Surrogate Modeling for Stochastic Simulators Using Statistical Approaches

Zhu Xujia

DISS. ETH NO. 28965

Surrogate Modeling for Stochastic Simulators Using Statistical Approaches

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by

Xujia Zhu
Dipl. Ing., Ecole Polytechnique
M.Sc., Technische Universität München
born on 15.01.1991
citizen of P.R. China

accepted on the recommendation of

Prof. Dr. Bruno Sudret, examiner
Prof. Dr. Josselin Garnier, co-examiner
Prof. Dr. Bozidar Stojadinović, co-examiner
Prof. Dr. Marco Broccardo, co-examiner
Dr. Stefano Marelli, co-examiner

2022-12-16

To my family and friends

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Sudret for having offered me the opportunity to work at his amazing group — Chair of Risk, Safety and Uncertainty Quantification at ETH Zurich. His continuous support, guidance, and instruction throughout these 5 years of my Ph.D. studies have not only led me to the scientific developments presented in this manuscript but also significantly influenced and shaped my personal and professional growth. Rigorous, patient, and constructive, he showed me a perfect example of an outstanding professor and scientist. Without him, I could have never reached this far in research.

I am very grateful to Prof. Garnier and Prof. Stojadinović for serving as members of my examination committee. Their feedback has helped me to improve the quality of the thesis.

I am eternally thankful to my colleagues at our chair. Stefano was always available to discuss scientific and "uncertain" topics. His rich experience in research, teaching, and life has encouraged and inspired me in all aspects of my Ph.D. Whenever I need help, he is willing to give advice as a friend. It is so nice to have had Nora work on the same project. Her critical thinking and solid mathematical background prompted plenty of fruitful discussions. Considerate and empathetic, she provided many invaluable suggestions for the thesis as well as for various problems I encountered. I am so happy to have worked and traveled with Paul. Like stochastic spectral embedding, he embeds a spectrum of talents: the director of the chair's video, an expert in Adobe products, a Lego crafter, a competitive gamer, and so on and so forth. It is always relaxing and entertaining to talk to him no matter the subject. with unwavering reliability, Moustapha optimizes a wide range of skills: research, supervision, travel, dance, etc. He has given very constructive comments on the first draft of the thesis, and he also generously volunteered to serve as my French tutor since I joined the chair. Emiliano was my officemate for two years. His expertise in copula and profound understanding of statistics have left me a treasure. Damar, the principal UQWorld developer, has inspired and deeply impressed me with his broad knowledge and humble attitude. Styfen is a big fan of programming in all languages, and I enjoy exchanging ideas with him on the development of artificial intelligence. Anderson is a thinker, and we have had tons of dazzling chats about research, politics, philosophy, and many other things. Katerina is always very enthusiastic. Passionate about machine learning, regular expression, and beach volleyball, she has injected vibrant energy into our working atmosphere. Adela is conscientious and warmhearted, and her stories (especially about her son Jakub) have the power to relax and cheer me up. Rui is an outstanding developer and football player. We have had countless discussions on work, courses, probability theory, and life. Christos, one of the main contributors to UQLab and UQCloud, has backed me up with his extraordinary IT magics. Philippe, a master of sensitivity analysis, brought numerous enjoyable conversations and joyful moments. I would like to thank the two assistants of the chair, Margaretha and Nicole warmly. Their support and help with all types of administrative documents have certainly relieved me a lot. It has also been a great pleasure to meet Biswarup and the visiting guests: Fritz, Henrique, René, Chiara, Max, Marco, and Tong during their stay in Zurich. I will never forget the time spent at the chair, which I will cherish forever.

In my Ph.D. studies, I was extremely lucky to meet many excellent professors and researchers from whom

I have learned a great deal. Sincere thanks to Prof. van der Geer. She was so kind to pick up my inquiry and have a long meeting with me about estimation theory despite the fact I was a newbie in this field. Moreover, her course on empirical processes helped me build my background knowledge of mathematical statistics. I am grateful to Prof. Broccardo for introducing the world of earthquake engineering to me. I enjoyed discussing with him the potential of advanced machine-learning methods for engineering applications. Many thanks to my collaborators Giuseppe, Nikos, and Saubin for sharing their expertise with me. Their invaluable contributions have played a significant role in the success of our collaborations.

My friends have constituted my family abroad throughout my long educational journey far from my family. Many thanks to Ao, Yanbo, Rui, Lionel, Sihan, Jeremy, Chongmo, Chenzhang, Yixuan, Xing, Yu, Hung-Sheng, Zhiyi, Qiao, Hongping, Feng, Bifeng, and Luchi. The list could have been extended much longer, but I have to stop at some point. Special thanks to my godfather Thierry for making me feel at home in France and helping me with cultural integration. Their company and encouragement have helped me navigate the numerous challenges that I encountered along the way.

Last but certainly not least, I want to express my wholehearted gratitude to my family. Standing by my side all the time, my girlfriend Ye has brought so much joy and surprise into my life. Her unwavering support and unconditional love have lifted me up during challenging times. Through the years and far away, I was unable to be with my parents during happy or sorrowful moments. Nonetheless, they have always been considerate and present, offering comfort and relief from afar. They might not understand the intricacies of my research, but this manuscript is the fruit of our joint efforts.

Xujia Zhu
March 17, 2023
in Zurich, Switzerland

Abstract

Nowadays, more and more complex interdependent infrastructures and networks are developed in engineering. The design and maintenance of such systems increasingly call for advanced computational models to optimize their performance and assess their reliability under various operational conditions. Unlike many conventional simulators that are deterministic, stochastic simulators feature intrinsic stochasticity. More precisely, they produce different results when run multiple times with a given set of input parameters. Due to this random nature, repeated model evaluations of the stochastic model with the same input value, called replications, are necessary to fully characterize the probability distribution of the associated model response.

For the purpose of optimization or uncertainty quantification (e.g., uncertainty propagation or sensitivity analysis), computational models typically need to be evaluated a large number of times. The additional layer of randomness due to the intrinsic stochasticity of stochastic simulators makes it even more computationally demanding to perform these complex analyses. A common practice to alleviate the prohibitive cost associated with expensive simulators is to build surrogate models, which behave similarly to the original model but are much cheaper to evaluate.

Contrary to the deterministic case, surrogate modeling of stochastic simulators has only emerged in the past decade. The main challenge in this field is that one model evaluation yields only a single realization of the random model response associated with the given input value. In other words, one run of a stochastic simulator provides proportionally much less information than that of a deterministic one.

This thesis focuses on developing efficient and accurate surrogate models to emulate the response distribution of stochastic simulators, combining statistical methods with state-of-the-art deterministic surrogate modeling techniques.

To this end, we propose two new approaches: the generalized lambda model (GLaM) and the stochastic polynomial chaos expansion (SPCE). The first one capitalizes on the use of the generalized lambda distribution to characterize the random nature of the simulator response. The distribution parameters are functions of the input variables and are represented by polynomial chaos expansions (PCEs). We explore replication-based methods to build GLaMs and improve their performance by an additional joint optimization of the overall likelihood function. We further elaborate this idea and develop a new method that does not require replications. Using this surrogate, we investigate sensitivity analysis for stochastic simulators.

The second class of stochastic surrogates, SPCE, overcomes the main shortcoming of GLaM, which is unable to represent multimodal distributions. In this more versatile stochastic emulator, we extend PCE by introducing an artificial latent variable to the expansion and an additive noise variable to mimic the intrinsic stochasticity of the simulator. We also propose an adaptive algorithm to construct the surrogate model without the need for replications.

For both stochastic surrogate models, we investigate basic theoretical properties of the primary estimation method. Analytical examples and engineering applications, including wind turbine design and seismic fragility analysis, are used to validate and illustrate the performance of the new approaches. Furthermore, these engi-

neering case studies provide valuable insights into the applicability of the developed framework to real-world industrial problems.

Résumé

De nos jours, de plus en plus d'infrastructures complexes et de réseaux interdépendants sont développés en ingénierie. La conception et la maintenance de ces systèmes font appel à des modèles numérique avancés afin d'optimiser leur performances et d'évaluer leur fiabilité dans diverses conditions opérationnelles. Contrairement à de nombreux simulateurs conventionnels qui sont déterministes, les simulateurs dits stochastiques possèdent une stochasticité intrinsèque. Plus précisément, ils produisent des résultats différents lorsqu'ils sont exécutés plusieurs fois avec les mêmes paramètres d'entrée. En raison de ce caractère aléatoire, des évaluations répétitives d'un modèle stochastique avec la même valeur d'entrée, appelées réplications, sont nécessaires pour caractériser entièrement la loi de probabilité de la réponse associée.

Dans un contexte d'optimisation ou de quantification des incertitudes (e.g., propagation des incertitudes ou analyse de sensibilité), les modèles numeriques doivent généralement être évalués un grand nombre de fois. La couche aléatoire supplémentaire due à la stochasticité intrinsèque des simulateurs stochastiques rend ces analyses encore plus exigeantes sur le plan des ressources informatiques. Une façon courante d'alléger les coûts de calculs prohibitifs consiste à construire des métamodèles, qui se comportent de manière similaire au modèle original, mais dont l'évaluation est beaucoup moins coûteuse.

Contrairement aux métamodèles développés pour les simulateurs déterministes, la métamodelisation des simulateurs stochastiques n'est apparue qu'au cours de la dernière décennie dans diverses applications d'ingénierie. L'enjeu principal dans ce domaine est qu'une évaluation du modèle ne produit qu'une seule réalisation de la réponse aléatoire associée à la valeur d'entrée. En d'autres termes, une exécution d'un simulateur stochastique fournit beaucoup moins d'information que dans le cas d'un simulateur déterministe.

Cette thèse est dédiée au développement de métamodèles efficaces et précis pour émuler la loi de probabilité de la réponse de simulateurs stochastiques, en combinant des méthodes statistiques avec des techniques de pointe de métamodelisation déterministe.

Pour cela, nous proposons deux métamodèles : le modèle « lambda généralisé » (MLaG) et les polynômes de chaos stochastiques (PCS). Le premier modèle capitalise sur l'utilisation de la loi lambda généralisée pour caractériser la réponse aléatoire. Les paramètres de la distribution sont des fonctions des variables d'entrée et sont représentés par polynômes de chaos. Nous explorons des méthodes basées sur la réplication pour construire des MLaGs et améliorer leurs performances par une optimisation conjointe supplémentaire de la fonction de vraisemblance globale. Nous approfondissons cette idée et développons une nouvelle méthode qui ne nécessite pas de réplications. À l'aide de ce métamodèle, nous étudions l'analyse de sensibilité pour simulateurs stochastiques.

La deuxième catégorie de métamodèles stochastiques PCS permet de représenter des distributions multimodales, ce que ne permet pas l'approache MLaG. Dans cet émulateur stochastique plus flexible, nous introduisons une variable latente artificielle dans l'expansion des polynômes de chaos et une variable de bruit additive pour imiter la stochasticité intrinsèque du simulateur. Nous proposons également un algorithme adaptatif pour construire ce modèle sans avoir besoin de réplications.

Pour les deux métamodèles stochastiques proposés, nous étudions certaines propriétés théoriques sur la

méthode d'estimation principale. Des exemples analytiques et des applications d'ingénierie, y compris la conception d'éoliennes et l'analyse de fragilité sismique, valident et illustrent la performance des nouvelles approches. En outre, ces études des cas d'ingénierie fournissent des indications précieuses sur l'applicabilité des méthodes développées à des problèmes industriels réels.

Contents

Our responsibility is to do what we can, learn what we can, improve the solutions, and pass them on.

Richard Phillips Feynman

1 Overview

1.1 Motivation

Understanding the world has always been a driving force of scientific progress, from ancient Egyptians to Greek philosophers (Grant, 2007), from "scientific revolution" to "Kelvin's clouds" (Thomson, 1901). Derived from philosophy, science nowadays factors in knowledge in the form of principles and laws, usually obtained by observations and experiments. Based on these discoveries and reasonable assumptions, mathematical models are developed to represent real-life scenarios with mathematical concepts and language. For example, one of the fundamentals of fluid dynamics is the Navier–Stokes equations (Landau and Lifshitz, 1987). This system of partial differential equations lies on the continuum assumption and encodes mathematically the conservation of mass, momentum, and energy. In addition, assumptions (usually based on experimental data) on the boundary and initial conditions and constitutive laws (e.g., Newtonian fluid) that link the stress field to flow velocity should be introduced to depict the operational conditions and physical properties.

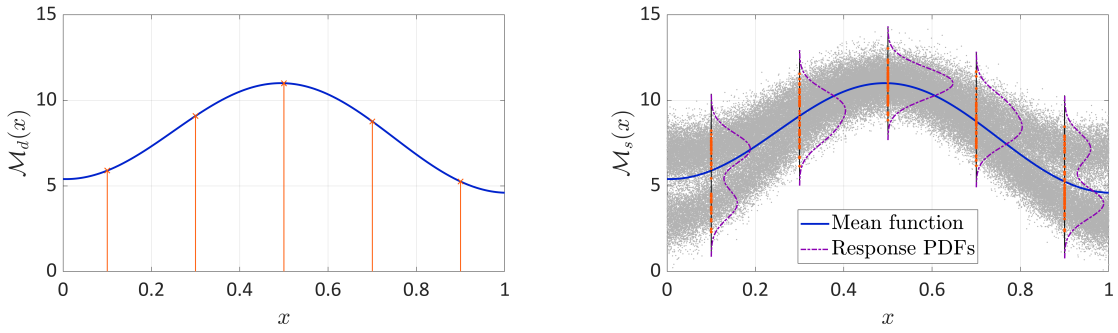
To study the behavior of a given system, one needs to solve the equations of its mathematical representation. However, complex problems usually do not have analytical closed-form solutions. To this end, *numerical analysis* has been developed to study the properties of the solution and tackle it numerically (Allaire, 2005). Typical methods applied to problems governed by partial differential equations include finite difference methods (Smith, 1985), finite element methods (Zienkiewicz et al., 2013), and finite volume methods (LeVeque, 2002). The mathematical model together with the numerical solver is a computational model, also known as a *simulator*, as it simulates the system behavior. In the last century, the swift development of computer architecture and computational software has substantially fueled computational models that can accurately mirror the underlying real-world system. Therewith, scientists and engineers can perform "numerical experiments" *in silico*. Nowadays, simulators are indispensable in all fields of science and engineering, e.g., physics, chemistry, biology, economics, mechanical and civil engineering.

By nature, a computational model takes a set of variables called the *input* and maps them to the *output*. The input consists typically of parameters from the mathematical model that characterize the system, such as the boundary/load conditions and constitutive material properties. The output contains some important

engineering quantities of the system in response to the input, e.g., displacement, deformation, and internal forces.

The input-output relation of a classical simulator is usually deterministic: for a given set of values of the input variables, the corresponding output value is unique. In other words, running repeatedly a deterministic simulator for the same input parameters will always give the same outcome. As an example, the mechanical behavior (output) of a well-defined structure is deterministic with respect to the external loads (input).

In contrast to their deterministic counterparts, *stochastic simulators*, as the name suggests, exhibit a stochastic behavior: for a fixed set of input parameters, several model evaluations produce different output values. More precisely, the model response is random given the input value, and each model run returns only a single possible outcome. As an illustration, Figure 1.1 compares a one-dimensional deterministic simulator with a stochastic one.



(a) Deterministic simulator

(b) Stochastic simulator

Figure 1.1: Comparisons of deterministic and stochastic simulators.

The data scattering observed in the stochastic simulator (Fig. 1.1b) seems unrealistic at first glance. However, real-world experiments are stochastic, and we can never be sure to obtain the exact same results even by fixing the same experimental conditions. This is because some noise or relevant variables cannot be identified or controlled, and thus they are not taken into account a priori. These factors are usually random and thus inject uncertainty into the predictions. To replicate this behavior in virtual experiments, one introduces some intrinsic stochasticity on top of the physical parameters, which makes the simulator stochastic. Despite this complex behavior, stochastic simulators are broadly developed and applied, especially in economics and social science, where stochastic processes are commonly used in modeling. In engineering, notwithstanding that most computational models are deterministic, stochastic simulators are increasingly deployed in different fields. Two examples of modern engineering are described below.

- **Wind turbine simulation**: The robust design of wind turbines requires analyzing the structural components under diverse wind conditions across the lifespan of the turbine. A typical simulator in this field consists of two submodels, namely the wind generator and the aero-servo-elastic simulator. The wind generator takes as input some macroscopic descriptors of the wind climate, such as the mean wind velocity, the turbulence intensity, and the wind shear exponent, which characterizes the variation of the mean velocity with altitude. These characteristic values are usually combined with a power spectral model to generate a coherent stochastic wind field (both in time and space; Jonkman, 2009). Then, the turbulent inflow is injected into the subsequent aero-servo-elastic model to simulate the complex multi-physics

scenario including mutual interactions of wind inflow, aerodynamics, structural dynamics, and control systems. Because of the stochastic wind generator, a given set of macroscopic wind parameters does not determine a unique wind profile, and thus the associated structural response is random.

- **Seismic fragility analysis**: In performance-based earthquake engineering, seismic loads are characterized by some summary statistics of ground motions, called *intensity measures*. A non-exhaustive list of conventional intensity measures includes peak ground acceleration, spectral acceleration, peak ground velocity, and Arias intensity. Because these parameters are only filtered quantities, they cannot uniquely determine the detailed time series. In other words, one can find (or generate) infinite earthquake signals that share the same values of intensity measures. Consequently, the level of damage incurred by a structure under various earthquakes sharing the same intensity measures is a random variable, rather than a deterministic value.

Because of the random behavior of the model response, it is generally necessary to repeatedly evaluate a stochastic simulator with the same input to fully characterize the probability distribution of the associated output. Such a repetitive procedure is called *replication* in the stochastic simulation literature. In addition, a simulator needs to be run for various configurations of the system (which entails various values of the input) for the purpose of optimization or uncertainty quantification. These two issues altogether require a large number of model runs for performing complex analyses with stochastic simulators. This is practically intractable for high-fidelity models, for which a single run may take hours or days.

To alleviate the computational burden, *surrogate models*, also known as *emulators* or *metamodels*, can be constructed as a proxy of the original simulator. Such models mimic the input-output relation of the target simulator but are much cheaper and easier to evaluate. In the past decades, surrogate modeling of deterministic models has gained increasing attention (Vapnik, 1995; Rasmussen and Williams, 2006; Blatman and Sudret, 2011; Chevreuil et al., 2015). These surrogate models consist in approximating the deterministic mapping of the simulator and have been successfully applied across different disciplines (see, e.g., Forrester et al., 2008; Asher et al., 2015; Harenberg et al., 2019; Tröndle et al., 2020; Lauvernet and Helbert, 2020) enabling complex analyses involving expensive simulations. Due to their stochastic nature, however, stochastic simulators cannot be directly emulated by deterministic surrogate models. Indeed, the development of stochastic emulators has only emerged recently in the field of uncertainty quantification and still remains in its infancy.

1.2 Contribution

The objective of the manuscript is to develop accurate and efficient surrogate modeling methods for stochastic simulators. In particular, the surrogate models should fulfill the following requirements for general-purpose use:

- **Full representation**: to yield the full probability distribution of the model response for any input value, as well as some important probabilistic quantities of interest (as functions of the input), e.g., mean, variance, and quantiles.
- **Flexibility**: to cover a wide range of distributions, i.e., without introducing restrictive assumptions (e.g., normality) on the response distribution.

As indicated above, the response distribution is of main interest in this thesis. In addition, the surrogate model should also be able to provide some summary quantities of the distribution, either as a direct feature or through simple post-processing, e.g., by sampling the emulator.

For practical implementations and applications, the model construction process should satisfy the following requirements:

- **Non-intrusiveness**: to treat the stochastic simulator as a *black box* and build the surrogate in a data-driven manner, i.e., without adapting or modifying the computational model itself.
- **Adaptivity**: to construct the surrogate model adaptively under a finite amount of available data.
- **Versatility**: to be compatible with training samples generated following different schemes, especially when replications are not available.

Surrogate models of interest should be as flexible as possible to have wide applicability. However, when building them from finite data, the more flexible they are, the more data are necessary to ensure an accurate approximation. This is the so-called *bias–variance trade-off* in statistical learning (Hastie et al., 2001). As a result, a crucial point is to find a suitable model flexibility to maximize its performance under finite samples.

To address these requirements, we have developed mainly two surrogate models in this Ph.D. thesis: the generalized lambda model (GLaM) and the stochastic polynomial chaos expansion (SPCE).

GLaM capitalizes on the generalized lambda distribution (GLD) to approximate the response distribution. Such a choice allows us to cover a wide range of probability distributions (especially unimodal distributions) while remaining in a parametric setup. The distribution parameters, as functions of the input, are represented by polynomial chaos expansions (PCEs). By combining GLD and PCE, the generalized lambda model inherits their respective advantages and meets the aforementioned desired properties. To build up such a surrogate in a non-intrusive and adaptive manner, we have developed replication-based approaches and further improved their performance through a joint optimization step, thus leveraging all the available information. This method is not restricted to GLaM but can be adapted to other surrogates with parametric features. To fulfill the last requirement (i.e., versatility), we have extended the method by combining advanced statistical learning techniques to bypass the need for replications.

The performance of GLaM is limited by GLD, as the latter is unable to represent multimodal distributions. To cope with more complex problems, we have proposed the stochastic polynomial chaos expansion. This surrogate model does not make any assumption on the shape of the response distribution. Instead, it introduces an artificial latent variable and a noise variable on top of the well-defined input to mimic the intrinsic stochasticity of the simulator. More precisely, the emulator is expressed as a PCE of both the input and the latent variables plus the noise term. The model exhibits high flexibility and allows for approximating accurately both unimodal and multimodal distributions. As a consequence of the properties of PCE, several quantities of interest can be calculated analytically from the model. Besides, one can easily sample the emulated response distribution, thanks to the efficient PCE representation. To construct the model following the requirements, we have developed an adaptive algorithm that does not require replications.

The developed methods have been validated on several analytical examples and applications from various fields, including the two engineering examples presented in Section 1.1. Furthermore, we have investigated the topic of sensitivity analysis for stochastic simulators using the surrogate models.

The research of this thesis has produced six published international journal papers (Zhu and Sudret, 2020, 2021c,d, 2023; Tsokanas et al., 2021; Saubin et al., 2021), one submitted journal article (currently under review; Zhu et al., 2023), three peer-reviewed conference papers with accompanying talks (Zhu and Sudret, 2019c, 2022b; Zhu et al., 2022), seven conference presentations (Zhu and Sudret, 2019d,b, 2021f,b,e,a, 2022a), and two conference posters (Zhu and Sudret, 2018, 2019a).

1.2.1 Journal papers

- Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275. DOI:10.1615/Int.J.UncertaintyQuantification.2020033029.
- Zhu, X. and Sudret, B. (2021). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380. DOI:10.1137/20M1337302.
- Zhu, X. and Sudret, B. (2021). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815. DOI:10.1016/j.ress.2021.107815.
- Zhu, X. and Sudret, B. (2023). Stochastic polynomial chaos expansions to emulate stochastic simulators. *International Journal for Uncertainty Quantification*, 13:31–52. DOI:10.1615/Int.J.UncertaintyQuantification.2022042912.
- Zhu, X., Broccardo, M., and Sudret, B. (2023). Seismic fragility analysis using stochastic polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 72:103413. DOI:10.1016/j.probengmech.2023.103413.
- Tsokanas, N., Zhu, X., Abbiati, G., Marelli, S., Sudret, B., and Stojadinović, B. (2021). A global sensitivity analysis framework for hybrid simulation with stochastic substructures. *Frontiers in Built Environment*, 7:1–12. DOI:10.3389/fbuil.2021.778716.
- Saubin, M., de Mita, S., Zhu, X., Sudret, B., and Halkett, F. (2021). Impact of ploidy and pathogen life cycle on resistance durability. *Peer Community Journal*, 1:1–12. DOI:10.24072/pcjournal.10.

1.2.2 Conference papers

- Zhu, X. and Sudret, B. (2020). Surrogating the response PDF of stochastic simulators using generalized lambda distributions. *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13), Seoul, South Korea.* May 2019. DOI:10.22725/ICASP13.086.
- Zhu, X. and Sudret, B. (2022). Introducing latent variables in polynomial chaos expansions to surrogate stochastic simulators. *Proceedings of the 13th International Conference on Structural Safety and Reliability (ICOSSAR 2021-2022). Shanghai, China.* DOI:10.3929/ethz-b-000572535.
- Zhu, X., Broccardo, M., and Sudret, B. (2022). Use of generalized lambda models for seismic fragility analysis. *Proceedings of the 8th International Symposium on Reliability Engineering and Risk Management (ISRERM 2022), Hannover, Germany.* DOI:10.3929/ethz-b-000551727.

1.2.3 Talks (speaker is underlined)

- Zhu, X. and Sudret, B. (2019). Use of generalized lambda distributions to emulate stochastic simulators. *3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019), Crete Island, Greece.* June 2019. DOI:10.3929/ethz-b-000352754.

- Zhu, X. and Sudret, B. (2019). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. In *9th International Conference on Sensitivity Analysis of Model Output (SAMO 2019), Universitat Oberta de Catalunya.* October 2019. DOI:10.3929/ethz-b-000394777.
- Zhu, X. and Sudret, B. (2021). Stochastic polynomial chaos expansions for emulating stochastic simulators. *Workshop on Stochastic Simulators, Paris, France.* March 2021. DOI:10.3929/ethz-b-000526108.
- Zhu, X. and Sudret, B. (2021). Emulating the response distribution of stochastic simulators. *MASCOT-NUM Workshop (MASCOT-NUM 2021), Aussois, France.* April 2021. DOI:10.3929/ethz-b-000501633.
- Zhu, X. and Sudret, B. (2021). Metamodels of stochastic simulators using polynomial chaos expansions with latent variables. *Engineering Mechanics Institute Conference and Probabilistic Mechanics and Reliability Conference (EMI/PMC 2021), Columbia University, New-York (USA)* May, 2021. DOI:10.3929/ethz-b-000525701.
- Zhu, X. and Sudret, B. (2021). Construction of sparse polynomial chaos surrogate models for simulators with mixed continuous and categorical variables. *4th International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNECOMP 2021), Athens, Greece.* June 2021. DOI:10.3929/ethz-b-000493029.
- Zhu, X. and Sudret, B. (2022). Extension of polynomial chaos expansions to the metamodeling of stochastic simulators. *SIAM Conference on Uncertainty Quantification (UQ 2022), Atlanta, GA, USA,* April 2022. DOI:10.3929/ethz-b-000542493.

1.2.4 Conference posters

- Zhu, X. and Sudret, B. (2018). Surrogating the response PDF of stochastic simulators using parametric & semi-parametric representations. *MASCOT-NUM Workshop (MASCOT-NUM 2018), Nantes, France.* March 2018. DOI:10.3929/ethz-b-000309591.
- Zhu, X. and Sudret, B. (2019). Emulating the response PDF of stochastic simulators using sparse generalized lambda models. *MASCOT-NUM Workshop (MASCOT-NUM 2019), Rueil-Malmaison, France.* March 2019. DOI:10.3929/ethz-b-000336526.

1.3 Outline

The dissertation is organized following the format of a cumulative thesis with three parts. Part I provides a general introduction to the topic, including a comprehensive literature review on all the methods that are further used in Part II. Part II presents our main contribution to the development of stochastic surrogate models. A few related discussions and collaborations are presented in the appendices. The content of each chapter is briefly summarized as follows.

Chapter 2 recaps the classical uncertainty quantification framework for deterministic computational models. To set up the mathematical foundations for a self-consistent presentation throughout the thesis, this chapter offers a summary of probability theory with a focus on the concept of random variables and probability distributions. Then, it introduces some classical methods for uncertainty propagation and discusses the principles of sensitivity analysis. Finally, two conventional surrogate models developed for emulating deterministic simulators, namely polynomial chaos expansions (PCEs) and Gaussian processes, are presented with a special emphasis on PCE, as it is applied across the main developments of the thesis. Note that an overall rather mathematical

formalism is chosen to facilitate the subsequent discussions and to bring more insights into uncertainty quantification.

Chapter 3 provides a thorough review of the state-of-the-art developments for surrogate modeling of stochastic simulators. Depending on their nature, the methods are grouped into three categories: statistical, replication-based, and random field approaches. This chapter starts with a brief presentation of the essential elements of statistical learning, which serve as the basis for statistical approaches. Then, it presents the statistical models for estimating some probabilistic quantities of interest of the response distribution, e.g., mean, variance, and quantiles. More importantly, it summarizes the methods developed for fitting the entire response distribution, including parametric and nonparametric models from classical statistics as well as related developments from machine learning. Finally, the major ideas of the replication-based and random field approaches are discussed to complete the review.

Chapter 4 presents a novel stochastic surrogate model called *generalized lambda model* (GLaM) and a few replication-based methods used in its construction. As a major component of GLaM, the probabilistic properties of generalized lambda distributions are studied. In this model, the distribution parameters are functions of the input and are modeled by PCE. Two local inference methods are first tested within the replication-based framework to construct the surrogate model. Then, an additional joint optimization step is developed to improve the accuracy of the replication-based approaches. The performance of the different fitting methods is compared on various examples and a case study from wind turbine design.

In Chapter 5, the idea of the joint optimization with all the available data is extended to a stand-alone target to deal with the cases where replications are not available. The statistical properties of the estimator are investigated, which offers a theoretical justification for this choice. For practical use with finite samples, some statistical methods are combined to adaptively select the basis functions for the PCE representations of the distribution parameters. The resulting estimation procedure does not require replications and yields accurate surrogate models.

With the help of the surrogate model, we look into sensitivity analysis for stochastic simulators in Chapter 6. A review of the state-of-the-art extensions of the popular variance-based indices to stochastic simulators is given. We provide some insights into the different sensitivity indices and offer a general guideline to the practitioners. Moreover, this chapter illustrates the effectiveness of GLaM on some examples for estimating the indices that only rely on the statistical dependence between the model input and output.

To bypass the limit of GLaM, which is unable to represent multimodal distributions, a more versatile surrogate model called *stochastic polynomial chaos expansion* (SPCE) is proposed in Chapter 7. This model introduces an artificial latent variable to imitate the random behavior of the stochastic simulator and a noise variable to smooth out the response distribution. An adaptive algorithm is developed to fit the model from finite samples without the need for replications. The benchmark examples showcase the excellent performance of the novel method compared with other state-of-the-art statistical models.

In Chapter 8, SPCE is applied to perform simulation-based seismic fragility analysis. In contrast to classical methods where the intensity measure needs to be obtained from the earthquake time series, we follow a new framework where engineering meaningful parameters of the ground motion model are selected as intensity measures. This allows for bridging the ground motion modeling and the fragility analysis. In this application, SPCE is used to quantify the statistical dependence between the input and output of the stochastic simulator, and it exhibits high accuracy for estimating not only the response distribution but also the fragility function.

Chapter 9 concludes the main findings and contributions of the thesis. The limitations of the developed methods are pointed out and several paths of improvement are suggested for future research on the topic of surrogate modeling of stochastic simulators.

Finally, some complementary materials and two journal papers to which we contributed through data analysis and discussions on stochastic simulators are presented as appendices. Appendix A discusses the role of replications in building the surrogate models proposed in this thesis. It also shows the consistency of maximum likelihood estimation for SPCE, thus providing some theoretical insight into SPCE developed in Chapter 7. Appendix B introduces a global sensitivity analysis framework for hybrid simulations with the help of GLaM. Appendix C develops an agent-based stochastic simulator to study the impact of the ploidy and the life cycle of pathogens on the resistance durability of plants.

Part I

Introduction

Not only does God play dice, but he sometimes throws them where they cannot be seen.

Stephen Hawking

2

Uncertainty quantification of deterministic models

Design and maintenance of modern engineering systems (e.g., power plants and wind turbines) require examining the system behavior under various conditions and operational scenarios. This is necessary to help engineers optimize the performance, assess the risk, and guarantee the reliability of the system. However, such analysis cannot merely rely on experiments, as only a limited number of experiments are feasible due to both time constraints and monetary costs. In this respect, computational models have been developed to virtually reflect engineering phenomena and processes (Winsberg, 2019).

A computational model contains parameters introduced in the mathematical modeling of the underlying system. In engineering, these parameters typically characterize the geometry of the structure under consideration (e.g., dimension of different components), material properties (e.g., elasticity or elasto-plastic constitutive laws), boundary conditions (e.g., of Dirichlet or Neumann type), and initial conditions for time-dependent problems. Through numerical algorithms, a computational model outputs a set of quantities (e.g., displacement and stress fields) that describe the system behavior. As a result, a simulator can be considered as a map that takes the system parameters as input and returns the abovementioned output quantities.

Simulators used in engineering are usually deterministic, in the sense that running them with a given set of input parameters will always return the same values. Mathematically, such a model is a function, that is,

$$\begin{aligned} \mathcal{M}_d : \mathcal{D}_{\boldsymbol{X}} &\to \mathcal{D}_{\boldsymbol{Y}}, \\ \boldsymbol{x} &\mapsto \mathcal{M}_d(\boldsymbol{x}), \end{aligned} \tag{2.1}$$

where the subscript $_d$ stands for deterministic, $\mathcal{D}_{\boldsymbol{X}}$ is the domain of definition of the input parameters, and $\mathcal{D}_{\boldsymbol{Y}}$ denotes the output range.

In this thesis, we consider simulators with a finite number M of scalar input parameters. They are collected in an M-dimensional vector $\boldsymbol{x}$, i.e., $\mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$. The performance of an engineering system is typically assessed by a small number of meaningful quantities instead of the entire solution field (in time and space). For example, the maximum deflection of a bridge is usually regarded as an indicator of its serviceability (Eurocode, 2004), and the absorbed kinetic energy of a car under crash impact serves as a key objective to optimize its crash-related

components (Moustapha, 2016). In this respect, we consider the case where a single aggregated quantity can represent the system performance. Thus, we suppose here that $\mathcal{D}_{\boldsymbol{Y}} \subset \mathbb{R}$.[1]

2.1 Uncertainty quantification

With the increasing power of computer hardware and software, the predictive capacity of a high-fidelity simulator is limited by the accuracy of the mathematical representation of the engineering system. On the one hand, several assumptions and simplifications (e.g., continuum assumption and idealized constitutive laws) are introduced in the modeling process to approximate the engineering scenario, which leads to a modeling discrepancy, called the *modeling error*. On the other hand, the model parameters are not perfectly determined and are affected by so-called *parametric uncertainties*. Sources of uncertainty can be classified into two categories: *epistemic* and *aleatory* uncertainty (Der Kiureghian and Ditlevsen, 2009). The epistemic uncertainty (originated from the Greek word $\epsilon\pi\iota\sigma\tau\eta\mu\eta$ meaning "knowledge") is due to a lack of knowledge. This can be caused by imprecise measurements or scarce data, e.g., the dimension of a beam measured by a device with a certain measurement tolerance. Such type of uncertainty can be reduced by acquiring more knowledge, e.g., by using a more accurate device or by collecting more data. In contrast, aleatory uncertainty (originated from the Latin word "alea" standing for "dice") corresponds to the irreducible intrinsic variability in the parameters, such as the outcome of a dice throw.

The modeling error is examined by validation studies on experimental data (Kennedy and O'Hagan, 2001). It is not within the scope of this manuscript, as we focus on studying the impact of parametric uncertainties.

Although the idea of determinism, such as Laplace's demon,[2] is elegant and attractive to scientists and engineers, uncertainties are ubiquitous, and one cannot avoid dealing with them in engineering. In this respect, uncertainty quantification has been developed to tackle rigorously the uncertain aspects (Sullivan, 2015; Ghanem, 2017; Soize, 2017). Figure 2.1 summarizes the general framework proposed by de Rocquigny (2006) and Sudret (2007) for uncertainty quantification.

- **Step A** defines the computational model representing the engineering system of interest. This involves considering all the relevant elements in the modeling process: combining possible sub-models and identifying the input and output across the model, as formulated in Eq. (2.1).
- **Step B** consists in quantifying the sources of uncertainty. The parameters whose exact values are unknown are identified and modeled. Common tools for modeling uncertainty include probability theory (Jaynes, 2003), possibility theory (Zadeh, 1999), and imprecise probability theory (Schöbi, 2017). The quantitative representation (e.g., probability distributions, fuzzy sets, intervals) can be prescribed based on expert judgment or calibrated from available data.
- **Step C** propagates the uncertainty from the input parameters to the output through the computational model. In the probabilistic context, the output is a random variable and is fully characterized by a prob-

[1] For vector-valued output, we can simply consider it as a set of scalar-valued functions.

[2] Laplace's demon, or originally "démon de Laplace" in French, is defined by Laplace (Laplace, 1814): "An intellect which at a certain moment would know all forces that set nature in motion, and all positions of all items of which nature is composed, if this intellect were also vast enough to submit these data to analysis, it would embrace in a single formula the movements of the greatest bodies of the universe and those of the tiniest atom; for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes." (translated from French to English by Truscott, F.W. and Emory, F.L.).

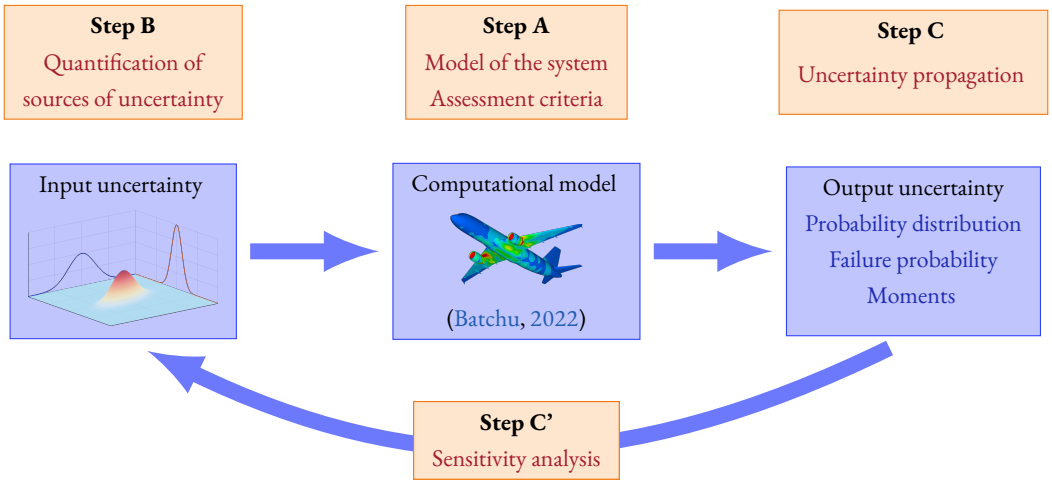


Figure 2.1: General framework of uncertainty quantification (derived from de Rocquigny, 2006; Sudret, 2007)

ability distribution. Depending on the focus of the application, however, certain summary quantities of the distribution are of more significance: quantiles and moments (e.g., mean and variance) for risk assessment (McNeil et al., 2005), and the probability of failure for reliability analysis (Melchers and Beck, 2018).

- **Step C'** complements the forward uncertainty propagation in **Step C** by assessing the contribution of the input uncertainty to that of the output (Saltelli et al., 2000). Sensitivity analysis guides engineers to find the most influential parameters to be investigated to efficiently reduce the output uncertainty and the least important variables that can be ignored at **Step C**. Moreover, it provides more insights into the computational model (e.g., how the input variables affect the model output) and thus helps understand the underlying phenomenon.

2.2 Probability theory in a nutshell

In this thesis, we use probability theory (Jacod and Protter, 2004) to model uncertainty. Within the probabilistic framework, we treat the exact values of the uncertain parameters as outcomes of a *random experiment* (such as flipping a coin). In particular, we represent the input parameters by a random vector $\boldsymbol{X}$, and the model output becomes a random variable $Y = \mathcal{M}_d(\boldsymbol{X})$. In this section, we briefly review the essential components of probability theory with a focus on random variables to lay the foundation for the following presentation. Importantly, the concepts recapitulated in this section are introduced with their rather formal mathematical definitions. This is meant to benefit readers, especially fresh researchers in the field, to understand the important mathematical tools in a rigorous way.

2.2.1 Probability space

Modern probability theory was established by Kolmogorov (1933) based on rigorous axioms and measure theory. At the root is the probability space given by the triplet $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the *sample space*, $\mathcal{F}$ is the collection of events, and $\mathbb{P}$ is the *probability measure*. In a random experiment, the sample space Ω contains all

the possible outcomes. In this respect, an event A is modeled as a subset of Ω, and whether it happens or not is determined by checking if A contains the outcome ω of an experiment. The collection of all the observable events (i.e., all the events that can be verified whether they occur or not in an experiment) forms the event algebra $\mathcal{F}$. In other words, $\mathcal{F}$ is the set of subsets of Ω. It is mathematically modeled as a *σ-algebra* on Ω, that is,

- $\Omega \in \mathcal{F}$;
- $\mathcal{F}$ is closed under complements, i.e., for all $A \in \mathcal{F}$, its complement $\Omega \backslash A \in \mathcal{F}$;
- $\mathcal{F}$ is closed under *countable* unions, i.e., for a sequences of events $A_n \in \mathcal{F}$ with $n \in \mathbb{N}$, $\bigcup_{n \in \mathbb{N}} A_n \in \mathcal{F}$.

For each event in $\mathcal{F}$, we need to assign a number that quantifies how "probable" it is to happen in an experiment. This is accomplished by introducing a probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$ that fulfills

- $\mathbb{P}(\Omega) = 1$;
- for a series of countably many events $(A_n)_{n \in \mathbb{N}}$ that are mutually disjoint, i.e., $A_n \bigcap A_m = \varnothing$ for $n \neq m \in \mathbb{N}$, $\mathbb{P}\left(\bigcup_n A_n\right) = \sum_{n \in \mathbb{N}} \mathbb{P}(A_n)$.

With the definition above, $(\Omega, \mathcal{F}, \mathbb{P})$ is a *measure space* with the "volume" of an event corresponding to its probability.

The dependence of events $A_1, A_2 \in \mathcal{F}$ is represented by the conditional probability $\mathbb{P}(A_1 \mid A_2)$ depicting the probability that event A_1 occurs given that event A_2 has happened. For $\mathbb{P}(A_2) \neq 0$, $\mathbb{P}(A_1 \mid A_2)$ is defined by

$$\mathbb{P}(A_1 \mid A_2) \overset{\text{def}}{=} \frac{\mathbb{P}(A_1 \bigcap A_2)}{\mathbb{P}(A_2)}. \tag{2.2}$$

Two events are said to be *independent* if

$$\mathbb{P}\left(A_1 \bigcap A_2\right) = \mathbb{P}(A_1)\,\mathbb{P}(A_2), \tag{2.3}$$

which implies that $\mathbb{P}(A_1 \mid A_2) = \mathbb{P}(A_1)$, i.e., the incidence of A_2 does not affect the probability of A_1.

2.2.2 Random variables

A random variable X is a *measurable* function that maps the probability space Ω to $(E, \mathcal{E})$ where E is the space containing the range of X (i.e., all the possible values that a random variable can take in a random experiment), and $\mathcal{E}$ is a collection of events related to X, a σ-algebra defined on E. Mathematically, X is defined as

$$\begin{aligned} X : (\Omega, \mathcal{F}) &\to (E, \mathcal{E}), \\ \omega &\mapsto X(\omega), \end{aligned} \tag{2.4}$$

where $X(\omega)$ is called a *realization* of X. In Eq. (2.4), we explicitly state the σ-algebras to emphasize that X is a measurable function that is defined by

$$\forall B \in \mathcal{E},\ X^{-1}(B) \overset{\text{def}}{=} \{\omega \in \Omega : X(\omega) \in B\} \in \mathcal{F}. \tag{2.5}$$

In this setup, E can be interpreted as a space of observation related to outcomes of X, and X being measurable implies that all the observable events (that constitute $\mathcal{E}$) can be "measured" with a probability. In other words, X induces a probability measure $\mathbb{P}_X$ on $(E, \mathcal{E})$ defined by

$$\mathbb{P}_X\left(X \in B\right) \stackrel{\text{def}}{=} \mathbb{P}\left(X^{-1}(B)\right). \tag{2.6}$$

$\mathbb{P}_X$ is called the *probability distribution* of X. When there is no ambiguity, we ignore the subscript and express directly $\mathbb{P}\left(X \in B\right)$ for simplicity.

2.2.3 Real-valued random variables

An uncertain parameter that takes real values is commonly modeled by a real-valued random variable mapping Ω to the real line, i.e., $E = \mathbb{R}$ with $\mathcal{E}$ being the Borel-algebra $\mathcal{B}(\mathbb{R})$.[3] This concept of Borel-algebra is quite abstract, but it can be understood as follows: all the intervals can be assigned a probability (i.e., the probability that the random variable falls into an interval can be assessed). Hence, $\mathbb{P}\left(X \in (-\infty, x]\right)$ is well-defined and gives rise to the definition of the cumulative distribution function (CDF) of X, that is,

$$F_X(x) \stackrel{\text{def}}{=} \mathbb{P}\left(X \in (-\infty, x]\right) = \mathbb{P}\left(X \leqslant x\right). \tag{2.7}$$

According to the definition of the probability measure $\mathbb{P}$, $F_{\boldsymbol{X}}$ has the following properties:

- F_X is non-decreasing, i.e., $\forall x_1 < x_2$, $F_X(x_1) \leqslant F_X(x_2)$;
- F_X is right-continuous, i.e., $\forall x_0 \in \mathbb{R}$, $\lim\limits_{x \to x_0^+} F_X(x) = F_X(x_0)$;
- $\lim\limits_{x \to -\infty} F_X(x) = 0$ and $\lim\limits_{x \to +\infty} F_X(x) = 1$.

A function $F : \mathbb{R} \to [0, 1]$ having the three properties above is a CDF for a unique probability measure $\mathbb{P}_X$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. In other words, F_X fully characterizes the probability distribution $\mathbb{P}_X$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$.

Because F_X is a non-decreasing function, its *generalized inverse* denoted by Q_X exists (Embrechts and Hofert, 2013) and is given by

$$Q_X(\alpha) \stackrel{\text{def}}{=} \inf\left\{x \in \mathbb{R} : F_X(x) \geqslant \alpha\right\}. \tag{2.8}$$

Q_X is called the *quantile function* of X, and the quantity $Q_X(\alpha)$ is called the α-*quantile* of X.

The measurable space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ is naturally equipped with the Lebesgue measure μ, which can be seen as a measure of the length of intervals. If $\mathbb{P}_X$ is absolutely continuous with respect to μ,[4] there exists a function $f_X : \mathbb{R} \to [0, +\infty)$ such that

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\, \mu(\mathrm{d}t) = \int_{-\infty}^{x} f_X(t)\, \mathrm{d}t, \tag{2.9}$$

where $\mu(\mathrm{d}t)$ is usually denoted by $\mathrm{d}t$ for simplicity. f_X is called the probability density function (PDF) of X, and can be defined by

$$f_X(x) \stackrel{\text{def}}{=} \frac{\mathrm{d}F_X(x)}{\mathrm{d}x}. \tag{2.10}$$

[3] Borel σ-algebra is the smallest σ-algebra containing all the open sets.
[4] This means $\forall B \in \mathcal{B}(\mathbb{R})$, $\mu(B) = 0$ implies $\mathbb{P}_X\left(X \in B\right) = 0$.

Following the properties of CDF, a PDF f_X fulfills $f_X(x) \geqslant 0$ and $\int_{\mathbb{R}} f_X(x)\,\mathrm{d}x = 1$. Conversely, any function $f : \mathbb{R} \to \mathbb{R}$ that has these two properties is a valid PDF. For a given f_X, one can calculate the CDF using Eq. (2.9). Therefore, f_X also fully characterizes the probability distribution of X.

As a random variable is a measurable function, one can define its Lebesgue integral

$$\mathbb{E}[X] \overset{\text{def}}{=} \int_{\Omega} X(\omega)\,\mathbb{P}(\mathrm{d}\omega)\,. \tag{2.11}$$

The integration operator $\mathbb{E}$ is called *expectation*. The integral can also be calculated by

$$\mathbb{E}[X] = \int_{\mathbb{R}} x\,\mathbb{P}_X(\mathrm{d}x) = \int_{\mathbb{R}} x\,\mathrm{d}F_X(x). \tag{2.12}$$

The first equality is obtained by a change of variables and the second equality follows the definition of the Lebesgue–Stieltjes integral (Bogachev, 2007). In addition, if X has a PDF, Eq. (2.11) can be explicitly calculated as

$$\mathbb{E}[X] = \int_{\mathbb{R}} x f(x)\,\mathrm{d}x. \tag{2.13}$$

If $\mathbb{E}[|X|] < +\infty$, X is called *integrable*. Let $\mathcal{L}^p(\Omega, \mathcal{F}, \mathbb{P})$ be the space of measurable functions whose absolute p-th power is integrable, i.e., $\mathbb{E}[|X|^p] < +\infty$. Based on Eq. (2.11), different types of moments of a random variable are defined and summarized in Table 2.1.

Table 2.1: Definition of different types of moments.

Moment	Central moment	Standardized moment
$\mathbb{E}[X^r]$	$\mathbb{E}[(X - \mathbb{E}[X])^r]$	$\dfrac{\mathbb{E}[(X - \mathbb{E}[X])^r]}{\mathbb{E}[(X - \mathbb{E}[X])^2]^{\frac{r}{2}}}$

In particular, some specific quantities in Table 2.1 are commonly used in practice to characterize a random variable. The first moment is known as the *expected* or *mean* value $m_X \overset{\text{def}}{=} \mathbb{E}[X]$, which shows the average value of X over $\mathbb{R}$. Moreover, this quantity is the best constant that approximates $X \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$ in the mean-squared sense, that is,

$$m_X = \arg\min_{m \in \mathbb{R}} \mathbb{E}\left[(X - m)^2\right]. \tag{2.14}$$

The second central moment is called the *variance* $\mathrm{Var}[X] \overset{\text{def}}{=} \mathbb{E}[(X - \mathbb{E}[X])^2]$, which is equal to the minimum value of the objective function in Eq. (2.14). Its square root is called the *standard deviation* $\sigma_X \overset{\text{def}}{=} \sqrt{\mathrm{Var}[X]}$, which measures the variability of X deviating from its mean m_X. The ratio between σ_X and m_X, namely σ_X / m_X, is called the *coefficient of variation*, which characterizes the extent of variability in relation to the expected value. The third and fourth standardized moments are known as *skewness* and *kurtosis*, respectively, which describe the level of asymmetry and tail behaviors of a probability distribution.

A probability distribution characterized by a fixed number of parameters is called a *parametric distribution*. Table 2.2 lists some parametric distributions that are widely used to model continuous uncertain parameters.

Table 2.2: Some important parametric distributions. $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the PDF of the standard normal distribution, B is the beta function, and Γ is the gamma function.

