, *J. Am. Statist. Assoc.*, **58**(301):13–30 (1963). (Cited on page 8.)

[67] S. Hosder, R.W. Walters, and M. Balch, Point-collocation nonintrusive polynomial chaos method for stochastic computational fluid dynamics, *AIAA J.*, **48**(12):2721–2729, 2010. (Cited on page 9.)

[68] T. Ishigami and T. Homma, An importance quantification technique in uncertainty analysis for computer models, In *Proceedings First International Symposium on Uncertainty Modeling and Analysis, 3-5 December 1990, College Park MD*, pp. 398–403, 1990. (Cited on page 109.)

[69] A. Janon, T. Klein, A. Lagnoux, M. Nodet, and C. Prieur, Asymptotic normality and efficiency of two Sobol index estimators, *ESAIM: PS*, **18**:342–364, 2014. (Cited on page 120.)

[70] P. Jensen and J. Bard, Algorithms for constrained optimization, Supplement to: *Operations Research Models and Methods*, John Wiley & Sons, Hoboken NJ, 2003. (Cited on page 19.)

[71] A. I. Jordan, A. Mühlemann, and J. F. Ziegal, Characterizing the optimal solutions to the isotonic regression problem for identifiable functionals, *Ann. Inst. Stat. Math.*, 2021. (Cited on page 51.)

[72] P.-H. T. Kamga, B. Li, M. M. McKerns, L. H. Nguyen, M. Ortiz, H. Owhadi, and T. J. Sullivan, Optimal uncertainty quantification with model uncertainty and legacy data, *J. Mech. Phys. Solids*, **72**:1–19, 2014. (Cited on page 4.)

[73] B. Kanna and S. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *J. Mech. Des.*, **116**(2):405–411, 1994. (Cited on page 19.)

[74] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.*, **3**(6):422–440, 2021. (Cited on page 73.)

[75] A. Kidane, A. Lashgari, B. Li, M. M. McKerns, M. Ortiz, H. Owhadi, G. Ravichandran, M. Stalzerand, and T. Sullivan, Rigorous model-based uncertainty quantification with application to terminal ballistics, part I: Systems with controllable inputs and small scatter, *J. Mech. Phys. Solids*, **60**(5):983–1001, 2012. (Cited on pages 4, 9 and 137.)

[76] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, 2017. (Cited on page 92.)

[77] J. Kleijnen, Kriging metamodeling in simulation: a review, *Eur. J. Oper. Res.*, **192**(3):707–716, 2009. (Cited on page 73.)

[78] D. D. Kosambi, Statistics in function space, *J. Indian Math. Soc.*, **7**:76–88, 1943. (Cited on page 73.)

[79] D. G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, *J. South. Afr. Inst. Min. Metall.*, **52**(6):119–139, 1951. (Cited on pages 73 and 77.)

[80] D. G. Krige, A statistical analysis of some of the borehole values in the Orange Free State Goldfield, *J. South. Afr. Inst. Min. Metall.*, **53**(3):47–64, 1952. (Cited on pages 73 and 77.)

[81] J. Laurenceau and P. Sagaut, Building efficient response surfaces of aerodynamic functions with Kriging and Cokriging, *AIAA J.*, **46**(2):498–507, 2008. (Cited on page 78.)

[82] M. Lazareff and J. Peter, CFD quality assessment in NIPCM studies with "spiral" convergence criterion for DoE refinement, In *20th AIAA Computational Fluid Dynamics Conference, 27-30 June 2011, Honolulu HI*, AIAA paper #2011-3864, 2011. (Cited on page 28.)

[83] M. Lazareff, Towards autonomous discovery of stiff structures in CFD Design of Experiment space, from random-walk automaton steered by the "spiral" criterion, *Comput. Fluids*, **99**:67–82, 2014. (Cited on pages 18 and 28.)

[84] M. Ledoux, *The Concentration-of-Measure Phenomenon*, Mathematical Surveys and Monographs **89**, American Mathematical Society, Providence RI, 2001. (Cited on page 8.)

[85] J. de Leeuw, K. Hornik, and P. Mair, Isotone optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and active set methods, *J. Stat. Software*, **32**(5):1–24, 2009. (Cited on page 51.)

[86] L. Le Gratiet and J. Garnier, Recursive co-Kriging model for design of computer experiments with multiple levels of fidelity, *Int. J. Uncertain. Quantif.*, **4**(5):365–386, 2014. (Cited on page 78.)