Name	Parameters	PDF	Mean	Variance	Skewness	Kurtosis
Normal $\mathcal{N}(\mu, \sigma^2)$	$\mu \in \mathbb{R}, \sigma > 0$	$\frac{1}{\sigma}\varphi(\frac{x-\mu}{\sigma})$	μ	σ^2	0	3
Lognormal $\mathcal{LN}(\lambda, \zeta)$	$\lambda \in \mathbb{R}, \zeta > 0$	$\frac{1}{\zeta x}\varphi\left(\frac{\ln(x)-\lambda}{\zeta}\right)$	$e^{\lambda+\frac{\zeta^2}{2}}$	$e^{2\lambda+\zeta^2}\sqrt{e^{\zeta^2}-1}$	$(e^{\zeta^2}+2)\sqrt{e^{\zeta^2}-1}$	$e^{4\zeta^2}+2e^{3\zeta^2}+3e^{2\zeta^2}-3$
Uniform $\mathcal{U}(a, b)$	$a < b \in \mathbb{R}$	$\frac{x-a}{b-a}\mathbb{1}_{[a,b]}(x)$	$\frac{1}{2}(b-a)$	$\frac{1}{12}(b-a)^2$	0	1.8
Beta $\mathrm{Beta}(\alpha, \beta)$	$\alpha, \beta > 0$	$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$	$\frac{2(\beta-\alpha)\sqrt{\alpha+\beta+1}}{(\alpha+\beta+1)\sqrt{\alpha\beta}}$	$\frac{3(\alpha+\beta+1)\left(2(\alpha-\beta)^2-\alpha\beta(\alpha+\beta+2)\right)}{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)}$
Gamma $\mathrm{Gamma}(\theta, k)$	$\theta, k > 0$	$\frac{x^{k-1}e^{-x/\theta}}{\Gamma(k)\theta^k}$	$k\theta$	$k\theta^2$	$\frac{2}{\sqrt{k}}$	$\frac{6}{k}+3$

2.2.4 Vector-valued random variables (random vectors)

In engineering applications, multiple uncertain resources are usually identified and should be jointly taken into account in the uncertainty quantification framework presented in Fig. 2.1. To this end, we can group them into a random vector. More precisely, an M dimensional random vector is a collection of M real-valued random variables $\boldsymbol{X} = (X_1, \ldots, X_M)^\intercal$, where the superscript $^\intercal$ denotes the transposition (so $\boldsymbol{X}$ is a column vector). Equivalently, we can also define a random vector by taking $E = \mathbb{R}^M$ and $\mathcal{E} = \mathcal{B}(\mathbb{R}^M)$ (the Borel σ-algebra of $\mathbb{R}^M$) in the generic definition of a random variable in Eq. (2.4), i.e., $\boldsymbol{X}$ can be seen as a vector-valued random variable.

2.2.4.1 Joint properties

The probability distribution of $\boldsymbol{X}$ is described by the joint CDF given by

$$F_{\boldsymbol{X}}(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{P}_{\boldsymbol{X}}(\boldsymbol{X} \leqslant \boldsymbol{x}) = \mathbb{P}\left(\bigcap_{j=1}^{M} X_j^{-1}((-\infty, x_j])\right), \tag{2.15}$$

where $\mathbb{P}_{\boldsymbol{X}}$ is the induced probability measure on $(\mathbb{R}^M, \mathcal{B}(\mathbb{R}^M))$, and $\boldsymbol{X} \leqslant \boldsymbol{x}$ stands for $\{X_1 \leqslant x_1, \ldots, X_M \leqslant x_M\}$. An important feature of the joint CDF is that if we split the random vector $\boldsymbol{X}$ into two random subvectors $\boldsymbol{X}_{\mathbf{u}}$ and $\boldsymbol{X}_{\mathbf{v}}$) with $\mathbf{u} \subset \{1, \ldots, M\}$ and $\mathbf{v} = \{1, \ldots, M\} \backslash \mathbf{u}$, the joint CDF of the random subvector $\boldsymbol{X}_{\mathbf{u}}$ can be computed by

$$F_{\boldsymbol{X}_{\mathbf{u}}}(\boldsymbol{x}_{\mathbf{u}}) = \lim_{\boldsymbol{x}_{\mathbf{v}} \to +\infty} F_{\boldsymbol{X}}(\boldsymbol{x}). \tag{2.16}$$

The operation in Eq. (2.16) is called *marginalization*. By this, one can obtain the probability distribution F_{X_j} of each individual component X_j, which is called the *marginal distribution*.

Similar to Eqs. (2.9) and (2.10), if $\mathbb{P}_{\boldsymbol{X}}$ is absolutely continuous with respect to the Lebesgue measure μ on $(\mathbb{R}^M, \mathcal{B}(\mathbb{R}^M))$, $\boldsymbol{X}$ has a joint PDF $f_{\boldsymbol{X}}$ that satisfies

$$\forall B \in \mathcal{B}(\mathbb{R}^M),\ \mathbb{P}_{\boldsymbol{X}}(\boldsymbol{X} \in B) = \int_B f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{2.17}$$

By taking B as the Cartesian product of the segment $(\infty, x_j]$ of each dimension, namely, $(-\infty, x_1] \times \ldots \times (-\infty, x_M]$ denoted by $(-\infty, \boldsymbol{x}]$ for short, we obtain

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{(-\infty, \boldsymbol{x}]} f_{\boldsymbol{X}}(\boldsymbol{t}) \mathrm{d}\boldsymbol{t}, \tag{2.18}$$

Therefore, $f_{\boldsymbol{X}}$ is sometimes defined by

$$f_{\boldsymbol{X}}(\boldsymbol{x}) \overset{\text{def}}{=} \frac{\partial^M F_{\boldsymbol{X}}(\boldsymbol{x})}{\partial x_1 \cdots \partial x_M}, \tag{2.19}$$

if the right-hand side exists.

From Eq. (2.17), the joint PDF is non-negative, and its integral over $\mathbb{R}^M$ is equal to 1. Conversely, if a function f has these two properties, it is a joint PDF of a certain random vector whose probability measure is defined by Eq. (2.17).

Using the property of the joint CDF in Eq. (2.15), the joint PDF of a random subvector $\boldsymbol{X}_{\mathbf{u}}$ can be obtained by marginalizing the effect of $\boldsymbol{X}_{\mathbf{v}}$:

$$f_{\boldsymbol{X}_{\mathbf{u}}}(\boldsymbol{x}_{\mathbf{u}}) = \int_{\mathbb{R}^{|\mathbf{v}|}} f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}_{\mathbf{v}}, \tag{2.20}$$

where $|\cdot|$ denotes the cardinality of a set.

As each component of $\boldsymbol{X}$ is a random variable, we can calculate the moments defined in Table 2.1 individually and then group them together, such as the mean vector $\boldsymbol{m}_{\boldsymbol{X}} = \mathbb{E}[\boldsymbol{X}] \overset{\text{def}}{=} (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_M])^{\mathsf{T}}$ (for $X_1, \ldots, X_M \in \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$) and variance vector $\boldsymbol{v}_{\boldsymbol{X}} = \mathrm{Var}[\boldsymbol{X}] \overset{\text{def}}{=} (\mathrm{Var}[X_1], \ldots, \mathrm{Var}[X_M])^{\mathsf{T}}$ (for $X_1, \ldots, X_M \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$).

In addition, several quantities that summarize the probabilistic relations among the components can be defined using the expectation operator $\mathbb{E}$, among others, covariance and correlation. The covariance of two random variables $X_j, X_k \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$ is given by

$$\mathrm{Cov}[X_j, X_l] \overset{\text{def}}{=} \mathbb{E}[(X_j - \mathbb{E}[X_j])(X_l - \mathbb{E}[X_l])] = \mathbb{E}[X_j X_l] - \mathbb{E}[X_j]\mathbb{E}[X_l]. \tag{2.21}$$

This quantity characterizes the co-variability of X_j and X_l. We can group the covariance of all the pairs of $\boldsymbol{X}$ into a matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}}$. Using vector notation, the covariance matrix is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{X}} \overset{\text{def}}{=} \mathbb{E}\left[(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^{\mathsf{T}}\right], \tag{2.22}$$

with its diagonal elements being the variance of each component $\Sigma_{j,j} = \mathrm{Var}[X_j]$, and its off-diagonal elements corresponding to the covariance $\Sigma_{j,l} = \mathrm{Cov}[X_j, X_l]$.

Based on the covariance in Eq. (2.21), the correlation coefficient of two random variables X_j and X_l is defined by

$$\mathrm{Corr}[X_j, X_k] \overset{\text{def}}{=} \frac{\mathrm{Cov}[X_j, X_k]}{\sqrt{\mathrm{Var}[X_j]}\sqrt{\mathrm{Var}[X_k]}}. \tag{2.23}$$

Following its definition, the correlation coefficient is a normalized version of the covariance and always takes values in $[-1, 1]$. This quantity essentially measures the linear relation between two random variables. In the extreme case, if $\mathrm{Corr}[X_j, X_k]$ is equal to 1 or -1, X_j and X_k are said to be perfectly correlated, and we can

find a linear relation between them, i.e., $X_j = aX_k + b$ with $a > 0$ for $\mathrm{Corr}[X_j, X_k] = 1$ and $a < 0$ for $\mathrm{Corr}[X_j, X_k] = -1$.

2.2.4.2 Conditional properties

Effectively, one may collect information on some variables and update the probability distribution of the others in $\boldsymbol{X}$. This is particularly useful if it is difficult to measure some quantities directly, but it is relatively simple to gauge certain related variables. The update procedure is achieved by using conditional probability distributions. Moreover, conditional distribution is an important concept when studying stochastic simulators in Chapter 3.

Intuitively, we can use Eq. (2.2) to form the probability measure of $\boldsymbol{X}_{\mathbf{u}}$ conditioned on the values of the other random variables $\boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}$. However, this is only well-defined for $\mathbb{P}(\boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}) > 0$ but becomes ill-posed if $\mathbb{P}(\boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}) = 0$. A rigorous definition of conditional distribution is complex and beyond the scope of the thesis, and interested readers are referred to Kolmogorov (1933), Durrett (2019), and Billingsley (1995) for a more detailed presentation. For simplicity, we assume in the sequel that the random vector has a joint PDF, as is the case for most engineering problems.

The conditional probability distribution of $\boldsymbol{X}_{\mathbf{u}}$ for $\boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}$ is fully characterized by the conditional PDF defined by

$$f_{\boldsymbol{X}_{\mathbf{u}}|\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{u}} \mid \boldsymbol{x}_{\mathbf{v}}) \overset{\text{def}}{=} \begin{cases} \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{f_{\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{v}})} & \text{if} \quad f_{\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{v}}) > 0, \\ 0 & \text{if} \quad f_{\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{v}}) = 0, \end{cases} \tag{2.24}$$

where $f_{\boldsymbol{X}_{\mathbf{v}}}$ is the joint PDF of $\boldsymbol{X}_{\mathbf{v}}$. Eq. (2.20) shows that $f_{\boldsymbol{X}_{\mathbf{u}}|\boldsymbol{X}_{\mathbf{v}}}$ is a valid joint PDF for all $\boldsymbol{x}_{\mathbf{v}}$ such that $f_{\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{v}}) > 0$. Similar to Eq. (2.20), we can compute the conditional PDF for any subset of $\boldsymbol{X}_{\mathbf{u}}$ by marginalization.

From Eq. (2.17), the conditional CDF is given by integrating the conditional PDF:

$$F_{\boldsymbol{X}_{\mathbf{u}}|\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{u}} \mid \boldsymbol{x}_{\mathbf{v}}) = \int_{(-\infty, \boldsymbol{x}_{\mathbf{u}}]} f_{\boldsymbol{X}_{\mathbf{u}}|\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{t} \mid \boldsymbol{x}_{\mathbf{v}}) \, \mathrm{d}\boldsymbol{t}. \tag{2.25}$$

For a random variable X_j with $j \in \mathbf{u}$ and $X_j \in \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$, its conditional expectation given $\boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}$ can be computed in a similar manner to Eq. (2.13), that is,

$$\mathbb{E}[X_j \mid \boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}] = \int_{\mathbb{R}} x_j \, f_{X_j|\boldsymbol{X}_{\mathbf{v}}}(x_j \mid \boldsymbol{x}_{\mathbf{v}}) \, \mathrm{d}x_j, \tag{2.26}$$

where $f_{X_j|\boldsymbol{X}_{\mathbf{v}}}$ is the conditional PDF of X_j. This gives a function of the conditioning value

$$m_{X_j|\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{x}_{\mathbf{v}}) \overset{\text{def}}{=} \mathbb{E}[X_j \mid \boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}], \tag{2.27}$$

which is called the *expected mean function*.

Based on $m_{X_j|\boldsymbol{X}_{\mathbf{v}}}$, we define the conditional expectation of X_j given $\boldsymbol{X}_{\mathbf{v}}$

$$\mathbb{E}[X_j \mid \boldsymbol{X}_{\mathbf{v}}] \overset{\text{def}}{=} m_{X_j|\boldsymbol{X}_{\mathbf{v}}}(\boldsymbol{X}_{\mathbf{v}}). \tag{2.28}$$

Note that we do not condition on a specific value $\boldsymbol{x}_\mathbf{v}$ but keep $\boldsymbol{X}_\mathbf{v}$ random.[5] As a result, $\mathbb{E}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right]$ is a function of $\boldsymbol{X}_\mathbf{v}$ and thus a random variable. The expectation of this random variable can be calculated by

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right]\right] &= \int_{\mathbb{R}^{|\mathbf{v}|}} \left(\int_{\mathbb{R}} x_j f_{X_j|\boldsymbol{X}_\mathbf{v}}(x_j \mid \boldsymbol{x}_\mathbf{v})\,\mathrm{d}x_j\right) f_{\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_\mathbf{v})\,\mathrm{d}\boldsymbol{x}_\mathbf{v} \\ &= \int_{\mathbb{R}} x_j f_{X_j}(x_j)\mathrm{d}x_j = \mathbb{E}\left[X_j\right],\end{aligned} \tag{2.29}$$

which is called the *law of total expectation*. The interpretation is straightforward: the expected value of a random variable is the average of its expected values on the segments of a partition formed by $\boldsymbol{X}_\mathbf{v}$.[6]

Following Eqs. (2.14) and (2.29), it is straightforward to show that the conditional mean function fulfills

$$m_{X_j|\boldsymbol{X}_\mathbf{v}} = \arg\min_{g\in\mathcal{G}} \mathbb{E}\left[\left(X_j - g\left(\boldsymbol{X}_\mathbf{v}\right)\right)^2\right], \tag{2.30}$$

where the feasible set $\mathcal{G}$ contains all the measurable functions that map $\mathbb{R}^{|\mathbf{v}|}$ to $\mathbb{R}$.

Similar to Eq. (2.27), one can define the conditional variance of X_j given $\boldsymbol{X}_\mathbf{v} = \boldsymbol{x}_\mathbf{v}$ by

$$v_{X_j|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_\mathbf{v}) \overset{\text{def}}{=} \mathbb{E}\left[\left(X_j - m_{X_j|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_\mathbf{v})\right)^2\right], \tag{2.31}$$

and the conditional variance of X_j given $\boldsymbol{X}_\mathbf{v}$ is

$$\mathrm{Var}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right] \overset{\text{def}}{=} v_{X_j|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{X}_\mathbf{v}) = \mathbb{E}\left[\left(X_j - \mathbb{E}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right]\right)^2 \,\middle|\, \boldsymbol{X}_\mathbf{v}\right]. \tag{2.32}$$

The law of total variance states that

$$\mathrm{Var}\left[X_j\right] = \mathbb{E}\left[\mathrm{Var}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right]\right] + \mathrm{Var}\left[\mathbb{E}\left[X_j \mid \boldsymbol{X}_\mathbf{v}\right]\right], \tag{2.33}$$

which decomposes the variance of X_j into two parts. By considering that $\boldsymbol{X}_\mathbf{v}$ generates a partition of Ω, the first part in Eq. (2.33) is the average of the variances within each segment, and the second part corresponds to the variance of the segmental average values.

2.2.5 Independence

The independence of events in Eq. (2.3) can be extended to real-valued or vector-valued random variables. $\boldsymbol{X}_\mathbf{u}$ and $\boldsymbol{X}_\mathbf{v}$ are called independent if

$$\forall B_1 \in \mathcal{B}(\mathbb{R}^{|\mathbf{u}|}), B_2 \in \mathcal{B}(\mathbb{R}^{|\mathbf{v}|}),\ \mathbb{P}\left(\boldsymbol{X}_\mathbf{u} \in B_1, \boldsymbol{X}_\mathbf{v} \in B_2\right) = \mathbb{P}\left(\boldsymbol{X}_\mathbf{u} \in B_1\right)\mathbb{P}\left(\boldsymbol{X}_\mathbf{v} \in B_2\right). \tag{2.34}$$

In this case, the joint PDF can be expressed as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}_\mathbf{u}}(\boldsymbol{x}_\mathbf{u})\, f_{\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_\mathbf{v}). \tag{2.35}$$

[5] The formal definition of conditional expectation relies on conditioning on the σ-algebra generated by $\boldsymbol{X}_\mathbf{v}$.

[6] Intuitively, this partition of the sample space can be seen as $\bigcup_{\boldsymbol{x}_\mathbf{v}\in\mathbb{R}^{|\mathbf{v}|}} \{\omega \in \Omega : \boldsymbol{X}_\mathbf{v}(\omega) = \boldsymbol{x}_\mathbf{v}\}$ (it is formally defined by the σ-algebra generated by $\boldsymbol{X}_\mathbf{v}$).

Following Eq. (2.24), the conditional distribution $f_{\boldsymbol{X}_\mathbf{u}|\boldsymbol{X}_\mathbf{v}}$ is equal to the unconditional one, $f_{\boldsymbol{X}_\mathbf{u}}$ in this case, which implies that knowledge of $\boldsymbol{X}_\mathbf{v}$ does not provide additional information on $\boldsymbol{X}_\mathbf{u}$.

Multiple random variables $X_1, \ldots, X_M$ are called *mutually independent* if

$$\forall B_1, \ldots, B_M \in \mathcal{B}(\mathbb{R}),\ \mathbb{P}\left(X_1 \in B_1, \ldots, X_M \in B_M\right) = \prod_{j=1}^{M} \mathbb{P}\left(X_j \in B_j\right). \tag{2.36}$$

Therefore, the joint PDF of $\boldsymbol{X}$ in Eq. (2.17) can be factorized as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{j=1}^{M} f_{X_j}(x_j). \tag{2.37}$$

By considering random vectors as vector-valued random variables, mutual independence can be extended to random vectors. If a set of random variables (or random vectors) are mutually independent and follow the same probability distribution, they are said to be *independent and identically distributed* (i.i.d.).

From Eqs. (2.24) and (2.35), we can define the *conditional independence*. Let $\mathbf{u}_1$ be a subset of $\mathbf{u}$ and $\mathbf{u}_2 = \mathbf{u} \backslash \mathbf{u}_1$. The two subvectors $\boldsymbol{X}_{\mathbf{u}_1}$ and $\boldsymbol{X}_{\mathbf{u}_2}$ are conditional independent given $\boldsymbol{X}_\mathbf{v}$ if

$$f_{\boldsymbol{X}_\mathbf{u}|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_\mathbf{u} \mid \boldsymbol{x}_\mathbf{v}) = f_{\boldsymbol{X}_{\mathbf{u}_1}|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_{\mathbf{u}_1} \mid \boldsymbol{x}_\mathbf{v})\, f_{\boldsymbol{X}_{\mathbf{u}_2}|\boldsymbol{X}_\mathbf{v}}(\boldsymbol{x}_{\mathbf{u}_2} \mid \boldsymbol{x}_\mathbf{v}). \tag{2.38}$$

2.2.6 Convergence of random variables

As random variables are functions and characterized by their probability distributions, there exist several different notions of convergence. These concepts are very important in statistics for assessing the property of estimators.

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ be a sequence of random vectors taking values in $\mathbb{R}^M$. Such a sequence converges to the random variable $\boldsymbol{X}$

- almost surely (a.s.), which is denoted by $\boldsymbol{X}_N \xrightarrow{\text{a.s.}} \boldsymbol{X}$, if

$$\mathbb{P}\left(\lim_{N\to+\infty} \boldsymbol{X}_N = \boldsymbol{X}\right) = 1; \tag{2.39}$$

- in probability, which is denoted by $\boldsymbol{X}_N \xrightarrow{P} \boldsymbol{X}$, if

$$\forall \varepsilon > 0,\ \lim_{N\to+\infty} \mathbb{P}\left(\|\boldsymbol{X}_N - \boldsymbol{X}\|_2 > \varepsilon\right) = 0, \tag{2.40}$$

  where $\|\cdot\|_2$ stands for the Euclidean norm on $\mathbb{R}^M$;

- in $\mathcal{L}^p$ (or in p-th mean), which is denoted by $\boldsymbol{X}_N \xrightarrow{\mathcal{L}^p} \boldsymbol{X}$, if

$$\lim_{N\to+\infty} \mathbb{E}\left[\|\boldsymbol{X}_N - \boldsymbol{X}\|_2^p\right] = 0; \tag{2.41}$$

- in distribution, which is denoted by $\boldsymbol{X}_N \xrightarrow{d} \boldsymbol{X}$, if

$$\lim_{N \to +\infty} F_{\boldsymbol{X}_N}(\boldsymbol{x}) = F_{\boldsymbol{X}}(\boldsymbol{x}), \tag{2.42}$$

  for all $\boldsymbol{x}$ at which $F_{\boldsymbol{X}}(\boldsymbol{x})$ is continuous.

The relations among the different convergences are summarized in Section 2.2.6. The definition of almost sure convergence is similar to the almost everywhere convergence for functions. The convergence in $\mathcal{L}^p$ relies on the norm on $\mathcal{L}^p(\Omega, \mathcal{F}, \mathbb{P})$, i.e., $\|\boldsymbol{X}\|_{\mathcal{L}^p} \overset{\text{def}}{=} (\mathbb{E}[\|\boldsymbol{X}\|_2^p])^{1/p}$. If for all $N > 0$ $\|\boldsymbol{X}_N\|_2$ is dominated by a random variable Z with $\mathbb{E}[Z^p] < +\infty$, the almost sure convergence implies that in $\mathcal{L}^p$, based on the dominated convergence theorem. The convergence in probability is weaker, and both almost sure convergence and convergence in $\mathcal{L}^p$ imply convergence in probability. Finally, the convergence in distribution focuses only on the probability distribution without considering the behavior of random variables as functions. This is the weakest and is implied by the other three types of convergence.

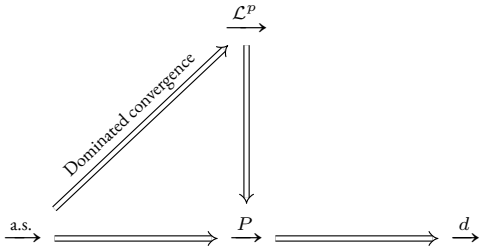


Figure 2.2: Relations among different types of convergence.

2.3 Uncertainty propagation

By means of the tools from probability theory in Section 2.2, the uncertain parameters in a computation model are identified and modeled by a random vector. The detailed probabilistic description of $\mathbb{P}_{\boldsymbol{X}}$ can be determined by design codes (Joint Committee on Structural Safety, 2002), statistical inferences (James et al., 2014), expert knowledge (O'Hagan, 2019), or a combination these techniques (Sudret, 2007).

Propagating the uncertainty in the input parameters through the model Eq. (2.1) leads to the output being a random variable, namely,

$$\begin{aligned} Y : \Omega &\to \mathbb{R}, \\ \omega &\mapsto Y(\omega) \overset{\text{def}}{=} \mathcal{M}_d \circ \boldsymbol{X}(\omega). \end{aligned} \tag{2.43}$$

Eq. (2.43) is well-defined if the domain of definition $\mathcal{D}_{\boldsymbol{X}}$ of $\mathcal{M}_d$ contains the range of $\boldsymbol{X}$. In this case, the image of $\boldsymbol{X}$ belongs to $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}))$ where $\mathcal{B}(\mathcal{D}_{\boldsymbol{X}})$ is the Borel algebra of $\mathcal{D}_{\boldsymbol{X}}$. With this restriction, we can work only with $\mathcal{D}_{\boldsymbol{X}}$ instead of $\mathbb{R}^M$.

The composition $\mathcal{M}_d \circ \boldsymbol{X}$ is assumed to be measurable, and thus it defines a probability measure denoted by $\mathbb{P}_Y$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. As $\mathbb{P}_{\boldsymbol{X}}$ is prescribed when quantifying the source of uncertainty, we can ignore the abstract probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and work directly on the probability space $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}), \mathbb{P}_{\boldsymbol{X}})$. To complete this

simplification, the computational model $\mathcal{M}_d$ is assumed to be a measurable function from $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}))$ to $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$.[7] This implies that any observable event of Y results from an observable event of $\boldsymbol{X}$. Therefore, the probability measure $\mathbb{P}_Y$ is given by

$$\mathbb{P}_Y\left(Y \in B\right) \overset{\text{def}}{=} \mathbb{P}_{\boldsymbol{X}}\left(\boldsymbol{X} \in \mathcal{M}_d^{-1}(B)\right), \tag{2.44}$$

where $\mathcal{M}_d^{-1}(B) \overset{\text{def}}{=} \{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} : \mathcal{M}_d(\boldsymbol{x}) \in B\}$.

The objective of uncertainty quantification is to study $\mathbb{P}_Y$. Following Eqs. (2.11) and (2.12), the expectation of Y is given by

$$\mathbb{E}\left[Y\right] = \int_{\mathbb{R}} y\, \mathbb{P}_Y\left(\mathrm{d}y\right) = \int_{\mathcal{D}_{\boldsymbol{X}}} \mathcal{M}_d(\boldsymbol{x})\, \mathbb{P}_{\boldsymbol{X}}\left(\mathrm{d}\boldsymbol{x}\right). \tag{2.45}$$

Similarly, for any measurable function $g : \mathbb{R} \to \mathbb{R}$, the expectation of the random variable $g(Y)$ can be calculated by

$$\mathbb{E}\left[g(Y)\right] = \int_{\mathbb{R}} g(y)\, \mathbb{P}_Y\left(\mathrm{d}y\right) = \int_{\mathcal{D}_{\boldsymbol{X}}} g \circ \mathcal{M}_d(\boldsymbol{x})\, \mathbb{P}_{\boldsymbol{X}}\left(\mathrm{d}\boldsymbol{x}\right). \tag{2.46}$$

Eq. (2.46) is a more general form of Eq. (2.45). Many characteristic quantities of Y can be expressed as in Eq. (2.46). For example, the variance of Y corresponds to

$$g(y) = \left(y - \mathbb{E}\left[Y\right]\right)^2, \tag{2.47}$$

and the probability that Y exceeds a given threshold δ_0 uses

$$g(y) = \mathbb{1}_{[\delta_0, \infty)}(y), \tag{2.48}$$

where $\mathbb{1}$ is an indicator function, since $\mathbb{P}\left(Y \geqslant \delta_0\right) = \mathbb{E}\left[\mathbb{1}_{[\delta_0, \infty)}(Y)\right]$.

For $\boldsymbol{X}$ with a joint PDF, Eq. (2.45) can be calculated by the integral

$$\mathbb{E}\left[g(Y)\right] = \int_{\mathcal{D}_{\boldsymbol{X}}} g \circ \mathcal{M}_d(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{2.49}$$

Evaluating the expectation in Eq. (2.46) is practically difficult. In the rest of this section, we review some classical methods developed for uncertainty propagation. In Section 2.3.1, we present the perturbation method, which is relatively simple but has a limited estimation power. A more generic method — Monte Carlo simulation — is introduced in Section 2.3.2. There, we also recap some sampling methods, which are used for constructing surrogate models throughout the manuscript. In Section 2.3.3, we present the method of Gaussian quadrature, which is later applied in Chapter 7 for evaluating one-dimensional integrals.

2.3.1 Perturbation method

The perturbation method consists in approximating the computational model $\mathcal{M}_d$ by a Taylor series expansion around the mean vector $\boldsymbol{m}_{\boldsymbol{X}}$ of the input $\boldsymbol{X}$ (Handa and Andersson, 1981). It is mainly used to estimate the mean and variance of Y.

[7] This assumption is stronger than Eq. (2.1), as measurability also relies on the associated σ-algebras.

Recall that the Taylor series expansion of $\mathcal{M}_d$ around a given value $\boldsymbol{x}_0$ reads

$$\mathcal{M}_d(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{x}_0) + (\boldsymbol{x}-\boldsymbol{x}_0)^\intercal \nabla \mathcal{M}_d(\boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_0)^\intercal \nabla^2 \mathcal{M}_d(\boldsymbol{x}_0)(\boldsymbol{x}-\boldsymbol{x}_0) + o\left(\|\boldsymbol{x}-\boldsymbol{x}_0\|_2^2\right), \quad (2.50)$$

where $o(\cdot)$ stands for "decreasing to zero at a faster rate than $(\cdot)$", $\nabla \mathcal{M}_d(\boldsymbol{x}_0)$ is the gradient at $\boldsymbol{x}_0$, and $\nabla^2 \mathcal{M}_d(\boldsymbol{x}_0)$ is the Hessian matrix defined as follows:

$$\nabla \mathcal{M}_d(\boldsymbol{x}_0) \overset{\text{def}}{=} \left.\begin{pmatrix} \frac{\partial \mathcal{M}_d(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial \mathcal{M}_d(\boldsymbol{x})}{\partial x_M} \end{pmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}_0}, \quad \nabla^2 \mathcal{M}_d(\boldsymbol{x}_0) \overset{\text{def}}{=} \left.\begin{pmatrix} \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_1 \partial x_M} \\ \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial \partial x_2 x_1} & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial^2 x_2} & \cdots & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_2 \partial x_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_M \partial x_1} & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial x_M \partial x_2} & \cdots & \frac{\partial^2 \mathcal{M}_d(\boldsymbol{x})}{\partial^2 x_M} \end{pmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}_0}. \quad (2.51)$$

As the mean vector $\boldsymbol{m}_{\boldsymbol{X}}$ represents the average behavior of $\boldsymbol{X}$, $\boldsymbol{X}$ can be considered varying around $\boldsymbol{m}_{\boldsymbol{X}}$. Hence, the latter is selected as $\boldsymbol{x}_0$ in the expansion. In addition, we truncate this expansion up to the second order and represent $\mathcal{M}_d$ by

$$\begin{aligned} \mathcal{M}_d(\boldsymbol{X}) \approx \tilde{\mathcal{M}}_d(\boldsymbol{X}) =& \mathcal{M}(\boldsymbol{m}_{\boldsymbol{X}}) + (\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) \\ &+ \frac{1}{2}(\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla^2 \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})(\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}}). \end{aligned} \quad (2.52)$$

By taking the expectation of Eq. (2.52), we approximate the expected value of Y by

$$\begin{aligned} \mathbb{E}[Y] &\approx \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) + \mathbb{E}\left[\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}}\right]^\intercal \nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) + \frac{1}{2}\mathbb{E}\left[(\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla^2 \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})(\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})\right] \\ &= \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) + 0 + \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})(\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla^2 \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})\right) \\ &= \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) + \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{X}} \nabla^2 \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})\right), \end{aligned} \quad (2.53)$$

where $\operatorname{Tr}(\cdot)$ denotes the trace of a matrix, and $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is the covariance matrix of $\boldsymbol{X}$ as defined in Eq. (2.22). Because $\mathbb{E}[\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}}] = 0$, the computational model evaluated at $\boldsymbol{m}_{\boldsymbol{X}}$ yields a first-order approximation $\mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})$ to $\mathbb{E}[Y]$. The additional term provides an enhancement based on the covariance matrix of $\boldsymbol{X}$ and the second-order derivatives of $\mathcal{M}_d$ evaluated at $\boldsymbol{m}_{\boldsymbol{X}}$.

To calculate the variance, we use only the first-order expansion in Eq. (2.52) (i.e., ignoring the second-order term), which gives

$$\begin{aligned} \operatorname{Var}[Y] &\approx \operatorname{Var}\left[\mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}) + (\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})\right] \\ &= \mathbb{E}\left[\left((\boldsymbol{X}-\boldsymbol{m}_{\boldsymbol{X}})^\intercal \nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})\right)^2\right] \\ &= \left(\nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}})\right)^\intercal \boldsymbol{\Sigma}_{\boldsymbol{X}} \nabla \mathcal{M}_d(\boldsymbol{m}_{\boldsymbol{X}}). \end{aligned} \quad (2.54)$$

Eqs. (2.53) and (2.54) are quite simple as they only requires evaluating the first- and second-order derivatives of $\mathcal{M}_d$ at the mean vector $\boldsymbol{m}_{\boldsymbol{X}}$. This comes at the cost of accuracy: if the computational model shows a strong nonlinearity in the region where $\boldsymbol{X}$ mostly likely lies, the first-order approximation would fail and the second-

order correction may not be enough. The idea to represent the computational model with a more accurate proxy gives rise to the development of *surrogate models*, which are presented in Section 2.5.

2.3.2 Monte Carlo simulation

Monte Carlo simulation is a general framework to study the probabilistic properties of random quantities. This method can be applied without any assumptions about the regularity of the computational model.

2.3.2.1 Estimation

The main idea of Monte Carlo simulation is to generate a set of realizations of Y and then study the sample statistics. To this end, we can see the computational model as a sampler for Y: one first generates a set of samples $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ following the probability distribution of $\boldsymbol{X}$ and then evaluates the computational model to obtain samples $\mathcal{Y} = \left\{y^{(1)}, \ldots, y^{(N)}\right\}$ for Y with $y^{(i)} = \mathcal{M}_d\left(\boldsymbol{x}^{(i)}\right)$.

The samples of $\boldsymbol{X}$ are typically generated independently, so the realizations of Y are also independent. From a statistical perspective, $\mathcal{X}$ can be seen as a single realization of i.i.d. random variables $\left\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(N)}\right\}$, and the same holds for $\mathcal{Y}$ with $\left\{Y^{(1)}, \ldots, Y^{(N)}\right\}$. The average of the latter is again a random variable given by

$$\overline{Y_N} = \frac{\sum_{i=1}^{N} Y^{(i)}}{N}, \tag{2.55}$$

which is called the *empirical mean* of Y. Following the *(strong) law of large numbers* (Loève, 1977), the random variable $\overline{Y_N}$ converges almost surely to $\mathbb{E}[Y]$, i.e., $\overline{Y_N} \xrightarrow{\text{a.s.}} \mathbb{E}[Y]$ (see the definition in Section 2.2.6). In other words, even though $\overline{Y_N}$ is random, it converges to the deterministic value $\mathbb{E}[Y]$ with increasing N. As a result, $\overline{Y_N}$ is commonly used to estimate $\mathbb{E}[Y]$.

The estimation uncertainty can be assessed using the *central limit theorem* (Billingsley, 1995), which states that

$$\sqrt{N}\left(\overline{Y_N} - \mathbb{E}[Y]\right) \xrightarrow{d} \mathcal{N}\left(0, \operatorname{Var}[Y]\right), \tag{2.56}$$

where $\mathcal{N}\left(0, \operatorname{Var}[Y]\right)$ stands for the normal distribution with zero mean and variance $\operatorname{Var}[Y]$. As Eq. (2.56) suggests, Monte Carlo estimators are usually said to have a convergence rate of $1/\sqrt{N}$.

Similar results can be obtained for the general case in Eq. (2.46) by considering $G^{(i)} = g\left(Y^{(i)}\right)$. In particular, let us define the *empirical distribution function* (similar to Eq. (2.48)):

$$F_N(y) \overset{\text{def}}{=} \sum_{i=1}^{N} \mathbb{1}_{(-\infty, y]}\left(Y^{(i)}\right). \tag{2.57}$$

By the law of large numbers, $F_N(y)$ converges almost surely to the CDF $F_Y(y)$ of Y for any given value y as $N \to +\infty$. Furthermore, the *Glivenko–Cantelli theorem* (Tucker, 1959) provides an even stronger convergence:

$$\sup_y |F_N(y) - F_Y(y)| \xrightarrow{\text{a.s.}} 0. \tag{2.58}$$

The estimator for the variance is given by

$$\hat{V}_N = \frac{\sum_{i=1}^{N} \left(Y^{(i)} - \overline{Y_N}\right)^2}{N}, \tag{2.59}$$

which is not a simple sample average and thus cannot be plugged directly into the framework of Eqs. (2.46) and (2.55) (as the mean value is unknown but represented by the empirical mean $\overline{Y_N}$). Nevertheless, Eq. (2.59) can be recast as

$$\hat{V}_N = \frac{\sum_{i=1}^{N} \left(Y^{(i)}\right)^2}{N} - \left(\overline{Y_N}\right)^2 = \overline{Y_N^2} - \left(\overline{Y_N}\right)^2, \tag{2.60}$$

where $\overline{Y_N^2}$ is the average of sample squares. The first term converges almost surely to $\mathbb{E}\left[Y^2\right]$ by the law of large numbers, and the second term $\left(\overline{Y_N}\right)^2$ to $\mathbb{E}\left[Y\right]^2$ by continuity, which implies $\hat{V} \xrightarrow{\text{a.s.}} \operatorname{Var}\left[Y\right]$. Note that the denominator N is sometimes replaced by $N-1$ to obtain an *unbiased estimator*, whose expectation is equal to the target quantity, i.e., $\mathbb{E}\left[\frac{N}{N-1}\hat{V}\right] = \operatorname{Var}\left[Y\right]$.

In general, for any function g whose discontinuous points form a set of zero probability, the *continuous mapping theorem* (van der Vaart, 1998, Theorem 2.3) guarantees that

$$\tilde{\boldsymbol{Y}}_N \xrightarrow{\text{a.s.}} \boldsymbol{y}_0 \implies g\left(\tilde{\boldsymbol{Y}}_N\right) \xrightarrow{\text{a.s.}} g(\boldsymbol{y}_0). \tag{2.61}$$

By taking $\tilde{\boldsymbol{Y}}_N = \left(\overline{Y_N^2}, \overline{Y_N}\right)^{\mathsf{T}}$ and $g(\boldsymbol{y}) = y_1 - y_2^2$ and applying the above theorem, we obtain the almost sure convergence of $\hat{V}_N$ to $\operatorname{Var}\left[Y\right]$. Finally, the asymptotic behavior of such kinds of estimators can be addressed by the δ-method (van der Vaart, 1998, Theorem 3.1). It states that if g is differentiable at $\boldsymbol{y}_0$ and $\tilde{\boldsymbol{Y}}_N$ satisfies

$$\sqrt{N}\left(\tilde{\boldsymbol{Y}}_N - \boldsymbol{y}_0\right) \xrightarrow{d} \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{Y}}}\right), \tag{2.62}$$

where $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{Y}}}$ is called the *asymptotic covariance* of $\tilde{\boldsymbol{Y}}_N$, the asymptotic behavior of $g\left(\tilde{\boldsymbol{Y}}_N\right)$ can be depicted by

$$\sqrt{N}\left(g\left(\tilde{\boldsymbol{Y}}_N\right) - g(\boldsymbol{y}_0)\right) \xrightarrow{d} \mathcal{N}\left(0, \left(\nabla g(\boldsymbol{y}_0)\right)^{\mathsf{T}} \boldsymbol{\Sigma}_{\tilde{\boldsymbol{Y}}} \nabla g(\boldsymbol{y}_0)\right). \tag{2.63}$$

2.3.2.2 Sampling

To perform Monte Carlo estimation, one needs to generate samples of $\boldsymbol{X}$. In practice, computer programs are deterministic in nature, and thus programs are only implemented to simulate *pseudo-random* samples, namely a set of numbers that look as if they were random.

Many methods focus on generating samples of $\boldsymbol{U}$ that are uniformly distributed in $[0,1]^M$ (Gentle, 2003). They can be classified into three groups. The first group called *Monte Carlo sampling* consists of congruential generators, which calculate the next sample point by a function of existing samples (Matsumoto and Nishimura, 1998). The initial value at the start of the recursion is called the *random seed*. Therefore, one can obtain the exact same sequence of numbers by controlling the random seed. Otherwise, if the random seed is not fixed, the algorithm will generate a different sequence upon each simulation, which shows certain "randomness" in the generation process. In contrast, the second type of approach called *quasi-Monte Carlo sampling* produces a deterministic sequence. Sacrificing the pseudo-randomness, this sequence tries to be "representative" of the sample space in the sense that it has a low discrepancy with respect to a certain space-filling measure. Several

such sequences can be found in the literature, among others, Halton sequences (Halton, 1960) and Sobol' sequences (Sobol', 1967). The third category is *stratified sampling*, which consists in dividing the space $[0,1]^M$ into subdomains and generating samples (e.g., based on the congruential generators) within each of them. This procedure ensures that the samples will not be close to one another and cover more uniformly the sample space. The most popular method of this type is called *Latin hypercube sampling* (LHS; McKay et al., 1979).

In many applications, the sources of uncertainty do not follow independent uniform distributions. Therefore, the simulated sequence (as described above) needs to be *transformed* into numbers that are close to what might be expected when sampling from the target distribution.

For $\boldsymbol{X}$ with mutually independent components, it is simply necessary to sample each marginal distribution individually. This corresponds to transforming each uniform component $U_j \in \mathcal{U}(0,1)$ of $\boldsymbol{U}$ to the random variable X_j with the target distribution. To this end, we can apply an important property of the quantile function defined in Eq. (2.8): $Q_{X_j}(U_j)$ follows the same probability distribution as X_j (Embrechts and Hofert, 2013). In other words,

$$\boldsymbol{X} \overset{d}{=} \left(Q_{X_1}(U_1), \ldots, Q_{X_1}(U_M)\right)^{\intercal}, \tag{2.64}$$

where $\overset{d}{=}$ denotes that the equality is in distribution, i.e., both sides of the equation follow the same probability distribution.

For $\boldsymbol{X}$ with dependent components, the quantile transform is not directly applicable, as the component-wise transform cannot create the desired dependence structure. Nevertheless, the joint distribution of $\boldsymbol{X}$ can be factorized using conditional distributions defined in Section 2.2.4.2:

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = F_{X_1}(x_1) F_{X_2|X_1}(x_2 \mid x_1) \ldots F_{X_M|X_1,\ldots,X_{M-1}}(x_M \mid x_1, \ldots, x_{M-1}). \tag{2.65}$$

Therefore, we can apply sequentially the quantile transform to independent realizations $\boldsymbol{u} = (u_1, \ldots, u_M)^{\intercal}$ of $\boldsymbol{U}$ to sample $\boldsymbol{X}$, that is,

$$x_1 = Q_{X_1}(u_1),\ x_2 = Q_{X_2|X_1}(u_1 \mid x_1),\ \ldots,\ x_M = Q_{X_M|X_1,\ldots,X_{M-1}}(u_M \mid x_1, \ldots, x_{M-1}), \tag{2.66}$$

where we express directly the relation in terms of realizations to avoid possible ambiguity. We denote this transform by $\boldsymbol{x} = \tilde{\mathcal{R}}_{\boldsymbol{X}}(\boldsymbol{u})$, and it is also known as the *inverse Rosenblatt transform* (Rosenblatt, 1952) in probability theory.

2.3.3 Gaussian quadrature

According to the definition in Eq. (2.49), calculating the properties of Y corresponds to evaluating integrals. One numerical technique to approximate these integrals is *Gaussian quadrature*, which calculates a weighted sum of the function values at a finite set of points.

For computing the expectation of $g(X)$ for a single random variable X with PDF f_X, the N_Q-point Gaussian rule is given by

$$\mathbb{E}\left[g(X)\right] = \int_{\mathbb{R}} g(x) f_X(x) \mathrm{d}x \approx \sum_{k=1}^{N_Q} w_i\, g\left(x^{(i)}\right), \tag{2.67}$$

where N_Q is the number of integration points, $x^{(i)}$ is the i-th integration point, and w_i is the associated weight.

A Gaussian quadrature rule is designed such that the integral is exact for any polynomials of degree less than $2N_Q - 1$, and thus the integration points and weights depend only on f_X. This is equivalent to ensuring the exact integration of monomials up to degree $2N_Q - 1$. Therefore, N_Q controls the accuracy of the numerical integration. Intuitively, one can solve a system of nonlinear equations to calculate $\left\{\left(x^{(i)}, w_i\right) : i = 1, \ldots, N_Q\right\}$ using the moments of X up to the order $2N_Q - 1$ (as moments of X are integrals of monomials of X according to their definition in Table 2.1). A more practical way to generate integration points and weights can be found in Abramowitz and Stegun (1970) and Golub and Welsch (1969).

For $\boldsymbol{X}$ with mutually independent components, the joint PDF can be factorized into a product of marginal distributions Eq. (2.37). In this case, one can first establish the quadrature rule for each marginal PDF f_{X_j} and then group them to form multidimensional quadrature points, the so-called *tensorized* scheme:

$$\mathbb{E}\left[g(\boldsymbol{X})\right] = \int_{\mathbb{R}} g(x) \prod_{j=1}^{M} f_{X_j}(x_j) \mathrm{d}\boldsymbol{x} \approx \sum_{i_1}^{N_Q} \sum_{i_2}^{N_Q} \ldots \sum_{i_M}^{N_Q} \left(\prod_{j=1}^{M} w_{i_j} \right) g\left(\boldsymbol{x}^{(\boldsymbol{i})}\right), \tag{2.68}$$

where $\boldsymbol{x}^{(\boldsymbol{i})} = \left(x_1^{(i_1)}, \ldots, x_M^{(i_M)}\right)^{\intercal}$. For $\boldsymbol{X}$ with a general dependent structure, one can apply the inverse Rosenblatt transform in Eq. (2.66) and evaluate the integral by

$$\mathbb{E}\left[g(\boldsymbol{X})\right] = \mathbb{E}\left[g\left(\tilde{\mathcal{R}}_{\boldsymbol{X}}(\boldsymbol{U})\right)\right] = \int_{[0,1]^M} g \circ \tilde{\mathcal{R}}_{\boldsymbol{X}}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} \approx \sum_{i_1}^{N_Q} \sum_{i_2}^{N_Q} \ldots \sum_{i_M}^{N_Q} \left(\prod_{j=1}^{M} w_{i_j} \right) g\left(\tilde{\mathcal{R}}_{\boldsymbol{X}}\left(\boldsymbol{u}^{(\boldsymbol{i})}\right)\right), \tag{2.69}$$

where the integration points and weights are calculated with respect to the uniform distribution between $[0, 1]$.

Owing to the tensorized scheme, which takes a full tensor product of the integration points, the quadrature method can require a large number of model evaluations if used in high-dimensional problems. For example, by fixing N_Q points per dimension, we need to evaluate the computational model N_Q^M times. This number grows dramatically by increasing N_Q, which controls the accuracy of the approximation, and the dimensionality of $\boldsymbol{X}$.

2.4 Global sensitivity analysis

Uncertainty propagation aims at studying the probabilistic properties of the output Y. Global sensitivity analysis instead concentrates on quantifying the impact of each or a group of uncertain parameters on the output uncertainty.

Let $\mathbf{A} = \{1, \ldots, M\}$ be the set of all the input indices. Similar to Section 2.2, for the following analysis, we split the input vector into two subvectors $\boldsymbol{X}_{\mathbf{u}}$ and $\boldsymbol{X}_{\mathbf{v}}$), where $\mathbf{v} = \mathbf{A} \backslash \mathbf{u}$.

2.4.1 Sobol' indices

Sobol' analysis is one of the most popular sensitivity analysis methods (Alexanderian et al., 2012; Brown et al., 2013; Wagner et al., 2020; Abbiati et al., 2021b) and has been extensively studied (Sobol', 1993; Homma and Saltelli, 1996; Saltelli et al., 2000). This method relies on the *Hoeffding–Sobol decomposition* (Hoeffding, 1948;

Sobol', 1993) of the model $\mathcal{M}_d$ into functions of increasing dimensionality, which offers a decomposition of the variance of Y. Therefore, it is also called variance-based sensitivity analysis.

The main assumption of this method is that $\boldsymbol{X}$ has mutually independent components and Y has a finite variance, i.e., $\mathcal{M}_d \in \mathcal{L}^2\left(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}), \bigotimes_{j=1}^{M} \mathbb{P}_{X_j}\right)$. Hence, this method is only applicable to independent uncertain sources.

Under this assumption, the function $\mathcal{M}_d$ can be uniquely decomposed into

$$\begin{aligned}\mathcal{M}_d(\boldsymbol{x}) &= \mathcal{M}_0 + \sum_{i=1}^{M} \mathcal{M}_i(x_i) + \sum_{1 \leqslant j < k \leqslant M} \mathcal{M}_{j,k}(x_j, x_k) + \ldots + \mathcal{M}_{1,\ldots,M}(x_1, \ldots, x_M), \\ &= \sum_{\mathbf{w} \in \mathcal{P}(\mathbf{A})} \mathcal{M}_{\mathbf{w}}(\boldsymbol{x}_{\mathbf{w}}),\end{aligned} \tag{2.70}$$

where $\mathcal{P}(\mathbf{A})$ is the power set of $\mathbf{A}$ (set of all the subsets of $\mathbf{A}$). Moreover, $\mathcal{M}_0$ corresponds to $\mathcal{M}_\varnothing$, which is a constant, and $\mathcal{M}_{\mathbf{w}}(\boldsymbol{x}_{\mathbf{w}})$ is the function associated with the subset of input variables $\boldsymbol{x}_{\mathbf{w}}$ defined by $\mathbf{w} \subset \{1, \ldots, M\}$. For example, $\mathbf{w} = \{1, 2\}$ corresponds to $\mathcal{M}_{1,2}(x_1, x_2)$. Each component $\mathcal{M}_{\mathbf{w}}(\boldsymbol{x}_{\mathbf{w}})$ is defined by conditional expectations (see Section 2.2.4.2), that is,

$$\mathcal{M}_{\mathbf{w}}(\boldsymbol{x}_{\mathbf{w}}) \overset{\text{def}}{=} \sum_{\mathbf{t} \in \mathcal{P}(\mathbf{w})} (-1)^{|\mathbf{w}|-|\mathbf{t}|} \mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{t}} = \boldsymbol{x}_{\mathbf{t}}\right], \tag{2.71}$$

which implies that $\mathcal{M}_0 = \mathbb{E}[Y]$ and $\mathbb{E}\left[\mathcal{M}_{\mathbf{w}}(\boldsymbol{X}_{\mathbf{w}})\right] = 0$ for all $\mathbf{w} \in \mathcal{P}(\mathbf{A}) \backslash \varnothing$. Moreover, the functions $\{\mathcal{M}_{\mathbf{w}} : \mathbf{w} \in \mathcal{P}(\mathbf{A})\}$ are mutually orthogonal, i.e.,

$$\forall \mathbf{w} \neq \mathbf{t}, \quad \mathbb{E}\left[\mathcal{M}_{\mathbf{w}}(\boldsymbol{X}_{\mathbf{w}}) \mathcal{M}_{\mathbf{t}}(\boldsymbol{X}_{\mathbf{t}})\right] = 0. \tag{2.72}$$

Therefore, the variance of the model output can be computed as

$$\operatorname{Var}[Y] = \mathbb{E}\left[(Y - \mathcal{M}_0)^2\right] = \sum_{\mathbf{w} \in \mathcal{P}(\mathbf{A}) \backslash \varnothing} \operatorname{Var}\left[\mathcal{M}_{\mathbf{w}}\left(\boldsymbol{X}_{\mathbf{w}}\right)\right]. \tag{2.73}$$

Let $V_{\mathbf{w}} \overset{\text{def}}{=} \operatorname{Var}\left[\mathcal{M}_{\mathbf{w}}\left(\boldsymbol{X}_{\mathbf{w}}\right)\right]$. Following the variance decomposition, the Sobol' indices are defined by

$$S_{\mathbf{w}} \overset{\text{def}}{=} \frac{V_{\mathbf{w}}}{\operatorname{Var}[Y]}. \tag{2.74}$$

For $|\mathbf{w}| = 1$, we obtain the *first-order Sobol' indices* $\{S_j : j \in \mathbf{A}\}$, which represent the main effect of each input variable. *Higher-order indices* defined by $|\mathbf{w}| \geqslant 2$ quantify the pure interactive effect within a given group of input variables. Because of the additive structure of Eq. (2.73), the sum of all the Sobol' indices is equal to 1, i.e., $\sum_{\mathbf{w} \in \mathcal{P}(\mathbf{A})} S_{\mathbf{w}} = 1$.

To assess the overall contribution of X_j to the output variance, Homma and Saltelli (1996) introduced the *total Sobol' index*

$$S_{T_j} \overset{\text{def}}{=} \sum_{\substack{\mathbf{w} \in \mathcal{P}(\mathbf{A}) \\ \mathbf{w} \ni j}} S_{\mathbf{w}} = 1 - \sum_{\mathbf{w} \in \mathcal{P}(\mathbf{A} \backslash \{j\})} S_{\mathbf{w}}. \tag{2.75}$$

It is worth remarking that the sum of all the total Sobol' indices are greater or equal to 1, i.e., $\sum_{j=1}^{M} S_{T_j} \geqslant 1$.

The equality holds if and only if there is no interactive effect among the input variables, meaning that all the higher-order Sobol' indices are 0.

Similar to the first-order and total Sobol' indices, we can define two indices associated with a group of variables $\boldsymbol{X}_{\mathbf{u}}$

$$S_{\{\mathbf{u}\}} \overset{\text{def}}{=} \sum_{\mathbf{w}\in\mathcal{P}(\mathbf{u})} S_{\mathbf{w}}, \quad S_{T_{\{\mathbf{u}\}}} \overset{\text{def}}{=} \sum_{\substack{\mathbf{w}\subset\mathbf{A} \\ \mathbf{w}\cap\mathbf{u}\neq\varnothing}} S_{\mathbf{w}} = 1 - S_{\{\mathbf{v}\}}, \tag{2.76}$$

which quantify the effect of the variables contained in $\boldsymbol{X}_{\mathbf{u}}$.

According to the definition of $\mathcal{M}_{\mathbf{w}}(\boldsymbol{x}_{\mathbf{w}})$ in Eq. (2.71), $V_{\mathbf{u}}$ can be expressed as

$$V_{\mathbf{u}} \overset{\text{def}}{=} \operatorname{Var}\left[\mathcal{M}_{\mathbf{u}}\left(\boldsymbol{X}_{\mathbf{u}}\right)\right] = \sum_{\mathbf{w}\subset\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{w}|} \operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{w}}\right]\right]. \tag{2.77}$$

Hence, the Sobol' indices defined in Eq. (2.76) can be calculated by conditional variances, that is,[8]

$$S_{\{\mathbf{u}\}} = \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\right]}{\operatorname{Var}\left[Y\right]}, \quad S_{T_{\{\mathbf{u}\}}} = 1 - \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{v}}\right]\right]}{\operatorname{Var}\left[Y\right]}. \tag{2.78}$$

2.4.1.1 Monte Carlo estimation

To compute the Sobol' indices, one needs to estimate $\operatorname{Var}\left[Y\right]$ and conditional variance of the type $\operatorname{Var}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\right]$. To apply the Monte Carlo estimator in Eq. (2.61), these quantities should be expressed in form of expectations. The variance of Y can be calculated by $\operatorname{Var}\left[Y\right] = \mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2$. To tackle the conditional variance, we define an auxiliary random variable

$$Y_{\mathbf{u}} = \mathcal{M}(\boldsymbol{X}_{\mathbf{u}}, \tilde{\boldsymbol{X}}_{\mathbf{v}}), \tag{2.79}$$

where $\tilde{\boldsymbol{X}}_{\mathbf{v}}$ is an i.i.d. copy of $\boldsymbol{X}_{\mathbf{v}}$. In other words, $Y_{\mathbf{u}}$ shares the same argument $\boldsymbol{X}_{\mathbf{u}}$ as Y, but they differ from $\tilde{\boldsymbol{X}}_{\mathbf{v}}$ and $\boldsymbol{X}_{\mathbf{v}}$, which are i.i.d. Under this setup, Y and $Y_{\mathbf{u}}$ follow the same probability distribution. Moreover, for any given value $X_{\mathbf{u}} = \boldsymbol{x}_{\mathbf{u}}$, Y and $Y_{\mathbf{u}}$ also follow the same probability distribution, and they are conditionally independent given $\boldsymbol{X}_{\mathbf{u}}$.

With the help of $Y_{\mathbf{u}}$, we can derive $\operatorname{Var}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\right]$ as

$$\begin{aligned}
\operatorname{Var}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\right] &= \mathbb{E}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]^2\right] - \mathbb{E}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\right]^2 \\
&= \mathbb{E}\left[\mathbb{E}\left[Y \mid X_{\mathbf{u}}\right]\mathbb{E}\left[Y_{\mathbf{u}} \mid X_{\mathbf{u}}\right]\right] - \mathbb{E}\left[Y\right]^2 \\
&= \mathbb{E}\left[\mathbb{E}\left[YY_{\mathbf{u}} \mid X_{\mathbf{u}}\right]\right] - \mathbb{E}\left[Y\right]^2 = \mathbb{E}\left[YY_{\mathbf{u}}\right] - \mathbb{E}\left[Y\right]^2 = \operatorname{Cov}\left[Y, Y_{\mathbf{u}}\right]
\end{aligned} \tag{2.80}$$

As a result, we can sample $\boldsymbol{X}_{\mathbf{u}}$, $\boldsymbol{X}_{\mathbf{v}}$ and $\tilde{\boldsymbol{X}}_{\mathbf{v}}$ to get samples of Y and $Y_{\mathbf{u}}$. Following Homma and Saltelli (1996), we obtain the estimator

$$\hat{S}_{\{\mathbf{u}\},N} = \frac{\frac{1}{N}\sum_{i=1}^{N} Y^{(i)} Y_{\mathbf{u}}^{(i)} - \left(\sum_{i=1}^{N} \frac{1}{N} Y^{(i)}\right)^2}{\sum_{i=1}^{N} \frac{1}{N} \left(Y^{(i)}\right)^2 - \left(\sum_{i=1}^{N} \frac{1}{N} Y^{(i)}\right)^2} \tag{2.81}$$

[8] For the first-order and total Sobol' indices of a variable X_j, it is sufficient to take $\mathbf{u} = \{j\}$.

Furthermore, leveraging the property that Y and $Y_{\mathbf{u}}$ have the same probability distribution, Janon et al. (2014) proposed another estimator

$$\hat{S}_{\{\mathbf{u}\},N} = \frac{\frac{1}{N}\sum_{i=1}^{N} Y^{(i)} Y_{\mathbf{u}}^{(i)} - \left(\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(Y^{(i)} + Y_{\mathbf{u}}^{(i)}\right)\right)^2}{\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(\left(Y^{(i)}\right)^2 + \left(Y_{\mathbf{u}}^{(i)}\right)^2\right) - \left(\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(Y^{(i)} + Y_{\mathbf{u}}^{(i)}\right)\right)^2}. \tag{2.82}$$

Following Eq. (2.61), both Eq. (2.81) and Eq. (2.82) converges almost surely to the target Sobol' index. Based on Eq. (2.63), Eq. (2.82) is shown to yield a smaller asymptotic variance than Eq. (2.81) (Janon et al., 2014), which is equivalent to say that Eq. (2.82) is asymptotically more efficient than Eq. (2.81).

2.4.2 A general framework

The Sobol' indices deal only with the allocation of output variance to the input. In this section, we consider a more general framework (Borgonovo et al., 2014) based on which many other types of sensitivity indices have been developed.

Consider a contrast measure $d(\cdot,\cdot)$ for comparing the discrepancy between probability measures. More specifically, $d(\mathbb{P}_1, \mathbb{P}_2)$ quantifies the dissimilarity of the probability measure $\mathbb{P}_2$ in respect to $\mathbb{P}_1$. Within this perimeter, $d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_{\mathbf{u}}=\boldsymbol{x}_{\mathbf{u}}}\right)$ characterizes how the knowledge of $\boldsymbol{X}_{\mathbf{u}} = \boldsymbol{x}_{\mathbf{u}}$ can change the probability of Y. $d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_{\mathbf{u}}=\boldsymbol{x}_{\mathbf{u}}}\right)$ is a function of the conditioning value $\boldsymbol{x}_{\mathbf{u}}$. By randomizing $\boldsymbol{x}_{\mathbf{u}}$ (similar to Eqs. (2.28) and (2.31)), we obtain a random variable $d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_{\mathbf{u}}}\right)$. The expected value of the latter can be used to define a sensitivity index $S^d_{\{\mathbf{u}\}}$ of the group $\boldsymbol{X}_{\mathbf{u}}$

$$S^d_{\{\mathbf{u}\}} \overset{\text{def}}{=} \mathbb{E}\left[d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_{\mathbf{u}}}\right)\right], \tag{2.83}$$

which indicates the average changes of the probability distribution of Y by knowing the value of $\boldsymbol{X}_{\mathbf{u}}$.

The first-order Sobol' indices can be defined as a special case of this general construction. Let us define $d(\cdot,\cdot)$ as the difference of the variance of the probability measures, that is,

$$d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_{\mathbf{u}}}\right) \overset{\text{def}}{=} \mathrm{Var}\left[Y\right] - \mathrm{Var}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]. \tag{2.84}$$

The associated index for $\mathbf{u} = \{j\}$ reads

$$S^d_j = \mathbb{E}\left[d\left(\mathbb{P}_Y, \mathbb{P}_{Y|X_j}\right)\right] = \mathrm{Var}\left[Y\right] - \mathbb{E}\left[\mathrm{Var}\left[Y \mid X_j\right]\right] = \mathrm{Var}\left[\mathbb{E}\left[Y \mid X_j\right]\right], \tag{2.85}$$

where the last equality comes from the law of total variance Eq. (2.33). This index measures how much one can reduce on average the variance of Y if the value of X_j is known. Dividing S^d_j by $\mathrm{Var}\left[Y\right]$ provides the first-order Sobol' index of X_j. Alternatively, we can divide $\mathrm{Var}\left[Y\right]$ in the definition of the contrast measure in Eq. (2.84) to obtain directly the first-order Sobol' index.

In information theory, *entropy* (MacKay, 2013) is usually used to describe the amount of information (or uncertainty) of a random variable. Replacing the variance $\mathrm{Var}\left[\cdot\right]$ by the entropy $H(\cdot)$ in Eq. (2.84), we obtain

another sensitivity index

$$S^d_{\{\mathbf{u}\}} = \mathbb{E}\left[d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_\mathbf{u}}\right)\right] = \mathbb{E}\left[H(Y) - H\left(Y \mid \boldsymbol{X}_\mathbf{u}\right)\right] = I\left(Y; \boldsymbol{X}_\mathbf{u}\right), \tag{2.86}$$

which is called the *mutual information* $I(Y; \boldsymbol{X}_\mathbf{u})$. This index is widely used for variable selection in decision trees where $I(Y; \boldsymbol{X}_\mathbf{u})$ is referred to as the "information gain" (Rokach and Maimon, 2005). The mutual information can be normalized by the entropy of Y yielding a quantity within the range $[0, 1]$.

As variance and entropy are merely characteristic values of a probability measure, Borgonovo (2007) proposed a dissimilarity metric $d(\cdot, \cdot)$ to take into account the whole probability distribution: it is defined as half of the L^1 distance between the PDFs, i.e.,

$$d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_\mathbf{u}=\boldsymbol{x}_\mathbf{u}}\right) = \frac{1}{2}\left\|f_Y - f_{Y|\boldsymbol{X}_\mathbf{u}}\right\|_{L^1} = \frac{1}{2}\int_{\mathbb{R}} \left|f_Y(y) - f_{Y|\boldsymbol{X}_\mathbf{u}}\left(y \mid \boldsymbol{x}_\mathbf{u}\right)\right| \mathrm{d}y, \tag{2.87}$$

which gives the Borgonovo index

$$S^d_{\{\mathbf{u}\}} = \mathbb{E}\left[d\left(\mathbb{P}_Y, \mathbb{P}_{Y|\boldsymbol{X}_\mathbf{u}}\right)\right] = \frac{1}{2}\mathbb{E}\left[\left\|f_Y - f_{Y|\boldsymbol{X}_\mathbf{u}}\right\|_{L^1}\right]. \tag{2.88}$$

Recent research in sensitivity analysis consists in looking for other suitable contrast measures. For example, da Veiga (2021) proposed embedding probability distributions in a reproducing kernel Hilbert space. The norm in the embedding space induces a contrast metric, which allows for a similar decomposition as in Eq. (2.73). Besides, the mutual information in Eq. (2.86) can also be extended to the embedding space and thus defines another sensitivity measure. Borgonovo et al. (2022) suggested using the Wasserstein-type metrics developed in optimal transport to represent the difference between probability measures.

2.5 Surrogate models

Performing uncertainty quantification with Monte Carlo simulation typically requires a large number of model runs to produce a reliable estimate due to the slow convergence behavior. However, high-fidelity simulators can be computationally demanding and costly to evaluate, e.g., a single model run can take hours to days. It is therefore impracticable to study directly these expensive models in the context of uncertainty quantification.

The high complexity of a computational model is usually due to the sophisticated representation of the physical processes and accurate numerical solvers. To reduce the computational cost, one can approximate directly the input-output relation of the original simulator with a non-physical surrogate model

$$\mathcal{M}_d(\boldsymbol{x}) \approx \mathcal{M}^{\mathrm{s}}_d(\boldsymbol{x}). \tag{2.89}$$

By construction, the formulation of $\mathcal{M}^{\mathrm{s}}_d$ is much simpler compared to the original simulator, and thus $\mathcal{M}^{\mathrm{s}}_d$ is much cheaper to evaluate. With $\mathcal{M}^{\mathrm{s}}_d$, one can then perform a large-scale Monte Carlo simulation to study the physical model $\mathcal{M}_d$.

Surrogate modeling consists in approximating the computational model by a function from a specific function space. Depending on how the function space is represented, surrogate models can be classified into two groups. The first one consists in parameterizing or defining the function space by a countable set of parame-

ters. Popular models of this type include polynomial chaos expansions (Ghanem and Spanos, 2003; Berveiller et al., 2006), wavelet expansions (Meyer, 1993; Mallat, 2009), and artificial neural networks (LeCun et al., 2015; Bishop, 1995; Cartwright, 2015). In contrast, the second group of methods defines the function space by a certain regularity. This type of model is mostly developed in statistics and applied to surrogate modeling, including smoothing splines (Craven and Wahba, 1978; Friedman, 1991), tree-based methods (Breiman, 2001; Loh, 2014), kernel regression (Schölkopf and Smola, 2002), Gaussian processes (Rasmussen and Williams, 2006; Bachoc et al., 2014; Lataniotis et al., 2018), and support vector regression (Vapnik, 1995; Moustapha et al., 2018).

In this thesis, we only focus on the *non-intrusive* methods to construct surrogate models. In this framework, the detailed structure of the computational model is not considered, and one only treats it as a black box: by feeding in an input value, it returns the model response. Hence, the only necessary operation to carry out with the simulator is model evaluation. More precisely, the simulator is evaluated on a set of selected input values $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\right\}$ called the *experimental design* (ED), and the associated model outcomes are collected into $\mathcal{Y} = \left\{y^{(1)}, \dots, y^{(N)}\right\}$ with $y^{(i)} = \mathcal{M}_d\left(\boldsymbol{x}^{(i)}\right)$. Conventional methods to generate the ED includes Monte Carlo sampling, quasi-Monte Carlo sampling, and Latin hypercube sampling as reviewed in Section 2.3.2.2. Non-intrusive methods consist in building surrogate models only from the generated input-output pairs $(\mathcal{X}, \mathcal{Y})$ without modifying or adapting the simulator. The decoupling of computational simulation and surrogate modeling is a very important feature of non-intrusive methods. This facilitates the application of surrogate models to all kinds of simulators, especially to computer codes that involve large-scale computations or substructures that are confidential or not publicly available.

Both Monte Carlo simulation in Section 2.3.2 and non-intrusive surrogate modeling rely on sampling and model evaluations. The main advantage of using surrogate models is that their construction takes into account the regularity of the simulator. For example, if the input-output relation of a simulator happens to be linear despite its complex underlying formulation, only $M+1$ points are needed to build an accurate surrogate model, whereas Monte Carlo simulation ignores the model behavior and only works with the output samples.

In this section, we review two of the most popular and widely used surrogate models in uncertainty quantification, namely polynomial chaos expansions and Gaussian processes.

2.5.1 Polynomial chaos expansions

2.5.1.1 Spectral representation

We assume that the input vector $\boldsymbol{X}$ possesses a joint PDF $f_{\boldsymbol{X}}$ and that the random model output variable Y has a finite variance. This is equivalent to assuming $\mathcal{M}_d \in \mathcal{L}^2\left(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}), \mathbb{P}_{\boldsymbol{X}}\right)$ with the probability measure in Eq. (2.17). Let us further introduce the Hilbert space $\boldsymbol{\mathcal{H}} = \mathcal{L}^2\left(\mathcal{D}_{\boldsymbol{X}}, \mathcal{B}(\mathcal{D}_{\boldsymbol{X}}), \mathbb{P}_{\boldsymbol{X}}\right)$ with the inner product

$$\langle g, h\rangle_{\boldsymbol{\mathcal{H}}} = \mathbb{E}\left[g(\boldsymbol{X})h(\boldsymbol{X})\right] = \int_{\mathcal{D}_{\boldsymbol{X}}} g(\boldsymbol{x})h(\boldsymbol{x})\mathbb{P}_{\boldsymbol{X}}(\mathrm{d}\boldsymbol{x}) = \int_{\mathcal{D}_{\boldsymbol{X}}} g(\boldsymbol{x})h(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{2.90}$$

This inner product induces a norm $\|\cdot\|_{\boldsymbol{\mathcal{H}}}$ that reads

$$\|h\|_{\boldsymbol{\mathcal{H}}} \overset{\text{def}}{=} \sqrt{\langle h, h\rangle_{\boldsymbol{\mathcal{H}}}} = \sqrt{\int_{\mathcal{D}_{\boldsymbol{X}}} h^2(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}}. \tag{2.91}$$

According to (Brezis, 2011, Theorem 4.13), $\boldsymbol{\mathcal{H}}$ is *separable*, meaning that $\boldsymbol{\mathcal{H}}$ has a countable dense subset. This implies that $\boldsymbol{\mathcal{H}}$ is a separable Hilbert space (Kantorovich and Akilov, 1982) with a countable orthonormal basis. Denoting the latter by $\{\psi_\gamma : \gamma \in \mathbb{N}\}$, it satisfies

$$\langle \psi_{\gamma_1}, \psi_{\gamma_2} \rangle_{\boldsymbol{\mathcal{H}}} = \delta_{\gamma_1,\gamma_2}, \tag{2.92}$$

where δ is the Kronecker delta defined by $\delta_{\gamma_1,\gamma_2} = 1$ if $\gamma_1 = \gamma_2$ and $\delta_{\gamma_1,\gamma_2} = 0$ otherwise. Most importantly, any function in $\boldsymbol{\mathcal{H}}$ can be represented by a unique expansion onto the basis, and thus the computational model $\mathcal{M}_d$ is expressed by

$$\mathcal{M}_d(\boldsymbol{x}) = \sum_{\gamma \in \mathbb{N}} c_\gamma \psi_\gamma(\boldsymbol{x}), \text{ with } c_\gamma \overset{\text{def}}{=} \langle \mathcal{M}_d, \psi_\gamma \rangle_{\boldsymbol{\mathcal{H}}}, \tag{2.93}$$

where $\boldsymbol{c}$ are the coefficients. The equality stands for the convergence of the infinite series to $\mathcal{M}_d$ in the mean-square sense ($\mathcal{L}^2$), as defined in Section 2.2.6. Since this convergence in $\mathcal{L}^2$ implies the convergence in distribution, one can study the probabilistic properties of Y from the expansion.

2.5.1.2 Polynomial basis

To represent $\mathcal{M}_d$ by a spectral expansion in Eq. (2.93), one needs to construct an appropriate basis $\{\psi_\gamma : \gamma \in \mathbb{N}\}$. In the following presentation, we assume that the input vector $\boldsymbol{X}$ has mutually independent components. The case of dependent input is addressed later on in this section.

Similar to $\boldsymbol{\mathcal{H}}$, the space $\mathcal{H}_j = \mathcal{L}^2\left(\mathbb{R}, \mathcal{B}(\mathbb{R}), \mathbb{P}_{X_j}\right)$ is also a separable Hilbert space with the inner product

$$\langle g, h \rangle_{\mathcal{H}_j} = \mathbb{E}\left[g(X_j)h(X_j)\right] = \int_{\mathcal{D}_{X_j}} g(x_j)h(x_j)\mathbb{P}_{X_j}(\mathrm{d}x_j) = \int_{\mathcal{D}_{X_j}} g(x_j)h(x_j)f_{X_j}(x_j)\mathrm{d}x_j. \tag{2.94}$$

We denote its orthonormal basis by $\left\{\phi_{\alpha_j}^{(j)} : \alpha_j \in \mathbb{N}\right\}$.

For independent input variables, the probability measure $\mathbb{P}_{\boldsymbol{X}}$ can be factorized as a product of the marginal measures. Therefore, there is an isomorphism between $\boldsymbol{\mathcal{H}}$ and the tensor product space $\bigotimes_{j=1}^{M} \mathcal{H}_j$ (Reed and Simon, 1972). In other words, we can form an orthonormal basis for $\boldsymbol{\mathcal{H}}$ by a tensor product of the orthonormal bases associated with $\mathcal{H}_j$ for $j = 1, \ldots, M$. More precisely, the basis is defined by

$$\left\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^M\right\} \text{ with } \psi_{\boldsymbol{\alpha}} = \bigotimes_{j=1}^{M} \phi_{\alpha_j}^{(j)}, \text{ that is, } \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j), \tag{2.95}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^\intercal \in \mathbb{N}^M$ is called a *multi-index* and gathers the indices of the univariate basis functions.

With Eq. (2.95), the construction of a basis for $\boldsymbol{\mathcal{H}}$ reduces to constructing an orthonormal basis for each $\mathcal{H}_j$. Various types of univariate bases can be found in the literature on surrogate models, e.g., polynomials (Xiu and Karniadakis, 2002), wavelets (Meyer, 1993; Mallat, 2009), Fourier series (Trigub and Belinsky, 2004; Millman et al., 2005), and Poincaré basis (Roustant et al., 2017; Lüthen et al., 2022c), which has recently been developed for sensitivity analysis. Among the possible choices, arguably the most common in engineering applications is polynomials.

Following Riesz (1923), if the marginal distribution f_{X_j} can be uniquely determined by its moments,[9] the set of polynomials is dense in $\mathcal{H}_j$. This condition holds for many classical parametric distributions: normal, uniform, gamma, beta, etc. As a result, one can find an orthonormal polynomial basis $\left\{\phi_{\alpha_j}^{(j)} : \alpha_j \in \mathbb{N}\right\}$ for $\mathcal{H}_j$. Here, α_j stands for the polynomial degree. In particular, for conventional distributions such as normal, uniform, Gamma, and beta (see their definition in Table 2.2), the associated univariate orthogonal polynomials are the well-known Hermite, Legendre, Laguerre, and Jacobi polynomials, respectively. A detailed description of these polynomials can be found in Xiu and Karniadakis (2002). For any other marginal distributions that fulfill the moment condition, such a basis can be computed numerically through the *Stieltjes procedure* (Gautschi, 2004).

If a marginal distribution does not allow for a polynomial basis (e.g., lognormal distribution), one can transform the variable X to another random variable Υ with a well-behaved distribution (e.g., normal, uniform). For a continuous random variable X, $U \overset{\text{def}}{=} F_X(X)$ is a random variable following a uniform distribution in $[0,1]$ (Embrechts and Hofert, 2013). Combining it with the quantile transform in Eq. (2.64), one can transform X to Υ with a desired continuous distribution F_Υ by

$$\Upsilon = \mathcal{T}(X) \overset{\text{def}}{=} Q_\Upsilon\left(F_X(X)\right). \tag{2.96}$$

Thus, a univariate basis for X is given by

$$\phi_\alpha = \tilde{\phi}_\alpha \circ \mathcal{T}. \tag{2.97}$$

where $\left\{\tilde{\phi}_\alpha : \alpha \in \mathbb{N}\right\}$ is the polynomial basis associated with the probability distribution of Υ.

If $\boldsymbol{X}$ has dependent components, the tensor product in Eq. (2.95) does generally not produce an orthonormal basis, and the condition for multivariate polynomials being dense in $\boldsymbol{\mathcal{H}}$ is more complicated (see Freud, 1971; Ernst et al., 2012). To circumvent this problem, we can apply a similar procedure as in Eq. (2.96) to transform $\boldsymbol{X}$ into an auxiliary vector $\boldsymbol{\Upsilon} = \boldsymbol{\mathcal{T}}(\boldsymbol{X})$ with independent components (e.g., a standard normal vector; Torre et al., 2019a). The main tool is the *Rosenblatt transform* (Rosenblatt, 1952). Let $\boldsymbol{u} = \mathcal{R}_{\boldsymbol{X}}(\boldsymbol{x})$ with the transform $\mathcal{R}_{\boldsymbol{X}}$ given by

$$u_1 = F_{X_1}(x_1),\ u_2 = F_{X_2|X_1}(x_2 \mid x_1),\ \ldots,\ u_M = F_{X_M|X_1,\ldots,X_{M-1}}(u_M \mid x_1,\ldots,x_{M-1}). \tag{2.98}$$

The random vector $\boldsymbol{U} = \mathcal{R}_{\boldsymbol{X}}(\boldsymbol{X})$ has mutually independent and uniformly distributed components in $[0,1]$. By applying the quantile transform to each component of $\boldsymbol{U}$, we can obtain a random vector $\boldsymbol{\Upsilon}$ with desired marginal distributions. Denote the overall transform as $\boldsymbol{\mathcal{T}} = \tilde{\mathcal{R}}_{\boldsymbol{\Upsilon}} \circ \mathcal{R}_{\boldsymbol{X}}$, where $\tilde{\mathcal{R}}_{\boldsymbol{\Upsilon}}$ follows the definition of the inverse Rosenblatt transform in Eq. (2.66). The basis is then defined with respect to the auxiliary variables, and the associated orthonormal basis function for $\boldsymbol{X}$ is given by

$$\psi_{\boldsymbol{\alpha}} \overset{\text{def}}{=} \bigotimes_{j=1}^{M} \tilde{\phi}_{\alpha_j}^{(j)} \circ \boldsymbol{\mathcal{T}} \implies \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \tilde{\phi}_{\alpha_j}^{(j)}(v_j) \text{ with } \boldsymbol{v} = \boldsymbol{\mathcal{T}}(\boldsymbol{x}). \tag{2.99}$$

[9] This is called the *Hamburger moment problem* (Freud, 1971).

With the polynomial basis $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^M\}$ defined in this section, the spectral representation of $\mathcal{M}_d$ is

$$\mathcal{M}_d(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{2.100}$$

which is called the *polynomial chaos expansion* (PCE) of $\mathcal{M}_d$.

2.5.1.3 Truncation schemes

For practical implementation, it is necessary to truncate the infinite sum in Eq. (2.100) to a finite series to handle the expansion. This consists in defining a finite subset $\mathcal{A} \subset \mathbb{N}^M$ of multi-indices corresponding to the basis functions used in the expansion, which is equivalent to fixing the coefficients to 0 for unselected functions. The approximation error of the truncated PCE compared to the full representation is given by

$$\left\| \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) - \mathcal{M}_d(\boldsymbol{X}) \right\|_{\mathcal{H}}^2 = \left\| \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M\setminus\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \right\|_{\mathcal{H}}^2 = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M\setminus\mathcal{A}} c_{\boldsymbol{\alpha}}^2. \tag{2.101}$$

Hence, the truncated model is accurate if most of the significant terms are included in $\mathcal{A}$.

For many engineering applications, computational models are usually smooth and do not exhibit significant high-order polynomial behaviors. As a result, the most common truncation scheme is to include basis functions with the total degree lower than a given value p, that is,

$$\mathcal{A}^{p,M} = \left\{ \boldsymbol{\alpha} \subset \mathbb{N}^M : \sum_{j}^{M} \alpha_j \leqslant p \right\}. \tag{2.102}$$

The total number of the resulting basis functions is

$$\left|\mathcal{A}^{p,M}\right| = \frac{(p+M)!}{p!\,M!}. \tag{2.103}$$

This number grows very fast with increasing maximum degree p and the number of input variables M. For example, $M = 5$ and $p = 5$ produce 252 terms, whereas $M = 10$ and $p = 10$ result in 184,756 functions in the truncated set. Working with a huge number of basis functions means that we need to estimate the same amount of unknown coefficients (see Section 2.5.1.4), which is impracticable for expensive simulators.

To effectively reduce the cardinality of $\mathcal{A}^{p,M}$, one can assume that the *sparsity-of-effects principle* (Montgomery, 2004) applies to the computational model, as first observed by Blatman (2009). According to this heuristic, most physical systems are dominated by some main effects and low-order interactions, whereas contributions of most high-order interactions are negligible.[10] Therefore, we can limit the interactions between input variables by restricting $\|\boldsymbol{\alpha}\|_0 \leqslant R$ where $\|\boldsymbol{\alpha}\|_0$ counts the non-zero entries of $\boldsymbol{\alpha}$, and R is called the *maximum interaction order* (Blatman, 2009). Another application of the principle leads to the *hyperbolic (q-norm)*

[10] This principle is examined and discussed in the field of factorial designs (Li et al., 2006). In this context, the high-order interactions correspond to the joint effect of three or more input variables.

truncation scheme (Blatman and Sudret, 2011):

$$\mathcal{A}^{p,q,M} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M : \|\boldsymbol{\alpha}\|_q \overset{\text{def}}{=} \left(\sum_{i=1}^{M} |\alpha_i|^q \right)^{\frac{1}{q}} \leqslant p \right\}, \tag{2.104}$$

where $q \leqslant 1$ defines the *quasi*-norm $\|\cdot\|_q$.[11] Using the q-norm, this truncation scheme offers a versatile control of interactions among high-degree polynomials. Note that $q = 1$ leads to $\mathcal{A}^{p,M}$ defined in Eq. (2.102).

2.5.1.4 Model construction

Once the truncated basis functions are selected, what remains is to calculate the coefficients $\boldsymbol{c}_{\mathcal{A}}$ (i.e., the components of $\boldsymbol{c}$ restricted to the truncated set $\mathcal{A}$) to build a PCE surrogate model. For most simulators, analytical closed-form solutions to evaluate the expectation in Eq. (2.93) do not exist. Following the non-intrusive philosophy, the coefficients should instead be estimated from a set of evaluations $(\mathcal{X}, \mathcal{Y})$ of the computational model.

To this end, several methods have been developed. The first one is to compute the expectation in Eq. (2.93) numerically by Gaussian quadrature presented in Section 2.3.3. In this case, $\mathcal{X}$ corresponds to the set of integration points. To alleviate the excessive number of model runs associated with the tensorized scheme for relatively high dimensional problems ($M \geqslant 5$), Smolyak's quadrature rule (Smolyak, 1963; Le Matre et al., 2004) can be adopted to construct a sparse grid. Alternatively, one can employ the Monte Carlo estimation of Eq. (2.93) by taking the empirical mean from an ED generated based on the distribution of $\boldsymbol{X}$ (Ghiocel and Ghanem, 2002).

As an alternative to quadrature, regression methods have gained significant attention for constructing PCEs non-intrusively in the last decade (Berveiller et al., 2006; Blatman and Sudret, 2010, 2011; Sargsyan et al., 2014). We recap the principle of this approach in this section.

In Eq. (2.93), each coefficient $c_{\boldsymbol{\alpha}}$ is defined by projecting $\mathcal{M}_d$ onto $\psi_{\boldsymbol{\alpha}}$, which is equivalent to

$$c_{\boldsymbol{\alpha}} = \arg\min_{a} \mathbb{E}\left[\left(\mathcal{M}_d(\boldsymbol{X}) - a\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \right)^2 \right]. \tag{2.105}$$

Similarly, the coefficients $\boldsymbol{c}_{\mathcal{A}}$ associated with a set of basis functions $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}\}$ can be seen as the coordinates of $\mathcal{M}_d$ projected onto the space spanned by the associated basis functions, namely

$$\boldsymbol{c}_{\mathcal{A}} = \arg\min_{\boldsymbol{a}} \mathbb{E}\left[\left(\mathcal{M}_d(\boldsymbol{X}) - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \right)^2 \right]. \tag{2.106}$$

Because of the orthogonality of the basis, solving Eq. (2.106) leads to the same result as solving Eq. (2.105) for each coefficient individually.

Let the experimental design $\mathcal{X}$ be generated by sampling the distribution of $\boldsymbol{X}$. The *ordinary least-squares* (OLS) estimation is based on approximating the expectation in Eq. (2.106) by the empirical mean, i.e.,

$$\hat{\boldsymbol{c}}_N = \frac{1}{N} \arg\min_{\boldsymbol{c}} \sum_{i=1}^{N} \left(y^{(i)} - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}^{(i)} \right) \right)^2. \tag{2.107}$$

[11] $\|\cdot\|_q$ with $q < 1$ is not a norm because it does not satisfy the triangle inequality.

By re-indexing the basis functions, we can write $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}\}$ as $\{\psi_\gamma : \gamma = 1, \ldots, |\mathcal{A}|\}$. Let us define the design matrix $\boldsymbol{\Psi}$ and the response vector $\boldsymbol{y}$ by

$$\boldsymbol{\Psi} \overset{\text{def}}{=} \begin{pmatrix} \psi_1\left(\boldsymbol{x}^{(1)}\right) & \psi_2\left(\boldsymbol{x}^{(1)}\right) & \cdots & \psi_{|\mathcal{A}|}\left(\boldsymbol{x}^{(1)}\right) \\ \psi_1\left(\boldsymbol{x}^{(2)}\right) & \psi_2\left(\boldsymbol{x}^{(2)}\right) & \cdots & \psi_{|\mathcal{A}|}\left(\boldsymbol{x}^{(2)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1\left(\boldsymbol{x}^{(N)}\right) & \psi_2\left(\boldsymbol{x}^{(N)}\right) & \cdots & \psi_{|\mathcal{A}|}\left(\boldsymbol{x}^{(N)}\right) \end{pmatrix}, \quad \boldsymbol{y} \overset{\text{def}}{=} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix}. \tag{2.108}$$

Then, Eq. (2.107) can be expressed in matrix notation as

$$\hat{\boldsymbol{c}}_N = \frac{1}{N} \arg\min_{\boldsymbol{c}} \left(\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}\right)^\intercal \left(\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}\right), \tag{2.109}$$

and the solution to this problem is given analytically by

$$\hat{\boldsymbol{c}}_N = \left(\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Psi}^\intercal \boldsymbol{y} = \left(\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}\right)^{-1} \left(\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{y}\right). \tag{2.110}$$

Note that it can easily be shown that $\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{y}$ is the empirical projection of $\mathcal{M}_d$ onto each selected basis function, i.e., solving the empirical solution to Eq. (2.105). Consequently, even though Eq. (2.105) and Eq. (2.106) are equivalent, their empirical version would generally lead to different estimators, as $\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}$ is not the identity matrix. However, each element of the matrix $\frac{1}{N}\left(\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}\right)$ reads

$$\frac{1}{N}\left(\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}\right)_{\gamma_1,\gamma_2} = \frac{1}{N}\sum_{i=1}^{N} \psi_{\gamma_1}\left(\boldsymbol{x}^{(i)}\right) \psi_{\gamma_2}\left(\boldsymbol{x}^{(i)}\right), \tag{2.111}$$

which is the empirical mean of $\langle \psi_{\gamma_1}, \psi_{\gamma_2} \rangle_{\boldsymbol{\mathcal{H}}} = \delta_{\gamma_1,\gamma_2}$. Because of the law of large numbers, $\frac{1}{N}\left(\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi}\right)$ converges almost surely to the identity matrix $\boldsymbol{I}_{|\mathcal{A}|}$ of size $|\mathcal{A}|$, i.e.,

$$\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{\Psi} \xrightarrow{\text{a.s.}} \boldsymbol{I}_{|\mathcal{A}|}. \tag{2.112}$$

Since matrix inversion is a continuous function and $\frac{1}{N}\boldsymbol{\Psi}^\intercal \boldsymbol{y}$ converges almost surely to $\boldsymbol{c}_{\mathcal{A}}$, the OLS estimator in Eq. (2.110) is *consistent*, meaning that

$$\hat{\boldsymbol{c}}_N \xrightarrow{\text{a.s.}} \boldsymbol{c}_{\mathcal{A}}. \tag{2.113}$$

Eq. (2.113) offers a theoretical justification for using Eq. (2.110). In practice, however, we can only afford a limited number of model runs. When there are only a few available data points but many coefficients to be estimated, Eq. (2.110) is prone to *overfitting*, which means that the fitted model performs well on the given data set (e.g., interpolates the data points) but fails to represent the global behavior of $\mathcal{M}_d$. As a rule-of-thumb, it is usually necessary to have N between $2\,|\mathcal{A}|$ and $3\,|\mathcal{A}|$ to achieve an accurate estimate (Fajraoui et al., 2017).

Due to this requirement, OLS becomes prohibitive in the case of high-dimensional or highly nonlinear problems. Applying the hyperbolic scheme and limiting the interaction order shown in Section 2.5.1.3 may still result in numerous coefficients (e.g., restricting the hyperbolic truncated set $\mathcal{A}^{10,10,0.5}$ with up to third-order interactions still produces 671 terms). To overcome this issue, sparse PCEs have been developed. The main idea

is to minimize Eq. (2.107) while selecting only the most significant basis functions among a candidate set (e.g., generated by applying the truncation schemes in Section 2.5.1.3). Various sparse algorithms can be found in the literature, such as penalized least-squares with least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), least-angle regression (LAR; Efron et al., 2004), orthogonal matching pursuit (Tropp and Gilbert, 2007; Doostan and Owhadi, 2011), stepwise regression (Blatman and Sudret, 2010; Abraham et al., 2017), Bayesian compressive sensing (Babacan et al., 2010; Sargsyan et al., 2014), etc. We refer to Lüthen et al. (2021, 2022a) for a thorough review of sparse PCEs.

In this thesis, we employ LAR to progressively enrich the set of selected basis functions, and we adopt the hybrid algorithm introduced in Blatman and Sudret (2011) to build the final surrogate.

2.5.1.5 Post-processing

After constructing a PCE surrogate model

$$\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \hat{c}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{2.114}$$

where $\hat{\boldsymbol{c}}$ are the estimated coefficients, we can easily perform a large-scale Monte Carlo simulation on $\mathcal{M}^{\mathrm{PCE}}$ for uncertainty quantification analysis, as it is computationally inexpensive. In addition, because of the use of an orthonormal basis, several important quantities of the model output can be computed directly by post-processing the coefficients.

According to the construction of the polynomial basis, the basis function $\psi_{\mathbf{0}}$ with $\mathbf{0} = (0,\dots,0)^{\intercal}$ is always constant and equal to 1. As a result, the mean value of $\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X})$ is equal to the coefficient $\hat{c}_{\mathbf{0}}$, i.e.,

$$\mathbb{E}\left[\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X})\right] = \langle \mathcal{M}^{\mathrm{PCE}}, 1\rangle_{\mathcal{H}} = \hat{c}_{\mathbf{0}}. \tag{2.115}$$

Similarly, the variance can be calculated by

$$\mathrm{Var}\left[\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X})\right] = \mathbb{E}\left[\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\{\mathbf{0}\}} \hat{c}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\right)^2\right] = \left\|\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\{\mathbf{0}\}} \hat{c}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\right\|_{\mathcal{H}}^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\{\mathbf{0}\}} \hat{c}_{\boldsymbol{\alpha}}^2. \tag{2.116}$$

The computation of high-order moments requires evaluating quantities of the form $\mathbb{E}\left[\prod_{k=1}^{r}\psi_{\boldsymbol{\alpha}_k}(\boldsymbol{X})\right]$, where r is the target order of moment. Analytical solutions for $r > 2$ are only available for certain types of polynomials, e.g., Hermite polynomials (Sudret et al., 2006).

Furthermore, PCE has a close link to the Hoeffding–Sobol decomposition presented in Eq. (2.70) (Sudret, 2007). Indeed, one can re-arrange a PCE and group the summands to express the PCE in the form of Eq. (2.70). More precisely, each element $\mathcal{M}_{\mathbf{w}}^{\mathrm{PCE}}$ in the decomposition is given by

$$\mathcal{M}_{\mathbf{w}}^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{w}}} \hat{c}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{2.117}$$

where $\mathcal{A}_{\mathbf{w}} = \{\boldsymbol{\alpha}\in\mathcal{A} : \forall j\in\mathbf{w}, \alpha_j \neq 0 \text{ and } \forall k\in\mathbf{A}\setminus\mathbf{w}, \alpha_k = 0\}$.

As a result, the Sobol' index in Eq. (2.74) can be calculated by

$$S_{\mathbf{w}}^{\mathrm{PCE}} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{w}}} \hat{c}_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} \hat{c}_{\boldsymbol{\alpha}}^2}, \tag{2.118}$$

and thus first-order, higher-order, and total Sobol' indices can all be computed analytically based on Eq. (2.76) and Eq. (2.118).

2.5.2 Gaussian processes

Gaussian process modeling, also known as Kriging (named after Krige, 1951), is a nonparametric technique that interpolates the data points. It has been originally developed in geostatistics and then elaborated to a general model (Santner et al., 2003). To understand this method, we first introduce the concept of random fields, which is also used in the next chapter to model stochastic simulators.

A real-valued *random field* (also known as *random process*) is defined as a collection of real-valued random variables $\{Y_{\boldsymbol{t}} : \boldsymbol{t} \in \mathcal{D}_{\boldsymbol{T}}\}$ indexed by $\boldsymbol{t}$ with $\mathcal{D}_{\boldsymbol{T}} \subset \mathbb{R}^M$ denoting the *index set*. For random vectors, $\mathcal{D}_{\boldsymbol{T}}$ is a subset of $\mathbb{N}$, whereas for random fields $\mathcal{D}_{\boldsymbol{T}}$ can contain intervals (e.g., $\mathcal{D}_{\boldsymbol{T}} = [0,1]^M$). As a result, a realization of a random field $Y_{\boldsymbol{t}}(\omega)$ is a function of its index, also called a *trajectory*. Because $\mathcal{D}_{\boldsymbol{T}}$ is uncountable, the probability measure of a random field cannot be defined as for random vectors in Eq. (2.15). In general, there are two ways to define and characterize a random field.

- The first one is to directly define it as a function-valued random variable as in Section 2.2.2 with the functions defined on $\mathcal{D}_{\boldsymbol{T}}$.[12] For example, we can define a random field having continuous trajectories on $[0,1]^d$ by using orthogonal polynomials (of $\boldsymbol{t}$) in Section 2.5.1.2 with random coefficients. Describing the joint probability distribution of the (countable) set of coefficients would then fully characterize the random field.
- The second one is through finite-dimensional distributions. In this case, one focuses on describing the probability distribution of any finite subset of random variables $\{Y_{\boldsymbol{t}_1}, \ldots, Y_{\boldsymbol{t}_n}\}$. A typical example is the *Gaussian random field* where any finite-dimensional random vector $(Y_{\boldsymbol{t}_1}, \ldots, Y_{\boldsymbol{t}_n})^{\intercal}$ follows a multivariate Gaussian distribution.

Gaussian process modeling of a deterministic function assumes that the target function $\mathcal{M}_d$ is a realization of a Gaussian random field $\{Y_{\boldsymbol{x}} : \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}\}$ that is indexed by the input variables. As a result, the value of $\mathcal{M}_d(\boldsymbol{x})$ is a specific realization of $Y_{\boldsymbol{x}}$.

A Gaussian random field is fully specified by its mean and auto-covariance function

$$m(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{E}\left[Y_{\boldsymbol{x}}\right], \; K(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Cov}\left[Y_{\boldsymbol{x}}, Y_{\boldsymbol{x}'}\right]. \tag{2.119}$$

Any finite subset of variables $\boldsymbol{Y}_n = (Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_n})^{\intercal}$ follows a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{m}_n, \boldsymbol{\Sigma}_n)$ with mean vector $\boldsymbol{m}_n = (m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_n))^{\intercal}$ and covariance matrix $\boldsymbol{\Sigma}_n$ where $(\boldsymbol{\Sigma}_n)_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

[12] In this case, E is a certain function space endowed with a specific σ-algebra $\mathcal{E}$.

More precisely, the joint PDF is given by

$$f_{\boldsymbol{Y}_n}(\boldsymbol{y}_n) = \frac{1}{\sqrt{2\pi \det(\boldsymbol{\Sigma}_n)}} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}_n - \boldsymbol{m}_n\right)^\intercal \boldsymbol{\Sigma}_n^{-1}\left(\boldsymbol{y}_n - \boldsymbol{m}_n\right)\right). \tag{2.120}$$

When evaluating the computational model on the ED $\mathcal{X}$ of size N, we obtain a realization $\boldsymbol{y} = \left(y^{(1)}, \ldots, y^{(N)}\right)^\intercal$ of $\boldsymbol{Y}_N = \left(Y_{\boldsymbol{x}^{(1)}}, \ldots, Y_{\boldsymbol{x}^{(N)}}\right)^\intercal$. Conditioned on this information (using the conditional distribution defined in Section 2.2.4.2 for $\boldsymbol{Y}_N = \boldsymbol{y}$), for any new point $\boldsymbol{x}$, $Y_{\boldsymbol{x}}$ is still a Gaussian random variable $Y_{\boldsymbol{x}} \sim \mathcal{N}(m_N(\boldsymbol{x}), \sigma_N^2(\boldsymbol{x}))$ with

$$\begin{aligned} m_N(\boldsymbol{x}) &= m(\boldsymbol{x}) + \boldsymbol{K}_{\mathcal{X}}^\intercal(\boldsymbol{x})\boldsymbol{\Sigma}_N^{-1}\left(\boldsymbol{y} - \boldsymbol{m}_N\right), \\ \sigma_N^2(\boldsymbol{x}) &= K(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{K}_{\mathcal{X}}^\intercal(\boldsymbol{x})\boldsymbol{\Sigma}_N^{-1}\boldsymbol{K}_{\mathcal{X}}(\boldsymbol{x}), \end{aligned} \tag{2.121}$$

where $\boldsymbol{m}_N$ and $\boldsymbol{\Sigma}_N$ are the mean vector and covariance matrix of the Gaussian random vector $\boldsymbol{Y}_N$ defined above, and $\boldsymbol{K}_{\mathcal{X}}(\boldsymbol{x}) = \left(K\left(\boldsymbol{x}, \boldsymbol{x}^{(1)}\right), \ldots, K\left(\boldsymbol{x}, \boldsymbol{x}^{(N)}\right)\right)^\intercal$ is the covariance between $Y_{\boldsymbol{x}}$ and $\boldsymbol{Y}_N$. It can be easily shown that for any $\boldsymbol{x}^{(i)} \in \mathcal{X}$, $m_N\left(\boldsymbol{x}^{(i)}\right) = y^{(i)}$ and $\sigma_N^2\left(\boldsymbol{x}^{(i)}\right) = 0$, which implies that the surrogate model interpolates the data. Moreover, the mean function is usually used to represent the surrogate model response, and the variance function shows some uncertainty in the estimation.

2.5.2.1 Model construction

To use Gaussian processes for surrogate modeling, the mean and auto-covariance functions need to be estimated from data, since they are unknown *a priori*.

Typically, the mean function is represented by a linear combination of N_f prescribed functions $\boldsymbol{f}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), \ldots, f_{N_f}(\boldsymbol{x})\right)$, that is,

$$m(\boldsymbol{x}) = \boldsymbol{\beta}^\intercal \boldsymbol{f}\left(\boldsymbol{x}\right) \tag{2.122}$$

where $\boldsymbol{\beta}$ is the associated coefficient vector. Common choices of $\boldsymbol{f}$ are $\boldsymbol{f}(\boldsymbol{x}) = 0$ (i.e., the mean function is fixed to 0), constant function $\boldsymbol{f}(\boldsymbol{x}) = 1$, linear functions $\boldsymbol{f}(\boldsymbol{x}) = (1, x_1, \ldots, x_M)^\intercal$, or PCE basis defined in Section 2.5.1.2 (Schöbi et al., 2015).

The auto-covariance function by nature is a *kernel* (Rasmussen and Williams, 2006), meaning that it is symmetric, and that for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{D}_{\boldsymbol{X}}$ and $a_1, \ldots, a_n \in \mathbb{R}$ it satisfies

$$\sum_{i,j} a_i a_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geqslant 0. \tag{2.123}$$

Conversely, any valid kernel (i.e., symmetric and which verifies Eq. (2.123)) is a valid auto-covariance function. As a result, one can choose any kernel $K(\cdot, \cdot)$ in the modeling. In practice, the auto-covariance function is usually factorized by

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 R(\boldsymbol{x}, \boldsymbol{x}'), \tag{2.124}$$

where $R(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Corr}\left[Y_{\boldsymbol{x}}, Y_{\boldsymbol{x}'}\right]$ is the correlation function, and σ^2 is the variance of the marginal distributions of the Gaussian process (i.e., under this factorization, all the marginal distributions have the same variance). By its definition, the correlation function is usually modeled by a normalized kernel $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ (i.e., $R(\boldsymbol{x}, \boldsymbol{x}; \boldsymbol{\theta}) = 1$) parameterized by $\boldsymbol{\theta}$. Because the correlation coefficient quantifies the co-variability of

two random variables deviating from their mean values, R is actually a similarity measure. Most computational models show a certain level of regularity: the closer the input values, the more similar the associated model responses become. This inspires the definition of R as a function of the difference between the points $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = R(\boldsymbol{x} - \boldsymbol{x}'; \boldsymbol{\theta})$, such as the isotropic Gaussian kernel

$$R(\boldsymbol{x}, \boldsymbol{x}'; \theta) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\theta^2}\right), \tag{2.125}$$

where $\|\cdot\|_2$ is the Euclidean norm in $\mathbb{R}^M$. More generally, the multivariate correlation function R can be constructed based on one or several one-dimensional kernels (e.g., exponential, Gaussian, Matérn functions; Rasmussen and Williams, 2006; Dubourg, 2011; Santner et al., 2003).

After parametrizing the mean and covariance functions as described above, all that remains is to estimate $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\theta}$.

Let $\boldsymbol{F}$ be the matrix containing the evaluations of the prescribed functions $\boldsymbol{f}$ on the ED $\mathcal{X}$, i.e., $F_{i,j} = f_j\left(\boldsymbol{x}^{(i)}\right)$. Let $\boldsymbol{R}_{\boldsymbol{\theta}}$ be the correlation matrix $\left(\boldsymbol{R}_{\boldsymbol{\theta}}\right)_{i,j} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right)$. Under this matrix notation, $\boldsymbol{Y}_N$ follows a multivariate Gaussian distribution with mean $\boldsymbol{F}\boldsymbol{\beta}$ and covariance matrix $\sigma^2\boldsymbol{R}_{\boldsymbol{\theta}}$. Following Eq. (2.120), the log-likelihood is given by

$$L_R(\boldsymbol{\beta}, \sigma, \boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) = \frac{N}{2}\log(2\pi) - \frac{\log\left(\det \boldsymbol{R}_{\boldsymbol{\theta}}\right)}{2} + \frac{N}{2}\log\left(\sigma^2\right) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})^{\intercal}\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{2.126}$$

By maximizing the likelihood, we can solve analytically for $\boldsymbol{\beta}$ and σ^2:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} &= \left(\boldsymbol{F}^{\intercal}\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^{\intercal}\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\boldsymbol{y}, \\ \hat{\sigma}_{\boldsymbol{\theta}}^2 &= \frac{1}{N}\left(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}\right)^{\intercal}\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\left(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}\right), \end{aligned} \tag{2.127}$$

where the subscript $_{\boldsymbol{\theta}}$ denotes that the associated quantity depends on the kernel parameters $\boldsymbol{\theta}$. The latter can be estimated by plugging Eq. (2.127) into Eq. (2.126) and again maximizing the likelihood. Alternatively, $\boldsymbol{\theta}$ can be treated as a set of *hyperparameters* and thereafter tuned separately from $\boldsymbol{\beta}$ and σ^2 (MacKay, 1999; Bachoc, 2013). We refer to Santner et al. (2003) and Rasmussen and Williams (2006) for a summary of different estimation methods.

2.6 Summary

In this chapter, we briefly reviewed the general framework and essential ingredients of uncertainty quantification for deterministic computational models. The computational model at the core of the analysis is deterministic. It maps a given set of input parameters that describe the modeled phenomena to the response. For real-world applications, the input parameters are often uncertain. Probability theory is our mathematical tool of choice to model uncertain quantities, which are represented by random variables and characterized by a joint probability distribution. By propagating the input uncertainty, the model output is a random variable whose probabilistic properties are of interest. Sensitivity analysis aims at studying how the output uncertainty can be allocated to the input variables. Monte Carlo simulation is a general method for uncertainty quantification but

can require a large number of model evaluations. To work with computationally expensive simulators, easy-to-evaluate surrogate models, such as polynomial chaos expansions or Gaussian processes, can be built on a small set of model runs to emulate the deterministic input-output map. Because of their simplicity, surrogate models can be used in a large-scale Monte Carlo simulation to perform uncertainty quantification analysis.