[87] O. Le Maître and O. Knio, *Spectral Methods for Uncertainty Quantification. With Applications to Computational Fluid Dynamics*, Springer, Dordrecht, 2010. (Cited on pages 3, 39, 73 and 103.)

[88] T. Leonard and J. S. J. Hsu, *Bayesian Methods: An Analysis for Statisticians and Interdisciplinary Researchers*, Cambridge University Press, Cambridge, 2001. (Cited on page 3.)

[89] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag, New York NY, 2004. (Cited on page 2.)

[90] D. Liu, A. Litvinenko, C. Schillings, and V. Schulz, Quantification of airfoil geometry-induced aerodynamic uncertainties—Comparison of approaches, *SIAM/ASA J. Uncertain. Quantif.*, **5**(1):334–352, 2017. (Cited on page 78.)

[91] Z. Liu, D. Lesselier, B. Sudret, and J. Wiart, Surrogate modeling of indoor down-link human exposure based on sparse polynomial chaos expansion, *Int. J. Uncertain. Quantif.*, **10**(2):145–163, 2020. (Cited on page 91.)

[92] L.J. Lucas, H. Owhadi, and M. Ortiz, Rigorous verification, validation, uncertainty quantification and certification through concentration-of-measure inequalities, *Comput. Methods Appl. Mech. Engng.*, **197**(51-52):4591–4609, 2008. (Cited on pages 8, 9 and 10.)

[93] A. Marrel, B. Iooss, B. Laurent, and O. Roustant, Calculations of Sobol indices for the Gaussian process metamodel, *Reliab. Eng. Syst. Safety*, **94**(3):742–751, 2009. (Cited on page 109.)

[94] D. Maruyama, D. Liu, and S. Görtz, General introduction to surrogate model-based approaches to UQ, In *Uncertainty Management for Robust Industrial Design in Aeronautics (C. Hirsch, D. Wunsch, J. Szumbarski, Ł. Łaniewski-Wołłk, J. Pons-Prats, eds.)*, pp. 203–211, Springer, Cham, 2019. (Cited on page 78.)

[95] B. Matérn, *Spatial Variation*, Springer-Verlag, New York NY, 1960. (Cited on page 86.)

[96] L. Mathelin and K. A. Gallivan, A compressed sensing approach for partial differential equations with random input data, *Commun. Comput. Phys.*, **12**(4):919–954, 2012. (Cited on page 106.)

[97] L. Mathelin, L. Pastur, and O. Le Maître, A compressed-sensing approach for closed-loop optimal control of nonlinear systems, *Theor. Comput. Fluid Dyn.*, **26**(1-4):319–337, 2012. (Cited on page 73.)

[98] G. Matheron, *Traité de Géostatistique Appliquée, Tome I*, Mémoires du Bureau de Recherches Géologiques et Minières, no. 14. Éditions Technip, Paris, 1962. (Cited on pages 73 and 77.)

[99] G. Matheron, *Traité de Géostatistique Appliquée, Tome II: Le Krigeage*, Mémoires du Bureau de Recherches Géologiques et Minières, no. 24, Éditions B.R.G.M., Paris, 1963. (Cited on pages 73 and 77.)

[100] C. McDiarmid, On the method of bounded differences, *Surveys in Combinatorics, 1989: Invited Papers at the Twelfth British Combinatorial Conference (J. Siemons, ed.)*, pp. 148–188, Cambridge University Press, Cambridge, 1989. (Cited on pages 8 and 11.)

[101] M. M. McKerns, P. Hung, and M. A. G. Aivazis, Mystic: a simple model-independent inversion framework, 2009. (Cited on pages 12, 18, 24, 33, 36, 37, 43, 48 and 72.)

[102] M. M. McKerns, L. Strand, T. J. Sullivan, P. Hung, and M. A. G. Aivazis, Building a framework for predictive science, In *Proceedings of the 10th Python in Science Conference SciPy 2011 (S. van der Walt and J. Millman, eds.)*, pp. 67–78, 2011. (Cited on pages 12, 18, 24, 33, 36, 37, 43, 48, 68 and 72.)

[103] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, *AIAA J.*, **32**(8):1598–1605, 1994. (Cited on page 28.)

[104] C. A. Micchelli and T. J. Rivlin, A survey of optimal recovery, In *Optimal Estimation in Approximation Theory (C. A. Micchelli and T. J. Rivlin, eds.)*, pp 1–54, Springer, Boston MA, 1977. (Cited on page 75.)