The fact that the polynomial is an approximation does not necessarily detract from its usefulness because all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind.

George Box

3

Stochastic surrogate models: state of the art

As presented in the previous chapter, computational models considered in classical uncertainty quantification are deterministic, and the uncertainty only comes from the input variables. In contrast, stochastic simulators contain intrinsic stochasticity, and the model response remains random even for a fixed set of input parameters (as we illustrated in Fig. 1.1). Mathematically, a stochastic simulator can be expressed by

$$\begin{aligned} \mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\to \mathbb{R}, \\ (\boldsymbol{x}, \omega) &\mapsto \mathcal{M}_s(\boldsymbol{x}, \omega), \end{aligned} \tag{3.1}$$

where Ω is the sample space as defined in Section 2.2.1. We denote the response random variable by $Y_{\boldsymbol{x}}(\omega) \overset{\text{def}}{=} \mathcal{M}_s(\boldsymbol{x}, \omega)$.

The intrinsic stochasticity comes from the fact that the input variables provide only partial information about the model response, and other relevant variables, called *latent variables* and denoted by $\boldsymbol{\Xi}$, are not explicitly taken into account and remain random. In other words, a stochastic simulator embeds a deterministic model $\mathcal{M}_d : (\boldsymbol{x}, \boldsymbol{\xi}) \mapsto \mathcal{M}_d(\boldsymbol{x}, \boldsymbol{\xi})$, but $\boldsymbol{\xi}$ is uncertain (modeled by $\boldsymbol{\Xi}$) and not identified as part of the input, as illustrated in Fig. 3.1. When fixing the input vector $\boldsymbol{x}$ and letting the latent variables vary in their domain of definition, multiple simulations with the same input $\boldsymbol{x}$ yield different results.

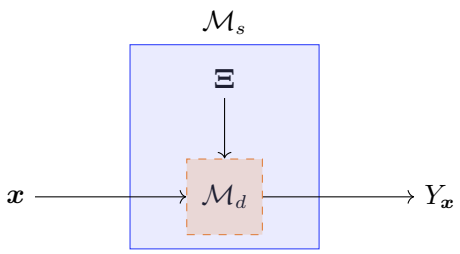


Figure 3.1: Schematic representation of stochastic simulators.

In this thesis, we simplify the notation in Eq. (3.1) by explicitly expressing the source of intrinsic stochasticity

as latent variables, that is,

$$Y_{\boldsymbol{x}} = \mathcal{M}_d(\boldsymbol{x}, \boldsymbol{\Xi}). \tag{3.2}$$

Each model evaluation for $\boldsymbol{x}$ generates a realization $\boldsymbol{\xi}$ of $\boldsymbol{\Xi}$ and produces a realization $\mathcal{M}_d(\boldsymbol{x}, \boldsymbol{\xi})$ of $Y_{\boldsymbol{x}}$. This simplification is introduced to avoid additional definition on the probability space when modeling the random input $\boldsymbol{X}$. Furthermore, we assume that the latent variables $\boldsymbol{\Xi}$ are independent of $\boldsymbol{X}$. When this is not the case, one can always apply the inverse Rosenblatt transform Eq. (2.66) to represent $\boldsymbol{\Xi}$ as a deterministic function J of $\boldsymbol{X}$ and a set of random variables $\tilde{\boldsymbol{\Xi}}$ that are independent of $\boldsymbol{X}$, i.e., $\boldsymbol{\Xi} = J\left(\boldsymbol{X}, \tilde{\boldsymbol{\Xi}}\right)$. The underlying simulator is then given by $\tilde{\mathcal{M}}_d\left(\boldsymbol{x}, \tilde{\boldsymbol{\xi}}\right) \overset{\text{def}}{=} \mathcal{M}_d\left(\boldsymbol{x}, J\left(\boldsymbol{x}, \tilde{\boldsymbol{\xi}}\right)\right)$.

Since computers are deterministic machines, stochastic codes are always implemented following Fig. 3.1. In this case, the distribution of the latent variables is known. Moreover, their values can be controlled and assessed. In principle, one can include $\boldsymbol{\Xi}$ in the input and use the classical uncertainty quantification framework presented in Chapter 2. However, this is not always feasible or meaningful in many cases.

First, in some applications (e.g., wind turbine simulations [Abdallah et al., 2019]), the uncertain sources can be extremely high-dimensional, especially when random processes (which contain an infinite number of random variables) are considered. Therefore, surrogate modeling of such a computational model directly is intractable. To this end, we can extract a finite number of dominant features as input and disregard the residuals as latent variables (Lataniotis, 2019), which transforms the deterministic simulator into a stochastic one.

Second, the latent variables can correspond to some parameters without significant physical meaning or interest. A typical example is agent-based models, which simulate the behavior of a system consisting of a large number of discrete agents under certain conditions or interventions, e.g., the spread of an infectious disease within a population (Cuevas, 2020). There, each agent behaves independently (in the sense that they do not react in the exact same way) and interacts within the population. Studying the individual's effect on the population's overall behavior is much less important than looking at the aggregated macroscopic properties of the population, as practitioners cannot intervene in the precise behavior of every agent in reality. For instance, during the outbreak of Covid-19, policymakers and researchers are more interested in the effectiveness of limiting the contact rate and boosting the vaccination rate of the whole population to contain the spread of the disease (Shattock et al., 2022), instead of focusing on a specific individual. Therefore, only some important macroscopic quantities are modeled as input, and the other agent-related variables are considered latent. Moreover, validation of such a simulator consists in comparing the statistics of the simulated data with the experimental data (Windrum et al., 2007). This implies that the statistical properties of $Y_{\boldsymbol{x}}$ are carefully modeled and validated, but the precise data generation process involving specific values of latent variables is less important. Therefore, it is practical to not include the latent variables in the input.

Third, some uncertain sources may not be accessed or even controlled. This mainly happens in simulations with experimental components such as hybrid simulations (Moustafa and Mosalam, 2015; Tsokanas et al., 2021). In this case, one cannot identify nor control all the relevant variables that affect the value of the model response. Consequently, working with these models in a deterministic manner is impossible. In an even broader sense, experiments can be seen as stochastic simulators: the input contains only a part of the relevant variables identified by the experimentalists, whereas all the others are considered latent.

For deterministic simulators, one model run produces all the information of the response corresponding to the given set of input parameters. Due to the intrinsic randomness, however, evaluating a stochastic simu-

lator once yields only a single realization of the associated response random variable. Hence, it is necessary to *repeatedly* run the simulator for the given set of input parameters to fully characterize the response distribution. The number of simulations could easily become intractable if the distribution of multiple values of the input should be investigated, such as in optimization problems. In the context of uncertainty quantification, Monte Carlo simulation is always applicable, but its slow convergence and the additional layer of randomness due to the intrinsic stochasticity always prevent its application to computationally expensive models.

Due to their deterministic nature, conventional surrogate models developed for deterministic simulators cannot be directly applied to emulate stochastic ones. For instance, polynomial chaos expansions (PCEs) in Section 2.5.1 represent a simulator by a deterministic polynomial function of the input variables, and Gaussian processes in Section 2.5.2 interpolate the data.

In this chapter, we provide a thorough review of different methods that can be used to emulate stochastic simulators. In Section 3.1, we go through the ingredients of statistical models. Methods of this type require the minimum control of computational models (i.e., it does not require replications nor controlling the intrinsic stochasticity), and thus they are the main focus of the review. In Section 3.2, we present replication-based methods that rely on replications to characterize the response distribution. In Section 3.3, we introduce random field approaches, which consist in representing a stochastic simulator by a random field. Finally, we summarize the existing methods and discuss their shortcomings in Section 3.4.

3.1 Statistical models

Let us define the random variable $Y = \mathcal{M}_d(\boldsymbol{X}, \boldsymbol{\Xi})$ that aggregates all the randomness from both the input and the latent variables. The model response $Y_{\boldsymbol{x}}$ for a given input value $\boldsymbol{x}$ is equivalent to Y conditioned on $\boldsymbol{X} = \boldsymbol{x}$. Therefore, the model response for any $\boldsymbol{x}$ can be characterized by some conditional quantities (e.g., conditional mean, variance, quantiles) and fully depicted by the conditional distribution.

Following the non-intrusive requirement and without controlling the intrinsic stochasticity, one evaluation of a stochastic model for $\boldsymbol{x}$ simply produces one realization of Y conditioned on $\boldsymbol{x}$. In other words, the stochastic simulator can be seen as a conditional sampler. Moreover, if we sample $\boldsymbol{X}$ using its probability distribution and then evaluate once the stochastic simulator, the input and output we obtain are a sample from the joint distribution of $(\boldsymbol{X}, Y)$.

The question of approximating stochastic models has arisen recently in engineering applications, but it is a rather classical task in statistics, as real data are never exact and always contain unknown and uncontrollable latent variables. In statistical learning (Hastie et al., 2001), samples are typically considered independent from the joint distribution of $(\boldsymbol{X}, Y)$, which fits perfectly into the framework of non-intrusive surrogate modeling. More precisely, we can generate the data in the following way. We first create an experimental design (ED) $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\right\}$ of size N following the distribution of $\boldsymbol{X}$. Then, we evaluate the stochastic simulator only *once* for each point $\boldsymbol{x}^{(i)} \in \mathcal{X}$ without controlling the latent variables (by default realizations of the latent variables $\boldsymbol{\Xi}$ are generated independently for each model evaluation), i.e., $y^{(i)} = \mathcal{M}_d\left(\boldsymbol{x}^{(i)}, \boldsymbol{\xi}^{(i)}\right)$. We group the associated model responses into $\mathcal{Y} = \left\{y^{(1)}, \dots, y^{(N)}\right\}$. The resulting data set $(\mathcal{X}, \mathcal{Y}) \overset{\text{def}}{=} \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right) : i = 1, \dots, N\right\}$ contains independent samples of $(\boldsymbol{X}, Y)$.

In this section, we recap the principles of statistical learning in Sections 3.1.1 to 3.1.3. More importantly,

we review the statistical models that have been developed to emulate several of the conditional quantities in Section 3.1.4 and estimate the entire conditional distribution in Section 3.1.5 from $(\mathcal{X}, \mathcal{Y})$.

3.1.1 Statistical modeling

In statistical modeling, assumptions are needed to characterize the unknown joint distribution of $(\boldsymbol{X}, Y)$. In the context of uncertainty quantification, the distribution of $\boldsymbol{X}$ is usually known (see Section 2.1), and we are interested in the conditional properties of Y. Hence, the assumptions discussed here are introduced to depict the conditional probability measure $\mathbb{P}_{Y|\boldsymbol{X}}$.

Depending on the prior knowledge and the final goal of the practitioner, assumptions can be made on *two levels*. The first one is about some characteristics of the conditional distribution for any given input value. For instance, the conditional distribution is assumed to be Gaussian or to have a finite mean (if only the mean is of interest). The second level corresponds to how some properties of the conditional distribution vary within the domain of the input variables. For example, we can assume that the conditional mean of Y given $\boldsymbol{X} = \boldsymbol{x}$ is a linear function of $\boldsymbol{x}$. These two levels of assumptions can be jointly or separately proposed by the modeler.

In general, statistical assumptions form a class of (conditional) probability measures $\mathcal{P}$ indexed by a set of parameters, say $\boldsymbol{c}$, from some parameter space $\mathcal{C}$. A statistical model is an instance in $\mathcal{P}$ to approximate the ground truth $\mathbb{P}_{Y|\boldsymbol{X}}$, which can be accurately represented for a certain set of parameters $\boldsymbol{c}_0$. Depending on the statistical assumptions, the parameter space $\mathcal{C}$ can be a subspace of $\mathbb{R}^d$ (e.g., linear regression) but it can be more general, or even infinite-dimensional (e.g., nonparametric models).

As an example, let us assume that the conditional distribution is Gaussian with its mean being an affine function of $\boldsymbol{x}$ and a constant variance that does not depend on $\boldsymbol{x}$. Based on these assumptions, the model is expressed as

$$Y_{\boldsymbol{x}} = \beta_0 + \sum_{j=1}^{M} \beta_j x_j + \epsilon, \tag{3.3}$$

where ϵ is a Gaussian random variable with zero-mean and variance σ^2, and it is independent of $\boldsymbol{X}$. Eq. (3.3) is a classical linear model and widely used in economics and social science. The model parameters here are $\boldsymbol{\beta}$ and σ. If we do not know the parametric form of the conditional mean function but only assume that it is smooth with a certain regularity (e.g., Hölder class; Tsybakov, 2009), the parameter $\boldsymbol{c}$ of this model is a function (that satisfies the regularity condition), and the associated parameter space has infinite dimensions.

In some applications, one is only interested in some conditional quantities of interest (QoIs), which are deterministic functions of $\boldsymbol{x}$ (e.g., the conditional mean and quantiles, see Section 3.1.4).[1] Besides, following certain statistical assumptions (e.g., the conditional distribution belongs to a specific parametric family, see Section 3.1.5.1), $\mathbb{P}_{Y|\boldsymbol{X}}$ can be determined by one or a few deterministic functions $\boldsymbol{g}_0(\cdot)$ of $\boldsymbol{x}$. Hence, characterizing $\mathbb{P}_{Y|\boldsymbol{X}}$ by $\mathcal{P}$ is reduced to choosing an appropriate class of functions $\boldsymbol{\mathcal{G}} = \{\boldsymbol{g}_{\boldsymbol{c}} : \boldsymbol{c} \in \mathcal{C}\}$ to represent $\boldsymbol{g}_0$. For example, Eq. (3.3) uses a linear model to emulate the conditional mean function. As $\boldsymbol{g}_0$ is deterministic, surrogate models like PCEs in Section 2.5.1, artificial neural networks (LeCun et al., 2015; Bishop, 1995; Cartwright, 2015), smoothing splines (Craven and Wahba, 1978; Friedman, 1991) or kernel-based models (Schölkopf and

[1] In this case, these functions do not uniquely determine the conditional probability measure. Since they are the targets of the estimation, we do not distinguish the conditional probability measures if the latter share the same such functions.

Smola, 2002) can be applied to the modeling.[2]

Once $\mathcal{G}$ is selected, we need to estimate a set of parameters (such as $\boldsymbol{\beta}$ and σ in Eq. (3.3)) from data to build up the surrogate model to characterize $\mathbb{P}_{Y|\boldsymbol{X}}$.

3.1.2 M-estimators

Based on the statistical assumptions, various estimation methods (van der Vaart, 1998) can be used. In this thesis, we present the *M-estimator*, which is simple yet the most popular estimation method used in modern statistical learning (e.g., both least-squares and maximum likelihood estimators belong to this family).

Let us first define a *loss* function $\ell(\boldsymbol{c}; \boldsymbol{x}, y)$ which represents a certain "risk" if we fix the model parameters to $\boldsymbol{c}$ and $(\boldsymbol{X}, Y)$ takes the value $(\boldsymbol{x}, y)$. We want to find a set of parameters $\boldsymbol{c}_0 \in \mathcal{C}$ such that the expected risk

$$L(\boldsymbol{c}) = \mathbb{E}\left[\ell(\boldsymbol{c}; \boldsymbol{X}, Y)\right] \tag{3.4}$$

is minimized, i.e.,

$$\boldsymbol{c}_0 = \arg\min_{\boldsymbol{c}} \mathbb{E}\left[\ell(\boldsymbol{c}; \boldsymbol{X}, Y)\right], \tag{3.5}$$

where the expectation is taken with respect to the true joint distribution of $\boldsymbol{X}$ and Y. The loss function is generally designed such that if $\mathbb{P}_{Y|\boldsymbol{X}}$ satisfies the statistical assumptions (i.e., $\mathbb{P}_{Y|\boldsymbol{X}} \in \mathcal{P}$), its associated parameters $\boldsymbol{c}_0$ are the solution to Eq. (3.5) (see different types of loss functions in Section 3.1.4 and Section 3.1.5).

Because the expectation in Eq. (3.5) cannot be analytically evaluated, we replace it with its empirical version given by

$$L_N(\boldsymbol{c}) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)}\right), \tag{3.6}$$

and the associated estimator is defined by

$$\hat{\boldsymbol{c}}_N = \arg\min_{\boldsymbol{c}} L_N(\boldsymbol{c}), \tag{3.7}$$

where the subscript $_N$ denotes the sample size. $\hat{\boldsymbol{c}}_N$ in Eq. (3.7) is called an *M-estimator* with M standing for minimization (or maximization if the negative loss is considered).

For $\mathcal{C}$ being infinite-dimensional or containing (finitely) too many parameters compared to the number of data points, simply optimizing Eq. (3.7) on a finite data set would fail as Eq. (3.7) can be ill-posed. Depending on the statistical models, this problem can be addressed by choosing certain penalty functions (Tibshirani, 1996; Zou and Hastie, 2005; Fan and Li, 2001; Craven and Wahba, 1978; Schölkopf and Smola, 2002) to regularize the estimation.

As a short summary, the parameterization $\mathcal{C}$ and the loss function are at the core of statistical learning: the former encodes the statistical assumptions, and the latter fosters a group of estimation methods.

[2] The application of Gaussian process modeling for stochastic simulators needs specific assumptions, see Section 3.1.5.3.

3.1.3 Assessment of the model performance

After fitting the model, it is necessary to evaluate its performance. With the loss function, we can define the generalization error as the expected loss for the estimated parameter $\hat{\boldsymbol{c}}_N$, that is,

$$\varepsilon_N^{\text{gen}} = \mathbb{E}\left[\ell(\hat{\boldsymbol{c}}_N; \boldsymbol{X}, \boldsymbol{Y}) \mid \mathcal{X}, \mathcal{Y}\right]. \tag{3.8}$$

The conditional expectation above indicates that the model is built on the given data $(\mathcal{X}, \mathcal{Y})$, which is called the *training set*.

To evaluate the expectation in Eq. (3.8), one could compute the average loss on the training set. However, because the model was built on the same data, this is not a good indicator of its general performance: indeed, the model may simply interpolate the data but can hardly be generalized to unseen data. Corrections to the training loss can be derived to estimate the expected generalization error, i.e., $\mathbb{E}\left[\varepsilon_N^{\text{gen}}\right]$, by treating the training samples as random variables and taking the expectation with respect to their distribution. For instance, we can apply some information criteria (Konishi and Kitagawa, 2008) for likelihood-based estimators and certain scaling factors to ordinary least-squares with orthogonal basis (Chapelle et al., 2002).

A more robust way is to generate a separate data set $(\mathcal{X}_{\text{test}}, \mathcal{Y}_{\text{test}})$, called *test set*, following the joint distribution of $(\boldsymbol{X}, Y)$, and calculate the average loss Eq. (3.8) using this data set. However, for expensive computational models, this is not feasible as it involves carrying out new simulations. Furthermore, this approach uses only part of the available data to train the surrogate model.

A more general approach to assess the performance of a statistical model is *cross-validation* (CV; Hastie et al., 2001). The N_{cv}-fold CV procedure is illustrated in Fig. 3.2 and described as follows. First, the data $(\mathcal{X}, \mathcal{Y})$ are randomly partitioned into N_{cv} equal-sized groups $\{V_k : k = 1, \ldots, N_{\text{cv}}\}$ (so each group contains N/N_{cv} data points by assuming that N is divisible of N_{cv}). For $k \in \{1, \ldots, N_{\text{cv}}\}$, we pick the k-th group V_k as the validation set and the other $N_{\text{cv}} - 1$ folds denoted by $V_{\sim k}$ as the training set. By applying the estimation method to $V_{\sim k}$, we obtain $\hat{\boldsymbol{c}}^{(k)}$. Then, the loss function of the fitted model is evaluated on V_k, hence assessing its *out-of-sample* performance by

$$\hat{\varepsilon}_k = \frac{1}{N/N_{\text{cv}}} \sum_{(\boldsymbol{x}, y) \in V_k} \ell\left(\hat{\boldsymbol{c}}^{(k)}; \boldsymbol{x}, y\right). \tag{3.9}$$

We repeat this procedure for each group of the partition $\{V_k : k = 1, \ldots, N_{\text{cv}}\}$, and take the average of the respective scores to estimate the generalized performance[3] of the model, that is,

$$\hat{\varepsilon}_{\text{cv}} = \frac{1}{N_{\text{cv}}} \sum_{k=1}^{N_{\text{cv}}} \hat{\varepsilon}_k. \tag{3.10}$$

For $N_{\text{cv}} = N$, the number of CV-folds is equal to the size of the data set. This case is called *leave-one-out* (LOO) cross-validation, and the associated loss is denoted by $\hat{\varepsilon}_{\text{LOO}}$. From Fig. 3.2, the calculation of $\hat{\varepsilon}_{\text{LOO}}$ can be very costly since it is required to construct N different models. Nevertheless, $\hat{\varepsilon}_{\text{LOO}}$ is widely used in some regression problems with a mean-squared loss, such as Eq. (2.107), because an analytical formula can be derived

[3] The cross-validation error is an estimator of the expected generalization error $\mathbb{E}\left[\varepsilon_{\text{gen}}\right]$ where the expectation is taken with respect to the training data of size $N - N_{\text{cv}}$. However, under certain stability conditions (Bousquet and Elisseeff, 2002), $\hat{\varepsilon}_{\text{cv}}$ is a good approximation to $\varepsilon_N^{\text{gen}}$.

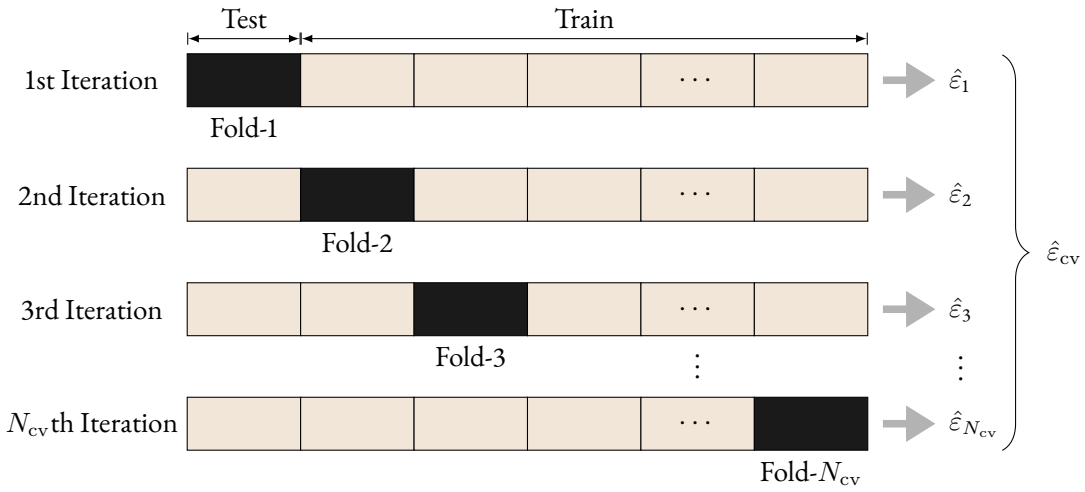


Figure 3.2: N_{cv}-fold cross-validation.

without refitting the model at each time. More precisely, if the fitted response at training points $\mathcal{X}$ is

$$\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}, \tag{3.11}$$

where $\boldsymbol{H}$ is a matrix independent of $\boldsymbol{y}$ and satisfies $\boldsymbol{H}\mathbf{1} = \mathbf{1}$ with $\mathbf{1}$ denoting the vector having all its elements being 1, we can compute the LOO error by

$$\varepsilon_{\text{LOO}} = \sum_{i=1}^{N} \frac{\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{1 - H_{i,i}}, \tag{3.12}$$

where $H_{i,i}$ is the i-th diagonal element of $\boldsymbol{H}$ (Hastie and Tibshirani, 1990). This is the case for PCE built with ordinary least-squares (OLS) in Eq. (2.110), where $\boldsymbol{H} = \boldsymbol{\Psi}\left(\boldsymbol{\Psi}^{\intercal}\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^{\intercal}$.

3.1.4 QoIs estimation

In this section, we review the statistical models applied to the estimation of some conditional properties of Y given $\boldsymbol{X}$, namely the conditional mean function in Section 3.1.4.1, conditional variance function in Section 3.1.4.2, and conditional quantiles in Section 3.1.4.3. As these are deterministic functions of $\boldsymbol{x}$, the models mentioned in Section 3.1.1 can be used for modeling. Therefore, we focus on discussing the loss functions that can be applied to the estimation. It is worth remarking that the presented methods do not assume a specific type of conditional distribution of Y given $\boldsymbol{X}$.[4]

[4]Gaussian process modeling indeed requires assumptions on the type of the distribution of $\mathbb{P}_{Y|\boldsymbol{X}}$: this case is discussed in Section 3.1.5.3.

3.1.4.1 Mean estimation

The most important quantity related to $\mathbb{P}_{Y|\boldsymbol{X}}$ is the conditional mean function $m_{Y|\boldsymbol{X}}$. Estimation of this function is a very classical task in regression where the stochastic model is expressed as

$$Y_{\boldsymbol{x}} = m_{Y|\boldsymbol{X}}(\boldsymbol{x}) + \epsilon, \tag{3.13}$$

where $\mathbb{E}[\epsilon \mid \boldsymbol{X}] = 0$.

Let $\mathcal{G} = \{g_{\boldsymbol{c}} : \boldsymbol{c} \in \mathcal{C}\}$ be a class of functions used to represent $m_{Y|\boldsymbol{X}}$. Following the property in Eq. (2.30), we can use the quadratic loss and obtain the mean-squared error:

$$\ell(\boldsymbol{c}; \boldsymbol{x}, y) = (y - g_{\boldsymbol{c}}(x))^2, \; L_N(\boldsymbol{c}) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - g_{\boldsymbol{c}}\left(\boldsymbol{x}^{(i)}\right)\right)^2. \tag{3.14}$$

As pointed out in Section 3.1.2, some regularization terms can be added to $L_N(\boldsymbol{c})$ to enable the use of complex models. Moreover, Eq. (3.14) offers a systematic way to estimate any conditional expected quantities of Y. More precisely, for any function h, the conditional mean of $h(Y)$ can be tackled by applying the method to the data $(\mathcal{X}, h(\mathcal{Y}))$.

3.1.4.2 Joint mean-variance estimation

The conditional variance is another very important quantity: it characterizes the variability of Y depending on the input variables. If the conditional variance function $v_{Y|\boldsymbol{X}}$ is a constant, the stochastic simulator is called *homoskedastic*. Otherwise, it is called *heteroskedastic*. In the presence of heteroskedasticity, a more appropriate loss function can be constructed to estimate $m_{Y|\boldsymbol{X}}$, that is,

$$L_N(\boldsymbol{c}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left(y^{(i)} - g_{\boldsymbol{c}}\left(\boldsymbol{x}^{(i)}\right)\right)^2}{v\left(\boldsymbol{x}^{(i)}\right)}, \tag{3.15}$$

which is called the weighted mean-squared error (Hastie et al., 2001). As a result, the conditional variance function is sometimes estimated (even when not of interest) to yield a better estimate of the conditional mean function.

Following its definition, the conditional variance function can be expressed as

$$v_{Y|\boldsymbol{X}}(\boldsymbol{x}) = \mathrm{Var}[Y \mid \boldsymbol{X} = \boldsymbol{x}] = \mathbb{E}\left[Y^2 \,\middle|\, \boldsymbol{X} = \boldsymbol{x}\right] - \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]^2. \tag{3.16}$$

Therefore, one can model and estimate separately the conditional mean functions of Y^2 and Y, respectively. The conditional variance function can then be estimated by $\hat{v}_{Y|\boldsymbol{X}}(\boldsymbol{x}) = \hat{m}_{Y^2|\boldsymbol{X}}(\boldsymbol{x}) - \hat{m}_{Y|\boldsymbol{X}}(\boldsymbol{x})^2$ (Härdle and Tsybakov, 1997). However, there is no guarantee that the estimated conditional variance is always positive for any $\boldsymbol{x}$ because the related two conditional mean functions are estimated independently.

Another approach consists in modeling directly the conditional variance function and forming a loss function based on residuals from the mean estimation (Harvey, 1976; Amemiya, 1977; Fan and Yao, 1998). In the first step, we estimate the conditional mean function as in Section 3.1.4.1. If this consistently estimates $m_{Y|\boldsymbol{X}}$, then the residual $r^{(i)} = y^{(i)} - \hat{m}_{Y|\boldsymbol{X}}\left(\boldsymbol{x}^{(i)}\right)$ is a good proxy of $y^{(i)} - m_{Y|\boldsymbol{X}}\left(\boldsymbol{x}^{(i)}\right)$, which is a sample of

$Y - m_{Y|\boldsymbol{X}}(\boldsymbol{X})$. Using the property

$$v_{Y|\boldsymbol{X}}(\boldsymbol{x}) = \mathbb{E}\left[\left(Y - m_{Y|\boldsymbol{X}}(\boldsymbol{x})\right)^2 \,\middle|\, \boldsymbol{X} = \boldsymbol{x}\right], \tag{3.17}$$

we can infer the conditional variance function by regressing the squared residuals (Amemiya, 1977). Following the same line of thought, some methods propose constructing the conditional variance function from certain transforms of the residuals (see Davidian and Carroll, 1987 for a summary).

After fitting the conditional variance function, we can estimate again the conditional mean function using weighted least-squares in Eq. (3.15) and iterate the procedure. This method that alternatively updates the conditional mean and variance functions is called the *feasible generalized least-squares* (FGLS; Wooldridge, 2013).

3.1.4.3 Quantile regression

Quantile regression is a very powerful tool developed in econometrics (Koenker and Bassett, 1978) to estimate conditional quantiles. Following the definition of quantiles, the conditional α-quantile is given by

$$q_\alpha(\boldsymbol{x}) = \inf\left\{q \in \mathbb{R} : F_{Y|\boldsymbol{X}}(q \mid \boldsymbol{x}) \geqslant \alpha\right\}. \tag{3.18}$$

As in Section 3.1.4.1, various classes of functions $\mathcal{G} = \{g_{\boldsymbol{c}} : \boldsymbol{c} \in \mathcal{C}\}$ can be chosen to represent q_α (see Koenker, 2017; Torossian et al., 2020 for comparisons of different models).

The main essence of quantile regression is the loss function, that is,

$$\ell\left(\boldsymbol{c}; \boldsymbol{x}, y\right) = \rho_\alpha\left(y - g_{\boldsymbol{c}}(\boldsymbol{x})\right), \tag{3.19}$$

where ρ_α is called the *check function* (also known as *pinball loss*) defined by

$$\rho_\alpha(t) = \begin{cases} (\alpha - 1)\, t & \text{if } t < 0, \\ \alpha\, t & \text{if } t \geqslant 0. \end{cases} \tag{3.20}$$

Fig. 3.3 compares graphically the check function with the quadratic function (used for the mean estimation). We observe that ρ_α is piecewise linear with a discontinuity point at $t = 0$.

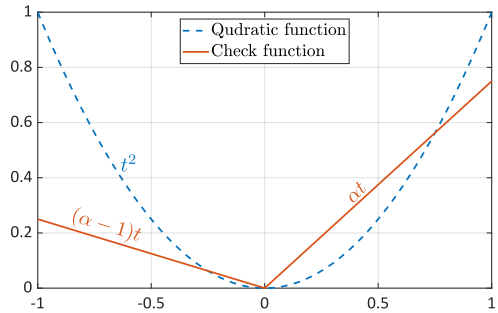


Figure 3.3: Comparison of the quadratic loss and check loss

This loss function is designed such that the α-quantile of the random variable Y solves the following opti-

mization problem

$$Q_Y(\alpha) = \arg\min_{q\in\mathbb{R}} \mathbb{E}\left[\rho_\alpha(Y-q)\right]. \tag{3.21}$$

Therefore, for any given $\boldsymbol{x}$, $q_\alpha(\boldsymbol{x})$ minimizes the conditional version of Eq. (3.21), and thus the conditional α-quantile function can be calculated by

$$q_\alpha(\cdot) = \arg\min_{g\in\mathcal{G}} \mathbb{E}\left[\mathbb{E}\left[\rho_\alpha(Y-g(\boldsymbol{X}))\,|\,\boldsymbol{X}\right]\right] = \arg\min_{g\in\mathcal{G}} \mathbb{E}\left[\rho_\alpha(Y-g(\boldsymbol{X}))\right]. \tag{3.22}$$

3.1.5 Distribution estimation

Not all the conditional quantities can be directly estimated via some loss functions, e.g., entropy and super-quantiles (Acerbi, 2002). Besides, estimating separately the conditional quantities presented in the previous sections can produce incompatible results. For instance, if we apply the quantile regression in Section 3.1.4.3 for various values of α, the estimated conditional quantiles may not generally follow the right order, i.e., lower quantiles can take larger values than higher quantiles (He, 1997).[5]

In order to fully represent $\mathbb{P}_{Y|\boldsymbol{X}}$ and estimate all the conditional quantities in a consistent manner, we need a statistical model to emulate the entire conditional distribution. Considering problems where the response Y has a continuous distribution, it is assumed that the conditional probability density function (PDF) $f_{Y|\boldsymbol{X}}$ exists, and the latter is the function to estimate in this section.

As discussed in Section 3.1.1, two types of assumptions can be made regarding the conditional distribution: (i) how it behaves for any given value of $\boldsymbol{x}$, and (ii) how it varies as a function of $\boldsymbol{x}$ within the input domain $\mathcal{D}_{\boldsymbol{X}}$. We review the methods developed for estimating the conditional PDF by grouping different types of statistical assumptions in the remaining part of this section.

3.1.5.1 Parametric conditional distribution

The most common way to model $f_{Y|\boldsymbol{X}}$ is to assume that the conditional distribution belongs to a certain parametric family of distributions (corresponding to the first type of assumptions). For most applications in statistics, the response distribution is assumed to be Gaussian, which is fully characterized by its conditional mean and variance functions. Other common parametric assumptions consist in using a specific type of distribution from the exponential family (McCullagh and Nelder, 1989), e.g., Bernoulli distribution (for classification problems), Gamma distribution, beta distribution, etc.

Under the parametric assumption, the distribution parameters, say $\boldsymbol{g}_0$, are functions of the input variables. To represent this function, we introduce the second type of assumptions, which gives a class of functions $\boldsymbol{\mathcal{G}} = \{\boldsymbol{g}_{\boldsymbol{c}} : \boldsymbol{c}\in\mathcal{C}\}$ (similar to the conditional QoI in Section 3.1.4). As a result, $\mathcal{C}$ defines a class of conditional distributions which can be expressed by $\tilde{f}_{Y|\boldsymbol{X}}(y\mid\boldsymbol{x}) = \tilde{f}_{Y|\boldsymbol{X}}(y\mid\boldsymbol{g}_{\boldsymbol{c}}(\boldsymbol{x}))$.

The estimation of the model parameters usually follows the principle of maximum likelihood. Let us first define the Kullback–Leibler divergence between two PDFs f_1 and f_2

$$D_{\mathrm{KL}}(f_1\|f_2) = \int_{\mathbb{R}} \log\left(\frac{f_1(y)}{f_2(y)}\right) f_1(y)\mathrm{d}y = \mathbb{E}_1\left[\log\left(\frac{f_1(Y)}{f_2(Y)}\right)\right], \tag{3.23}$$

[5] This issue can be addressed by introducing constraints to bridge the independent optimization problems (Takeuchi et al., 2006). However, the joint optimization is very difficult to solve and does not improve the overall accuracy (Torossian et al., 2020).

where the expectation is taken with respect to the distribution defined by the PDF f_1. By applying the Jensen's inequality, we can show that $D_{\mathrm{KL}}(f_1 \| f_2) \geqslant 0$ and the equality is reached if and only if $f_2 = f_1$. In other words, the PDF f_1 solves the following optimization problem

$$f_1 = \arg\min_f \mathbb{E}_1\left[\log\left(\frac{f_1(Y)}{f(Y)}\right)\right] = \arg\min_f \mathbb{E}_1\left[\log\left(f_1(Y)\right)\right] - \mathbb{E}_1\left[\log\left(f(Y)\right)\right] = \arg\max_f \mathbb{E}_1\left[\log\left(f(Y)\right)\right], \tag{3.24}$$

where the optimization is performed among all the valid PDFs (i.e., non-negative functions with their integration over $\mathbb{R}$ equal to 1). Therefore, the conditional distribution for a given value $\boldsymbol{x}$ of the input solves the conditional version of Eq. (3.24), that is,

$$f_{Y|\boldsymbol{X}}(\cdot \mid \boldsymbol{x}) = \arg\max_f \mathbb{E}\left[\log\left(f(Y)\right) \mid \boldsymbol{X} = \boldsymbol{x}\right]. \tag{3.25}$$

By varying $\boldsymbol{x}$, the conditional distribution can be calculated by

$$f_{Y|\boldsymbol{X}} = \arg\max_f \mathbb{E}\left[\mathbb{E}\left[\log\left(f(Y \mid \boldsymbol{X})\right)\right] \mid \boldsymbol{X}\right] = \arg\max_f \mathbb{E}\left[\log\left(f(Y \mid \boldsymbol{X})\right)\right], \tag{3.26}$$

where the feasible set contains all the valid conditional PDFs (i.e., for any $\boldsymbol{x}$ such that $f_{\boldsymbol{X}}(\boldsymbol{x}) \neq 0$, $f(\cdot \mid \boldsymbol{x})$ is a valid PDF). Therefore, we can form a loss function as follows

$$\ell(\boldsymbol{c}; \boldsymbol{x}, y) = -\log\left(\tilde{f}_{Y|\boldsymbol{X}}\left(y \mid \boldsymbol{g}_{\boldsymbol{c}}(\boldsymbol{x})\right)\right), \tag{3.27}$$

where we add the negative sign to turn the maximization to minimization to comply with the concept "loss" as in Section 3.1.2. The M-estimator associated with Eq. (3.27) is called the *maximum likelihood estimator*.

Methods of this class usually rely on rather restrictive assumptions about the type of response distribution. The advantage is that the estimation is very efficient and accurate if the model response does follow (or is very close to) the prescribed distribution family. However, the applicability of these methods is limited to a small group of problems bounded by the assumptions.

3.1.5.2 Nonparametric estimation

To enable a more flexible representation, nonparametric models can be used. In this case, only assumptions on the smoothness of the response distribution and of its variations within $\mathcal{D}_{\boldsymbol{X}}$ are needed (i.e., no parametric assumptions).

Kernel smoothing is the most popular nonparametric approach (Silverman, 1986) to estimate (joint) PDFs. Using this technique, a natural way to estimate the conditional PDF is to first approximate the joint PDF $f_{\boldsymbol{X},Y}$ of $(\boldsymbol{X}, Y)$ and the PDF $f_{\boldsymbol{X}}$ of $\boldsymbol{X}$, and then to express the conditional PDF as the ratio of the two estimated functions (following Eq. (2.24)), that is,

$$\hat{f}_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{\hat{f}_{\boldsymbol{X},Y}(\boldsymbol{x}, y)}{\hat{f}_{\boldsymbol{X}}(\boldsymbol{x})} = \frac{\frac{1}{N}\sum_{i=1}^{N} K_Y\left(y, y^{(i)}; h_y\right) K_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}\right)}{\frac{1}{N}\sum_{i=1}^{N} K_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}\right)}, \tag{3.28}$$

where K_Y and $K_{\boldsymbol{X}}$ are kernels for Y and $\boldsymbol{X}$, and h_y and $\boldsymbol{h}$ are the associated parameters called *bandwidths*. To

guarantee that Eq. (3.28) produces a valid conditional PDF, K_Y should satisfy

$$\int_{\mathbb{R}} K_Y(y, y'; h_y)\mathrm{d}y = 1. \tag{3.29}$$

Eq. (3.28) is referred to as the *kernel conditional density estimator* (KCDE; Hayfield and Racine, 2008).

The estimator in Eq. (3.28) does not take into account the information of the PDF $f_{\boldsymbol{X}}$ of $\boldsymbol{X}$. To improve the accuracy of the estimation, we can replace the denominator in Eq. (3.28) by $f_{\boldsymbol{X}}$, as suggested in some literature (Lacour, 2015; Bertin et al., 2016):

$$\hat{f}_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{\frac{1}{N}\sum_{i=1}^{N} K_Y\left(y, y^{(i)}; h_y\right) K_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}\right)}{f_{\boldsymbol{X}}(\boldsymbol{x})}. \tag{3.30}$$

However, Eq. (3.28) does not produce a valid conditional PDF, as the integration over y is equal to

$$\int_{\mathbb{R}} \frac{\frac{1}{N}\sum_{i=1}^{N} K_Y\left(y, y^{(i)}; h_y\right) K_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}\right)}{f_{\boldsymbol{X}}(\boldsymbol{x})}\mathrm{d}y = \frac{\frac{1}{N}\sum_{i=1}^{N} K_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}\right)}{f_{\boldsymbol{X}}(\boldsymbol{x})}, \tag{3.31}$$

which is generally not 1. If we re-normalize Eq. (3.28) by the inverse of Eq. (3.31), we will obtain the exact same result as Eq. (3.28).

For PDF estimations, multivariate kernels can be constructed as a product of univariate kernels,

$$K_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{h}) = \prod_{j=1}^{M} K_{X_j}(x_j, {x'}_j; h_j), \tag{3.32}$$

and one-dimensional kernels are typically given by

$$K(x, x'; h) = \frac{1}{h} k\left(\frac{x - x'}{h}\right), \tag{3.33}$$

where $k(\cdot)$ is a PDF (Silverman, 1986). Common choices for k include uniform, Gaussian, Epanechnikov, etc. The selection of kernels depends on the statistical assumptions of the problem. For example, if the PDF is expected to be continuous and unbounded, Gaussian kernels are usually quite suitable.

For Eq. (3.28), we need to choose the bandwidths h_y and $\boldsymbol{h}$. This cannot be achieved by maximizing the likelihood since the loss function reaches $-\infty$ for $h_y = 0$, which makes the optimization problem ill-posed. One way to solve the problem is to introduce certain Gaussian assumptions (Silverman, 1986; Chen et al., 2001). A more robust method is to use cross-validation: we treat the bandwidths as hyperparameters and select their values such that the CV error in Section 3.1.3 is minimized. Holmes et al. (2007) proposed using the negative log-likelihood function in Eq. (3.27) as a loss and optimizing the LOO error. Another choice of the loss function that is widely used in nonparametric estimations corresponds to the integrated mean-squared error (Hall et al.,

2004). The conditional PDF can be calculated by

$$\begin{aligned} f_{Y|\boldsymbol{X}} &= \arg\min_f \int_{\mathcal{D}_{\boldsymbol{X}}} \int_{\mathbb{R}} \left(f(y \mid \boldsymbol{x}) - f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x})\right)^2 \mathrm{d}y \, f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \arg\min_f \int_{\mathcal{D}_{\boldsymbol{X}}} \int_{\mathbb{R}} f(y \mid \boldsymbol{x})^2 f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} - 2 \int_{\mathcal{D}_{\boldsymbol{X}}} \int_{\mathbb{R}} f(y \mid \boldsymbol{x}) f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \mathrm{d}y \\ &= \arg\min_f \int_{\mathcal{D}_{\boldsymbol{X}}} \int_{\mathbb{R}} f(y \mid \boldsymbol{x})^2 f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} - 2\mathbb{E}\left[f(Y \mid \boldsymbol{X})\right]. \end{aligned} \tag{3.34}$$

By modeling f as Eq. (3.28), we can tune the bandwidths by optimizing an approximation to the objective function in Eq. (3.34). The first term in Eq. (3.34) does not involve the unknown conditional distribution and can be easily computed (e.g., by Monte Carlo integration if an analytical solution is not available). The second term involves an expectation, but it cannot be replaced by its empirical version, since $h_y = 0$ would minimize the approximation of Eq. (3.34) and lead the loss function once again to $-\infty$. For a more robust estimation, we apply the leave-one-out cross-validation: first, for each point of the data set, we use the rest of the $N-1$ data to build the conditional PDF following Eq. (3.28) and evaluate its value on the hold-out data; second, we average these N validation values to get an estimate to $\mathbb{E}\left[f(Y \mid \boldsymbol{X})\right]$. Finally, the bandwidths are selected such that the CV estimation of the objective function in Eq. (3.34) is minimized.

Some other nonparametric models can also be found in the literature. Stone (1994) proposed representing the conditional PDF by splines, i.e., as a function of both $\boldsymbol{x}$ and y. Efromovich (2010) projected the conditional PDF onto a Fourier basis of both $\boldsymbol{x}$ and y. Similarly, Izbicki and Lee (2017) adopted a Fourier basis for y but used regression methods for estimating the coefficients (of the Fourier series) as functions of $\boldsymbol{x}$. Fan et al. (1996) suggested transforming approximately the conditional PDF estimation to a regression problem and applying local polynomials for the estimation.

Owing to their flexibility, a significant drawback of nonparametric models is that they suffer from the *curse-of-dimensionality*, meaning that the accuracy of the model decreases drastically with increasing dimension. For example, using a second-order kernel (e.g., Gaussian kernel), the estimator in Eq. (3.30) converges at a rate $N^{-\frac{2}{M+5}}$ to the conditional PDF (Hall et al., 2004).

3.1.5.3 Remarks on Gaussian process models

Because of their flexibility and built-in uncertainty quantification of the estimation, Gaussian processes have been extensively explored in statistical learning to model functions. In this section, we briefly comment on their usage for conditional PDF estimation. In this context, the mean function of the Gaussian random field is always set to zero, and thus such a random process is only characterized by its auto-covariance function.

If the conditional distribution is assumed to be Gaussian with a constant variance (i.e., homoskedastic), we can follow the regression setup in Eq. (3.13) with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ independent of $\boldsymbol{X}$. We model the conditional mean function as a realization of a Gaussian random field with the auto-covariance function $K_m(\cdot, \cdot; \boldsymbol{\theta})$. Because ϵ is additive and Gaussian, the stochastic simulator is also a realization of a Gaussian random field with

$$K(\boldsymbol{x}, \boldsymbol{x}') = K_m(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) + \sigma^2 \delta_{\boldsymbol{x}, \boldsymbol{x}'}. \tag{3.35}$$

Similarly, if the variance of ϵ is not a constant but a known function v of $\boldsymbol{x}$ (which is the conditional variance

function), the auto-covariance function becomes

$$K(\boldsymbol{x}, \boldsymbol{x}') = K_m(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) + v(\boldsymbol{x})\delta_{\boldsymbol{x},\boldsymbol{x}'}. \tag{3.36}$$

Therefore, we can use the same methods in Section 2.5.2 to construct the model and make predictions.

If the conditional variance function is unknown, we can model it as a realization of a lognormal random field, i.e., the exponential transform of a Gaussian random field, to ensure its positiveness. Under this construction, the stochastic simulator is not a realization of a Gaussian random field anymore, and we should look into more details of the formulation.

As a short recap, the model response Y for a given $\boldsymbol{x}$ follows a Gaussian distribution. The mean and variance functions are modeled as realizations of a Gaussian random field and exponential transform of another Gaussian random field, respectively. The two random fields are independent.

Following the properties of the Gaussian random field in Section 2.5.2, the joint distributions of the means $\boldsymbol{M}_{\mathcal{X}}$ and variances $\boldsymbol{V}_{\mathcal{X}}$ of Y at the ED $\mathcal{X}$ are known: $\boldsymbol{M}_{\mathcal{X}}$ follows a multivariate normal distribution with joint PDF denoted by $f_{\boldsymbol{M}_{\mathcal{X}}}$, and $\boldsymbol{V}_{\mathcal{X}}$ follows a multivariate lognormal distribution with joint PDF denoted by $f_{\boldsymbol{V}_{\mathcal{X}}}$. After observing the data $\mathcal{Y}$, the distribution of $\boldsymbol{M}_{\mathcal{X}}$ and $\boldsymbol{V}_{\mathcal{X}}$ can be updated to

$$\begin{aligned} f(\boldsymbol{m}, \boldsymbol{v} \mid \mathcal{X}, \mathcal{Y}) &\propto f_{Y|\boldsymbol{X}}(\mathcal{Y} \mid \boldsymbol{m}, \boldsymbol{v}) \cdot f_{\boldsymbol{M}_{\mathcal{X}}}(\boldsymbol{m}) \cdot f_{\boldsymbol{V}_{\mathcal{X}}}(\boldsymbol{v}) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi v_i}} \exp\left(-\frac{\left(y^{(i)} - m_i\right)^2}{2v_i}\right) \cdot f_{\boldsymbol{M}_{\mathcal{X}}}(\boldsymbol{m}) \cdot f_{\boldsymbol{V}_{\mathcal{X}}}(\boldsymbol{v}), \end{aligned} \tag{3.37}$$

where the first term $f_{Y|\boldsymbol{X}}(\mathcal{Y} \mid \boldsymbol{m}, \boldsymbol{v})$ corresponds to the likelihood.[6]

Applying again the properties of the Gaussian random field, given the data, the distribution of the mean and variance of $Y_{\boldsymbol{x}}$ for any input value $\boldsymbol{x}$ reads

$$f(\tilde{m}, \tilde{v} \mid \mathcal{X}, \mathcal{Y}) = \int f_{M_{\boldsymbol{x}}|\boldsymbol{M}_{\mathcal{X}}}(\tilde{m} \mid \boldsymbol{m})\, f_{V_{\boldsymbol{x}}|\boldsymbol{V}_{\mathcal{X}}}(\tilde{v} \mid \boldsymbol{v})\, f(\boldsymbol{m}, \boldsymbol{v} \mid \mathcal{X}, \mathcal{Y}) \mathrm{d}\boldsymbol{m}\mathrm{d}\boldsymbol{v}, \tag{3.38}$$

where $f_{M_{\boldsymbol{x}}|\boldsymbol{M}_{\mathcal{X}}}(\cdot \mid \boldsymbol{m})$ and $f_{V_{\boldsymbol{x}}|\boldsymbol{V}_{\mathcal{X}}}(\cdot \mid \boldsymbol{v})$ are the conditional PDFs of the two random fields evaluated at $\boldsymbol{x}$ conditioned on their values $\boldsymbol{m}$ and $\boldsymbol{v}$ on $\mathcal{X}$.

Eq. (3.38) is typically used to quantify the uncertainty for estimating the mean and variance of Y at $\boldsymbol{x}$. Moreover, we can aggregate this uncertainty and predict the distribution of $Y_{\boldsymbol{x}}$ conditioned on the data by

$$\begin{aligned} f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}, \mathcal{X}, \mathcal{Y}) &= \int f(y \mid \tilde{m}, \tilde{v})\, f(\tilde{m}, \tilde{v} \mid \mathcal{X}, \mathcal{Y}) \mathrm{d}m\mathrm{d}v \\ &= \int \frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{(y - \tilde{m})^2}{2\tilde{v}}\right) f(\tilde{m}, \tilde{v} \mid \mathcal{X}, \mathcal{Y}) \mathrm{d}m\mathrm{d}v. \end{aligned} \tag{3.39}$$

Eq. (3.38) and Eq. (3.39) follow directly from the conditioning properties of the random fields and the Gaussian assumption of the model response, respectively, whereas Eq. (3.37) carries the information of the data and is the central part of the model. In most cases, Eq. (3.37) cannot be obtained analytically. Goldberg et al.

[6] The updating procedure in Eq. (3.37) follows *Bayes' theorem*. In Bayesian terminology, $f_{\boldsymbol{M}_{\mathcal{X}}}(\boldsymbol{m}) \cdot f_{\boldsymbol{V}_{\mathcal{X}}}(\boldsymbol{v})$ is called the *prior distribution* (provided by the two independent random fields), and $f(\boldsymbol{m}, \boldsymbol{v} \mid \mathcal{X}, \mathcal{Y})$ is the *posterior distribution* given the available data.

(1997) proposed a method based on Markov chain Monte Carlo to sample Eq. (3.37). Other major developments consist in approximating the distribution Eq. (3.37) by variational inference (Lázaro-Gredilla and Titsias, 2011; Saul et al., 2016) or replacing the updated distribution of the variances $\boldsymbol{v}$ at $\mathcal{X}$ with (estimated) deterministic values (Kersting et al., 2007; Marrel et al., 2012; Binois et al., 2018).