[105] C. A. Micchelli and M. Pontil, Kernels for multi-task learning, In *NIPS'04: Proceedings of the 17th International Conference on Neural Information Processing*, 921–928, 2004. (Cited on pages 75 and 76.)

[106] E. H. Moore, *General Analysis Part II*, Memoirs of the American Philosophical Society, The American Philosophical Society, Philadelphia PA, 1939. (Cited on page 74.)

[107] Q. Mo and S. Li, New bounds on the restricted isometry constant δ_{2k}, *Appl. Comput. Harmon. Anal.*, **31**(3):460–468, 2011. (Cited on page 102.)

[108] T. Mukhopadhyay, S. Chakraborty, S. Dey, S. Adhikari, and R. Chowdhury, A critical assessment of Kriging model variants for high-fidelity uncertainty quantification in dynamics of composite shells, *Arch. Comput. Method. Eng.*, **24**(3):495–518, 2017. (Cited on page 77.)

[109] J. E. Nash and J. V. Sutcliffe, River flow forecasting through conceptual models part I — A discussion of principles, *J. Hydrol.*, **10**(3):282–290, 1970. (Cited on page 108.)

[110] T. Nguyen-Thien and T. Tran-Cong, Approximation of functions and their derivatives: A neural network implementation with applications, *Appl. Math. Model*, **23**(9):687–704, 1999. (Cited on page 73.)

[111] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Methods Eng.*, **17**:403–434, 2010. (Cited on page 73.)

[112] H. Owhadi, C. Scovel, T. Sullivan, M. M. McKerns, and M. Ortiz, Optimal Uncertainty Quantification, *SIAM Rev.*, **55**(2):271–345, 2013. (Cited on pages 4, 11, 12, 25, 33, 44, 48, 135 and 141.)

[113] H. Owhadi and C. Scovel, *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design*, Cambridge University Press, Cambridge, 2019. (Cited on page 75.)

[114] H. Owhadi and G. R. Yoo, Kernel Flows: From learning kernels from data into the abyss, *J. Comput. Phys.*, **389**:22–47, 2019. (Cited on pages 72, 86, 88, 94, 120, 134, 136 and 142.)

[115] H. Owhadi, Do ideas have shape? Plato's theory of forms as the continuous limit of artificial neural networks, arXiv:2008.03920, 2020. (Cited on pages 73, 75, 76 and 134.)

[116] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z., Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation in PyTorch, In *NIPS 2017 Workshop Autodiff*, 2017. (Cited on page 92.)

[117] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, Junjie, and S. Chintala, PyTorch: An imperative style, high-performance deep learning library, In *Advances in Neural Information Processing Systems 32 (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.)*, pp. 8024–8035, 2019. (Cited on page 92.)

[118] V. I. Paulsen and M. Raghupathi, *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, Cambridge University Press, Cambridge, 2016. (Cited on pages 73, 74, 96 and 97.)

[119] C. Prieur and S. Tarantola, Variance-based sensitivity analysis: theory and estimation algorithms, In *Handbook of Uncertainty Quantification (R. Ghanem, D. Higdon, and H. Owhadi, eds.)*, pp. 1217–1239, Springer, Cham, 2016. (Cited on page 120.)

[120] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, Surrogate-based analysis and optimization, *Prog. Aerosp. Sci.*, **41**(1):1–28, 2005. (Cited on page 73.)

[121] H. Rabitz, Ö. F. Aliş, J. Shorter, and K. Shim, Efficient input-output model representations, *Comput. Phys. Commun.*, **117**(1-2):11–20, 1999. (Cited on page 106.)

[122] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge MA, 2006. (Cited on pages 73 and 85.)

[123] S. Reich and C. J. Cotter, *Probabilistic Forecasting and Bayesian Data Assimilation*, Cambridge University Press, Cambridge, 2015. (Cited on page 137.)

[124] T. Robertson, F. T. Wright, and R. L. Dykstra, *Order Restricted Statistical Inference*, John Wiley & Sons, Chichester, 1988. (Cited on page 51.)

[125] H. H. Rosenbrock, An automatic method for finding the greatest or least value of a function, *The Computer Journal*, **3**(3):175–184, 1960. (Cited on page 114.)

[126] P. Ruhnau, A. Stahl, and C. Schnörr, On-line variational estimation of dynamical fluid flows with physics-based spatio-temporal regularization, In *Pattern Recognition: 28th DAGM Symposium, 12-14 September 2006, Berlin (K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer, eds.)*, pp. 444–454, Springer, Berlin, 2006. (Cited on page 137.)