Similarly, if the conditional distribution is parametric, we can model the distribution parameters as realizations of Gaussian random fields and follow the same procedure from Eq. (3.37) to Eq. (3.39) for predictions (Rasmussen and Williams, 2006; Chan and Dong, 2011; Saul et al., 2016). The only difference is to adapt the likelihood in Eq. (3.37) and the conditional distribution in Eq. (3.39) to the actual parametric distribution.[7]

When the type of the response distribution is unknown, one can model the conditional PDF as a realization of a *logistic Gaussian process* given by

$$\tilde{f}_{Y|\boldsymbol{X}} \sim \frac{\exp\left(W_{\boldsymbol{x},y}\right)}{\int_{\mathbb{R}} \exp\left(W_{\boldsymbol{x},y}\right) \mathrm{d}y}, \tag{3.40}$$

where $W_{\boldsymbol{x},y}$ is a Gaussian random field indexed by both $\boldsymbol{x}$ and y. The prediction can be made through a certain discretization of the random field $W_{\boldsymbol{x},y}$, which approximates the random field through a finite number of random variables. The distribution of the latter conditioned on the data can be calculated similarly to the parametric cases in Eq. (3.37) (see Lenk, 1991; Tokdar and Ghosh, 2007; Riihimäki and Vehtari, 2014; Gautier et al., 2021 for more details).

In general, Gaussian process models are very flexible, and they embed an intrinsic uncertainty quantification feature. However, employing such models for estimating conditional distributions being not homoskedastic Gaussian is typically time-demanding and requires large data set (Binois et al., 2018).

3.1.5.4 Generative models in deep learning

In recent years, generative models have been extensively investigated in deep learning to estimate probability distributions. For conditional distribution estimations, the conditional generative model reads

$$\boldsymbol{Y}_{\boldsymbol{x}} = g_{\boldsymbol{c}}(\boldsymbol{x}, \boldsymbol{Z}), \tag{3.41}$$

where $\boldsymbol{Z}$ are artificial latent variables introduced to represent intrinsic stochasticity, and $g_{\boldsymbol{c}}$ is parameterized by a deep neural network. Here, $\boldsymbol{Y}_{\boldsymbol{x}}$ is usually a high-dimensional output, e.g., a picture of a human face.

Dinh et al. (2017) proposed a special architecture of neural networks called *normalizing flow*, which is a bijective map from $\boldsymbol{Z}$ to $\boldsymbol{Y}$ (for a given $\boldsymbol{x}$). This allows expressing analytically the conditional PDF, which can thus be estimated by maximum likelihood. For general structures of neural networks, the conditional PDF of $\boldsymbol{Y}_{\boldsymbol{x}}$ is intractable (it requires integrating over the latent variables $\boldsymbol{Z}$). Kingma and Welling (2014) suggested optimizing the *evidence lower bound*, which is a lower bound of the intractable log-likelihood. Goodfellow et al. (2014) developed an adversarial strategy that trains simultaneously a discriminator and the generator Eq. (3.41).

[7] The framework presented here requires the likelihood Eq. (3.37). For M-estimators, however, we only have a loss function but the likelihood is generally unknown (as it depends on the unknown conditional PDF). Nevertheless, one can treat the loss function as a negative log-likelihood and construct a *pseudo* conditional PDF, which allows applying the Gaussian process model to represent the target function. As an example, Lum and Gelfand (2012) derived an asymmetric Laplace distribution from the check loss Eq. (3.19) and use it for quantile regressions. However, it should be kept in mind that the data may not come from the constructed distribution, and the confidence bounds produced by the model should be carefully interpreted. In this case, it is more suitable to see the Gaussian process as a kernel-based method (Schölkopf and Smola, 2002).

The discriminator is built for distinguishing the real data from the samples of the generator, while the latter is trained to "fool" the discriminator as much as possible. Assuming an optimal discriminator, the generator is optimized asymptotically in terms of the *Jensen-Shannon divergence* with respect to the underlying distribution. Finally, Ren et al. (2016) proposed embedding conditional distributions into a reproducing kernel Hilbert space (Song et al., 2009) and employed the metric in that space for fitting the model.

Methods developed in deep learning are designed particularly for big data and high-dimensional model outputs. The objective of these tasks is mainly to capture the dependence structure among the output components. For example, to generate human faces, the practitioner is mostly interested in the overall patterns instead of the probability distribution of a single pixel. The architecture of neural networks is difficult to design (so-called architecture engineering; Elsken et al., 2019), and conventional choices are typically over-parameterized with a huge number of unknowns (e.g., ResNets [He et al., 2016] contain over 10 million parameters). These complex models are black boxes and hard to interpret. In addition, they can be subject to numerical issues (Hanin, 2018; Bau et al., 2019), and the model construction is extremely time-consuming even with powerful hardware.

3.2 Replication-based approaches

The second category of methods is that of replication-based approaches. Unlike the statistical approaches in Section 3.1, where the input and latent variables are treated in the same way when generating the data, methods of this type capitalize on using replications to "separate" the intrinsic stochasticity.

Performing replications for a given input value $\boldsymbol{x}$ produces independent samples from the underlying conditional distribution at $\boldsymbol{x}$. These samples can be used in return to characterize or estimate the conditional distribution $f_{Y|\boldsymbol{X}}(\cdot \mid \boldsymbol{x})$. Collecting this information on the discrete points of the ED $\mathcal{X}$, we can extend it to the entire input space $\mathcal{D}_{\boldsymbol{X}}$ with standard regression methods. The basic ideas of replication-based approaches are summarized as follows:

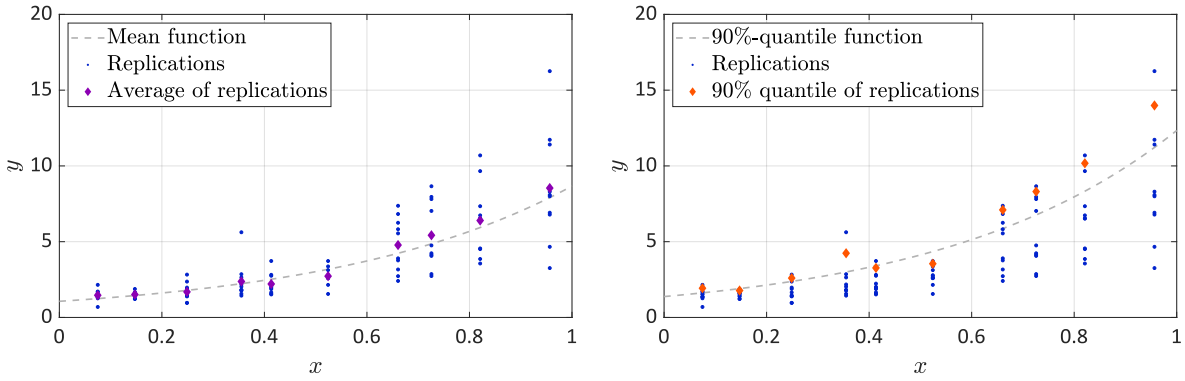
1. For each point $\boldsymbol{x}^{(i)}$ of the ED $\mathcal{X}$, the stochastic simulator is repeatedly evaluated $R^{(i)}$ times, and the results are collected in $\mathcal{Y}^{(i)} = \left\{y^{(i,1)}, \dots, y^{\left(i,R^{(i)}\right)}\right\}$.
2. From $\mathcal{Y}^{(i)}$, one can estimate certain characteristic quantities (e.g., QoIs, distribution parameters), say $\boldsymbol{\lambda}^{(i)}$, of the conditional distribution at $\boldsymbol{x}^{(i)}$.
3. The estimated values $\hat{\boldsymbol{\Lambda}} = \left\{\hat{\boldsymbol{\lambda}}^{(1)}, \dots, \hat{\boldsymbol{\lambda}}^{(N)}\right\}$ are treated as observations of the underlying function evaluated at $\mathcal{X}$ disturbed by Gaussian noises, i.e.,

$$\hat{\boldsymbol{\lambda}}^{(i)} = \boldsymbol{g}\left(\boldsymbol{x}^{(i)}\right) + \epsilon^{(i)}, \tag{3.42}$$

   with $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma_i^2)$. Therefore, the problem is reduced to estimating the conditional mean from data with additive Gaussian noises. This can be achieved by applying the standard regression methods in Section 3.1.4.1 and Section 3.1.5.3 to $(\mathcal{X}, \hat{\boldsymbol{\Lambda}})$.

For example, to estimate the conditional mean function, we calculate the average value of the replications for each point in $\mathcal{X}$. As illustrated in Fig. 3.4a, the empirical means scatter around the underlying conditional mean function, and they are much closer to the reference values than the raw data points. We can observe a

similar behavior for the 90%-quantiles on $\mathcal{X}$ in Fig. 3.4b. The regression model in Eq. (3.42) typically assumes that the local estimations based on replications are unbiased and follow normal distributions. This is generally not true (e.g., the empirical quantiles are biased). However, if the estimator is consistent and asymptotically normal, Eq. (3.42) is applicable when a relatively large number of replications are considered.



(a) Average values of replications

(b) 90%-quantiles of replications

Figure 3.4: Empirical mean values and 90%-quantiles of replications.

The first replication-based method, called *stochastic Kriging*, was developed by Ankenman et al. (2010) for estimating the conditional mean and variance functions. In this approach, Gaussian process regression with homoskedastic additive noise in Eq. (3.35) is applied to estimate the conditional variance function from the empirical variance of the replications (i.e., it assumes that the variance of the noise terms in Eq. (3.42) does not depend on $\boldsymbol{x}$). Evaluating the estimated conditional variance $\hat{v}^{(i)}$ on the ED point $\boldsymbol{x}^{(i)}$ can help provide an estimate $\frac{\hat{v}^{(i)}}{R^{(i)}}$ of the variance of the empirical mean of the replications. Using these variances as σ_i^2 in Eq. (3.42), the conditional mean function is estimated by applying the Gaussian process regression in Eq. (3.36) to the average values of replications.

Based on a similar idea, Plumlee and Tuo (2014) proposed using the Gaussian process regression model with homoskedastic additive noise in Eq. (3.35) to estimate the conditional quantiles from the empirical quantiles of the replications. Torossian et al. (2020) suggested improving the efficiency by applying Eq. (3.36) with the variance of the local inference estimated by bootstrapping.

For the conditional PDF estimation, Moutoussamy et al. (2015) applied the kernel density estimator to the replications. They developed two approaches to extend the conditional PDF estimated from replications on the discrete points in $\mathcal{X}$ to the entire input space $\mathcal{D}_{\boldsymbol{X}}$. In the first one, they chose the nonparametric estimator

$$\hat{f}_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{\sum_{i=1}^{N} K_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h}) \hat{f}_i(y)}{\sum_{i=1}^{N} K_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{h})}, \tag{3.43}$$

where $\hat{f}_i(y)$ is the estimated conditional PDF at $\boldsymbol{x}^{(i)}$ and $K_{\boldsymbol{X}}$ is a multivariate kernel. The second approach consists in looking for an appropriate parameterization of the conditional distribution. This is achieved by performing a functional principal component analysis to construct a basis of finite size and then projecting every $\hat{f}_i(y)$ onto the basis to get the associated coefficients. The latter are treated as $\hat{\boldsymbol{\lambda}}^{(i)}$ in the replication-based procedure described above and are employed to construct the coefficients as functions of $\boldsymbol{x}$ as in Eq. (3.42).

The main advantage of replication-based methods is that the estimation of any conditional QoI or PDF is

transformed into a standard regression problem. However, as the surrogate model is built sequentially from two separate steps, the available data are never used altogether. If the quantities estimated from replications are not sufficient statistics of the underlying conditional distribution, this two-step strategy would lead to a loss of information when estimating the conditional distribution. Besides, since the output of the first step is the input of the second step, the quality of the surrogate model depends strongly on the accuracy of the local inference. If there is a systematic bias in the estimation from replications, such as in the 90%-quantiles in Fig. 3.4b, the regression methods would not be able to filter it out. In order to have an accurate local estimate, a relatively large number of replications is necessary, especially for nonparametric estimators: Moutoussamy et al. (2015) considered 400 replications for the conditional PDF estimation, while Browne et al. (2016) used 10^4 replications to estimate quantiles. This may result in an unaffordable number of model runs when multiplied by the size of the ED.

3.3 Random field modeling

The last type of approach, which has recently been developed, capitalizes on representing a stochastic simulator by a random field (Azzi et al., 2019; Lüthen et al., 2022b).

According to its formulation, a stochastic simulator can be viewed as a random field (presented in Section 2.5.2) indexed by its input parameters $\boldsymbol{x}$. For a specific value $\boldsymbol{x}$, keeping the latent variables random would result in the output random variable $Y_{\boldsymbol{x}} = \mathcal{M}_d(\boldsymbol{x}, \boldsymbol{\Xi})$. By fixing the values of latent variables, the stochastic simulator is a function of $\boldsymbol{x}$, i.e., $\mathcal{M}_d(\cdot, \boldsymbol{\xi})$, which is a trajectory of the random field on $\mathcal{D}_{\boldsymbol{X}}$. As a result, using a random field to approximate the stochastic simulator allows one to consider not only the marginal distribution of $Y_{\boldsymbol{x}}$ but also the entire dependence structure, e.g., the joint distribution of any finite number of random variables $Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_n}$.

The main tool used in this section is the Karhunen–Loève expansion (Karhunen, 1947; Loève, 1978), which consists in decomposing a random field into a set of random variables and functions of the index parameters $\boldsymbol{x}$. Assuming that the auto-covariance function K is continuous and the index space $\mathcal{D}_{\boldsymbol{X}}$ is a closed and bounded subset of $\mathbb{R}^M$,[8] the random field $Y_{\boldsymbol{x}}$ can be represented by

$$Y_{\boldsymbol{x}} = m(\boldsymbol{x}) + \sum_{l=1}^{+\infty} \sqrt{\lambda_l}\, Z_l\, e_l(\boldsymbol{x}), \tag{3.44}$$

where $m_{Y|\boldsymbol{X}}(\boldsymbol{x}) = \mathbb{E}\left[Y_{\boldsymbol{x}}\right]$ is the conditional mean function and

$$\begin{aligned} \int_{\mathcal{D}_{\boldsymbol{X}}} K(\boldsymbol{x}, \boldsymbol{x}') e_l(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}' &= \lambda_l e_l(\boldsymbol{x}), \\ Z_l \overset{\text{def}}{=} \frac{1}{\sqrt{\lambda_l}} \int_{\mathcal{D}_{\boldsymbol{X}}} Y_{\boldsymbol{x}}\, e_l(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. & \end{aligned} \tag{3.45}$$

The functions $\{e_l : l \in \mathbb{N}\}$ are eigenfunctions of the integral operator with corresponding eigenvalues $\{\lambda_l : l \in \mathbb{N}\}$, and they form an orthonormal basis of $\mathcal{L}^2(\mathcal{D}_{\boldsymbol{X}})$. The coefficients $\{Z_l : l \in \mathbb{N}\}$ in the expansion are random

[8] The method was originally developed for the index space being an interval of type $[a, b]$ based on Mercer's theorem (Mercer, 1909; Loève, 1978). Its extension to $\mathcal{D}_{\boldsymbol{X}} \in \mathbb{R}^M$ has various versions (Ghanem and Spanos, 2003; Schwab and Todor, 2006; Minh et al., 2006). Here, we present a rather classical one (Steinwart and Scovel, 2012).

variables with zero-mean $\mathbb{E}[Z_l] = 0$ and unit-variance $\mathrm{Var}[Z_l] = 1$. Moreover, they are uncorrelated, i.e., $\mathrm{Cov}[Z_{l_1}, Z_{l_2}] = 0$ for $l_1 \neq l_2$.

By truncating the infinite series to N_l terms, the random field is approximated by

$$Y_{\boldsymbol{x}} \approx m(\boldsymbol{x}) + \sum_{l=1}^{N_l} \sqrt{\lambda_l}\, Z_l\, e_l(\boldsymbol{x}). \tag{3.46}$$

To build the surrogate model in Eq. (3.46), one needs to estimate the auto-covariance function and the joint distribution of $Z_l : l = 1, \ldots, N_l$. To this end, it is necessary to get access to the information revealing the dependence structure of the computational model as a random field. Evaluating the stochastic simulator with independent samples of $\boldsymbol{X}$ and $\boldsymbol{\Xi}$ always results in independent realizations of Y. In other words, such a sampling scheme breaks the dependence structure of the random field and thus is not suitable for random field approaches. To explore the dependence structure, one should fix the values of the latent variables, and run the stochastic simulators for realizations of $\boldsymbol{X}$, which provides the trajectories on a discrete set of points.

For each random seed, the model is run for the same ED points $\mathcal{X}$, which allows us to estimate the auto-covariance function on the discrete points of $\mathcal{X}$. Based on the empirical covariance matrix, Azzi et al. (2019) proposed two approaches to estimate the eigenvalues and eigenfunctions. The first one emulates directly the auto-covariance function by regressing the empirical covariance matrix and then solves Eq. (3.45). The second one performs a spectral decomposition of the empirical covariance matrix and then emulates the eigenvectors as functions of $\boldsymbol{x}$. Two potential problems may occur in practice: regressing the covariance matrix can lead to invalid auto-covariance functions, and emulating the eigenvectors will generally not provide orthonormal functions in $\mathcal{D}_{\boldsymbol{X}}$.

To solve the problem more robustly, Lüthen et al. (2022b) explored the use of sparse PCEs to approximate the trajectories directly (so the ED for different trajectories does not need to be the same). The empirical auto-covariance function is then estimated from the continuous trajectories. By considering the PDF of $\boldsymbol{X}$ in Eq. (3.45) (only independent uniform distributions were used by Azzi et al., 2019),[9] solving the eigenvalue problem is reduced to a discrete principal component analysis of the PCE coefficients, which can be efficiently tackled. When the basis functions are available, by projecting the emulated continuous trajectories onto the eigenfunctions, one can get samples of the $\{Z_l : l \in N_l\}$ and then apply statistical methods (e.g., vine copula inference [Czado, 2019]) to estimate the joint distribution of the random coefficients.

The methods presented in this section not only provide the conditional distribution but also capture the dependence structure of the stochastic simulator as a random field. It is essential to control the intrinsic stochasticity, which is infeasible in some cases, especially when working with experimental data (e.g., hybrid simulations). Besides, we need to introduce statistical assumptions (e.g., regularities) on the trajectories or on the auto-covariance function so that they can be inferred from discrete samples. In addition, we should also model and estimate the joint probability distribution of the random coefficients $\{Z_l : l = 1, \ldots, N_l\}$ from data whose size is equal to the number of available trajectories. Therefore, the accuracy of such a surrogate depends on the quality of two separate steps: the estimation of trajectories and the statistical inference of the joint distribution of $\{Z_l : l = 1, \ldots, N_l\}$.

[9] This is an extended version of Karhunen–Loève expansion as the index space has an additional probability measure. To make the expansion in Eq. (3.44) still possible, some assumptions on the auto-covariance function should be made, e.g., K is bounded (König, 1986, Theorem 3.a.1).

3.4 Discussion

In this chapter, we reviewed the state-of-the-art methods developed in various fields which can be adopted for emulating stochastic simulators.

For statistical models in Section 3.1, the stochastic simulator can be simply evaluated "as is" without controlling the random seed or requiring replications. These approaches generally consider all the data together to build up the model. Most of the statistical methods have been developed for estimating some conditional QoIs (mean, variance, quantiles). For conditional distribution estimation, however, the common practice relies either on quite restrictive assumptions on the distribution type or on nonparametric estimators which are flexible but require a large number of samples.

Replication-based approaches in Section 3.2 explore repeated model evaluations for the same input value to reduce the stochasticity and characterize the conditional properties. Based on the filtered quantities on the discrete points of $\mathcal{X}$, classical regression-based surrogate modeling methods for deterministic simulators can be used with minimum modifications to emulate the target functions. Within this framework, replications are indispensable. Furthermore, a large number of replications are necessary for conditional distribution estimation especially when nonparametric estimators are employed for local inference.

The third type of approach in Section 3.3 casts the stochastic simulator as a random field. On top of the conditional properties, it also accounts for the statistical dependence of the model response on the intrinsic stochasticity. The construction of the surrogate random field consists of two major steps: (i) emulating trajectories and the auto-covariance function of the stochastic simulator; (ii) applying Karhunen–Loève expansions to represent the intrinsic stochasticity through a finite number of random variables. By their nature, these approaches require the values of the latent variables to be fixed to evaluate trajectories, which is infeasible in some applications.

The objective of this thesis is to develop efficient surrogate modeling methods for emulating the conditional distribution of stochastic simulators, i.e., the dependence structure of the stochastic simulator as a random field is not of interest in the present work. To tackle a wide range of problems, we do not consider the possibility of controlling intrinsic stochasticity. Within this perimeter, we investigate in the following chapters flexible parametric models to bypass the restrictive assumptions for estimating the conditional distribution without going through the nonparametric framework. Replication-based methods are considered in the first place. Then, we focus on developing versatile models and methods that do not require replications.

Part II

Publications

4

Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions

This chapter is a post-print of

> Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal of Uncertainty Quantification*, 10:249–275. DOI:10.1615/Int.J.UncertaintyQuantification.2020033029.[1]

differing from the published paper only in terms of layout and formatting.

Author contributions. **X. Zhu**: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - Original Draft, Visualization. **B. Sudret**: Supervision, Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition.

Abstract

Due to limited computational power, performing uncertainty quantification analyses with complex computational models can be a challenging task. This is exacerbated in the context of stochastic simulators, the response of which to a given set of input parameters, rather than being a deterministic value, is a random variable with unknown probability density function (PDF). Of interest in this paper is the construction of a surrogate that can accurately predict this response PDF for any input parameters. We suggest using a flexible distribution family — the generalized lambda distribution — to approximate the response PDF. The associated distribution parameters are cast as functions of input parameters and represented by sparse polynomial chaos expansions.

To build such a surrogate model, we propose an approach based on a local inference of the response PDF at each point of the experimental design based on replicated model evaluations. Two versions of this framework are proposed and compared on analytical examples and case studies.

4.1 Introduction

Computer models, a.k.a. simulators, are nowadays widely used in the context of design optimization, uncertainty quantification and sensitivity analysis. A simulator is called *deterministic* if repeated runs with the same input parameters produce exactly the same output quantity of interest (QoI); for example, a finite element model of a structure with external load as input and stresses as output is a deterministic simulator. In contrast, a *stochastic simulator* provides different results when run repeatedly with the same input values. In other words, for a given vector of input parameters, the QoI of a stochastic simulator is a *random variable*, whose probability density function (PDF) is of interest. The reason for this intrinsic stochasticity is that some source of randomness inside the model, which can be represented by *latent variables*, is not taken explicitly into account within the input parameters. Therefore, if not all the relevant variables that uniquely determine the output can be fully specified, the model output remains random. Examples of stochastic simulators are encountered when evaluating the performance of a wind turbine under stochastic loads when only some characteristic values of the wind climate are known, or when predicting the price of an option in financial market with only historical data.

Such numerical models can be time-consuming: a single model evaluation may require minutes to hours of simulation, as it is the case for complex fluid dynamic codes. To alleviate the computational burden, surrogate models, a.k.a. emulators, have been successfully developed for deterministic simulators, such as Gaussian processes (Rasmussen and Williams, 2006) and polynomial chaos expansions (Xiu and Karniadakis, 2002; Ghanem and Spanos, 2003). The construction of surrogate models relies on a set of model evaluations, called the *experimental design* (ED). However, when it comes to stochastic simulators, one single model evaluation for a given vector of input parameters is incapable to fully characterize the associated QoI. As a result, repeated runs with the same input parameters, called *replications*, are necessary to obtain the resulting (unknown) probability distribution of the QoI. Consequently, standard surrogate modeling techniques cannot directly be applied to stochastic simulators, due to the very random nature of the output.

Large efforts have been dedicated to estimate summary scalar quantities of the random output as a function of the input parameters, such as the mean value (McCullagh and Nelder, 1989; Iooss and Ribatet, 2009; Ankenman et al., 2010), the standard deviation (Dacidian and Carroll, 1987; Fan and Yao, 1998; Marrel et al., 2012) and quantiles (Bhattacharya and Gangopadhyay, 1990; Plumlee and Tuo, 2014; Koenker, 2017). However, surrogate modeling for the entire response PDF of a stochastic code is a less mature field. Two types of approaches can be found in the literature. The first is known as the *statistical approach*. If the response PDF belongs to the exponential family, generalized linear models (GLM) can be efficiently applied (McCullagh and Nelder, 1989; Hastie and Tibshirani, 1990). When the probability distribution is arbitrary and no prior knowledge on its shape is available, nonparametric estimators may be considered, notably kernel density estimators (Fan and Gijbels, 1996; Hall et al., 2004) and projection estimators (Efromovich, 2010). Nonparametric estimators, however, suffer from the *curse of dimensionality* (Tsybakov, 2009), meaning that the necessary amount

of data needed to achieve sufficient accuracy increases drastically with increasing input dimensionality.

A second approach is the *replication-based* method, which capitalizes instead on available replications to represent the response distribution through a suitably general parametric distribution. The parameters of the latter are then treated as outputs of a deterministic simulator. Conventional deterministic surrogate modeling methods may then be used to emulate these parameters as functions of the input. Note that this approach was initially proposed to estimate summary statistics (Ankenman et al., 2010; Plumlee and Tuo, 2014). It has been extended to more general cases given the functional form of the parametric PDF by Moutoussamy et al. (2015). So far, nonparametric estimators have been used to estimate the distribution from replications (Moutoussamy et al., 2015; Browne et al., 2016). Thus, many replications are necessary, sometimes as many as 10^4 replications for each point of the experimental design, which severely limits the applicability of such an approach.

It is worthwhile to notice that the existing methods either assume a rather restrictive shape of the distribution or require a large number of model evaluations. The present paper aims at designing a replication-based approach which will reduce the necessary amount of replications. To this end, we propose approximating the response PDF of a stochastic simulator by *generalized lambda distributions* (GLDs; Karian and Dudewicz, 2000). Then, the distribution parameters are functions of the input parameters and further represented by polynomial chaos (PC) expansions (Ghanem and Spanos, 2003; Xiu, 2010). Note that we limit ourselves to non-intrusive PC methods in this paper, as the generation of data from the stochastic simulator is purely data-driven. To construct such a surrogate model, we then present two algorithms in this paper: the first one follows the general idea of the replication-based approach, while the second enriches the former with an additional optimization step.

The paper is organized as follows. Sections 4.2 and 4.3 introduce generalized lambda distributions and polynomial chaos expansions, respectively. In Section 4.4, we present our novel algorithms to infer the response PDF of a stochastic simulator based on limited replicated data. Section 4.5 validates the proposed methods through two toy examples, and Section 4.6 illustrates their performance on two applications, namely a stochastic differential equation case study and a wind turbine simulation. Finally, we summarize the main findings of the paper and provide outlooks for future research in Section 4.7.

4.2 Generalized lambda distributions

4.2.1 Formulation

The generalized lambda distribution is a highly flexible four-parameter probability distribution function designed to approximate most of the well-known parametric distributions (Karian and Dudewicz, 2000). Figure 4.1 illustrates how, with the proper choice of parameters, it can accurately approximate, normal, uniform, Student's t, exponential, lognormal, Weibull distributions, among others.

Instead of providing a direct parametrization of the PDF, the GLD parametrizes the *quantile function*, which is the inverse of the cumulative distribution function $Q = F^{-1}(u)$. Therefore, Q is a non-decreasing function defined in $[0, 1]$. In this paper, we consider the GLD of the Freimer–Kollia–Mudholkar–Lin (FKML)

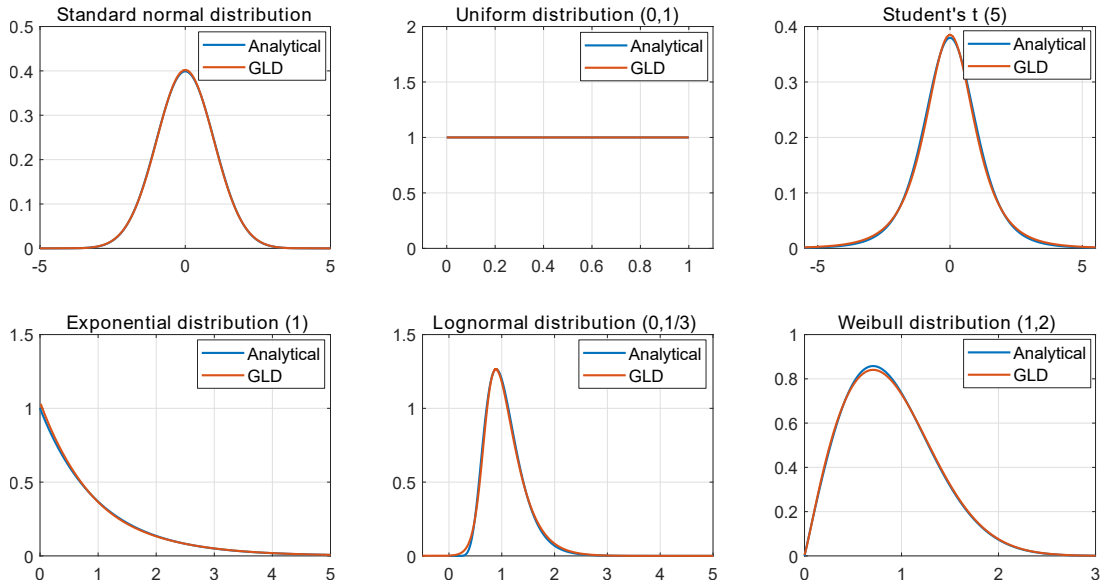


Figure 4.1: Visual comparison of GLD approximation of several common distributions.

family (Freimer et al., 1988), which is defined as:

$$Q(u) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4}\right), \tag{4.1}$$

where λ_1 is the location parameter, λ_2 is the scaling parameter, and λ_3 and λ_4 are shape parameters. To ensure valid quantile functions, it is only required that λ_2 be positive.

Parametrizing the quantile function is equivalent to modeling the inverse probability integral transform. More precisely, the random variable Y with Q as quantile function and the random variable $Q(U)$ with $U \sim \mathcal{U}(0,1)$ follow the same distribution. Therefore, the PDF $f_Y(y)$ of a random variable Y following a GLD can be calculated through a change of variables as follows:

$$f_Y(y) = \frac{f_U(u)}{Q'(u)} = \frac{\lambda_2}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}} \mathbb{1}_{[0,1]}(u), \text{ with } u = Q^{-1}(y), \tag{4.2}$$

where $\mathbb{1}_{[0,1]}$ is the indicator function. A closed form expression of Q^{-1}, and therefore of f_Y, is in general not available.

Figure 4.2 illustrates some PDF shapes of the FKML generalized lambda distributions in the (λ_3, λ_4) plane. It shows that distributions which belong to this family can cover a wide range of shapes that are determined by λ_3 and λ_4. For example, $\lambda_3 = \lambda_4$ produces symmetric PDFs, and $\lambda_3, \lambda_4 < 1$ yields bell-shaped distributions. Importantly, λ_3 and λ_4 control the support and the tail properties of the resulting PDF. The distribution has lower infinite support for $\lambda_3 \leqslant 0$ and upper infinite support for $\lambda_4 \leqslant 0$. In contrast, $\lambda_3 > 0$ implies that the PDF support is left-bounded and $\lambda_4 > 0$ corresponds to right-bounded distributions. More precisely, the support of the PDF, denoted by $\text{supp}\left(f_Y(y)\right) = [B_l, B_u]$, can be derived from Eq. (4.1) as follows:

$$B_l(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} -\infty, & \lambda_3 \leqslant 0 \\ \lambda_1 - \frac{1}{\lambda_2 \lambda_3}, & \lambda_3 > 0 \end{cases}, \quad B_u(\lambda_1, \lambda_2, \lambda_4) = \begin{cases} +\infty, & \lambda_4 \leqslant 0 \\ \lambda_1 + \frac{1}{\lambda_2 \lambda_4}, & \lambda_4 > 0 \end{cases}. \tag{4.3}$$

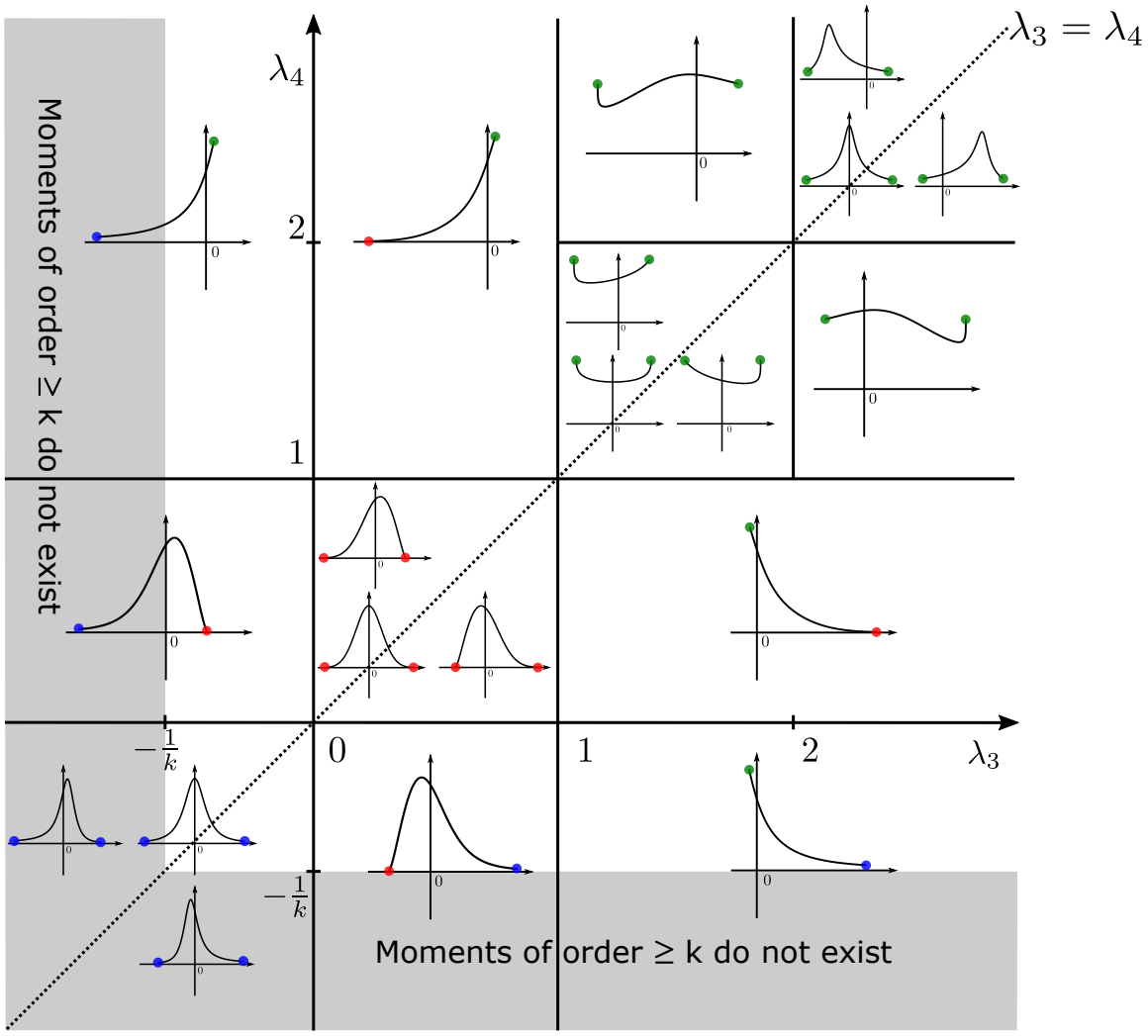


Figure 4.2: A graphical illustration of the shapes that can be represented by the FKML family of GLD as a function of λ_3 and λ_4. The values of λ_1 and λ_2 are set to 0 and 1, respectively. The dotted line is $\lambda_3 = \lambda_4$, which produces symmetric PDFs. The blue points indicate that the PDF has infinite support in the marked direction. In contrast, both the red and green points denote the boundary points of the PDF support. The PDF $f_Y(y) = 0$ on the red dots, whereas $f_Y(y) = 1$ on the green ones.

4.2.2 Estimation of $\boldsymbol{\lambda}$

Many estimation methods have been proposed to fit a generalized lambda distribution to data (Chalabi et al., 2011). Karian and Dudewicz (2010) and Corlu and Meterelliyoz (2016) compared different methods through exhaustive Monte Carlo simulations with various test cases. All of the estimators show comparable performance, and none of them is shown to always outperform the others. The performance depends on the shape of the true distribution, the sample size and the goodness-of-fit criterion used for comparison. In this paper, we choose to apply the method of moments that relies on matching the four moments: mean, variance, skewness, and kurtosis (Lakhany and Massuer, 2000), and the maximum likelihood estimation (Su, 2007).

4.2.2.1 Method of moments

Following Eq. (4.2), the expectation of any function $g(Y)$ can be calculated as

$$\mathbb{E}\left[g(Y)\right] = \mathbb{E}\left[g(Q(U))\right] = \int_0^1 g(Q(u))\mathrm{d}u. \tag{4.4}$$

Accordingly, the k^{th} moment is given by

$$\mathbb{E}\left[Y^k\right] = \int_0^1 \left(\lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right)\right)^k \mathrm{d}u,$$

which is then simplified as

$$\mathbb{E}\left[Y^k\right] = \int_0^1 \left(c + \frac{1}{\lambda_2}s(u)\right)^k \mathrm{d}u, \tag{4.5}$$

where $c \overset{\text{def}}{=} \lambda_1 - \frac{1}{\lambda_2\lambda_3} + \frac{1}{\lambda_2\lambda_4}$ and $s(u) \overset{\text{def}}{=} \frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4}$. To further elaborate Eq. (4.5), we calculate

$$v_k = \int_0^1 s(u)^k \mathrm{d}u = \sum_{j=0}^{k} \frac{(-1)^j}{\lambda_3^{k-j}\lambda_4^j}\binom{k}{j} \mathrm{B}(\lambda_3(k-j)+1, \lambda_4 j + 1), \tag{4.6}$$

where B denotes the beta function. With the help of Eq. (4.6), Eq. (4.5) can be calculated through binomial expansions. As a result, the mean, variance, skewness, and kurtosis are given by (see details in Lakhany and Massuer, 2000)

$$\mu = \mathbb{E}\left[Y\right] = \lambda_1 - \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3+1} - \frac{1}{\lambda_4+1}\right), \quad \sigma^2 = \mathbb{E}\left[(Y-\mu)^2\right] = \frac{(v_2 - v_1^2)}{\lambda_2^2}, \tag{4.7}$$

$$\delta = \mathbb{E}\left[\left(\frac{(Y-\mu)}{\sigma}\right)^3\right] = \frac{v_3 - 3v_1v_2 + 2v_1^3}{(v_2-v_1^2)^{\frac{3}{2}}}, \quad \kappa = \mathbb{E}\left[\left(\frac{(Y-\mu)}{\sigma}\right)^4\right] = \frac{v_4 - 4v_1v_3 + 6v_1^2v_2 - 3v_1^4}{(v_2-v_1)^2}. \tag{4.8}$$

The method of moments matches these four quantities to the associated empirical moments $\left(\hat{\mu}, \hat{\sigma}^2, \hat{\delta}, \hat{\kappa}\right)$ computed from the available sample set $\mathcal{Y} = \left\{y^{(1)} \ldots, y^{(N)}\right\}$. Since v_k is only a nonlinear function of λ_3 and λ_4, the skewness and kurtosis are also functions of only λ_3 and λ_4. Therefore, the fitting procedure first estimates λ_3, λ_4 solving Eq. (4.8), which can be replaced by an optimization problem shown in Eq. (4.9). The

remaining parameters, namely λ_1 and λ_2, are then estimated directly from Eq. (4.7).

$$\left(\hat{\lambda}_3, \hat{\lambda}_4\right) = \arg\min_{\lambda_3,\lambda_4} (\delta(\lambda_3,\lambda_4) - \hat{\delta})^2 + (\kappa(\lambda_3,\lambda_4) - \hat{\kappa})^2. \tag{4.9}$$

Note that for $\lambda_3 \leqslant -0.25$ or $\lambda_4 \leqslant -0.25$, the generalized lambda distribution has infinite fourth order moment as shown in Figure 4.2 for $k = 4$. Therefore, the method of moments only provides $\lambda_3 > -0.25$ and $\lambda_4 > -0.25$.

4.2.2.2 Maximum likelihood estimation

Since the PDF of the generalized lambda distribution is not explicitly given, the negative log-likelihood function can only be evaluated numerically according to Eq. (4.2):

$$l(\boldsymbol{\lambda}) = -\sum_{i=1}^{N} \log\left(\frac{\lambda_2}{u_i^{\lambda_3-1} + (1-u_i)^{\lambda_4-1}}\right), \tag{4.10}$$

where

$$u_i = Q^{-1}(y_i), \quad y_i = Q(u_i) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u_i^{\lambda_3}-1}{\lambda_3} - \frac{(1-u_i)^{\lambda_4}-1}{\lambda_4}\right). \tag{4.11}$$

The maximum likelihood method estimates the distribution parameters by minimizing the negative log-likelihood defined in Eq. (4.10):

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} l(\boldsymbol{\lambda}). \tag{4.12}$$

For a sample set of size N, each likelihood function evaluation requires solving N times the nonlinear equation Eq. (4.11). Consequently, the maximum likelihood estimation can be time-consuming for large data sets. To alleviate the computational burden, we propose the bisection method (Burden et al., 2015) to efficiently solve Eq. (4.11) using the property that $Q(u)$ is monotonic and defined in $[0,1]$.

4.3 Polynomial chaos expansions

4.3.1 Introduction

A deterministic simulator is a function $\mathcal{M}$ that maps a set of input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)^\intercal \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$ to the output quantity of interest $y \in \mathbb{R}$. In the context of uncertainty quantification, the input parameters are modeled by a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_M)^\intercal$ described by its joint distribution $f_{\boldsymbol{X}}$ with support $\mathcal{D}_{\boldsymbol{X}}$. Therefore, the uncertainty in the input variables propagates through the computational model to the output, which becomes a random variable denoted by $Y = \mathcal{M}(\boldsymbol{X})$.

Under the assumption that Y has finite variance, $\mathcal{M}$ belongs to the Hilbert space $\mathcal{H}$ of square-integrable functions with respect to the inner product $\langle u, v\rangle_{\mathcal{H}} = \mathbb{E}[u(\boldsymbol{X})v(\boldsymbol{X})] = \int_{\mathcal{D}_{\boldsymbol{X}}} u(\boldsymbol{x})v(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$. If the joint distribution $f_{\boldsymbol{X}}$ satisfies certain conditions (Ernst et al., 2012), the output random variable Y can be cast as the following spectral expansion:

$$Y = \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M} a_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{4.13}$$

where $a_{\boldsymbol{\alpha}}$ is the coefficient associated to the basis function $\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$. The latter can be obtained as a tensor product of univariate polynomials, each of them being orthogonal with respect to the probability measure $f_{X_i}(x_i)\mathrm{d}x_i$ of the $i-$th variable X_i:

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j). \tag{4.14}$$

Details about the construction of this generalized polynomial chaos expansion can be found in Xiu and Karniadakis (2002) and Sudret (2015).

4.3.2 Sparse PCE

The spectral expansion in Eq. (4.13) is an infinite series. In practice, truncation schemes must be adopted, which leads to approximating the computational model by a finite series defined by a finite multi-index subset $\mathcal{A} \subset \mathbb{N}^M$:

$$\mathcal{M}(\boldsymbol{x}) \approx \mathcal{M}^{PC}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} a_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}). \tag{4.15}$$

Once the set of candidates is selected, regression-based algorithms such as ordinary least squares (Berveiller et al., 2006) can be applied to the data $(\mathcal{X}, \mathcal{Y}) = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right), i = 1, \ldots, N\right\}$ to build the surrogate model. Here, $\mathcal{X}$ denotes the experimental design of the input variables, and $\mathcal{Y}$ are the associated model outputs. One common method to select $\mathcal{A}$ is the full basis of degree p, which contains all the PC basis functions the degree of which is lower than a given value p. However, it is well known that the classical "full" PC approximation suffers from the curse of dimensionality (Blatman and Sudret, 2011), due to the quick increase of the basis size with increasing input dimension or polynomial degree. To overcome this problem, sparse polynomial chaos expansions have been proposed, which select only the most important basis functions among a candidate set (Blatman and Sudret, 2010, 2011), before ordinary least squares are used to compute the coefficients. In the present work, we use the hybrid-LAR algorithm (Marelli and Sudret, 2019) implemented in the open source software UQLab (Marelli and Sudret, 2014) for building sparse PCE. The selection procedure of the algorithm is based on *least-angle regression* (LAR; Efron et al., 2004).

In the sequel, we will combine PCE with the local inference of generalized lambda distributions on each point of the experimental design with replications.

4.4 Infer-and-Fit algorithm and joint modeling

4.4.1 Introduction

We assume that the response PDF of the stochastic simulator for a given input realization $\boldsymbol{x}$ follows a generalized lambda distribution, with distribution parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^\intercal$ that are functions of $\boldsymbol{x}$:

$$Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{4.16}$$

Under appropriate assumptions discussed in Section 4.3, each component of $\boldsymbol{\lambda}(\boldsymbol{x})$ admits a PC representation. For the FKML family, $\lambda_2(\boldsymbol{x})$ is required to be positive (see Section 4.2), and thus the associated PC

approximation is built on the natural logarithm $\log\left(\lambda_2(\boldsymbol{x})\right)$. In a nutshell, $\boldsymbol{\lambda}(\boldsymbol{x})$ are decomposed as

$$\lambda_s\left(\boldsymbol{x}\right) \approx \lambda_s^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_s} a_{s,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad s=1,3,4, \tag{4.17}$$

$$\lambda_2\left(\boldsymbol{x}\right) \approx \lambda_2^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} a_{2,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right), \tag{4.18}$$

where $\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right)$ are the PC approximations of the unknown functions $\boldsymbol{\lambda}(\boldsymbol{x})$. The truncation sets $\{\mathcal{A}_s, s=1,2,3,4\}$ are to be defined, and the coefficients $a_{s,\boldsymbol{\alpha}}$ are the model parameters to be estimated from the samples. For the purpose of clarification, we explicitly express the model parameters $\boldsymbol{a}$ in the surrogate model $\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right)$ so as to emphasize that $\boldsymbol{a}$ are unknown and need to be estimated from the data.

4.4.2 Infer-and-Fit algorithm

To account for the intrinsic randomness, the stochastic simulator is repeatedly run R times for each point $\boldsymbol{x}^{(i)}$ of the experimental design $\mathcal{X}$, and the associated output is denoted by $\mathcal{Y}^{(i)} = \left\{y^{(i,1)}, y^{(i,2)}, \ldots, y^{(i,R)}\right\}$, where the upper index (i,r) refers to the output of the r^{th} replication for the i^{th} set of input parameters in the experimental design.

Following Moutoussamy et al. (2015) and Browne et al. (2016), one straightforward way to build a surrogate model is the Infer-and-Fit algorithm presented in Algorithm 4.1.

Algorithm 4.1 Infer-and-Fit algorithm

1: **for** $i \leftarrow 1, N$ **do**
2: $\quad \hat{\boldsymbol{\lambda}}^{(i)} \leftarrow \hat{\boldsymbol{\lambda}}\left(\mathcal{Y}^{(i)}\right)$
3: **end for**
4: $\hat{\boldsymbol{\Lambda}} \leftarrow \left(\hat{\boldsymbol{\lambda}}^{(1)}, \hat{\boldsymbol{\lambda}}^{(2)}, \ldots, \hat{\boldsymbol{\lambda}}^{(N)}\right)^{\intercal}$
5: $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\tilde{\boldsymbol{a}}) \leftarrow$ Hybrid-LAR $\left(\mathcal{X}, \hat{\boldsymbol{\Lambda}}\right)$

Function $\hat{\boldsymbol{\lambda}}(\cdot)$ in the second line of Algorithm 4.1 denotes an estimator of the distribution parameters based on the replications (see Section 4.2.2), and Hybrid-LAR in the last line is the algorithm (Blatman and Sudret, 2011) used to build sparse PCE for $\boldsymbol{\lambda}(\boldsymbol{x})$.

Algorithm 4.1 consists of two main steps. The first step is used to capture the intrinsic stochasticity through replications. More precisely, this inference step aims at providing an estimate $\hat{\boldsymbol{\lambda}}^{(i)}$ of the distribution parameters $\boldsymbol{\lambda}\left(\boldsymbol{x}^{(i)}\right)$ for each point of the experimental design $\mathcal{X}$. The second step independently builds four surrogate models for the distribution parameters, based on the estimated parameters at discrete points of the experimental design. For the local inference in the first step, we test both the method of moments and the maximum likelihood estimation (a detailed comparison is presented in Section 4.5). Besides, in the second step, we choose to use the hybrid-LAR for sparse PCE constructions, but Algorithm 4.1 is not bounded to this choice: any other regression methods such as ordinary least squares (Berveiller et al., 2006), orthogonal matching pursuit (Tropp and Gilbert, 2007), etc. can be used equivalently.

In practice, the estimator $\hat{\boldsymbol{\lambda}}^{(i)}$ is calculated from replications of finite size due to finite computational budget, and it is subject to noise. Consequently, the choice of the regression setting for building sparse PCE is advantageous because of its robustness to noise (Torre et al., 2019). However, the generalized lambda distribution is rather flexible, so that a few samples cannot guarantee an accurate estimation (Corlu and Meterelliyoz, 2016), and none of the existing methods have been proved to produce unbiased estimators. If a consistent bias is present in the estimation, the use of regression algorithms cannot filter it out. Moreover, the four parameters of the GLD, considered as functions of the input variables, are approximated by four PCE built independently. As a result, the Infer-and-Fit algorithm qualitatively requires many replications R to achieve a good estimate (quantitative results are shown in Section 4.5).

The two separate steps of Algorithm 4.1 may be seen as two successive, independent optimization problems. First, the four parameters of the GLD are optimally fitted for each point $\boldsymbol{x}^{(i)} \in \mathcal{X}$, leading to $\hat{\boldsymbol{\Lambda}}$. Second, coefficients of the PCE of each parameter $\lambda_s(\boldsymbol{x})$ are optimized based on the data collected in $\hat{\boldsymbol{\Lambda}}$, so as to minimize a mean squared error. Intuitively, it appears that these two successive optimizations are suboptimal. We propose to complement the Infer-and-Fit algorithm with a subsequent joint optimization.

4.4.3 Joint PCE-GLD fitting

To reduce the computational cost associated with the need for a large number of replications, we propose a similar approach as that of generalized linear models (McCullagh and Nelder, 1989). In this joint modeling method, PC coefficients $\boldsymbol{a}$ of the four λ_s's are calibrated from the original data $(\mathcal{X}, \mathcal{Y})$ through a maximum likelihood estimation, instead of being calibrated from the local estimates $\hat{\boldsymbol{\Lambda}} = \left\{\hat{\boldsymbol{\Lambda}}^{(1)}, \dots, \hat{\boldsymbol{\Lambda}}^{(N)}\right\}$, as shown in Algorithm 4.1.

To form such an estimator, a deeper insight into the nature of stochastic simulators is necessary. Running once the stochastic simulator for $\boldsymbol{x}$, the output value is a realization of the random variable $Y(\boldsymbol{x})$, which can also be written as $Y \mid \boldsymbol{X} = \boldsymbol{x}$. As a result, the stochastic simulator can be regarded as a *conditional sampler* with the response PDF $f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x})$. Therefore, we can write the joint distribution of $(\boldsymbol{X}, Y)$ as $f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) = f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x})\, f_{\boldsymbol{X}}(\boldsymbol{x})$. The GLD surrogate provides an approximation $f_{Y|\boldsymbol{X}}\left(y \middle| \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a})\right)$ to the conditional PDF. Therefore, the joint PDF of the GLD model is $f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a}) = f_{\boldsymbol{X}}(\boldsymbol{x}) f_{Y|\boldsymbol{X}}\left(y \middle| \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a})\right)$.