[127] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Design and analysis of computer experiments, *Stat. Sci.*, **4**(4):409–423, 1989. (Cited on page 73.)

[128] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer-Verlag, New York NY, 2003. (Cited on page 73.)

[129] É. Savin, A. Resmini, and J. Peter, Sparse polynomial surrogates for aerodynamic computations with random inputs, In *18th AIAA Non-Deterministic Approaches Conference, 4-8 January 2016, San Diego CA*, AIAA paper #2016-0433, 2016. (Cited on pages 99, 106, 118 and 120.)

[130] É. Savin and B. Faverjon, Computation of higher-order moments of generalized polynomial chaos expansions, *Int. J. Numer. Methods Eng.*, **111**(12):1192–1200, 2017. (Cited on page 106.)

[131] A. Schwaighofer, V. Tresp, and K. Yu, Learning Gaussian process kernels via hierarchical bayes, In *NIPS'04: Proceedings of the 17th International Conference on Neural Information Processing Systems*, pp. 1209–1216, December 2004. (Cited on page 86.)

[132] A. Serafino, B. Obert, and P. Cinnella, Multi-fidelity gradient-based strategy for robust optimization in computational fluid dynamics, *Algorithms*, **13**(10):248, 2020. (Cited on page 78.)

[133] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, Metamodels for computer-based engineering design: survey and recommendations, *Eng. Comput.*, **17**(2):129–150, 2001. (Cited on page 73.)

[134] M. Skibinsky, The range of the $(n+1)$th moment for distributions on $[0,1]$, *J. Appl. Prob.*, **4**(3):543–552, 1967. (Cited on page 15.)

[135] C. Soize and R. G. Ghanem, Physical systems with random uncertainties: Chaos representations with arbitrary probability measure, *SIAM J. Sci. Comput.*, **26**(2):395–410, 2004. (Cited on pages 39, 73 and 103.)

[136] P. R. Spalart and S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, In *30th Aerospace Sciences Meeting and Exhibit, 6-9 January 1992, Reno NV*, AIAA paper #1992-0439, 1992. (Cited on pages 65 and 118.)

[137] M. L. Stein, *Interpolation of Spatial Data*, Springer-Verlag, New York NY, 1999. (Cited on pages 85 and 86.)

[138] J. Stenger, F. Gamboa, M. Keller, and B. Iooss, Optimal Uncertainty Quantification of a risk measurement from a thermal-hydraulic code using canonical moments, *Int. J. Uncertain. Quantif.*, **10**(1):35–53, 2020. (Cited on pages 15 and 18.)

[139] J. Stenger, *Optimal Uncertainty Quantification of a Risk Measurement from a Computer Code*, PhD thesis, Université de Toulouse, October 2020. (Cited on pages 15, 17 and 18.)

[140] R. Storn and K. Price, Differential Evolution—A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, **11**(4):341–359, 1997. (Cited on pages 18, 21, 29, 40 and 68.)

[141] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, *Reliab. Eng. Syst. Saf.*, **93**(7):964–979, 2008. (Cited on page 106.)

[142] B. Sudret, Polynomial chaos expansions and stochastic finite element methods, In *Risk and Reliability in Geotechnical Engineering (K.-K. Phoon and J. Ching, eds.)*, pp. 624, CRC Press, Boca Raton FI, 2014. (Cited on page 108.)

[143] T. J. Sullivan, U. Topcu, M. M. McKerns, and H. Owhadi, Uncertainty quantification via codimension-one partitioning, *Int. J. Numer. Meth. Eng.*, **85**(12):1499–1521 (2011). (Cited on page 8.)

[144] T. J. Sullivan, M. M. McKerns, D. Meyer, F. Theil, H. Owhadi, and M. Ortiz, Optimal Uncertainty Quantification for legacy data observations of Lipschitz functions, *ESAIM: M2AN*, **47**(6):1657–1689, 2016. (Cited on page 4.)

[145] T. J. Sullivan, *Introduction to Uncertainty Quantification*, Springer, Cham, 2014. (Cited on pages 12, 13, 23 and 31.)

[146] G. Sun and S. Wang, A review of the artificial neural network surrogate modeling in aerodynamic design, *Proc. IMechE Part G: J. Aerospace Engineering*, **233**(16):5863–5872, 2019. (Cited on page 73.)