Minimizing the Kullback–Leibler divergence between $f_{\boldsymbol{X},Y}(\boldsymbol{x}, y)$ and $f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a})$ gives an appropriate approximation of the GLD surrogate to the underlying true model:

$$\boldsymbol{a}_0 = \arg\min_{\boldsymbol{a}} D_{KL}\left(f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \,\|\, f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a})\right), \tag{4.19}$$

where:

$$\begin{aligned} D_{KL}\left(f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \,\|\, f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a})\right) &= \int f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \log\left(\frac{f_{\boldsymbol{X},Y}(\boldsymbol{x}, y)}{f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a})}\right) \mathrm{d}\boldsymbol{x}\mathrm{d}y \\ &= -\int f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \log\left(f_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a})\right) \mathrm{d}\boldsymbol{x}\mathrm{d}y + \text{const.} \\ &= -\int f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \log\left(f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}; \boldsymbol{a})\, f_{\boldsymbol{X}}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}\mathrm{d}y + \text{const.} \end{aligned} \tag{4.20}$$

Since $f_{\boldsymbol{X}}$ does not contain the model parameters $\boldsymbol{a}$, Eq. (4.19) can be further simplified as

$$\boldsymbol{a}_0 = \arg\min_{\boldsymbol{a}} -\int f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) \log\left(f_{Y|\boldsymbol{X}}\left(y \mid \boldsymbol{\lambda}(\boldsymbol{x}; \boldsymbol{a})\right)\right) \mathrm{d}\boldsymbol{x}\mathrm{d}y. \tag{4.21}$$

Thus the model parameters $\boldsymbol{a}_0$ can be obtained by minimizing the following function:

$$l(\boldsymbol{a}) \overset{\text{def}}{=} -\mathbb{E}_{\boldsymbol{X},Y}\left[\log\left(f_{Y|\boldsymbol{X}}\left(Y\middle|\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{X}; \boldsymbol{a})\right)\right)\right]. \tag{4.22}$$

Note that if the true underlying model can be expressed as the form $f_{Y|\boldsymbol{X}}\left(y|\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a}_{\mathrm{true}})\right)$, $\boldsymbol{a}_0$ from Eq. (4.21) guarantees that $f_{Y|\boldsymbol{X}}\left(y|\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a}_0)\right)$ is the same as that of the true model $\forall \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$.

To estimate $\boldsymbol{a}_0$, the expectation in Eq. (4.22) is replaced by some estimator. In most cases, a sample based empirical average $\frac{1}{N}\sum_{i=1}^{N} \log\left(f_{Y|\boldsymbol{X}}\left(y^{(i)}\middle|\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{a}\right)\right)\right)$ is used, where $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ are drawn independently from the joint distribution $f_{\boldsymbol{X},Y}(\boldsymbol{x}, y)$. For a given r, $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i,r)}\right)\right\}_{i=1}^{N}$ are independent samples, and thus it is natural to consider the estimator

$$\hat{l}^{(r)}(\boldsymbol{a}) = \frac{1}{N}\sum_{i=1}^{N} -\log\left(f_{Y|\boldsymbol{X}}\left(y^{(i,r)}\middle|\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{a}\right)\right)\right) \tag{4.23}$$

to replace the expectation in Eq. (4.22). Note that $\left\{\hat{l}^{(r)}(\boldsymbol{a})\right\}_{r=1}^{R}$ are unbiased estimators of $l(\boldsymbol{a})$. Therefore, the following estimator $\hat{l}(\boldsymbol{a})$ is also unbiased:

$$\hat{l}(\boldsymbol{a}) = \frac{1}{R}\sum_{r=1}^{R} \hat{l}^{(r)}(\boldsymbol{a}). \tag{4.24}$$

For a given $\boldsymbol{a}$, $\left\{\hat{l}^{(r)}(\boldsymbol{a})\right\}_{r=1}^{R}$ have the same variance $\sigma^2(\boldsymbol{a})$, because they are the same estimator applied to different samples $\left\{\left(\boldsymbol{x}^{(1)}, y^{(1,r)}\right), \ldots, \left(\boldsymbol{x}^{(N)}, y^{(N,r)}\right)\right\}$, generated by the same scheme and indexed by r. Nevertheless, these estimators of $l(\boldsymbol{a})$ are not mutually independent due to the presence of replications. Hence, the variance of $\hat{l}(\boldsymbol{a})$ is calculated as follows:

$$\begin{aligned}\mathrm{Var}\left[\hat{l}(\boldsymbol{a})\right] &= \mathrm{Var}\left[\frac{1}{R}\sum_{r=1}^{R} \hat{l}^{(r)}(\boldsymbol{a})\right] \\ &= \frac{1}{R^2}\left(\sum_{r=1}^{R} \mathrm{Var}\left[\hat{l}^{(r)}(\boldsymbol{a})\right] + \sum_{r_1=1}^{R}\sum_{r_2\neq r_1} \mathrm{Cov}\left[\hat{l}^{(r_1)}(\boldsymbol{a}), \hat{l}^{(r_2)}(\boldsymbol{a})\right]\right).\end{aligned} \tag{4.25}$$

Using the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\mathrm{Var}\left[\hat{l}(\boldsymbol{a})\right] &\leqslant \frac{1}{R^2}\left(\sum_{r=1}^{R} \mathrm{Var}\left[\hat{l}^{(r)}(\boldsymbol{a})\right] + \sum_{r_1=1}^{R}\sum_{r_2\neq r_1} \sqrt{\mathrm{Var}\left[\hat{l}^{(r_1)}(\boldsymbol{a})\right]}\sqrt{\mathrm{Var}\left[\hat{l}^{(r_2)}(\boldsymbol{a})\right]}\right) \\ &= \frac{1}{R^2}\left(R\cdot\sigma^2(\boldsymbol{a}) + R(R-1)\cdot\sigma^2(\boldsymbol{a})\right) = \sigma^2(\boldsymbol{a}).\end{aligned} \tag{4.26}$$

The inequality becomes an equality if and only if $\hat{l}^{(r_1)}(\boldsymbol{a})$ is an affine function of $\hat{l}^{(r_2)}(\boldsymbol{a})$. In the context of stochastic simulators, $\hat{l}^{(r_1)}(\boldsymbol{a})$ is not a deterministic function of $\hat{l}^{(r_2)}(\boldsymbol{a})$. Therefore, for a given $\boldsymbol{a}$, $\hat{l}(\boldsymbol{a})$ has less

variance than any estimator of the set $\left\{\hat{l}^{(r)}(\boldsymbol{a})\right\}_{r=1}^{R}$, so it is a better estimator in terms of variance. Replacing the expectation in Eq. (4.22) by $\hat{l}(\boldsymbol{a})$, we end up with a new estimator:

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \hat{l}\left(\boldsymbol{a}\right), \tag{4.27}$$

where

$$\hat{l}\left(\boldsymbol{a}\right) \stackrel{\text{def}}{=} \sum_{i}^{N}\sum_{r}^{R} -\log\left(f_{Y|X}\left(y^{(i,r)}\middle|\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right)\right)\right). \tag{4.28}$$

This estimator by itself does not produce sparsity in the PC representations, meaning that the basis functions for $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{a})$ should be predefined before optimizing Eq. (4.27). To this end, we first exploit Algorithm 4.1 to identify a sparse truncation scheme $\{\mathcal{A}_s, s = 1,\ldots,4\}$ for each component of $\boldsymbol{\lambda}$ in terms of the input vector $\boldsymbol{x}$. Then, we keep this representation and optimize the associated coefficients over the $R \times N$ data points globally (by joint likelihood maximization Eq. (4.27)), instead of separately. Therefore, this new procedure, which is summarized in Algorithm 4.2, can be considered as a *refinement* of Algorithm 4.1, which is expected to improve the surrogate quality with respect to the number of available replications.

Algorithm 4.2 Joint PCE-GLD fitting

1: Apply Algorithm 4.1 to get the sparse PCE truncation schemes $\{\mathcal{A}_s, s = 1,\ldots,4\}$, and the associated coefficients $\tilde{\boldsymbol{a}}$

2: $\hat{\boldsymbol{a}} \leftarrow \arg\min_{\boldsymbol{a}} \hat{l}\left(\boldsymbol{a}\right)$, where $\hat{l}\left(\boldsymbol{a}\right)$ is defined in Eq. (4.28) and

$$\lambda_s^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_s} a_{s,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \quad s = 1,3,4 \tag{4.29}$$

$$\lambda_2^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} a_{2,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right) \tag{4.30}$$

In the second step of Algorithm 4.2, the log-likelihood $\hat{l}(\boldsymbol{a})$ needs to be evaluated with given PC coefficients $\boldsymbol{a}$ for each data point $\left(\boldsymbol{x}^{(i)}, y^{(i,r)}\right)$. The computation details are illustrated in Figure 4.3 and described here. The preliminary step (referred to as Step 0 in Figure 4.3) evaluates the basis functions $\{\psi_{\boldsymbol{\alpha}}, \boldsymbol{\alpha}\in\mathcal{A}_s\}$ at all $\boldsymbol{x}^{(i)}\in\mathcal{X}$. Step 1 calculates the distribution parameters $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right)$ according to Eqs. (4.29) and (4.30), in which the model parameters $\boldsymbol{a}$ are used. The two layers involved in this step (Layer 1 and Layer 2 in Figure 4.3) are not fully connected because the sparse basis sets are independently selected for each components of $\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x};\boldsymbol{a}\right)$ in Algorithm 4.1. Step 2 solves the nonlinear equation $u_{i,r} = Q^{-1}\left(y^{(i,r)}\right)$, where the current values of $\boldsymbol{\lambda}^{(i)}$'s are used, see Eq. (4.1). The nonlinear equation is explicitly written as

$$y^{(i,r)} = \lambda_1^{(i)} + \frac{1}{\lambda_2^{(i)}}\left(\frac{u_{i,r}^{\lambda_3^{(i)}}-1}{\lambda_3^{(i)}} - \frac{(1-u_{i,r})^{\lambda_4^{(i)}}-1}{\lambda_4^{(i)}}\right). \tag{4.31}$$

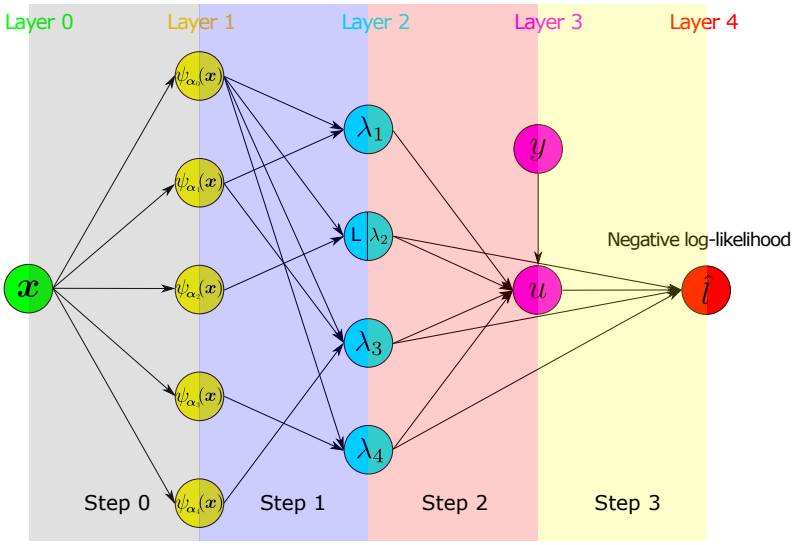


Figure 4.3: Flow chart of the negative log-likelihood calculation

Eventually, Step 3 computes the negative log-likelihood using Eq. (4.2). More precisely, we have

$$-\log\left(f_{Y|\boldsymbol{X}}\left(y^{(i,r)}\middle|\boldsymbol{\lambda}^{(i)}\right)\right)=\log\left(\frac{u_{i,r}^{\lambda_3^{(i)}-1}+(1-u_{i,r})^{\lambda_4^{(i)}-1}}{\lambda_2^{(i)}}\right). \tag{4.32}$$

The optimization problem in the second step of Algorithm 4.2 is not only highly nonlinear but also subject to complex constraints. As discussed in Section 4.2, the FKML family can produce PDFs with bounded support (see Eq. (4.3)), which implies that the negative log-likelihood function will take value $+\infty$ if the data are outside the support. To avoid numerical issues, constraints need to be introduced, and the complete optimization problem becomes

$$\hat{\boldsymbol{a}}=\arg\min_{\boldsymbol{a}}\hat{l}(\boldsymbol{a}), \tag{4.33}$$

$$\text{such that } \forall i \begin{cases} B_l\left(\lambda_1^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right),\lambda_2^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right),\lambda_3^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right)\right)\leqslant \min_r y^{(i,r)}, \\ B_u\left(\lambda_1^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right),\lambda_2^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right),\lambda_4^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{a}\right)\right)\geqslant \max_r y^{(i,r)}, \end{cases} \tag{4.34}$$

where B_l and B_u are computed from Eq. (4.3).

In general, the fact that the negative log-likelihood function can reach $+\infty$ is not a problem because Eq. (4.33) is a minimization problem. Therefore, we can always treat the optimization problem as unconstrained. However, numerical issues can occur when applying unconstrained derivative-based algorithms. For this reason, we choose to use the derivative-based algorithm *trust region without constraints* (Steihaug, 1983) in the first place. If it does not converge, which implies that some constraints are activated, the constrained (1+1)-CMA-ES algorithm (Arnold and Hansen, 2012) available in UQLab (Moustapha et al., 2019) is used instead.

For derivative-based algorithms, the choice of a relevant starting point is important to ensure convergence. In the proposed method, we use the coefficients resulting from the Infer-and-Fit algorithm as the starting point, namely $\tilde{\boldsymbol{a}}$. However, $\tilde{\boldsymbol{a}}$ is generally not guaranteed to be feasible. If it violates the inequality conditions in Eq. (4.34), additional operations are necessary to have a feasible starting point, see details in Section 4.a.1.

When applying derivative-based optimizers to solve Eq. (4.33), using finite difference to calculate gradients would be time-consuming and inaccurate. This is because the likelihood (Eq. (4.28)) is expensive to evaluate, especially considering that NR nonlinear equations (Eq. (4.31)) need to be solved. To alleviate the computational burden, we derived analytical expressions of the derivatives through implicit differentiations of Eqs. (4.1) and (4.2) and the chain rule (see Section 4.a.2 for details). Besides, the Hessian matrix (second order derivatives of $\hat{l}$ with respect to $\boldsymbol{a}$) has also been derived. As a result, each iteration of the trust region algorithm only evaluates once the likelihood function $\hat{l}(\boldsymbol{a})$.

4.5 Analytical examples

In this section, we investigate the performance of the Infer-and-Fit and of the joint modeling algorithm using two analytical examples. The examples are built such that the PDF of $Y(\boldsymbol{x})$ is known but does not follow the generalized lambda distribution, so as to test the flexibility of the proposed approaches. As an inference tool for the first algorithm, we apply both the method of moments and the maximum likelihood estimation to get the values $\hat{\boldsymbol{\lambda}}^{(i)}$ for each $\boldsymbol{x}^{(i)} \in \mathcal{X}$. The associated surrogate models built from the Infer-and-Fit algorithm are respectively denoted by *GLD MM* and *GLD MLE*. Similarly, the joint PCE-GLD algorithm provides another two models denoted by *GLD joint_MM* and *GLD joint_MLE*. Note that when building these two joint models following Algorithm 4.2, both of them rely on solving the optimization problem in Eq. (4.33). However, results are not identical because the sparse truncation sets $\{\mathcal{A}_s, s = 1, \dots, 4\}$ for $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a})$ as well as the starting points for the optimization generally differ.

The error measure between the emulated PDF and the true one is computed using the Hellinger distance, which is then averaged over all possible $\boldsymbol{x}$. More precisely, we define

$$\epsilon = \mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{HD}}\left(f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{X}), f_{Y|\boldsymbol{X}}\left(y \mid \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{X}; \hat{\boldsymbol{a}}\right)\right)\right)\right]. \tag{4.35}$$

It is reminded that the Hellinger distance between two continuous PDFs p and q reads

$$\begin{aligned} d_{\mathrm{HD}}\left(p(y), q(y)\right) &= \frac{1}{\sqrt{2}}\left\|\sqrt{p(y)} - \sqrt{q(y)}\right\|_2 \\ &= \sqrt{\frac{1}{2}\int_{-\infty}^{+\infty}\left(\sqrt{p(y)} - \sqrt{q(y)}\right)^2 \mathrm{d}y} = \sqrt{1 - \int_{-\infty}^{+\infty}\sqrt{p(y)q(y)}\mathrm{d}y}. \end{aligned} \tag{4.36}$$

Another natural choice for measuring the distance between two PDFs would have been the KL divergence. However, $D_{KL}(p\|q)$ tends to $+\infty$ if $\mathrm{supp}(p)\backslash\mathrm{supp}(q)$ has non zero probability with respect to p, which is not suitable for the comparison.

In practice, the integral in Eq. (4.36) is computed using numerical integration. In this paper, we restrict the integral interval from $(-\infty, +\infty)$ to $[Q_p^{0.001}, Q_p^{0.999}] \bigcup [Q_q^{0.001}, Q_q^{0.999}]$, where $Q_p^{0.001}$ and $Q_p^{0.999}$ denote the 0.1% and 99.9% quantiles of a random variable having p as PDF (similar notations are used for q). Note that this is feasible here because the two densities in Eq. (4.35) we want to compare have analytical expressions for the specific examples handled.

To calculate the expectation in Eq. (4.35), quasi-Monte Carlo simulation is used with $N_{\mathrm{test}} = 1{,}000$ samples generated by the Sobol' sequence (Sobol', 1967) in the input space. The Sobol' sequence sampler is also used

to draw the experimental design (ED). To study the performance of the proposed methods, data are generated for various combinations of the experimental design size N and the amount of replications R per ED point. Each scenario is run 100 times with independent experimental designs to account for statistical uncertainty. Error estimates for each scenario (N, R) are thus represented by box plots.

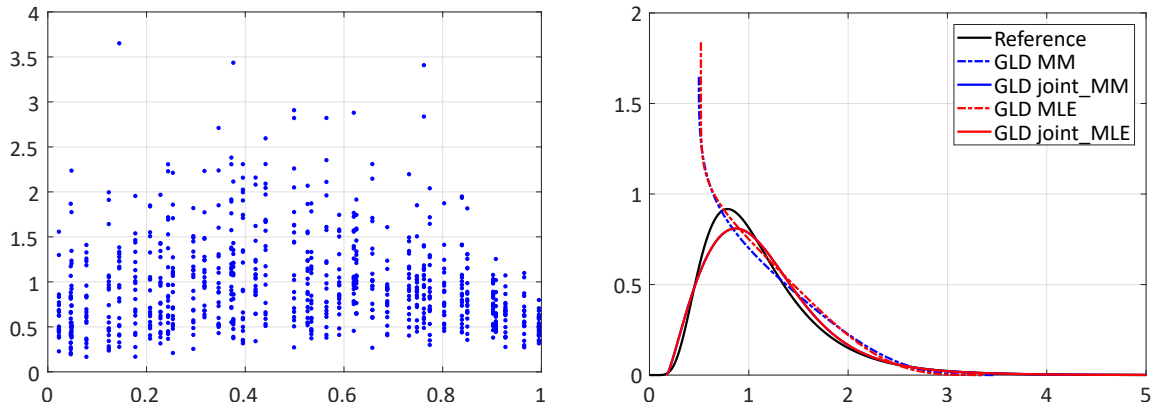
4.5.1 Example 1: a one-dimensional simulator

The first example is defined as follows:

$$Y(X,\omega) = \sin\left(\frac{2\pi}{3}X + \frac{\pi}{6}\right) \cdot \left(Z_1(\omega) \cdot Z_2(\omega)\right)^{\cos(X)}, \tag{4.37}$$

where $X \sim \mathcal{U}(0,1)$ is the input parameter, and $Z_1(\omega) \sim \mathcal{LN}(0, 0.25)$ and $Z_2(\omega) \sim \mathcal{LN}(0, 0.5)$ are latent variables following lognormal distributions. Under this definition, $Y(x,\omega)$ follows a lognormal distribution $\mathcal{LN}(\ell(x), \zeta(x))$ with $\ell(x) = \log\left(\sin\left(\frac{2\pi}{3}x + \frac{\pi}{6}\right)\right)$ and $\zeta(x) = \sqrt{\frac{3}{8}\cos(x)}$, $x \in [0,1]$. As mentioned in Section 4.2, the lognormal distribution, which is widely used in engineering, can be approximated by the generalized lambda distribution. The nonlinearity in its parameters leads to nonlinear functions of $\boldsymbol{\lambda}(\boldsymbol{x})$ in the GLD approximation, and thus the PC representations $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x})$ are also nonlinear.

Figure 4.4 shows one realization of an experimental design of $N = 40$ and $R = 20$ for each point and the predicted PDF of the four surrogate models for $X = 0.5$. We observe that the two models *GLD MM* and *GLD MLE* built using the Infer-and-Fit algorithm cannot capture the shape of the true distribution. In contrast, the two joint models *GLD joint_MM* and *GLD joint_MLE* produce a PDF that not only has the correct shape but also is an accurate approximation of the underlying distribution. We remark that with the data illustrated in Figure 4.4, *GLD joint_MM* and *GLD joint_MLE* are identical, implying that even though *GLD MM* and *GLD MLE* are different, their associated joint models can still be identical if they select the same sparse truncation sets $\{\mathcal{A}_s, s = 1, \ldots, 4\}$. Therefore, the selected algorithm is appears to be not too sensitive to the starting point.



(a) Generated data

(b) PDF prediction for $X = 0.5$

Figure 4.4: Example 1 — 40 ED points and 20 replications

Figures 4.5 to 4.6 show quantitative comparisons of the convergence behavior of the four models. It turns out that in general all the GLD models converge when increasing the size of experimental design N and the

number of replications R. Moreover, the two joint models outperform the models built of the Infer-and-Fit approach, especially when only a few replications are available.

In the case with only 20 replications (Figure 4.5), the convergence behavior of *GLD MM* and *GLD MLE* shows a weak dependence on N. This is because in the first step of Algorithm 4.1, estimators $\hat{\boldsymbol{\lambda}}^{(i)}$ from both the method of moments and the maximum likelihood estimation might be biased. Then regression used in the second step is not able to filter the bias. Moreover, a few replications lead to high variance of the estimators, which together with the bias explains the non convergent behavior of *GLD MM* and *GLD MLE*. When increasing the number of replications, the bias becomes less significant and the variance of the error decreases. Therefore, ϵ decreases with increasing N for *GLD MM* and *GLD MLE* in Figure 4.6.

In contrast, *GLD joint_MLE* exhibits a fast error decay even with a small number of replications. This is because all the available data are used at once to estimate the model parameters, which reduces both the bias and the variance. *GLD joint_MM* appears to provide less accurate PDF estimation than *GLD joint_MLE*, which is due to the less appropriate truncation scheme selected by *GLD MM*. Nevertheless, *GLD joint_MM* still improves the results of *GLD MM* and outperforms *GLD MLE*.

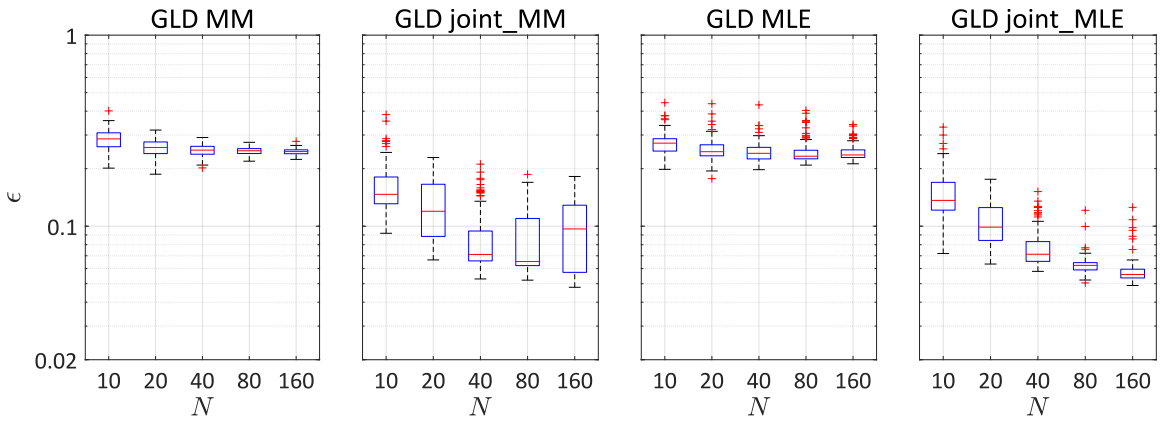


Figure 4.5: Example 1 — Hellinger distance between the surrogate model built with $R = 20$ and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

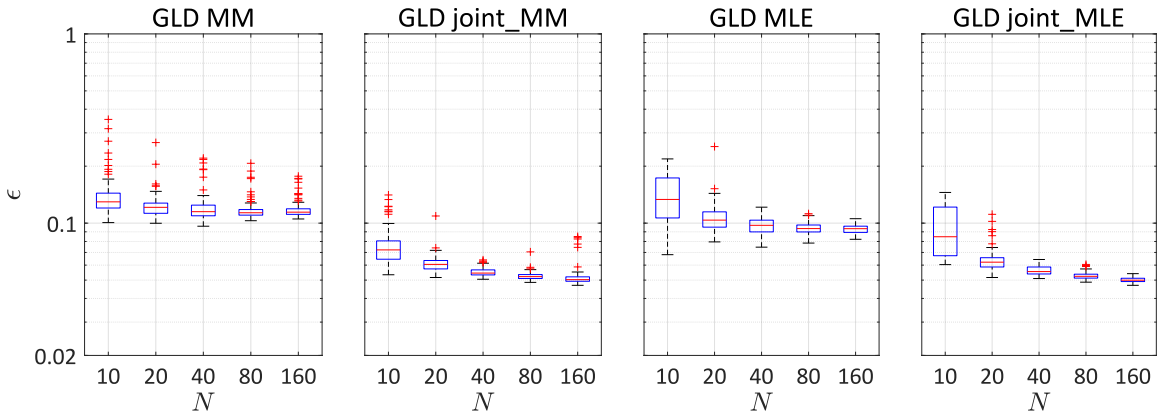


Figure 4.6: Example 1 — Hellinger distance between the surrogate model built with $R = 80$ and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

We have run the simulation for $N = \{10, 20, 40, 80, 160\}$ and $R = \{10, 20, 40, 80, 160\}$. Figure 4.7 sum-

marizes the total number of model runs NR (only up to 1,600) against the error measure. The results are consistent with what we have observed in the case of fixed number of replications. More precisely, the two models built with the joint modeling algorithm outperform those based on the Infer-and-Fit algorithm: with 400 models runs, *GLD joint_MM* and *GLD joint_MLE* provide more accurate PDF estimations than *GLD MM* and *GLD MLE* with 1,600 model evaluations.

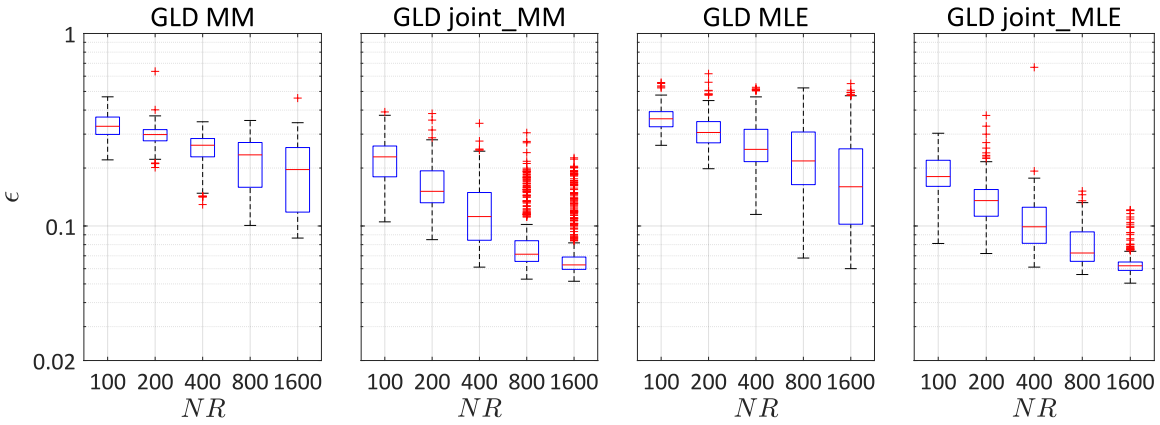


Figure 4.7: Example 1 — Hellinger distance between surrogate models built with different total number of model runs and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

4.5.2 Example 2: a five-dimensional simulator

The second analytical example is defined as follows:

$$Y(\boldsymbol{X}, \omega) = \mu(\boldsymbol{X}) + \sigma(\boldsymbol{X}) \cdot Z(\omega), \tag{4.38}$$

where $Z(\omega) \sim \mathcal{N}(0,1)$ is the latent variable that represents the source of randomness, and $\boldsymbol{X}$ is a five-dimensional random vector, with independent components having uniform distribution $\mathcal{U}(0,1)$. $Y(\boldsymbol{x})$ is a Gaussian random variable with mean $\mu(\boldsymbol{x})$ and standard deviation $\sigma(\boldsymbol{x})$. In this example, the mean function $\mu(\boldsymbol{x})$ reads

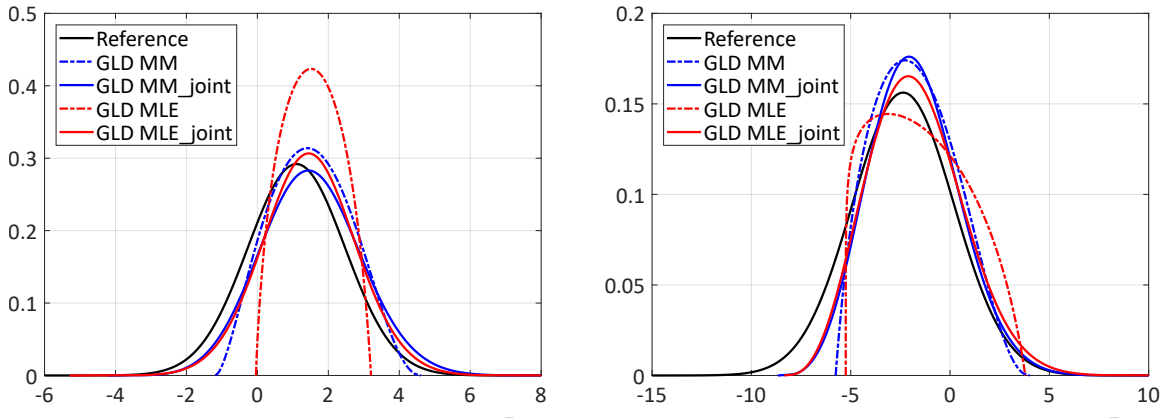
$$\mu(\boldsymbol{x}) = 3 - \sum_{j=1}^{5} j x_j + \frac{1}{5}\sum_{j=1}^{5} j x_j^3 + \frac{1}{15}\log\left(1 + \sum_{j=1}^{5} j(x_j^2 + x_j^4)\right) + x_1 x_2^2 - x_5 x_3 + x_2 x_4, \tag{4.39}$$

and the standard deviation $\sigma(\boldsymbol{x})$ is given by

$$\sigma(\boldsymbol{x}) = \exp\left(\frac{1}{4}\sum_{j=1}^{5} x_j\right), \tag{4.40}$$

which implies a strong heteroskedastic effect with a highly nonlinear mean function. This example is used to show the performance of the proposed methods in moderate dimensional problems.

Similar to the previous example, the GLD models demonstrate a convergent behavior, as illustrated in Figures 4.9 to 4.11. The two joint models yield more accurate estimates than those built with the Infer-and-Fit algorithm. In the case of a few replications, both *GLD MM* and *GLD MLE* fail to capture the shape of the PDF (Figure 4.8), and thus converge rather slowly with respect to N (see Figure 4.9). In contrast, the two



(a) PDF at $\boldsymbol{x} = (0.25, 0.25, 0.25, 0.25, 0.25)^{\intercal}$ (b) PDF at $\boldsymbol{x} = (0.75, 0.75, 0.75, 0.75, 0.75)^{\intercal}$

Figure 4.8: Example 2 — PDF predictions with an experimental design of size 50 and 25 replications.

joint models are less sensitive to the number of replications, and their performance mainly depends on the ED size. Unlike the first example, using the method of moment turns out to produce slightly more accurate estimates when applying the Infer-and-Fit algorithm. However, the parametric estimation methods employed to get $\hat{\boldsymbol{\lambda}}^{(i)}$ do not have a significant influence on the accuracy of the joint algorithm: *GLD joint_MM* and *GLD joint_MLE* show very similar convergence.

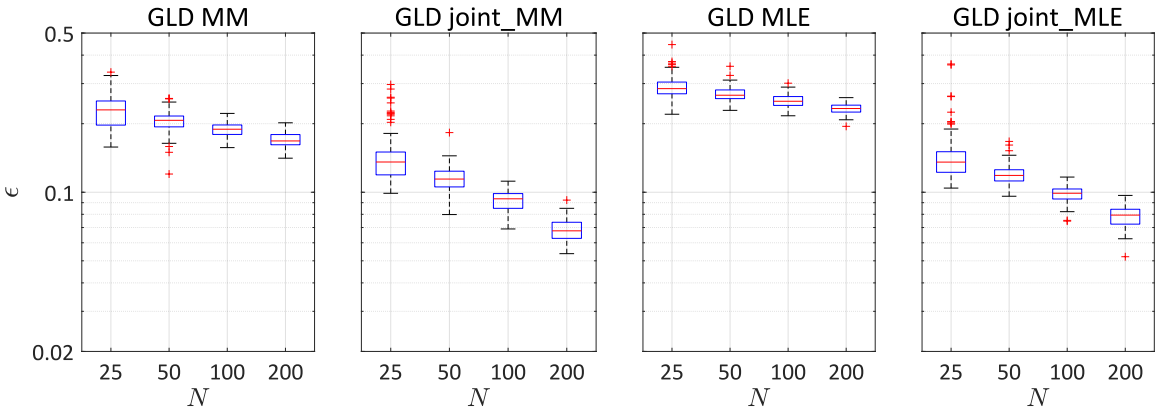


Figure 4.9: Example 2 — Hellinger distance between the surrogate model built with $R = 25$ and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale)

In this section, only the error measure based on the Hellinger distance is reported for convergence studies. Nevertheless, quantitative comparisons using other metrics such as the Kolmogorov–Smirnov distance, the mean value and the 95% quantile of the predicted distributions show similar trends.

4.6 Applications

4.6.1 Stochastic differential equation

Stochastic differential equations (SDEs) are widely used to model the evolution of complex systems in many fields, e.g., finance (McNeil et al., 2005), epidemics (Gray et al., 2011), and meteorology (Iversen et al., 2015).

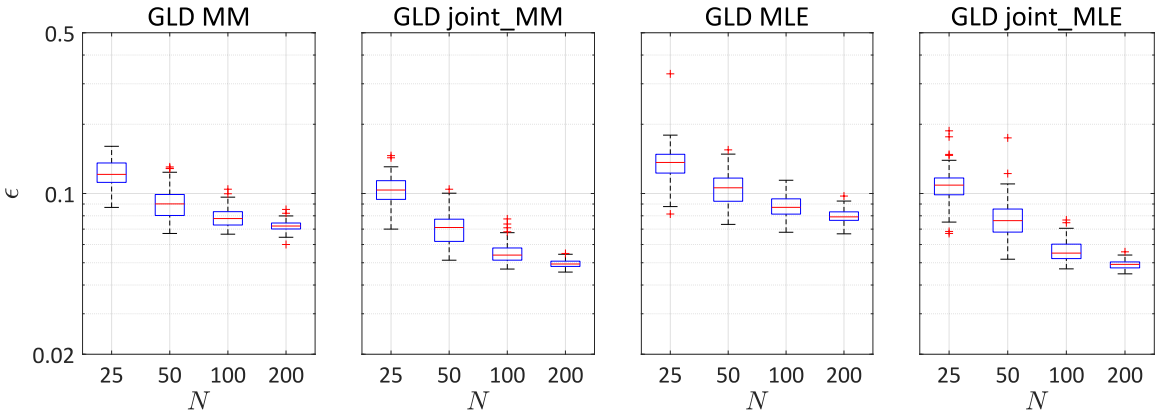


Figure 4.10: Example 2 — Hellinger distance between the surrogate model built with $R = 100$ and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale)

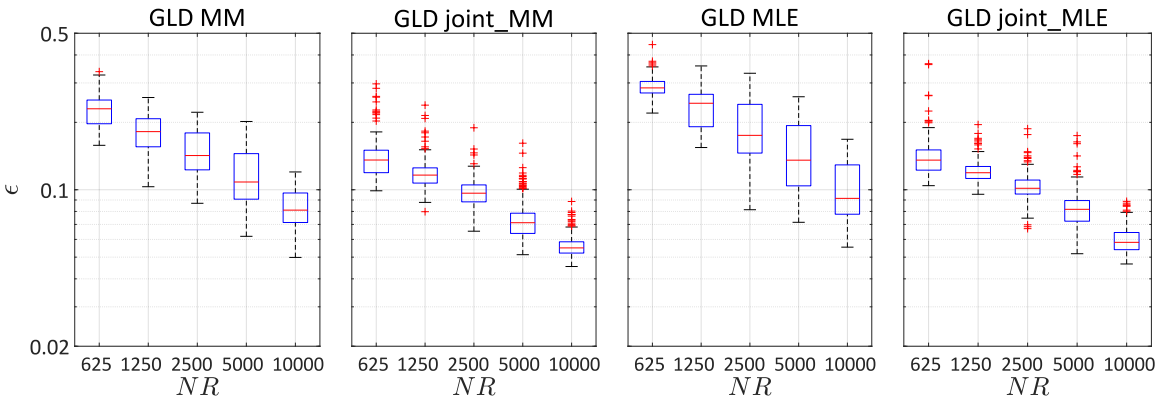


Figure 4.11: Example 2 — Hellinger distance between surrogate models built with different total number of model runs and the true response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

Due to the stochastic process (e.g., Wiener processes) involved in a SDE, the associated solution is also a stochastic process. As a results, when fixing the parameters of a SDE, any scalar-valued deterministic function of the solution process produces a random variable, which can be regarded as a stochastic simulator. In this case study, we consider the example proposed by Jimenez et al. (2017), the governing equation of which reads

$$\mathrm{d}Y_t = (X_1 - Y_t)\mathrm{d}t + (\nu Y_t + 1)X_2 \,\mathrm{d}W_t, \tag{4.41}$$

with the initial condition defined by

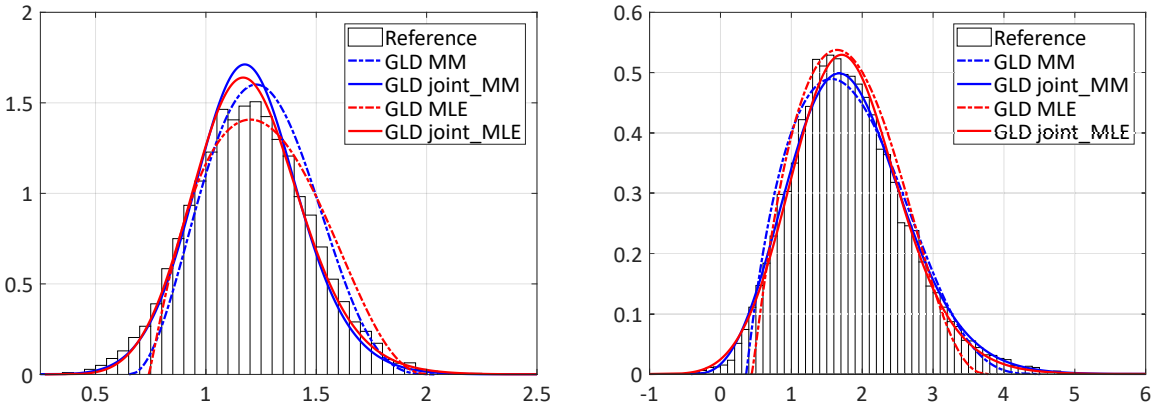
$$Y_0 = 0 \;\text{ almost surely.}$$

In this equation, $\boldsymbol{X} = (X_1, X_2)^{\intercal}$ are the SDE parameters, and W_t is a standard Wiener process that represents the source of randomness. We denote $Y_t(\boldsymbol{x})$ the solution of Eq. (4.41) for $\boldsymbol{X} = \boldsymbol{x}$, and we focus on the value of $Y_t(\boldsymbol{x})$ at $t = 10$, i.e., $Y_{10}(\boldsymbol{x})$ is the scalar QoI.

Note that the value of ν controls how the Wiener process affects the QoI: for $\nu = 0$, $\mathrm{d}W_t$ is multiplied with a constant, and thus the solution $Y_t(\boldsymbol{x})$ is a Gaussian process; whereas for $\nu \neq 0$, W_t interacts with the unknown process $Y_t(\boldsymbol{x})$, and the marginal distribution of $Y_t(\boldsymbol{x})$ does not have an analytical closed-form. We

set $\nu = 0.2$ in this study. To numerically solve Eq. (4.41), we apply the classical Euler–Maruyana method (Kloeden and Platen, 1992) with time step $\Delta t = 0.01$. Therefore, the discretized version of Eq. (4.41) has a large number of latent random variables $\boldsymbol{Z}$ equal to $\frac{10}{\Delta t} = 1{,}000$. This problem is representative of cases with low dimensionality in $\boldsymbol{X}$ and very large size of $\boldsymbol{Z}$.

The original definition of $\boldsymbol{X}$ proposed by Jimenez et al. (2017) follows $X_1 \sim \mathcal{U}(0.95, 1.15)$ and $X_2 \sim \mathcal{U}(0.02, 0.22)$. According to some preliminary tests, we found that under this setting, the response PDF is close to a normal distribution and does not vary significantly with respect to $\boldsymbol{x}$ because the range of definition of the input parameters is rather narrow. In order to have richer shapes for the output PDF of $Y_{10}(\boldsymbol{x})$ and challenge our algorithm, we choose $X_1 \sim \mathcal{U}(0.9, 2)$, $X_2 \sim \mathcal{U}(0.1, 1)$ in this paper. Thus, the response PDF can have normal-like shape and can also be right-skewed depending on $\boldsymbol{x}$.



(a) PDF at $X = (1.2, 0.3)^{\mathsf{T}}$ (b) PDF at $X = (1.8, 0.8)^{\mathsf{T}}$

Figure 4.12: Stochastic differential equation — PDF predictions with an experimental design of size 80 using 40 replications. The reference histogram is calculated based on 10,000 replications.

Figure 4.12 shows the results when applying the developed methods to an experimental design of $N = 80$ and $R = 40$. We observe that all the four models can generally well approximate the underlying distributions. Detailed comparison shows that the Infer-and-Fit algorithm is not able to correctly emulate the shape variation of the response PDF: when the underlying PDF is close to a normal distribution, both *GLD MM* and *GLD MLE* predict a slightly right-skewed PDF; whereas for positively skewed PDF, neither of them is able to accurately approximate the tail. In contrast, *GLD joint_MM* and *GLD joint_MLE* not only capture the shape variation but also better represent the underlying PDF.

Similar to the analytical examples in Section 4.5, we investigate the convergence behavior of the developed methods. Since the analytical PDF is not available, we use kernel density estimation (Wand and Jones, 1995) using 10,000 replications as the reference distribution for each point in the test set. The Hellinger distance between the predicted PDF and the reference is averaged over a test set $\mathcal{X}_{\text{test}}$ containing 100 points generated with a Sobol' sequence.

The convergence study of the four models is reported in Figures 4.13 to 4.14. As expected from the analytical examples, the joint modeling algorithm appears much more efficient. In particular, both *GLD joint_MM* and *GLD joint_MLE* yield an error around 0.07 in the case of 20 replications and 80 ED points, i.e., 1,600 model evaluations, whereas *GLD MM* and *GLD MLE* can barely achieve this accuracy even when 80 replications and 160 ED points, i.e., a total of 12,800 model evaluations, are available. More generally, the joint algorithm pro-

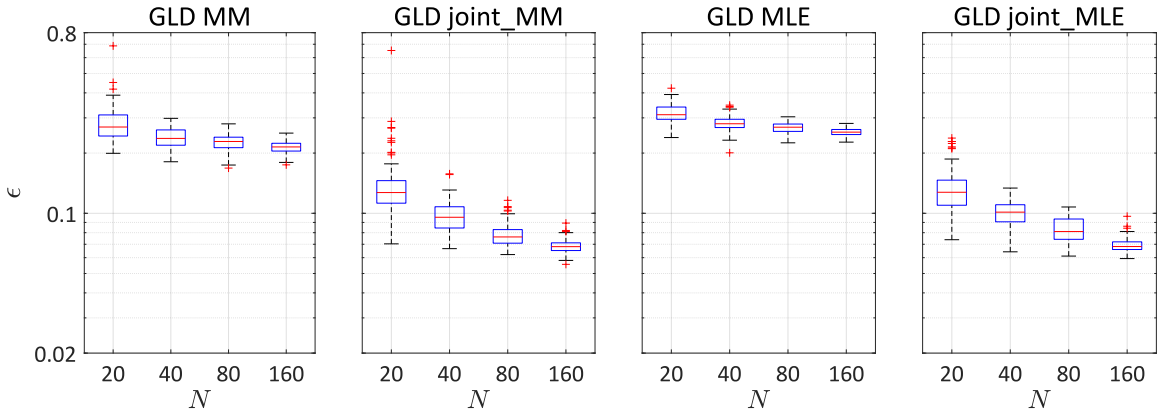


Figure 4.13: Stochastic differential equation — Hellinger distance between the surrogate model built with $R = 20$ and the reference response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

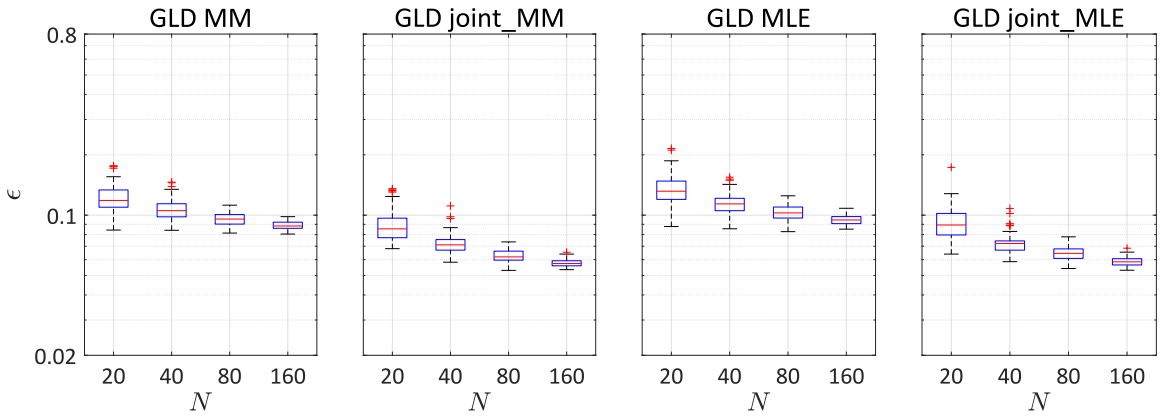


Figure 4.14: Stochastic differential equation — Hellinger distance between the surrogate model built with $R = 80$ and the reference response PDF, averaged over $\mathcal{X}_{\text{test}}$ (log-scale).

duce more accurate results than the Infer-and-Fit algorithm when a large number of replications are available.

4.6.2 Wind turbine design

In the wind turbine design process, structural components need to be analyzed under diverse environmental loads to assess their performance and reliability. Typical simulations consist of two parts, namely the generation of the external excitations (i.e., wind inflow) and the aero-servo-elastic simulation as illustrated in Figure 4.15. The latter refers to the complex multi-physics scenario including mutual interactions of wind inflow, aerodynamics, structural dynamics (elastic deflections) and control systems.

The wind field generator used in this study is TurbSim (Jonkman, 2009), which is a stochastic inflow turbulence simulator. It takes five macroscopic parameters as input: (1) the mean wind velocity U at reference altitude z_{ref}; (2) the turbulence intensity I, denoting the coefficient of variation of the wind time series, i.e., $I = \sigma/U$; (3) the wind shear exponent α, describing the variation of the mean wind velocity with the altitude according to the following equation:

$$U(z) = U \cdot \left(\frac{z}{z_{\text{ref}}}\right)^{\alpha}; \tag{4.42}$$

(4) the air density ρ and (5) the inclination angle β (see Abdallah et al., 2019 for details). Since these five parameters cannot fully determine a wind field, random seeds are used on top of these macroscopic parameters in TurbSim to generate a coherent turbulent three-dimensional velocity time series (Jonkman, 2009).

The wind turbine structure studied here is the reference 5 MW upwind turbine described in Jonkman et al. (2009). The aero-servo-elastic simulator is FAST (Jonkman et al., 2009), a deterministic computational model that takes inflow wind fields as input and calculates the structural response as output. However, due to the use of random seeds in the turbulent wind generation, simulations of wind turbines are stochastic with respect to the five input macroscopic parameters. In other words, fixing the five quantities described above, any number of three-dimensional wind fields can be simulated, each of which leads to a different response and predicted performance of the wind turbine. Note that this is also what happens in reality for wind turbines.

Of interest is the maximum flap-wise bending moment at the blade root M_b within the simulated time (10 minutes) for a given wind climate defined by the 5 macroscopic parameters, as illustrated in Figure 4.15. To build a stochastic surrogate, the Latin hypercube sampling (LHS) method (McKay et al., 1979) with rejection is used to create an experimental design of 485 points in dimension 5. More precisely, input samples are firstly generated by the LHS following Table 4.1, and then the samples that are outside the bounds that are calibrated from real wind climate are removed. The bounds are respectively defined in the (I, U) plane, (α, U) plane, and (α, I) plane, as illustrated in Figure 4.16 (Slot et al., 2020). Importantly, when the turbulence standard deviation $\sigma = I \cdot U$ is close to zero, the wind speed barely varies in time, and thus the response PDF is close to being degenerate, which can cause numerical problems. In this case, the simulator can be considered as deterministic and does not fit to the GLD framework. Hence, we introduce an additional bound in σ, and only samples with $\sigma > 0.05$ are simulated. Finally, the simulator is run 50 times for each set of input parameters as replications.

Considering the physical process, we chose to use the turbulence standard deviation σ rather than the turbulence intensity I to build the surrogate models. Hence, the input parameters are pre-processed as $\boldsymbol{X} = (U, \sigma, \alpha, \beta, \rho)^{\intercal}$ for training.