[147] L. Sun and J.-X. Wang, Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data, *Theor. Appl. Mech. Lett.*, **10**(3):161–169, 2020. (Cited on page 73.)

[148] T. Tao, 254A, Notes 1: Concentration of measure, https://terrytao.wordpress.com/2010/01/03/254a-notes-1-concentration-of-measure/, 2010. (Cited on page 8.)

[149] R. Thirumalainambi and J. Bardina, Training data requirement for a neural network to predict aerodynamic coefficients, In *Proceedings Volume 5102, Independent Component Analyses, Wavelets, and Neural Networks*, 2003. (Cited on page 73.)

[150] R. J. Tibshirani, H. Hoefling and R. Tibshirani, Nearly-isotonic regression, *Technometrics*, **53**(1):54–61, 2011. (Cited on page 51.)

[151] U. Topcu, L. J. Lucas, H. Owhadi and M. Ortiz, Rigorous uncertainty quantification without integral testing, *Reliab. Eng. Syst. Saf.*, **96**(9):1085–1091, 2011. (Cited on page 9.)

[152] E. van den Berg and M. P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, *SIAM J. Sci. Comput*, **31**(2):890–912, 2008. (Cited on pages 108 and 120.)

[153] E. van den Berg and M. P. Friedlander, Sparse optimization with least-squares constraints, *SIAM J. Optim.*, **21**(4):1201–1229, 2011. (Cited on pages 108 and 120.)

[154] P. Venkataraman, *Applied Optimization with MATLAB Programming, 2nd Edition*, John Wiley & Sons, Hoboken NJ, 2009. (Cited on page 19.)

[155] R. Wallach, B. Mattos, R Girardi, and M. Curvo, Aerodynamic coefficient prediction of transport aircraft using neural network, In *44th AIAA Aerospace Sciences Meeting and Exhibit, 09-12 January 2006, Reno, NV*, AIAA paper #2006-658, 2006. (Cited on page 73.)

[156] M. P. Wand and M. D. Jones, *Kernel Smoothing*, Chapman and Hall/CRC, Boca Raton FI, 1995. (Cited on pages 112, 117, 123, 126 and 130.)

[157] R. Ward, Compressed sensing with cross validation, *IEEE Trans. Inf. Theory*, **55**(12):5773–5782, 2009. (Cited on page 102.)

[158] N. Wiener, The homogeneous chaos, *Amer. J. Math.*, **60**(4):897–936, 1938. (Cited on page 103.)

[159] J. Weinmeister, X. Gao, and S. Roy, Analysis of a polynomial chaos-Kriging metamodel for uncertainty quantification in aerodynamics, *AIAA J.*, **57**(6):2280–2296, 2019. (Cited on page 77.)

[160] D. C. Wilcox, Formulation of the k–ω turbulence model revisited, *AIAA J.*, **46**(11):2823–2838, 2008. (Cited on page 28.)

[161] C. K. I. Williams and C. E. Rasmussen, Gaussian processes for regression, In *Advances in Neural Information Processing Systems 8 (D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, eds.)*, pp. 514–520, MIT Press, Cambridge MA, 1996. (Cited on page 86.)

[162] D. Xiu and G. E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, **24**(2):619–644, 2002. (Cited on pages 39, 73 and 103.)

[163] D. Xiu and J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.*, **27**(3):1118–1139, 2005. (Cited on page 106.)

[164] L. Yan, X. Duan, B. Liu, and J. Xu, Gaussian processes and polynomial chaos expansion for regression problem: Linkage via the RKHS and comparison via the KL divergence, *Entropy*, **20**(3):191, 2018. (Cited on pages 73 and 103.)

[165] M. Yoshimura, K. Shimoyama, T. Misaka, and S. Obayashi, Topology optimization of fluid problems using genetic algorithm assisted by the Kriging model, *Int. J. Numer. Methods Eng.*, **109**(4):514–532, 2017. (Cited on page 77.)

[166] X. Zhang, F. Xie, T. Ji, Z. Zhu, and Y. Zheng, Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization, *Comput. Methods Appl. Mech. Eng.*, **373**:113485, 2021. (Cited on page 73.)

[167] L. R. Zuhal, K. Zakaria, P. S. Palar, K. Shimoyama, and R. P. Liem, Polynomial-Chaos–Kriging with gradient information for surrogate modeling in aerodynamic design, *AIAA J.*, **59**(8):2950–2967, 2021. (Cited on page 78.)