Because of the sampling scheme, the input parameters U, I and α are not independent, which violates the independent assumption when building PC basis. One possibility to tackle this problem would be to use the Rosenblatt transform (Rosenblatt, 1952) to map the dependent inputs into a set of independent random variables and then build PC basis of the latter. However, Torre et al. (2019) shows that this approach, while yielding improved estimates of the output statistics, is typically detrimental to the accuracy of pointwise pre-

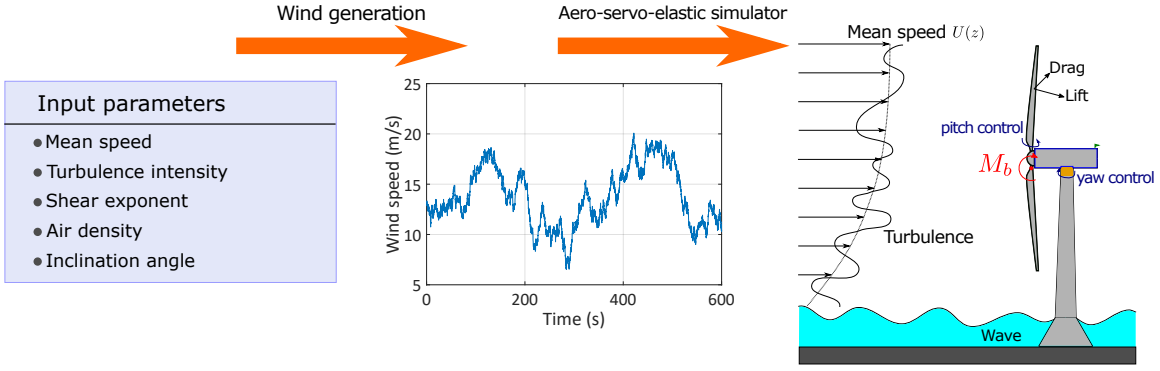
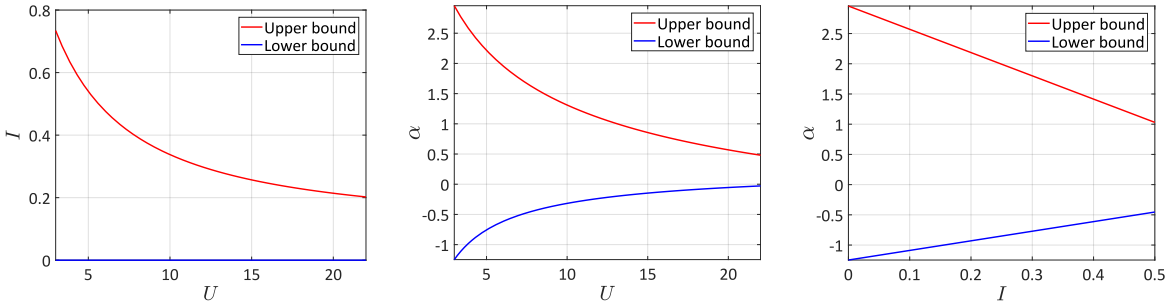


Figure 4.15: Wind turbine simulation scheme

Table 4.1: Wind turbine case study — description of the input variables

Name	Description	Distribution	Parameters
U	Mean speed (m)	Uniform	$[3, 22]$
I	Turbulence intensity	Uniform	$[0, 0.5]$
α	Shear exponent	Uniform	$[-2, 5]$
ρ	Air density (kg/m^3)	Uniform	$[0.8, 1.4]$
β	Declination angle (deg)	Uniform	$[-10, 10]$



(a) Bounds in (I, U) plane (b) Bounds in (α, U) plane (c) Bounds in (α, I) plane

Figure 4.16: Bounds on the physical parameters (U, I, α) calibrated from real wind data

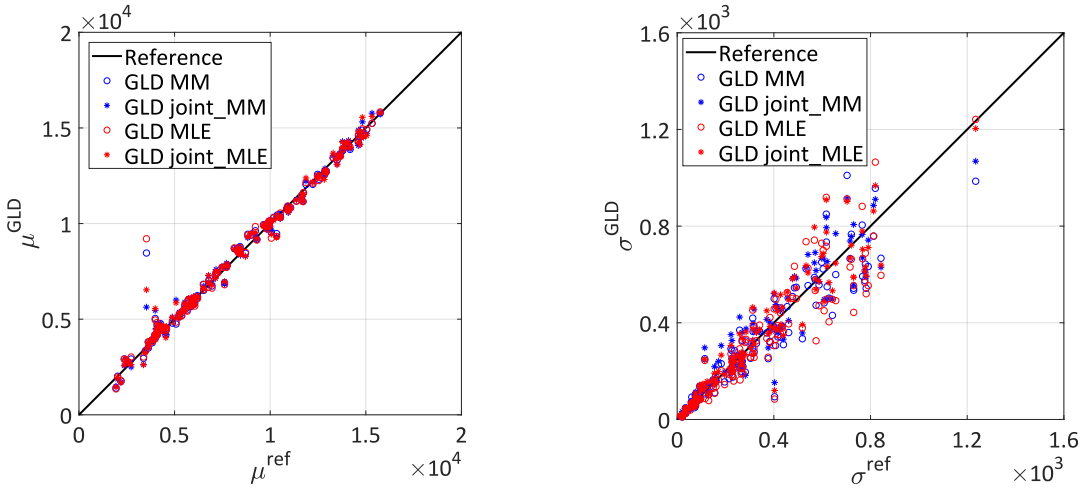
dictions. This is because the Rosenblatt transform is highly nonlinear, resulting in a transformed model whose PCE spectrum decays typically more slowly than the original one. Therefore, we ignore the dependence when building PC basis functions, and only the marginal distribution of each input variables is needed. Since the marginal distributions are difficult to be derived analytically due to the bounds, we apply the kernel density estimators to 10,000 samples generated according to the rejection sampling scheme described before. Note that the air density ρ and the inclination angle β are uniform variables, and they are independent from U, I and α. Therefore, we use Legendre polynomials as the associated univariate PC basis functions for these two variables.

Unlike the previous example in Section 4.6.1, the wind turbine simulation is costly, and thus we cannot run as many times the simulator as needed to have a reliable estimate of the error defined in Eq. (4.35). To assess the performance of the proposed methods, 120 samples of input parameters are generated by the same scheme as the training set. The simulator is repeatedly run 500 times for each test point. We then compare some sample statistics with those predicted by the stochastic emulators built by the developed methods. The former are considered as references. The metrics used for comparison are the mean, the standard deviation (std) and the 5%, 10%, 50%, 90% and 95% quantiles.

The results of the four GLD models are shown in Figure 4.17 and Figure 4.18. Comparisons of the scalar quantities show that all the four GLD models demonstrate a good fit to the simulated scenario. Among the scalar quantities, the mean and the quantiles are estimated with high accuracy, whereas the standard deviation estimation is relatively poor.

To have more quantitative comparison among the four GLD models, we define the normalized mean squared error:

$$\epsilon = \frac{\sum_{i=1}^{N_{\text{test}}} \left(q_{GLD}^{(i)} - \hat{q}^{(i)} \right)^2}{\sum_{i=1}^{N_{\text{test}}} \left(\hat{q}^{(i)} - \bar{\hat{q}} \right)^2}, \text{ with } \bar{\hat{q}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \hat{q}^{(i)}, \tag{4.43}$$



(a) Mean estimation

(b) Std estimation

Figure 4.17: Wind turbine case study — Comparison of the mean and the standard deviation estimation of the maximum flapwise bending moment (kN·m). The x-axis (reference) is the empirical quantity calculated from the 500 replications.

where q is the statistical quantity of interest (mentioned above), $q_{GLD}^{(i)}$ is the value predicted by the GLD model, and $\hat{q}^{(i)}$ denotes the estimated quantity (empirical mean, standard deviation and quantiles) based on the replications for $\boldsymbol{x}^{(i)}$.

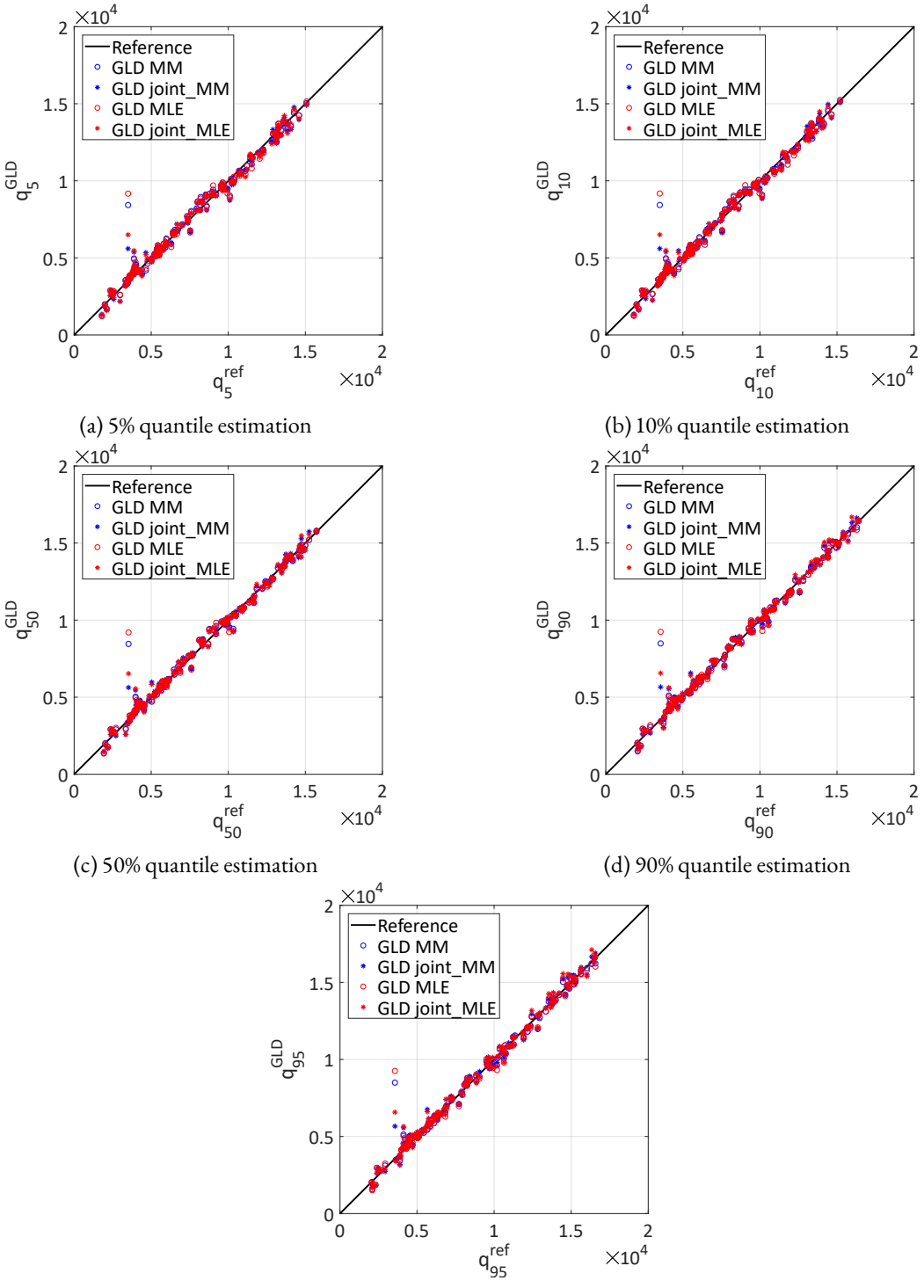
The errors associated with the scalar quantities are reported in Table 4.2. We can observe that the method of moments outperforms the maximum likelihood estimation for both the Infer-and-Fit and the joint model in terms of all the error measures used here. The joint models generally improve their associated Infer-and-Fit models, and *GLD joint_MM* provides the best estimates.

Table 4.2: Normalized mean squared error of various quantities in the test set. The best results among the four GLD models are highlighted in bold.

GLD models	mean	std	Q_{05}	Q_{10}	Q_{50}	Q_{90}	Q_{95}
GLD MM	0.0185	**0.1125**	0.0231	0.0221	0.0188	0.0166	0.0165
GLD joint_MM	**0.0099**	0.124	**0.0133**	**0.0154**	**0.0103**	**0.009**	**0.0091**
GLD MLE	0.0235	0.1488	0.0292	0.0280	0.0237	0.0214	0.0213
GLD joint_MLE	0.0128	0.1642	0.0167	0.0155	0.0131	0.0121	0.0125

Apart from the detailed quantitative comparison of the scalar quantities, we visualize also the PDF prediction in Figure 4.19 for two specific values of $\boldsymbol{x}$. The reference histograms are obtained based on the 500 replications. Since only 500 samples are available, the histograms are less smooth than those of the previous example in Section 4.6.1. We observe that all the four surrogate models can well capture the location of the underlying distribution. In addition, the two joint models demonstrate better performance on the shape approximation. For example, in Figure 4.19, the two Infer-and-Fit models produce narrower support than the range of the samples, whereas the support is accurately approximated by the two joint models.

As a conclusion, the GLD joint models allows for accurate prediction of the PDF of the maximum flapwise bending moment at the blade root at a total cost of about 24,000 runs of Turbsim+FAST. The calculation have been carried out on the ETH Euler cluster using 96 cores for a physical time of about 20.5 hours. Interestingly,



(a) 5% quantile estimation

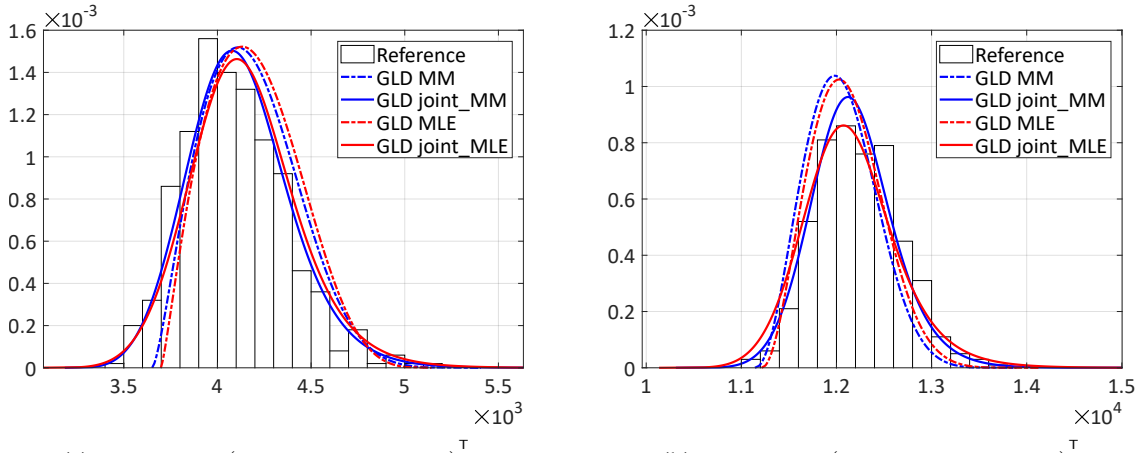
(b) 10% quantile estimation

(c) 50% quantile estimation

(d) 90% quantile estimation

(e) 95% quantile estimation

Figure 4.18: Wind turbine case study — Comparison of the quantiles estimation of the maximum flapwise bending moment (kN·m). The x-axis (reference) is the empirical quantity calculated from the 500 replications.



(a) PDF at $\boldsymbol{x} = (4.9, 1.2, 0.6, 0.9, 7.2)^\intercal$

(b) PDF at $\boldsymbol{x} = (11.8, 2.9, 0.3, 1.0, 0.2)^\intercal$

Figure 4.19: Wind turbine case study — PDF predictions with the experimental design of size 485 using 50 replications. The reference histogram is calculated based on 500 replications.

the 95% quantile, which is of interest for design assessment, is remarkably predicted all over the space of input parameters.

4.7 Conclusions

The aim of this paper is to build efficient and accurate surrogate models for stochastic simulators within the replication-based framework. Generalized lambda distributions are used to flexibly approximate the output PDF, while the relationship of their parameters with the inputs is approximated through polynomial chaos expansions. To construct surrogate models in a non-intrusive manner, we first proposed the Infer-and-Fit algorithm which consists of solving two consecutive problems. In the first step, the distribution parameters are inferred based on repeated model evaluations for each point of the experimental design. Then the estimated values are used to build a PCE surrogate model for each distribution parameters. The Infer-and-Fit algorithm allows us to use conventional regression techniques to construct PCE. However, this method is sensitive to the number of replications due to the two-step strategy, whereby the model responses are only used in the first step. In order to build accurate surrogate models even when a few replications are available, we proposed in a second part the joint modeling method described in Algorithm 4.2. This approach carries out one more optimization step after getting a first estimate from the Infer-and-Fit approach, which is used to provide sparse truncation sets for $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{a})$ and a starting point for the optimization. This enrichment allows us to use all the available data at once and provides a maximum likelihood estimator of the model parameters, namely the coefficients of the polynomial chaos expansions of the $\boldsymbol{\lambda}$'s. Due to the complexity of the likelihood function, this additional optimization problem can be expensive to solve. To alleviate the computational burden, we vectorized the implementation in the Matlab environment and derived analytically the gradient and the Hessian matrix of the objective function. As a result, we can efficiently apply derivative-based optimizers.

For the analytical examples in Section 4.5 and the stochastic differential equation case study in Section 4.6.1, the proposed two algorithms are both able to approximate the reference distributions with high accuracy, even though the data generation scheme does not follow the generalized lambda distribution. As expected, the joint

models show better performance when only a few replications are available. For the wind turbine application in Section 4.6.2, due to the cost of the simulator, only some important statistical scalar quantities are compared to a reference solution obtained from a large Monte Carlo simulation, whereas PDFs at selected input points are only compared visually. Both developed methods demonstrate high accuracy for the mean and quantiles estimation.

In all the examples and applications, joint models are observed to consistently improve the result of the associated models built with the Infer-and-Fit algorithm. Besides, for the parametric estimation in the first step of the Infer-and-Fit algorithm, the method of moments and maximum likelihood show comparable performance. This observation matches the conclusion in Corlu and Meterelliyoz (2016).

In the joint modeling method, the main role played by the replications is to obtain a truncation scheme for each component of $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{a})$ as well as to find an initial starting point for the following optimization step. Therefore, replications are not necessary if the basis functions of each distribution parameter are known or preselected. Work is in progress to improve the proposed method by combining feasible generalized least squares and sparse regressions for finding appropriate starting points and basis selections, which completely avoids the need for replications and thus drastically reduces the computational cost (Zhu and Sudret, 2021).

Acknowledgments

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524). The authors gratefully thank Dr. Imad Abdallah (ETH Zurich) and René Slot (Aalborg University) for extensive fruitful discussions on applications, and Nora Lüthen (ETH Zurich) for the computation of the wind turbine data.

4.a Appendix

4.a.1 Feasible starting point

For the additional optimization problem introduced in the joint algorithm, the coefficients $\tilde{\boldsymbol{a}}$ fitted from the Infer-and-Fit algorithm are chosen as an appropriate starting point for the optimization. However, as discussed in Section 4.4, the objective function $l(\boldsymbol{a})$ can take the value $+\infty$, and thus complex constraints are present, which is summarized in Eq. (4.34). Therefore, additional operations are necessary to have a feasible starting point if $\tilde{\boldsymbol{a}}$ does not satisfy the constraints.

It is observed from Eq. (4.3) that the lower (upper) bound of the support is a monotonic function of λ_3 (λ_4) for fixed λ_1 and λ_2. Therefore, reducing the coefficients $a_{3,\mathbf{0}}$ and $a_{4,\mathbf{0}}$ that are associated with the constant functions in Eq. (4.29) broadens the support of the response PDF for all $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. Based on this property, the following procedure is proposed to adjust $\tilde{\boldsymbol{a}}$ to be feasible:

1. Evaluate $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)};\tilde{\boldsymbol{a}}\right)$ for all $\boldsymbol{x}^{(i)} \in \mathcal{X}$
2. Collect the index i into the set I, whose associated $\lambda_3^{(i)}$ is positive

3. For all $i \in I$, calculate $\check{\lambda}_3^{(i)}$ such that the minimum value of the associated replication results located exactly on the lower bound. More precisely, according to Eq. (4.3), we have

$$\check{\lambda}_3^{(i)} = \frac{1}{\lambda_2^{(i)} \left(\lambda_1^{(i)} - \min_r y^{(i,r)} \right)} \tag{4.44}$$

4. Decrease the value of $\tilde{a}_{3,\mathbf{0}}$ so that $\lambda_3^{(i)} < \check{\lambda}_3^{(i)}$ for all $i \in I$.

This algorithm only deals with the constraints related to the lower bounds. The same method can be used for those from upper bounds, which consists in modifying the constant $\tilde{a}_{4,\mathbf{0}}$ based on $\tilde{\lambda}_4^{(i)}$.

4.A.2 Analytical derivations for Algorithm 4.2

In this section, we compute the analytical derivatives of the negative log-likelihood function l with respect to the model parameters $\boldsymbol{a}$. Since the objective function is a composition of several functions as shown in Figure 4.3, the derivatives can be calculated through the chain rule, which flows from Step 3 to Step 1. Starting from

$$l = \log \left(\frac{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}}{\lambda_2} \right), \tag{4.45}$$

we get the following partial derivatives:

$$\frac{\partial l}{\partial u} = \frac{(\lambda_3 - 1)u^{\lambda_3 - 2} - (\lambda_4 - 1)(1-u)^{\lambda_4 - 2}}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}}, \tag{4.46}$$

$$\frac{\partial l}{\partial \lambda_2} = -\frac{1}{\lambda_2}, \tag{4.47}$$

$$\frac{\partial l}{\partial \lambda_3} = \frac{u^{\lambda_3 - 1} \log(u)}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}}, \tag{4.48}$$

$$\frac{\partial l}{\partial \lambda_4} = \frac{(1-u)^{\lambda_4 - 1} \log(1-u)}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}}. \tag{4.49}$$

The differentiation at Step 2 is more complex because u is not an explicit function of $\boldsymbol{\lambda}$, and thus it involves derivatives of a highly nonlinear implicit function Eq. (4.1). Based on

$$y = Q(u) = \lambda_1 + \frac{1}{\lambda_2} \left(\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4} \right),$$

and because y is given, differentiating both side gives

$$0 = \mathrm{d} \left(\lambda_1 + \frac{1}{\lambda_2} \left(\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4} \right) \right),$$

where d stands for the total differentiation.

Expanding and rearranging the equations above, we have

$$\frac{\partial u}{\partial \lambda_1} = -\frac{\lambda_2}{u^{\lambda_3-1}+(1-u)^{\lambda_4-1}}, \tag{4.50}$$

$$\frac{\partial u}{\partial \lambda_2} = \frac{1}{\lambda_2\left(u^{\lambda_3-1}+(1-u)^{\lambda_4-1}\right)}\left(\frac{u^{\lambda_3}-1}{\lambda_3}-\frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right), \tag{4.51}$$

$$\frac{\partial u}{\partial \lambda_3} = \frac{u^{\lambda_3}-\lambda_3 u^{\lambda_3}\log(u)-1}{\lambda_3^2\left(u^{\lambda_3-1}+(1-u)^{\lambda_4-1}\right)}, \tag{4.52}$$

$$\frac{\partial u}{\partial \lambda_4} = \frac{-(1-u)^{\lambda_4}+\lambda_4(1-u)^{\lambda_4}\log(1-u)+1}{\lambda_4^2\left(u^{\lambda_3-1}+(1-u)^{\lambda_4-1}\right)}. \tag{4.53}$$

As illustrated in Figure 4.3, the derivatives of the negative log-likelihood function with respect to $\boldsymbol{\lambda}$ come from two parts: one is from the direct derivative (Eqs. (4.47) to (4.49)) in Step 3, the other part is contributed by the implicit differentiation (Eqs. (4.50) to (4.53)) in Step 2. As a result, we have

$$\frac{\mathrm{d}l}{\mathrm{d}\lambda_s} = \frac{\partial l}{\partial \lambda_s} + \frac{\partial l}{\partial u}\frac{\partial u}{\partial \lambda_s}, \quad s = 1,2,3,4. \tag{4.54}$$

Finally, the derivatives flow back to the model parameters $\boldsymbol{a}$ at Step 1 as follows:

$$\frac{\mathrm{d}l}{\mathrm{d}a_{s,\boldsymbol{\alpha}}} = \frac{\mathrm{d}l}{\mathrm{d}\lambda_s}\frac{\partial \lambda_s}{\partial a_{s,\boldsymbol{\alpha}}} = \frac{\mathrm{d}l}{\mathrm{d}\lambda_s}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad s = 1,3,4, \tag{4.55}$$

$$\frac{\mathrm{d}l}{\mathrm{d}a_{2,\boldsymbol{\alpha}}} = \frac{\mathrm{d}l}{\mathrm{d}\lambda_2}\frac{\partial \lambda_2}{\partial a_{2,\boldsymbol{\alpha}}} = \frac{\mathrm{d}l}{\mathrm{d}\lambda_2}\frac{1}{L'(\lambda_2)}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{4.56}$$

where L denotes the transform that is used to guarantee the positiveness of $\lambda_2(\boldsymbol{x})$. Recall that we chose to use $L(\lambda_2) = \log(\lambda_2)$ in this paper. Similar techniques can be used to derive the Hessian matrix of the negative log-likelihood function, which is necessary for the trust-region algorithm. Due to the lengthy derivation, we omit the result here. Eq. (4.54) calculates the derivatives of the log-likelihood function with respect to the distribution parameters $\boldsymbol{\lambda}$. Hence, it can be used in the maximum likelihood estimation of the distribution parameters of a random variable following a generalized lambda distribution.

References

Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators—application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67–77.

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58:371–382.

Arnold, D. V. and Hansen, N. (2012). A (1+1)-CMA-ES for constrained optimisation. In Soule, T. and Moore, J. H., editors, *Proceedings of the Genetic and Evolutionary Computation Conference 2012 (GECCO 2012) (Philadelphia, PA)*, pages 297–304.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15(1–3):81–92.

Bhattacharya, P. K. and Gangopadhyay, A. K. (1990). Kernel and nearest-neighbor estimation of a conditional quantile. *The Annals of Statistics*, 18(3):1400–1415.

Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Browne, T., Iooss, B., Le Gratiet, L., Lonchampt, J., and Rémy, E. (2016). Stochastic simulators based optimization by Gaussian process metamodels—application to maintenance investments planning issues. *Quality and Reliability Engineering International*, 32(6):2067–2080.

Burden, R. L., Faires, J. D., and Burden, A. M. (2015). *Numerical analysis*. Cengage Learning.

Chalabi, Y., Scott, D. J., and Würtz, D. (2011). The generalized lambda distribution as an alternative to model financial returns. Technical report, Eidgenössische Technische Hochschule and University of Auckland.

Corlu, C. G. and Meterelliyoz, M. (2016). Estimating the parameters of the generalized lambda distribution: Which method performs best? *Communications in Statistics - Simulation and Computation*, 45(7):2276–2296.

Dacidian, M. and Carroll, R. J. (1987). Variance function estimation. *Journal of the American Statistical Association*, 400:1079–1091.

Efromovich, S. (2010). Dimension reduction and adaptation in conditional density estimation. *Journal of the American Statistical Association*, 105:761–774.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32:407–499.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fan, J. and Yao, Q. W. (1998). Efficient estimation of conditional variance functions in stochastic regression. *Biometrika*, 85:645–660.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Communications in Statistics - Theory and Methods*, 17:3547–3567.

Ghanem, R. G. and Spanos, P. (2003). *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, Mineola, 2nd edition.

Gray, A., Greenhalgh, D., Hu, L., Mao, X., and Pan, J. (2011). A stochastic differential equation SIS epidemic model. *SIAM Journal on Applied Mathematics*, 71(3):876–902.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99:1015–1026.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94:1194–1204.

Iversen, E. B., Morales, J. M., Moller, J. K., Mao, X., and Madsen, H. (2015). Short-term probabilistic forecasting of wind speed using stochastic differential equations. *International Journal of Forecasting*, 32:981–990.

Jimenez, M. N., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 5:378–402.

Jonkman, B. J. (2009). TurbSim user's guide: version 1.50. Technical report, National Renewable Energy Laboratory, U.S. Department of Energy.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-MW reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory, U.S. Department of Energy.

Karian, Z. A. and Dudewicz, E. J. (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press.

Karian, Z. A. and Dudewicz, E. J. (2010). *Handbook of Fitting Statistical Distributions with R*. Taylor & Francis.

Kloeden, P. E. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer.

Koenker, R. (2017). Quantile regression: 40 years on. *Annual Review of Economics*, 9:155–176.

Lakhany, A. and Massuer, H. (2000). Estimating the parameters of the generalised lambda distribution. *Algo Research Quarterly*, 3(3):47–58.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proceedings of the 2nd International Conference on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Marelli, S. and Sudret, B. (2019). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-104.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22:833–847.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ.

Moustapha, M., Lataniotis, C., Wiederkehr, P., Wagner, P.-R., Wicaksono, D., Marelli, S., and Sudret, B. (2019). UQLib User Manual. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-201.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:116–155.

Plumlee, M. and Tuo, R. (2014). Building accurate emulators for stochastic simulations via quantile Kriging. *Technometrics*, 56:466–473.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23:470–472.

Slot, R. M. M., Sørensen, J. D., Sudret, B., Svenningsen, L., and Thogersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162.

Sobol', I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Computational Mathematics and Mathematical Physics*, 7:86–112.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637.

Su, S. (2007). Numerical maximum log likelihood estimation for generalized lambda distributions. *Computational Statistics & Data Analysis*, 51(8):3983–3998.

Sudret, B. (2015). Polynomial chaos expansions and stochastic finite element methods. In Phoon, K.-K. and Ching, J., editors, *Risk and Realiability in Geotechnical Engineering*, Risk and Reliability in Geotechnical Engineering, chapter 6, pages 265–300. Taylor and Francis.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623.

Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, Cambridge, New York.

Wand, M. and Jones, M. C. (1995). *Kernel smoothing*. Chapman & Hall/CRC.

Xiu, D. (2010). *Numerical methods for stochastic computations – A spectral method approach*. Princeton University press.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2021). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

5

Emulation of stochastic simulators using generalized lambda models

This chapter is a post-print of

> Zhu, X. and Sudret, B. (2021). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380. DOI:10.1137/20M1337302.[1]

differing from the published paper only in terms of layout and formatting.

Author contributions. **X. Zhu**: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - Original Draft, Visualization. **B. Sudret**: Supervision, Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition.

Abstract

Stochastic simulators are ubiquitous in many fields of applied sciences and engineering. In the context of uncertainty quantification and optimization, a large number of simulations is usually necessary, which becomes intractable for high-fidelity models. Thus surrogate models of stochastic simulators have been intensively investigated in the last decade. In this paper, we present a novel approach to surrogating the response distribution of a stochastic simulator which uses generalized lambda distributions, whose parameters are represented by polynomial chaos expansions of the model inputs. As opposed to most existing approaches, this new method does not require replicated runs of the simulator at each point of the experimental design. We propose a new fitting procedure which combines maximum conditional likelihood estimation with (modified) feasible generalized least-squares. We compare our method with state-of-the-art nonparametric kernel estimation on four different applications stemming from mathematical finance and epidemiology. Its performance is illustrated in terms of

[1]First published in SIAM/ASA Journal of Uncertainty Quantification in Volume 9, Issue 4, 2021, published by the Society for Industrial and Applied Mathematics (SIAM).

the accuracy of both the mean/variance of the stochastic simulator and the response distribution. As the proposed approach can also be used with experimental designs containing replications, we carry out a comparison on two of the examples, showing that replications do not necessarily help to get a better overall accuracy and may even worsen the results (at a fixed total number of runs of the simulator).

5.1 Introduction

With increasing demands on the functionality and performance of modern engineering systems, design and maintenance of complex products and structures require advanced computational models, a.k.a. simulators. They help assess the reliability and optimize the behavior of the system already at the design phase. Classical simulators are usually deterministic because they implement solvers for the governing equation of the system. Thus, repeated model evaluations with the same input parameters consistently result in the same value of the output quantities of interest (QoIs). In contrast, *stochastic simulators* contain intrinsic randomness, which leads to the QoI being a random variable conditioned on the given set of input parameters. In other words, each model evaluation with the same input values generates a realization of the response random variable that follows an unknown distribution. Formally, a stochastic simulator $\mathcal{M}_s$ can be expressed as

$$\begin{aligned} \mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\to \mathbb{R}, \\ (\boldsymbol{x}, \omega) &\mapsto \mathcal{M}_s(\boldsymbol{x}, \omega), \end{aligned} \tag{5.1}$$

where $\boldsymbol{x}$ is the input vector that belongs to the input space $\mathcal{D}_{\boldsymbol{X}}$, and Ω denotes the sample space of the probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$ that represents the internal source of randomness.

Stochastic simulators are widely used in modern engineering, finance, and medical sciences. Typical examples include evaluating the performance of a wind turbine under stochastic loads (Abdallah et al., 2019), predicting the price of an option in financial markets (Shreve, 2004), and the spread of a disease in epidemiology (Britton, 2010).

Due to the random nature of stochastic simulators, repeated model evaluations with the same input parameters, called hereinafter *replications*, are necessary to fully characterize the probability distribution of the corresponding QoI. In addition, uncertainty quantification and optimization problems typically require model evaluations for various sets of input parameters. Altogether, it is necessary to have a large number of model runs, which becomes intractable for costly models. To alleviate the computational burden, surrogate models, a.k.a. emulators, can be used to replace the original model. Such a model emulates the input-output relation of the simulator and is easy and cheap to evaluate.

Among several options for constructing surrogate models, this paper focuses on the so-called *nonintrusive* approaches. More precisely, the computational model is considered as a "black box" and is only required to be evaluated on a limited number of input values, called the *experimental design* (ED).

Three classes of methods can be found in the literature for emulating the entire response distribution of a stochastic code in a nonintrusive manner. The first one is the *random field approach*, which approximates the stochastic simulator by a random field. The definition in Eq. (5.1) implies that a stochastic simulator can be regarded as a random field indexed by its input variables. Controlling the intrinsic randomness allows one to get access to different trajectories of the simulator, which are deterministic functions of the input variables.

In practice, this is achieved by fixing the *random seed* inside the simulator. Evaluations of the trajectories over the experimental design can then be extended to continuous trajectories, either by classical surrogate methods (Jimenez et al., 2017) or through Karhunen–Loève expansions (Azzi et al., 2019). Since this approach requires the effective access to the random seed, it is only applicable to data generated in a specific way.

Another class of methods is the *replication-based approach*, which relies on using replications at all points of the experimental design to represent the response distribution through a suitable parametrization. The estimated distribution parameters are then treated as (noisy) outputs of a deterministic simulator. Then, conventional surrogate modeling methods, such as Gaussian processes (Rasmussen and Williams, 2006) and polynomial chaos expansions (PCEs; Blatman and Sudret, 2011), can emulate these parameters as a function of the model input (Moutoussamy et al., 2015; Browne et al., 2016). Because this approach employs two separate steps, the surrogate quality depends on the accuracy of the distribution estimation from replicates in the first step (Zhu and Sudret, 2020). Therefore, many replications are necessary, especially when nonparametric estimators are used for the local inference (Moutoussamy et al., 2015; Browne et al., 2016).

A third class of methods, known as the *statistical approach*, does not require replications or controlling the random seed. If the response distribution belongs to the exponential family, generalized linear models (McCullagh and Nelder, 1989) and generalized additive models (Hastie and Tibshirani, 1990) can be efficiently applied. When the QoI for a given set of input parameters follows an arbitrary distribution, nonparametric estimators can be considered, notably kernel density estimators (Fan and Gijbels, 1996; Hall et al., 2004) and projection estimators (Efromovich, 2010). However, it is well known that nonparametric estimators suffer from the *curse of dimensionality* (Tsybakov, 2009), meaning that the necessary amount of data increases drastically with increasing input dimensionality.

In a recent paper (Zhu and Sudret, 2020), we proposed a novel stochastic emulator called the *generalized lambda model* (GLaM). Such a surrogate model uses generalized lambda distributions (GLDs) to represent the response probability density function (PDF). The dependence of the distribution parameters on the input is modeled by PCEs. However, the methods developed in Zhu and Sudret (2020) rely on replications. In the present contribution, we propose a new statistical approach combining feasible generalized least-squares with maximum conditional likelihood estimations to get rid of the need for replications. Therefore, the proposed method is much more versatile in the sense that replications and seed controls are no longer necessary.

The paper is organized as follows. In Sections 5.2 and 5.3, we briefly review GLDs and PCEs, which are the two main elements constituting the GLaM. In Section 5.4, we recap the GLaM framework and introduce the maximum conditional likelihood estimator. Then, we present the algorithm developed to find an appropriate starting point to optimize the likelihood, and to design ad hoc truncation schemes for the PCEs of distribution parameters. In Section 5.5, we validate the proposed method on two analytical examples and two case studies in mathematical finance and epidemiology, respectively, to showcase its capability to tackle real problems. Finally, we summarize the main findings of the paper and provide an outlook for future research in Section 5.6.

5.2 Generalized lambda distributions

5.2.1 Formulation

The generalized lambda distribution (GLD) is a flexible probability distribution family. It is able to approximate most of the well-known parametric distributions (Freimer et al., 1988; Karian and Dudewicz, 2000), e.g., uniform, normal, Weibull, and Student's t distributions. The definition of a GLD relies on a parametrization of the *quantile function* $Q(u)$, which is a nondecreasing function defined on $[0,1]$. In this paper, we consider the GLD of the Freimer–Kollia–Mudholkar–Lin family (Freimer et al., 1988), which is defined by

$$Q(u;\boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right), \tag{5.2}$$

where $\boldsymbol{\lambda} = \{\lambda_l : l = 1,\dots,4\}$ are the four distribution parameters. More precisely, λ_1 is the location parameter, λ_2 is the scaling parameter, and λ_3 and λ_4 are the shape parameters. To ensure valid quantile functions (i.e., Q being nondecreasing on $u \in [0,1]$), it is required that λ_2 be positive. Based on the quantile function, the PDF $f_W(w;\boldsymbol{\lambda})$ of a random variable W following a GLD can be derived as

$$f_W(w;\boldsymbol{\lambda}) = \frac{1}{Q'(u;\boldsymbol{\lambda})} = \frac{\lambda_2}{u^{\lambda_3-1}+(1-u)^{\lambda_4-1}}\mathbb{1}_{[0,1]}(u), \text{ with } u = Q^{-1}(w;\boldsymbol{\lambda}), \tag{5.3}$$

where $Q'(u;\boldsymbol{\lambda})$ is the derivative of Q with respect to u, and $\mathbb{1}_{[0,1]}$ is the indicator function. A closed-form expression of Q^{-1}, and therefore of f_W, is in general not available, and thus the PDF is evaluated by solving the nonlinear equation Eq. (5.3) numerically.

5.2.2 Properties

GLDs cover a wide range of unimodal shapes, including bell-shaped, U-shaped, S-shaped and bounded-mode distributions, which is determined by λ_3 and λ_4, as illustrated in Figure 5.1 (Zhu and Sudret, 2020). For instance, $\lambda_3 = \lambda_4$ produces symmetric PDFs, and $\lambda_3, \lambda_4 < 1$ leads to bell-shaped distributions. Moreover, λ_3 and λ_4 are closely linked to the support and the tail properties of the corresponding PDF. $\lambda_3 > 0$ implies that the PDF support is left-bounded and $\lambda_4 > 0$ corresponds to right-bounded PDFs. Conversely, the distribution has lower infinite support for $\lambda_3 \leqslant 0$ and upper infinite support for $\lambda_4 \leqslant 0$. More precisely, the support of the PDF denoted by $\operatorname{supp}\left(f_W(w;\boldsymbol{\lambda})\right) = [B_l, B_u]$ is given by

$$B_l(\boldsymbol{\lambda}) = \begin{cases} -\infty, & \lambda_3 \leqslant 0, \\ \lambda_1 - \frac{1}{\lambda_2\lambda_3}, & \lambda_3 > 0, \end{cases} \quad B_u(\boldsymbol{\lambda}) = \begin{cases} +\infty, & \lambda_4 \leqslant 0, \\ \lambda_1 + \frac{1}{\lambda_2\lambda_4}, & \lambda_4 > 0. \end{cases} \tag{5.4}$$

Importantly, for $\lambda_3 < 0$ ($\lambda_4 < 0$), the left (resp., right) tail decays asymptotically as a power law, and thus the GLD family can also provide fat-tailed distributions. Due to this power law decay, for $\lambda_3 \leqslant -\frac{1}{k}$ or $\lambda_4 \leqslant -\frac{1}{k}$, moments of order greater than k do not exist. For $\lambda_3, \lambda_4 > -0.5$, the mean and variance exist and are given by

$$\mu = \mathbb{E}[W] = \lambda_1 - \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3+1} - \frac{1}{\lambda_4+1}\right), \quad v = \operatorname{Var}[W] = \frac{(d_2 - d_1^2)}{\lambda_2^2}, \tag{5.5}$$

where the two auxiliary variables d_1 and d_2 are defined by

$$\begin{aligned} d_1 &= \frac{1}{\lambda_3}\,\mathrm{B}(\lambda_3+1,1) - \frac{1}{\lambda_4}\,\mathrm{B}(1,\lambda_4+1), \\ d_2 &= \frac{1}{\lambda_3^2}\,\mathrm{B}(2\lambda_3+1,1) - \frac{2}{\lambda_3\lambda_4}\,\mathrm{B}(\lambda_3+1,\lambda_4+1) + \frac{1}{\lambda_4^2}\,\mathrm{B}(1,2\lambda_4+1), \end{aligned} \tag{5.6}$$

with B denoting the beta function.

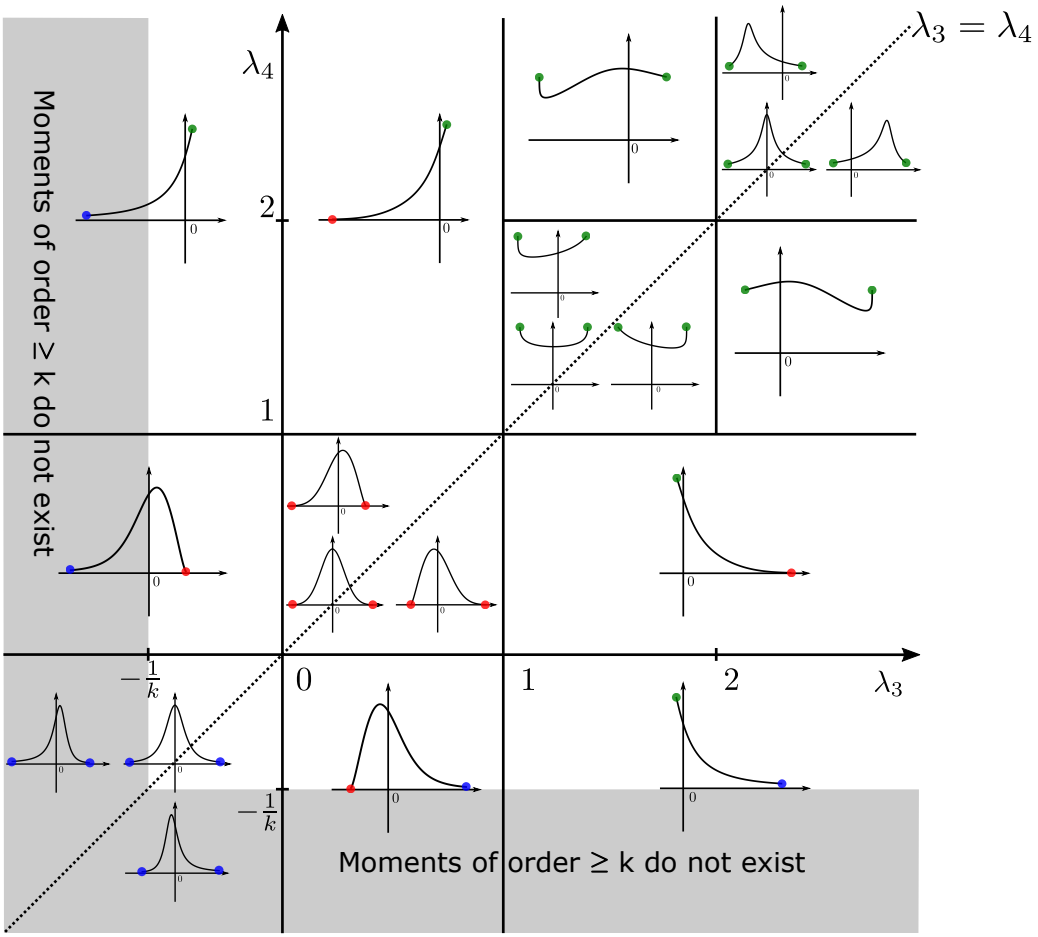


Figure 5.1: A graphical illustration of the PDF of the FKML family of GLD as a function of λ_3 and λ_4. The values of λ_1 and λ_2 are set to 0 and 1, respectively. The blue points indicate that the PDF has infinite support in the marked direction. In contrast, both the red and green points denote the boundary points of the PDF support. More precisely, the PDF $f_W(w) = 0$ on the red dots, whereas $f_W(w) = 1$ on the green ones.

5.3 Polynomial chaos expansions

Consider a deterministic computational model $\mathcal{M}_d(\boldsymbol{x})$ that maps a set of input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)^\mathsf{T} \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$ to the system response $z \in \mathbb{R}$. In the context of uncertainty quantification, the input variables are affected by uncertainty due to lack of knowledge or intrinsic variability (also called aleatory uncertainty). Therefore, they are modeled by random variables and grouped into a random vector $\boldsymbol{X}$ characterized by a joint PDF

$f_{\boldsymbol{X}}$. The uncertainty in the input variables propagates through the the model $\mathcal{M}_d$ to the output, which becomes a random variable denoted by $Z = \mathcal{M}_d(\boldsymbol{X})$.

Remark. *$f_{\boldsymbol{X}}$ is the joint PDF for the input variables, which is needed to define orthogonal polynomials as described below. It should not be confused with the stochasticity of the simulator addressed in the next sections.*

Provided that the output random variable Z has finite variance, $\mathcal{M}_d$ belongs to the Hilbert space $\mathcal{H}$ of square-integrable functions associated with the inner product

$$\langle u, v \rangle_{\mathcal{H}} \overset{\text{def}}{=} \mathbb{E}\left[u(\boldsymbol{X})v(\boldsymbol{X})\right] = \int_{\mathcal{D}_{\boldsymbol{X}}} u(\boldsymbol{x})v(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{5.7}$$

If the joint PDF $f_{\boldsymbol{X}}$ fulfills certain conditions (Ernst et al., 2012), the space spanned by multivariate polynomials is dense in $\mathcal{H}$. In other words, $\mathcal{H}$ is a separable Hilbert space admitting a polynomial basis.

In this study, we assume that $\boldsymbol{X}$ has mutually independent components, and thus the joint distribution $f_{\boldsymbol{X}}$ is expressed as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{j=1}^{M} f_{X_j}(x_j). \tag{5.8}$$

Let $\{\phi_k^{(j)} : k \in \mathbb{N}\}$ be the orthogonal polynomial basis with respect to the marginal distribution of f_{X_j}, i.e.,

$$\mathbb{E}\left[\phi_k^{(j)}(X_j)\,\phi_l^{(j)}(X_j)\right] = \delta_{kl}, \tag{5.9}$$

with δ being the Kronecker symbol defined by $\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise. Then, the multivariate orthogonal polynomial basis can be obtained as the tensor product of univariate polynomials (Soize and Ghanem, 2004):

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j), \tag{5.10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M) \in \mathbb{R}^M$ denotes the multi-index of degrees. Each component α_j indicates the polynomial degree of ϕ_{α_j} and thus of $\psi_{\boldsymbol{\alpha}}$ in the jth variable x_j. For some classical distributions, e.g., normal, uniform, exponential, the associated univariate orthogonal polynomials are well known as Hermite, Legendre, and Laguerre polynomials (Xiu and Karniadakis, 2002). For arbitrary marginal distributions, such a basis can be computed numerically through the *Stieltjes procedure* (Gautschi, 2004).

Following the construction defined in Eq. (5.10), $\{\psi_{\boldsymbol{\alpha}}(\cdot), \boldsymbol{\alpha} \in \mathbb{N}^M\}$ forms an orthogonal basis for $\mathcal{H}$. Thus, the random output Z can be represented by

$$Z = \mathcal{M}_d(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{5.11}$$

where $c_{\boldsymbol{\alpha}}$ is the coefficient associated with the basis function $\psi_{\boldsymbol{\alpha}}$. The spectral representation in Eq. (5.11) is a series with infinitely many terms. In practice, it is necessary to adopt truncation schemes to *approximate* $\mathcal{M}_d(\boldsymbol{x})$ with a finite series defined by a finite subset $\mathcal{A} \subset \mathbb{N}^M$ of multi-indices. A typical scheme is the hyperbolic (q-

norm) truncation scheme (Blatman and Sudret, 2010):

$$\mathcal{A}^{p,q,M} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M, \|\boldsymbol{\alpha}\|_q = \left(\sum_{i=1}^{M} |\alpha_i|^q \right)^{\frac{1}{q}} \leqslant p \right\}, \tag{5.12}$$

where p is the maximum total degree of polynomials, and $q \leqslant 1$ defines the quasi-norm $\|\cdot\|_q$. Note that with $q = 1$, we obtain the so-called full basis of total degree less than p.

For an arbitrary distribution $f_{\boldsymbol{X}}$ with dependent components of $\boldsymbol{X}$, the usual practice is to transform $\boldsymbol{X}$ into an auxiliary vector $\boldsymbol{\xi}$ with independent components (e.g., a standard normal vector) using the Nataf or Rosenblatt transform (Torre et al., 2019). Alternatively, polynomials orthogonal to the joint distribution may be computed on the fly using a numerical Gram–Schmidt orthogonalization (Jakeman et al., 2019).

5.4 Generalized lambda models (GLaMs)

5.4.1 Introduction

Because of their flexibility, we assume that the response random variable of a stochastic simulator for a given input vector $\boldsymbol{x}$ follows a GLD. Hence, the distribution parameters $\boldsymbol{\lambda}$ are functions of the input variables:

$$Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{5.13}$$

Under appropriate conditions discussed in Section 5.3, each component of $\boldsymbol{\lambda}(\boldsymbol{x})$ admits a spectral representation in terms of orthogonal polynomials. Recall that $\lambda_2(\boldsymbol{x})$ is required to be positive (see Section 5.2). Thus, we choose to build the associated PCE on the natural logarithm transform $\log\left(\lambda_2(\boldsymbol{x})\right)$. This results in the following approximations:

$$\lambda_l\left(\boldsymbol{x}\right) \approx \lambda_l^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_l} c_{l,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad l = 1, 3, 4, \tag{5.14}$$

$$\lambda_2\left(\boldsymbol{x}\right) \approx \lambda_2^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right) = \exp\left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_2} c_{2,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \right), \tag{5.15}$$

where $\boldsymbol{\mathcal{A}} = \{\mathcal{A}_l : l = 1, \ldots, 4\}$ are the truncation sets defining the basis functions, and $\boldsymbol{c} = \{c_{l,\boldsymbol{\alpha}} : l = 1, \ldots, 4,\ \boldsymbol{\alpha} \in \mathcal{A}_l\}$ are coefficients associated to the bases. For the purpose of clarity, we explicitly express $\boldsymbol{c}$ in the spectral approximations as in $\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right)$ to emphasize that $\boldsymbol{c}$ are the model parameters.

The generalized lambda model presented above is a statistical model. It involves two approximations. First, the response distribution of a stochastic simulator is approximated by GLDs. As illustrated in Figure 5.1, GLDs cover a wide range of unimodal shapes but cannot produce multimodal distributions. Thus, the GLD representation is appropriate when the response distribution stays unimodal. In this case, the flexibility of GLDs allows capturing the possible shape variation of the response distribution within a single parametric family. Second, the distribution parameters $\boldsymbol{\lambda}(\boldsymbol{x})$ seen as functions of $\boldsymbol{x}$ are represented by truncated polynomial chaos expansions. So they must belong to the Hilbert space of square-integrable functions with respect to $f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$.

5.4.2 Estimation of the model parameters

Given the truncation sets $\mathcal{A}$, the coefficients $\boldsymbol{c}$ need to be estimated from data to build the surrogate model. In this paper, as opposed to Zhu and Sudret (2020) and the vast majority of the literature on stochastic simulators, the simulator is required to be evaluated *only once* on the experimental design $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\right\}$, and the associated model responses are collected in $\mathcal{Y} = \left\{y^{(1)}, \dots, y^{(N)}\right\}$. To develop surrogate models in a nonintrusive manner, we propose using the maximum conditional likelihood estimator:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \mathsf{L}\left(\boldsymbol{c}\right), \tag{5.16}$$

where

$$\mathsf{L}\left(\boldsymbol{c}\right) = \sum_{i=1}^{N} \log\left(f^{\mathrm{GLD}}\left(y^{(i)}; \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{c}\right)\right)\right). \tag{5.17}$$

Here, f^{GLD} denotes the PDF of the GLD defined in Eq. (5.3), and $\mathcal{C}$ is the search space for $\boldsymbol{c}$. The estimator introduced in Eq. (5.17) can be derived from minimizing the Kullback–Leibler divergence between the surrogate PDF and the underlying true response PDF over $\mathcal{D}_{\boldsymbol{X}}$; see details in Zhu and Sudret (2020). The advantages of this estimation method are twofold. On the one hand, it removes the need for replications in the experimental design. On the other hand, if a GLaM for a certain choice of $\boldsymbol{c}$ can exactly represent the stochastic simulator, the proposed estimator is *consistent* under mild conditions, as shown in Theorem 5.1 (see Section 5.a.1 for a detailed proof).

Theorem 5.1. *Let* $(\boldsymbol{X}^{(1)}, Y^{(1)}), \dots, (\boldsymbol{X}^{(N)}, Y^{(N)})$ *be independent and identically distributed random variables following* $\boldsymbol{X} \sim P_{\boldsymbol{X}}$ *and* $Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{c}_0)\right)$. *If the following conditions are fulfilled, the estimator defined in Eq. (5.16) is consistent, that is,*

$$\hat{\boldsymbol{c}} \xrightarrow{a.s.} \boldsymbol{c}_0. \tag{5.18}$$

(i) $P_{\boldsymbol{X}}$ *is absolutely continuous with respect to the Lebesgue measure of* $\mathbb{R}^M$, *i.e., the joint PDF* $f_X(\boldsymbol{x})$ *is Lebesgue-measurable.*

(ii) f_X *has a compact support* $\mathcal{D}_{\boldsymbol{X}}$.

(iii) $\mathcal{C}$ *is compact, and* $\boldsymbol{c}_0 \in \mathcal{C}$.

(iv) There exists a set $A \subset \mathcal{D}_{\boldsymbol{X}}$ *with* $P_{\boldsymbol{X}}\left(\boldsymbol{X} \in A\right) > 0$ *such that* $\forall \boldsymbol{x} \in A$, $Y(\boldsymbol{x})$ *does not follow a uniform distribution.*

Most of the assumptions in the Theorem 5.1 are realistic, except the one that the true model can be exactly represented by a GLaM, which is rather technical to guarantee the consistency. In practice, we do not require the QoI for any input parameters following a GLD but assume that the response distribution can be well approximated by GLDs.

It is worth remarking that since a GLD can have very fat tails (see Section 5.2.2), solving the optimization problem may produce response PDFs with unexpected infinite moments when the model is trained on a small data set. To prevent too-fat tails (if no prior knowledge suggests it), we apply the threshold $\lambda_3^{\mathrm{PC}}(\boldsymbol{x}) = \max\left\{\lambda_3^{\mathrm{PC}}(\boldsymbol{x}; \hat{\boldsymbol{c}}), -0.3\right\}$ and $\lambda_4^{\mathrm{PC}}(\boldsymbol{x}) = \max\left\{\lambda_4^{\mathrm{PC}}(\boldsymbol{x}; \hat{\boldsymbol{c}}), -0.3\right\}$, which indicates that we enforce the surrogate PDFs to have finite moments up to order 3 (higher order moments may exist depending on $\hat{\boldsymbol{c}}$). Thresholds

larger than -0.3 (e.g., from -0.1 to 0) can be used if the response PDF is known to be light-tailed. Note that when enough data are available, these operations are unnecessary because the resulting model does not exceed the threshold. Although the thresholdings could have been imposed in the model definition in Eq. (5.14), they change the regularity of the optimization problem, and do not generally improve the performance according to our experience. Therefore, we only use them for postprocessing.

Remark 5.1. *While we consider the simulator to be evaluated only once for each point of the experimental design in this paper, the estimator defined in Eq. (5.16) is not limited to this type of data. When replications are available, the objective function can be reformulated to*

$$\mathsf{L}(\boldsymbol{c}) = \sum_{i=1}^{N} \frac{1}{R^{(i)}} \sum_{r=1}^{R^{(i)}} \log\left(f^{\mathrm{GLD}}\left(y^{(i,r)}; \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{c}\right)\right)\right), \tag{5.19}$$

where $R^{(i)}$ denotes the number of replications at point $\boldsymbol{x}^{(i)}$, and $y^{(i,r)}$ is the model response for $\boldsymbol{x}^{(i)}$ at the rth replication. In addition, if $R^{(i)}$ is constant for all points $\boldsymbol{x}^{(i)} \in \mathcal{X}$, Eq. (5.19) provides the same estimator as in our previous work (Zhu and Sudret, 2020).

5.4.3 Fitting procedure

In practice, the evaluation of $\mathsf{L}(\boldsymbol{c})$ is not straightforward because the PDF of GLDs does not have an explicit form as shown in Eq. (5.3). Details about the evaluation procedure are given in Zhu and Sudret (2020). Note that the optimization problem Eq. (5.16) is subject to complex inequality constraints due to the dependence of the PDF support on $\boldsymbol{\lambda}$ (see Eq. (5.4)). Given a starting point, we follow the optimization strategy developed in Zhu and Sudret (2020): We first apply the derivative-based *trust-region* optimization algorithm (Steihaug, 1983) without constraints. If none of the inequality constraints is activated at the optimum, we keep the results as the final estimates. Otherwise, the constrained (1+1)-CMA-ES algorithm (Arnold and Hansen, 2012) available in the software UQLab (Moustapha et al., 2019) is used instead.

Because $\mathsf{L}(\boldsymbol{c})$ is highly nonlinear, a good starting point is necessary to guarantee the convergence of the optimization algorithm. In this section, we introduce a robust method to find a suitable starting point.

According to Eq. (5.5), the mean $\mu(\boldsymbol{x})$ and the variance function $v(\boldsymbol{x})$ of a GLaM satisfy

$$\begin{aligned} \mu(\boldsymbol{x}) &= \lambda_1^{\mathrm{PC}}(\boldsymbol{x}) + \frac{1}{\lambda_2^{\mathrm{PC}}(\boldsymbol{x})} g\left(\lambda_3^{\mathrm{PC}}(\boldsymbol{x}), \lambda_4^{\mathrm{PC}}(\boldsymbol{x})\right), \\ \log\left(v(\boldsymbol{x})\right) &= -2\log\left(\lambda_2^{\mathrm{PC}}(\boldsymbol{x})\right) + h\left(\lambda_3^{\mathrm{PC}}(\boldsymbol{x}), \lambda_4^{\mathrm{PC}}(\boldsymbol{x})\right), \end{aligned} \tag{5.20}$$

where we group the dependence of μ and $\log(v)$ on λ_3 and λ_4 into g and h, respectively, for the purpose of simplicity. If $\lambda_3^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_4^{\mathrm{PC}}(\boldsymbol{x})$ do not vary strongly on $\mathcal{D}_{\boldsymbol{X}}$, we observe that the variations of the mean and the variance function are mostly dominated by the location parameter $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$ and the scale parameter $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$.

Recall that the spectral approximation for $\lambda_2(\boldsymbol{x})$ is on its logarithmic transform. If a PCE can be constructed for $\mu(\boldsymbol{x})$ and $-\frac{1}{2}\log\left(v(\boldsymbol{x})\right)$, the associated coefficients can be used as a preliminary guess for the coefficients of $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$, respectively. As a result, we first focus on estimating the mean and the variance

function as follows:

$$\mu(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_\mu} c_{\mu,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad v(\boldsymbol{x}) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_v} c_{v,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right),$$

where the form of the variance function implies a multiplicative *heteroskedastic* effect (see Harvey, 1976).

The mean estimation is a classical regression problem. However, since the variance function is also unknown and needs to be estimated, the heteroskedastic effect should be taken into account. Many methods have been developed in statistics and applied science to tackle heteroskedastic regression problems. They can be classified into two groups: one class of methods relies on repeated measurements at given input values (Sadler and Smith, 1985; Ankenman et al., 2010; Murcia et al., 2018) (replication-based), whereas a second class of methods jointly estimates both quantities by optimizing certain functions without the need for replications (Nelder and Pregibon, 1987; Davidian and Carroll, 1987; Goldberg et al., 1997; Marrel et al., 2012). Some studies (Davidian and Carroll, 1987; Marrel et al., 2012) have shown higher efficiency of the second class of methods over the former. This finding supports our pursuit for a replication-free approach. In particular, we opt for feasible generalized least-squares (FGLS; Wooldridge, 2013), which iteratively fits the mean and variance functions in an alternative way.

The details are described in Algorithm 5.1. In this algorithm, OLS denotes the use of ordinary least-squares, and WLS is weighted least-squares. $\hat{\boldsymbol{v}}$ corresponds to the set of estimated variances on the design points in $\mathcal{X}$ which are then used as weights in WLS to re-estimate $\boldsymbol{c}_\mu$.

Algorithm 5.1 Feasible generalized least-squares (FGLS)

1: $\hat{\boldsymbol{c}}_\mu \leftarrow \mathrm{OLS}(\mathcal{X}, \mathcal{Y})$
2: **for** $i \leftarrow 1, \ldots, N_{\mathrm{FGLS}}$ **do**
3: $\quad \hat{\boldsymbol{\mu}} \leftarrow \sum_{\boldsymbol{\alpha}\in\mathcal{A}_\mu} c_{\mu,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\mathcal{X})$
4: $\quad \tilde{\boldsymbol{r}} \leftarrow 2\log(|\mathcal{Y} - \hat{\boldsymbol{\mu}}|)$
5: $\quad \hat{\boldsymbol{c}}_v \leftarrow \mathrm{OLS}(\mathcal{X}, \tilde{\boldsymbol{r}})$
6: $\quad \hat{\boldsymbol{v}} = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_v} c_{v,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\mathcal{X})\right)$
7: $\quad \hat{\boldsymbol{c}}_\mu \leftarrow \mathrm{WLS}(\mathcal{X}, \mathcal{Y}, \hat{\boldsymbol{v}})$
8: **end for**
9: Output: $\hat{\boldsymbol{c}}_\mu, \hat{\boldsymbol{c}}_v$

After obtaining $\hat{\boldsymbol{c}}_\mu$ and $\hat{\boldsymbol{c}}_v$ from FGLS, we perform two rounds of the optimization procedure described at the beginning of this section to build the GLaM surrogate. First, we set the starting points as $\boldsymbol{c}_1 = \boldsymbol{c}_\mu$, $\boldsymbol{c}_2 = -\frac{1}{2}\boldsymbol{c}_v$, and $\lambda_3^{\mathrm{PC}}(\boldsymbol{x}) = \lambda_4^{\mathrm{PC}}(\boldsymbol{x}) = 0.13$, which corresponds to a normal-like shape. Then, we fit a GLaM with $\lambda_3^{\mathrm{PC}}(\boldsymbol{x})$ $\lambda_4^{\mathrm{PC}}(\boldsymbol{x})$ being only constant; i.e., the coefficients of nonconstant basis functions are kept as zeros during the fitting. Finally, we use the resulting estimates as a starting point and construct a final GLaM with all the considered basis functions by solving Eq. (5.17).

5.4.4 Truncation schemes

Provided that the bases of $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x})$ are given, we have presented a procedure to construct GLaMs from data in the previous section. However, there is generally no prior knowledge that would help select the truncation sets $\mathcal{A}_l$'s ab initio. In this section, we develop a method to determine a suitable hyperbolic truncation scheme $\mathcal{A}^{p,q,M}$ presented in Eq. (5.12) for each component of $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x})$.

As discussed in Section 5.2, $\lambda_3^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_4^{\mathrm{PC}}(\boldsymbol{x})$ control the shape variations of the response PDF. We assume that the shape does not vary in a strongly nonlinear way. Hence, the associated p can be set to a small value, e.g., $p = 1$, in practice. In contrast, $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$ require possibly larger degree p since their behavior is associated with the mean and the variance function, which might vary nonlinearly over $\mathcal{D}_{\boldsymbol{X}}$. To this end, we modify Algorithm 5.1 to adaptively find appropriate truncation schemes for $\mu(\boldsymbol{x})$ and $v(\boldsymbol{x})$, which are then used for $\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$, respectively.

Algorithm 5.2 Modified feasible generalized least-squares

Input: $(\mathcal{X}, \mathcal{Y}), \boldsymbol{p}_1, \boldsymbol{q}_1, \boldsymbol{p}_2, \boldsymbol{q}_2$
$\mathcal{A}_\mu,\ \hat{\boldsymbol{c}}_\mu \leftarrow \mathrm{AOLS}\left(\mathcal{X}, \mathcal{Y}, \boldsymbol{p}_1, \boldsymbol{q}_1\right)$
for $i \leftarrow 1, \ldots, N_{\mathrm{FGLS}}$ **do**
  $\hat{\boldsymbol{\mu}} \leftarrow \sum_{\boldsymbol{\alpha} \in \mathcal{A}_\mu} c_{m,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\mathcal{X})$
  $\tilde{\boldsymbol{r}} \leftarrow 2 \log\left(|\mathcal{Y} - \hat{\boldsymbol{\mu}}|\right)$
  $\mathcal{A}_v^i,\ \hat{\boldsymbol{c}}_v^i,\ \varepsilon_{\mathrm{LOO}}^i \leftarrow \mathrm{AOLS}\left(\mathcal{X}, \tilde{\boldsymbol{r}}, \boldsymbol{p}_2, \boldsymbol{q}_2\right)$
  $\hat{\boldsymbol{v}} \leftarrow \exp\left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_v} c_{v,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\mathcal{X})\right)$
  $\hat{\boldsymbol{c}}_\mu \leftarrow \mathrm{WLS}\left(\mathcal{X}, \mathcal{Y}, \mathcal{A}_\mu, \hat{\boldsymbol{v}}\right)$
end for
$i^* = \arg\min\left\{\varepsilon_{\mathrm{LOO}}^i : i = 1, \ldots, N_{\mathrm{FGLS}}\right\}$
Output: $\mathcal{A}_\mu, \hat{\boldsymbol{c}}_\mu^{i^*}, \mathcal{A}_v^{i^*}, \hat{\boldsymbol{c}}_v^{i^*}$

Algorithm 5.2 presents the modified FGLS. Instead of using OLS, we apply the *adaptive ordinary least-squares* with degree and q-norm adaptivity (referred to as AOLS; Marelli and Sudret, 2019). This algorithm builds a series of PCEs, each of which is obtained by applying OLS with the truncation set $\mathcal{A}^{p,q,M}$ defined by a particular combination of $p \in \boldsymbol{p}$ and $q \in \boldsymbol{q}$. Then, it selects the truncation scheme for which the associated PCE has the lowest *leave-one-out* error. In the modified FGLS, the truncation set $\mathcal{A}_\mu$ for $\mu(\boldsymbol{x})$ is selected only once (before the loop), whereas several truncation schemes $\{\mathcal{A}_v^i : i = 1, \ldots, N_{\mathrm{FGLS}}\}$ are obtained. We select the one corresponding to the smallest leave-one-out error on the expansion of the variance as the truncation set $\mathcal{A}_v$ for $v(\boldsymbol{x})$. After running Algorithm 5.2, we apply the two-round optimization strategy described in the previous section to build the GLaM corresponding to the selected truncation schemes.

There are several parameters to be determined in Algorithm 5.2. In the following examples and applications, we set the candidate degrees $\boldsymbol{p}_1 = \{0, \ldots, 10\}$ for $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$, and $\boldsymbol{p}_2 = \{0, \ldots, 5\}$ for $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$. $\boldsymbol{p}_1$ contains high degrees to approximate possibly highly nonlinear mean functions, the accuracy of which is crucial for basis selections for $\lambda_2(\boldsymbol{x})$ in Algorithm 5.2. $\boldsymbol{p}_2$ is set to have degrees up to 5, allowing relatively complex variations. The lists of q-norms are $\boldsymbol{q}_1 = \boldsymbol{q}_2 = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, which contains the full basis. The total number of FGLS iterations is set to $N_{\mathrm{FGLS}} = 10$ which, according to our experience, is enough to find an

appropriate truncated set for $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$.

5.5 Application examples

In this section, we validate the proposed algorithm on two analytical examples and two case studies in mathematical finance and epidemiology. In the four cases, the response distributions do not belong to a single parametric family, so as to test the flexibility of the proposed method. In addition, we compare the performance of GLaMs with the nonparametric kernel conditional density estimator from the package `np` (Hayfield and Racine, 2008) implemented in R. The latter performs a thorough leave-one-out cross-validation with a multistart strategy to choose the bandwidths (Hall et al., 2004), which is one of the state-of-the-art kernel estimation methods. The surrogate model built by this method is referred to as the kernel conditional density estimator (KCDE).

Alongside GLaM and KCDE, another surrogate model, the heteroskedastic Gaussian process (denoted by GP), is also considered. This model assumes that the response distribution is Gaussian, and the mean and variance functions are represented by Gaussian processes. We apply the method proposed by Binois et al. (2019) which adopts a sequential design strategy to actively balance the trade-off between replications and explorations. The algorithm is available in the package `hetGP` in R. However, due to the sequential design (the new points are added one by one), building such a surrogate can be very time-consuming (cf. Section 5.5.2 for details). Consequently, we present the comparisons with `hetGP` only for the first two examples.

Moreover, for comparison purposes, we consider another "Gaussian" surrogate model where we represent the response distribution with a normal distribution. The associated mean and variance, which are functions of the input $\boldsymbol{x}$, are not fitted to data but set to the *true* values of the simulator. In other words, this surrogate model should represent the "oracle" of Gaussian-type mean-variance surrogate models, such as the ones presented in Marrel et al. (2012) and Binois et al. (2018).

We use Latin hypercube sampling (McKay et al., 1979) to generate the experimental design for GLaM and KCDE. The stochastic simulator is only evaluated once for each vector of input parameters. The associated QoI values are used to construct surrogate models with the proposed estimation procedure in Section 5.4.3. In contrast, the construction of the GP relies on a sequential design strategy which adaptively find new points to evaluate (Binois et al., 2019). Hence, we use Latin hypercube sampling of 20% of the total number of model runs to initiate the process. Then, the algorithm proceeds by iteratively looking for points to evaluate and updating the surrogate.

To quantitatively assess the performance of the surrogate model, we define an error measure between the underlying model and the emulator by

$$\varepsilon = \mathbb{E}\left[d\left(Y(\boldsymbol{X}), \hat{Y}(\boldsymbol{X})\right)\right], \tag{5.21}$$

where $Y(\boldsymbol{X})$ is the model response, $\hat{Y}(\boldsymbol{X})$ corresponds to that of the surrogate, $d(Y_1, Y_2)$ denotes the contrast measure between the probability distributions of Y_1 and Y_2, and the expectation is taken with respect to $\boldsymbol{X}$. In this study, we use the *normalized Wasserstein distance*, defined by

$$d(Y_1, Y_2) = \frac{d_{\mathrm{WS}}(Y_1, Y_2)}{\sigma(Y_1)}, \tag{5.22}$$

where d_{WS} is the *Wasserstein distance of order two* (Villani, 2009) defined by

$$d_{\mathrm{WS}}\left(Y_1, Y_2\right) \stackrel{\text{def}}{=} \left\|Q_1 - Q_2\right\|_2 = \sqrt{\int_0^1 \left(Q_1(u) - Q_2(u)\right)^2 \mathrm{d}u}\,, \tag{5.23}$$

where Q_1 and Q_2 are the quantile functions of Y_1 and Y_2, respectively. As a summary, by combining Eq. (5.21) and Eq. (5.23) the global error reads

$$\varepsilon = \int_{\mathcal{D}_{\boldsymbol{X}}} \sqrt{\int_0^1 \left(Q_{Y(\boldsymbol{x})}(u) - Q_{\hat{Y}(\boldsymbol{x})}(u)\right)^2 \mathrm{d}u} \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{\sqrt{\mathrm{Var}\left[Y(\boldsymbol{x})\right]}} \mathrm{d}\boldsymbol{x}. \tag{5.24}$$

Following this definition, the standard deviation σ_{Y_1} can be seen as the Wasserstein distance between the distribution of Y_1 and a degenerate distribution concentrated at the mean value μ_{Y_1}. As a result, the Wasserstein distance normalized by the standard deviation can be interpreted as the ratio of the error related to emulating the distribution of Y_1 by that of Y_2, and to using the mean value μ_{Y_1} as a proxy of Y_1.

Because d_{WS} is invariant under translation, the normalized Wasserstein distance is invariant under both translation and scaling; that is,

$$\forall a \in \mathbb{R}\backslash 0, b \in \mathbb{R} \quad \frac{d_{\mathrm{WS}}\left(a\,Y_1 + b, a\,Y_2 + b\right)}{\sigma(a\,Y_1 + b)} = \frac{d_{\mathrm{WS}}\left(Y_1, Y_2\right)}{\sigma(Y_1)}. \tag{5.25}$$

To calculate the expectation in Eq. (5.21), we use Latin hypercube sampling to generate a test set $\mathcal{X}_{\mathrm{test}}$ of size $N_{\mathrm{test}} = 1{,}000$ in the input space. The normalized Wasserstein distance is calculated for each $\boldsymbol{x} \in \mathcal{X}_{\mathrm{test}}$ and then averaged by N_{test}.

For the last two case studies, the analytical response distribution of $Y(\boldsymbol{x})$ is unknown. To characterize it, we repeatedly evaluate the model 10^4 times for $\boldsymbol{x}$. In addition, we also compare some summarizing statistical quantity $b(\boldsymbol{x})$ of the model response $Y(\boldsymbol{x})$, such as the mean $\mathbb{E}\left[Y(\boldsymbol{x})\right]$ or variance $\mathrm{Var}\left[Y(\boldsymbol{x})\right]$, depending on the focus of the application. Note that $b(\boldsymbol{x})$ is a deterministic function of input variables, and we define the normalized mean-squared error by

$$\varepsilon_b = \frac{\sum_{i=1}^{N_{\mathrm{test}}} \left(b_S^{(i)} - \hat{b}^{(i)}\right)^2}{\sum_{i=1}^{N_{\mathrm{test}}} \left(\hat{b}^{(i)} - \bar{\hat{b}}\right)^2}, \text{ with } \bar{\hat{b}} = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \hat{b}^{(i)}, \tag{5.26}$$

where $b_S^{(i)}$ is the value predicted by the surrogate for $\boldsymbol{x}^{(i)} \in \mathcal{X}_{\mathrm{test}}$, and $\hat{b}^{(i)}$ denotes the quantity estimated from 10^4 replicated runs of the original stochastic simulator for $\boldsymbol{x}^{(i)}$. The error ε_b defined in Eq. (5.26) indicates how much of the variance of $b(\boldsymbol{X})$ cannot be explained by $b_S(\boldsymbol{X})$ estimated from surrogate model.

Experimental designs of various size $N \in \{250; 500; 1{,}000; 2{,}000; 4{,}000\}$ are investigated to study the convergence of the proposed method. Each scenario is run 50 times with independent experimental designs to account for statistical uncertainty in the random design for GLaM and KCDE. For GP, N corresponds to the total number of model runs. We repeat 10 times for each value of N (i.e., 10 heteroskedastic Gaussian processes are built using the same number of model runs). As a consequence, error estimates for each N are represented by box plots.

5.5.1 Example 1: a two-dimensional simulator

The first example is the *Black–Scholes* model used for stock prices (McNeil et al., 2005):

$$\mathrm{d}S_t = x_1 S_t \,\mathrm{d}t + x_2 S_t \,\mathrm{d}W_t, \tag{5.27}$$

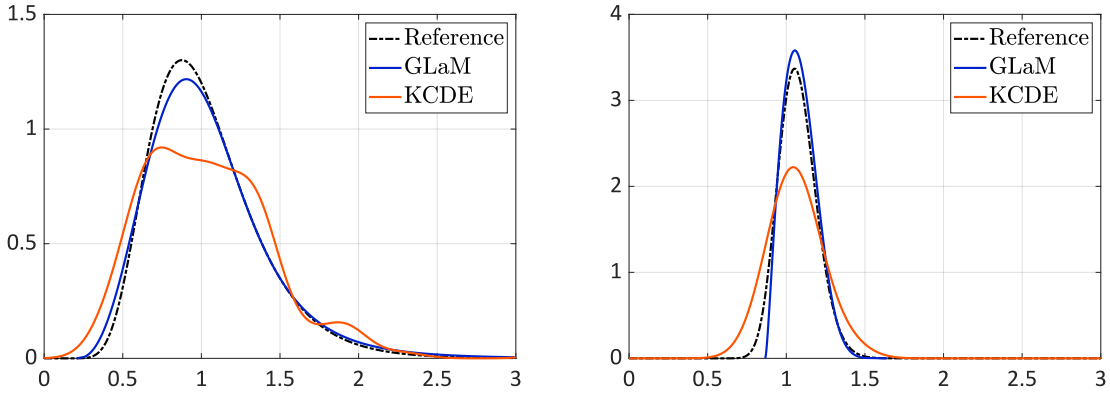
where $\boldsymbol{x} = (x_1, x_2)^\intercal$ are the input parameters, corresponding to the expected return rate and volatility of a stock, respectively. W_t is a standard Wiener process, which represents the source of stochasticity. Eq. (5.27) is a stochastic differential equation whose solution $S_t(\boldsymbol{x})$ is a stochastic process for given parameters $\boldsymbol{x}$. Note that we explicitly express $\boldsymbol{x}$ in $S_t(\boldsymbol{x})$ to emphasize that $\boldsymbol{x}$ are input parameters, but the stochastic equation is defined with respect to time. Without loss of generality, we set the initial condition to $S_0(\boldsymbol{x}) = 1$.

In this example, we are interested in $Y(\boldsymbol{x}) = S_1(\boldsymbol{x})$, which corresponds to the stock value in one year i.e., $t = 1$. We set $X_1 \sim \mathcal{U}(0, 0.1)$ and $X_2 \sim \mathcal{U}(0.1, 0.4)$ to represent the input uncertainty, where the ranges are selected based on parameters calibrated from real data (Reddy and Clinton, 2016).

The solution to Eq. (5.27) can be derived using Itô calculus (Shreve, 2004): $Y(\boldsymbol{x})$ follows a lognormal distribution defined by

$$Y(\boldsymbol{x}) \sim \mathcal{LN}\left(x_1 - \frac{x_2^2}{2}, x_2\right). \tag{5.28}$$

As the distribution of $Y(\boldsymbol{x})$ is known, it is not necessary to simulate the whole process $S_t(\boldsymbol{x})$ with time integration to evaluate $S_1(\boldsymbol{x})$. Instead, we can directly generate samples from the distribution defined in Eq. (5.28).

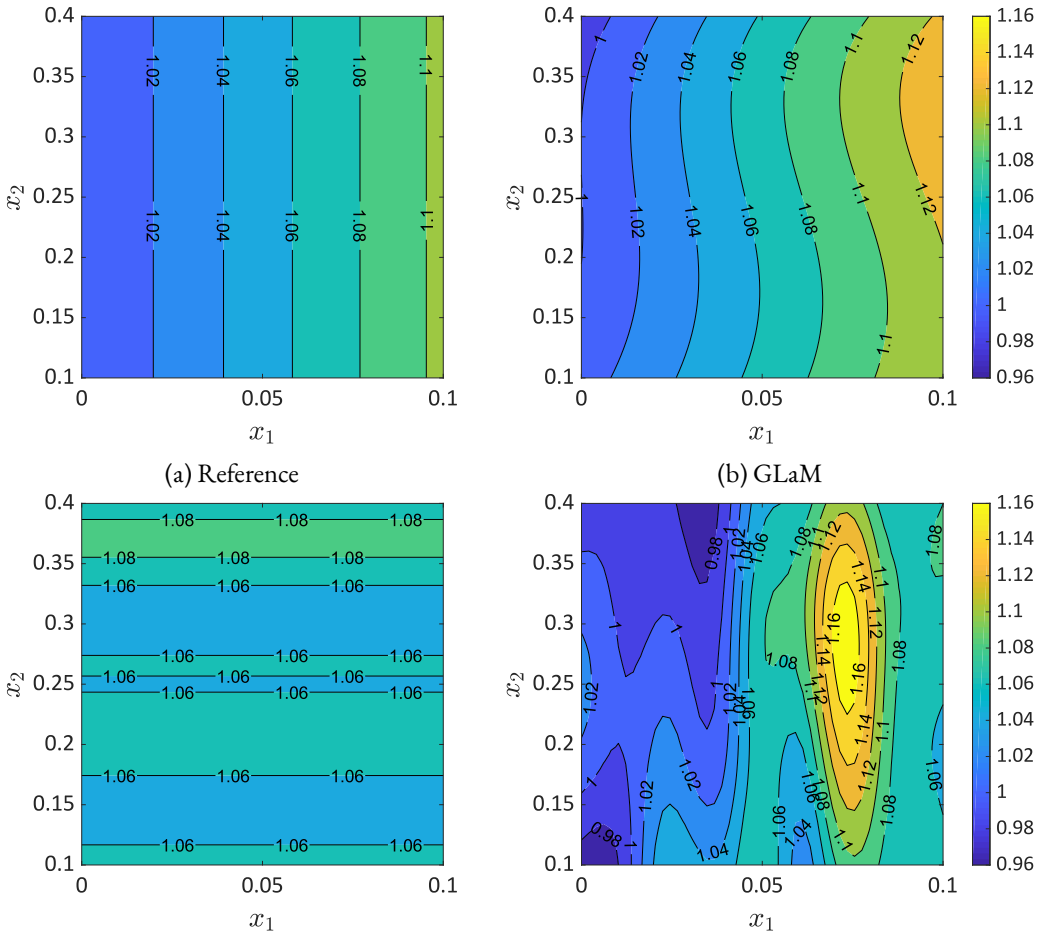


(a) PDF for $\boldsymbol{x} = (0.03, 0.33)^\intercal$ (b) PDF for $\boldsymbol{x} = (0.07, 0.11)^\intercal$

Figure 5.2: Example 1 — Comparisons of the emulated PDF, $N = 500$.

Figure 5.2 shows two PDFs predicted by a GLaM and a KCDE built on an experimental design of size $N = 500$. We observe that with 500 model runs, the KCDE yields PDFs with spurious oscillations and demonstrates relatively poor representation of the bulk. In contrast, the GLaM can better approximate the underlying response PDF in terms of both magnitude and shape variations. Figures 5.3 and 5.4 compare the mean and variance function predicted by the GLaM, KCDE, and GP. The analytical mean function following Eq. (5.28) is $\exp(x_1)$, which only depends on the first variable. The GLaM gives an accurate estimate of the mean function, whereas the KCDE captures a wrong dependence, and GP produces a rather complex structure. For the variance function, the GLaM yields a more detailed trend than the KCDE and GP.

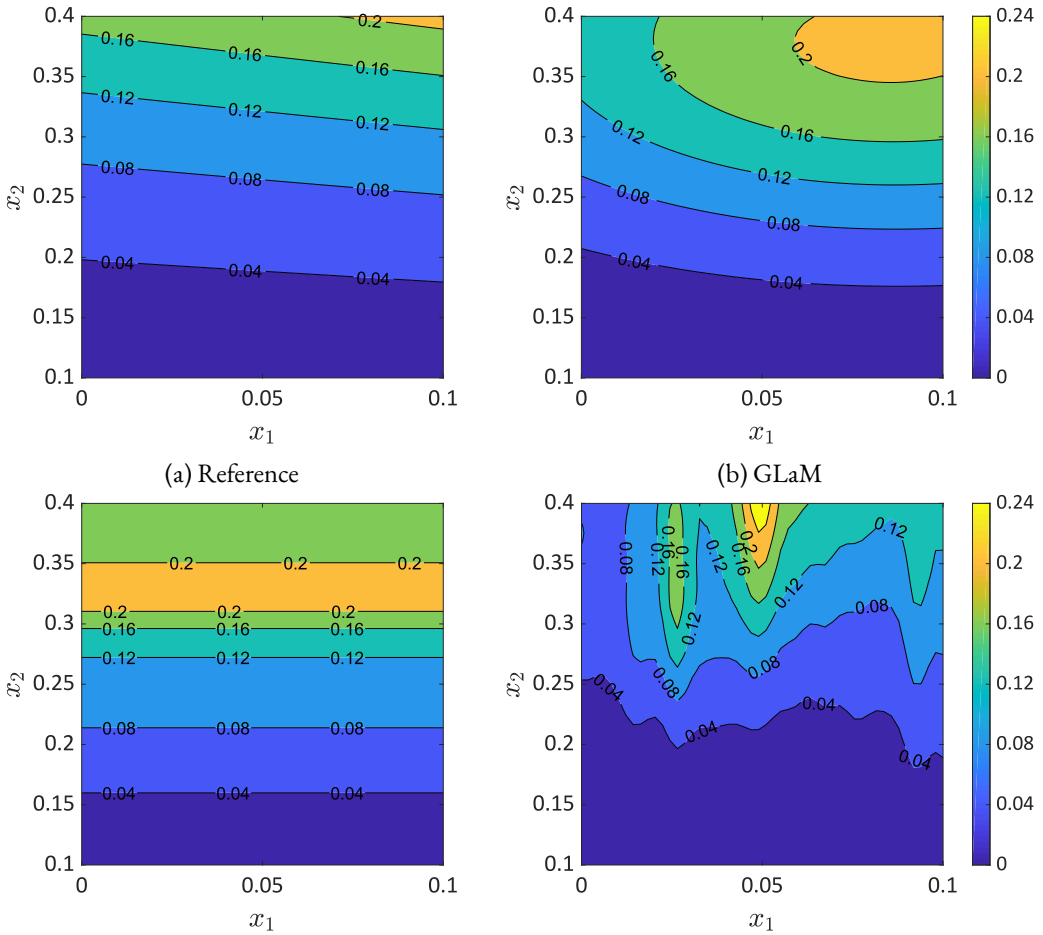
For quantitative comparisons, Figure 5.5 summarizes the error measure Eq. (5.21) with respect to the size



(a) Reference (b) GLaM

(c) KCDE (d) GP

Figure 5.3: Example 1 — Comparisons of the mean function estimation, $N = 500$.



(a) Reference (b) GLaM

(c) KCDE (d) GP

Figure 5.4: Example 1 — Comparisons of the variance function estimation, $N = 500$.

of experimental design. The accuracy of the oracle normal approximation is also reported (black dashed line). This error is only due to model misspecifications because we use the true mean and variance (however, the true response distribution is lognormal). The GP approach performs rather poorly and converges to the oracle normal approximation when the number of points in the experimental design increases. This means that it can accurately estimate the mean and variance functions for large data sets. However, due to the limitation of the Gaussian assumption, GP cannot further decrease the error. The average error of GLaMs built on $N = 500$ model runs are smaller than that of the normal approximation. For $N > 500$, GLaMs clearly provide more accurate results. KCDEs show a slow rate of convergence even in this example of dimension two. In contrast, GLaMs reveal high efficiency with a faster decrease of the errors. In terms of the average error, GLaMs outperform KCDEs for all sizes of experimental design. Furthermore, GLaMs yield an average error near 0.1 for $N = 1{,}000$, which can be hardly achieved by KCDEs even with four times more model runs.

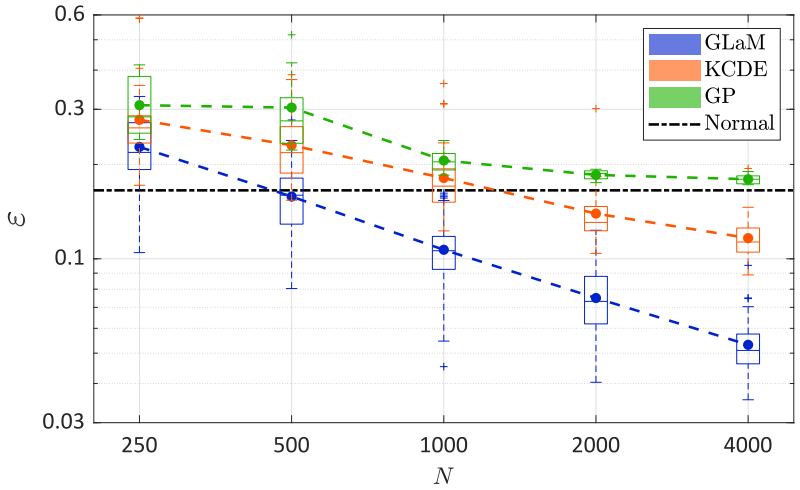


Figure 5.5: Example 1 — Comparison of the convergence between GLaMs and KCDEs in terms of the normalized Wasserstein distance as a function of the size of the experimental design. The dashed lines denote the average value over 50 repetitions of the full analysis. The green box plots and associated dashed lines correspond to the errors of the heteroskedastic Gaussian Process with sequential design (10 repetitions for each size of the experimental design). The black dash-dotted line represents the error of the model assuming that the response distribution is normal with the true mean and variance.

5.5.2 Example 2: a five-dimensional simulator

The second example is given by

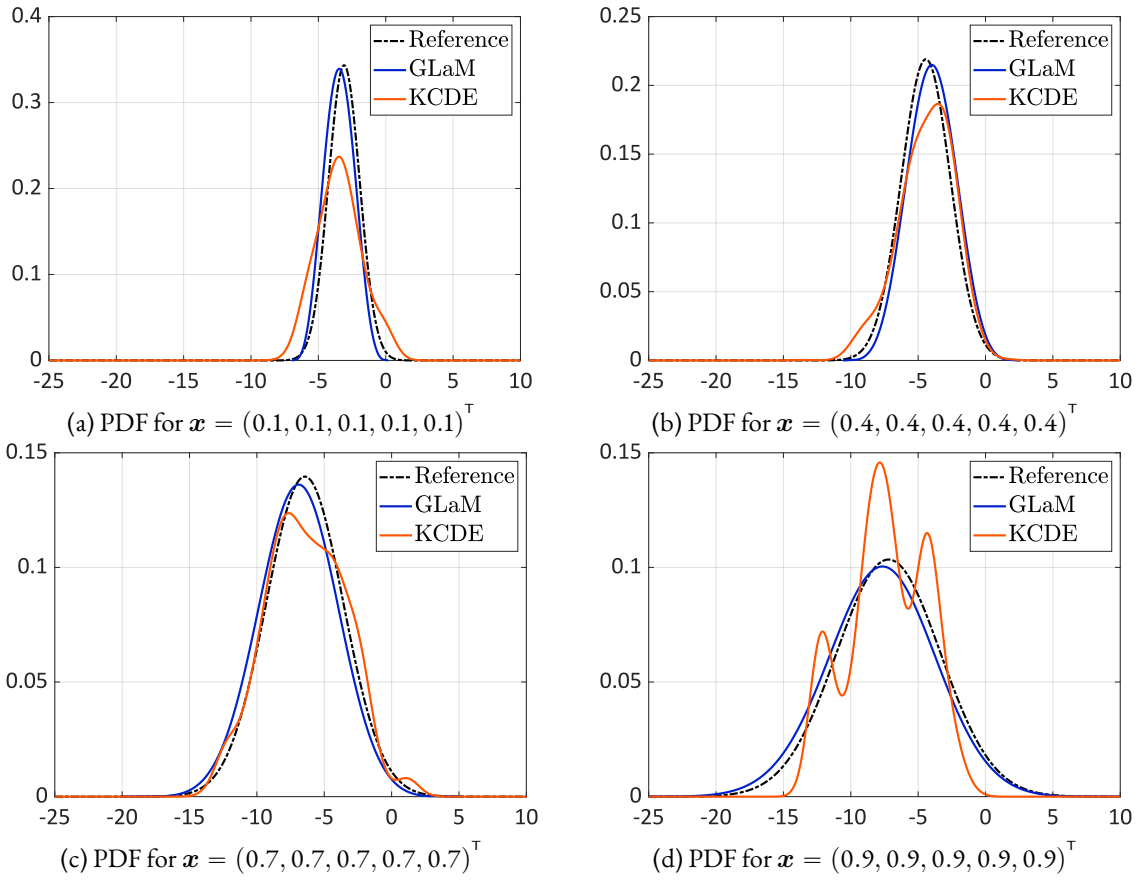
$$Y(\boldsymbol{x}) = \mathcal{M}_s(\boldsymbol{x}, \omega) = \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x}) \cdot Z(\omega), \tag{5.29}$$

where $\boldsymbol{X} \sim \mathcal{U}([0,1]^5)$ are the input variables, and $Z \sim \mathcal{N}(0,1)$ is the latent variable that introduces the stochasticity. The simulator has an input dimension of $M = 5$, which is used to show the performance of the proposed method in a moderate-dimensional problem. By definition, $Y(\boldsymbol{x})$ is a Gaussian random variable with

mean $\mu(\boldsymbol{x})$ and standard deviation $\sigma(\boldsymbol{x})$ which are defined by

$$\begin{aligned}\mu(\boldsymbol{x}) &= 3 - \sum_{j=1}^{5} j\, x_j + \frac{1}{5}\sum_{j=1}^{5} j\, x_j^3 + \frac{1}{15}\sum_{j=1}^{5} j\, \log\left((x_j^2 + x_j^4)\right) + x_1\, x_2^2 - x_5\, x_3 + x_2\, x_4, \\ \sigma(\boldsymbol{x}) &= \exp\left(\frac{1}{10}\sum_{j=1}^{5} j\, x_j\right),\end{aligned} \tag{5.30}$$

Thus, this example has a nonlinear mean function and a strong heteroskedastic effect: the variance varies between 1 and 20.



(a) PDF for $\boldsymbol{x} = (0.1, 0.1, 0.1, 0.1, 0.1)^{\mathsf{T}}$

(b) PDF for $\boldsymbol{x} = (0.4, 0.4, 0.4, 0.4, 0.4)^{\mathsf{T}}$

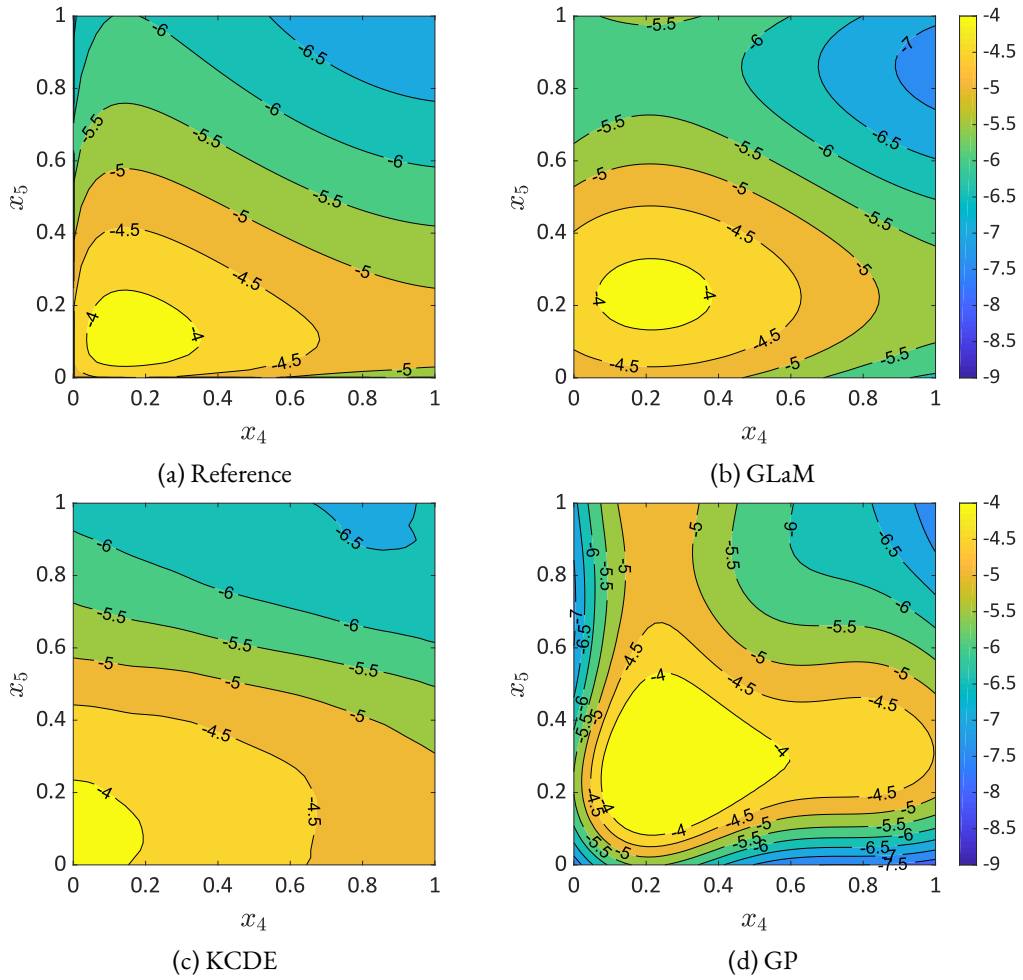
(c) PDF for $\boldsymbol{x} = (0.7, 0.7, 0.7, 0.7, 0.7)^{\mathsf{T}}$

(d) PDF for $\boldsymbol{x} = (0.9, 0.9, 0.9, 0.9, 0.9)^{\mathsf{T}}$

Figure 5.6: Example 2 — Comparisons of the emulated PDF, $N = 1{,}000$. Variance values 1.35, 3.32, 8.17, 14.88 from (a) to (d)

Figure 5.6 compares the model response PDFs (with different variances) for four input values with those predicted by a GLaM and a KCDE built upon 1,000 model runs. The results show that the GLaM correctly identifies the shape of the underlying normal distribution among all possible shapes of the GLD. Moreover, it yields a better approximation to the reference PDF, whereas KCDE tends to "wiggle" in Figure 5.6d (high variance) and overestimate the spread in Figure 5.6a (low variance). Figures 5.7 and 5.8 illustrate the mean and variance function predicted by the GLaM, KCDE, and GP in the $x_4 - x_5$ plan with all the other variables fixed at their expected value. The results show that the GLaM provides more accurate estimates for both functions.

Similar to the first example, we perform a convergence study for $N \in \{250; 500; 1{,}000; 2{,}000; 4{,}000\}$, the results of which are shown in Fig. 5.9. The underlying response distribution is Gaussian, and thus the oracle



(a) Reference (b) GLaM

(c) KCDE (d) GP

Figure 5.7: Example 2 — Comparisons of the mean function estimation in the plan $x_4 - x_5$ with all the other input fixed at their expected value. The surrogate models are fitted to an ED with $N = 1{,}000$.

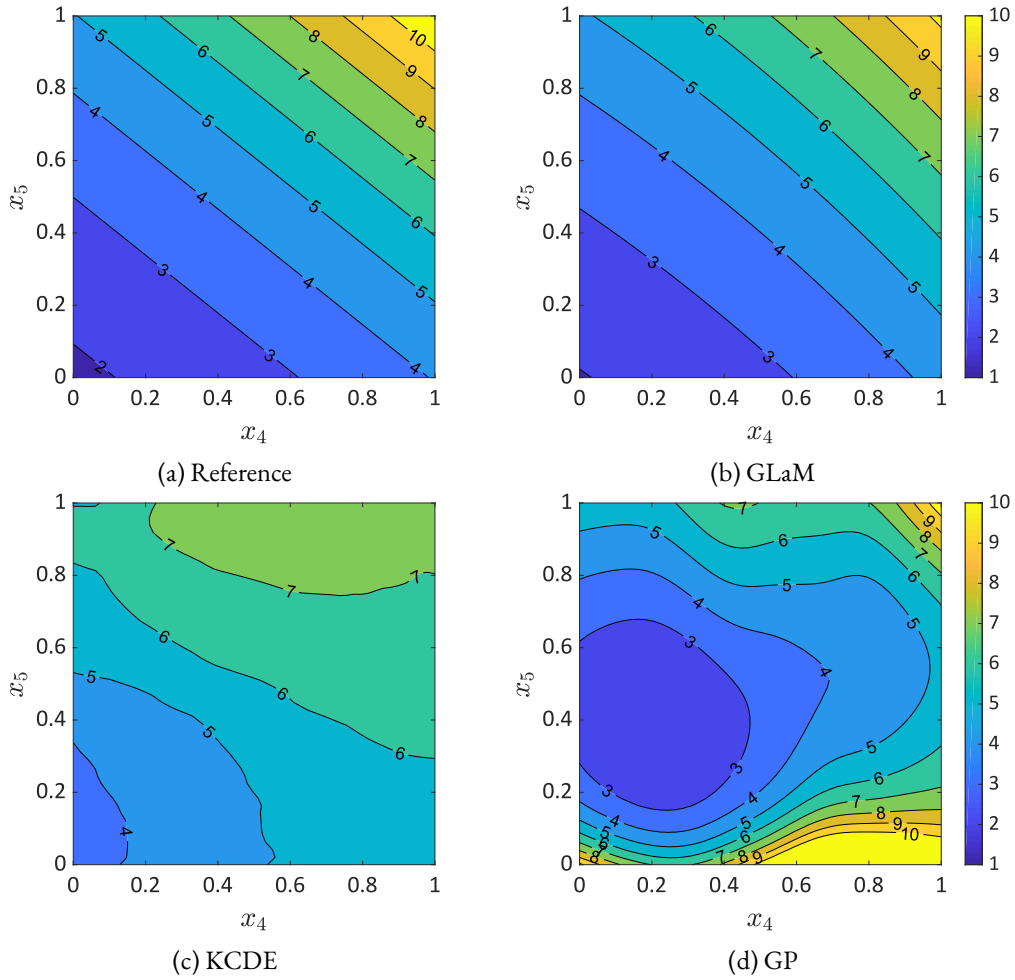


Figure 5.8: Example 2 — Comparisons of the variance function estimation in the plan $x_4 - x_5$ with all the other input fixed at their expected value. The surrogate models are fitted to an ED with $N = 1{,}000$.

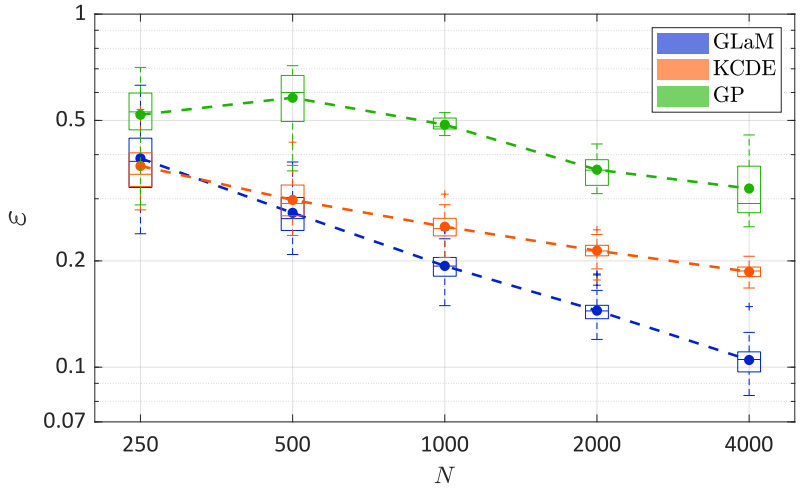


Figure 5.9: Example 2 — Comparison of the convergence between GLaMs and KCDEs in terms of the normalized Wasserstein distance as a function of the size of the experimental design. The dashed lines denote the average value over 50 repetitions of the full analysis. The green box plots and associated dashed lines correspond to the errors of the heteroskedastic Gaussian Process with sequential design (10 repetitions for each size of the experimental design). The "oracle" normal model has an error $\varepsilon = 0$ that is not plotted here.

normal approximation has $\varepsilon = 0$, which is not reported in the figure. Surprisingly, GP gives the worst results. This may be understood as follows: the updating criterion of the sequential design targets at minimizing the *integrated mean-squared error*. The latter mainly focuses on improving the mean estimation (as illustrated in Figs. 5.7 and 5.8), yet both the mean and variance contribute to the Wasserstein distance Eq. (5.23). Also, this example is a five-dimensional problem, which results in more parameters to estimate for GP. In the case of small N, namely $N = 250$, both the GLaMs and KCDEs perform poorly, with the GLaMs showing a similar average error but higher variability. This is explained as follows. Because of the use of AOLS in the modified FGLS procedure, we observe that the total number of coefficients of GLaMs to be estimated varies between 19 to 39 for $N = 250$. Since the GLD is very flexible, a relatively large data set is necessary to provide enough evidence of the underlying PDF shape. Consequently, a small N can lead to overfitting for high-dimensional $\boldsymbol{c}$, but good surrogates can be obtained for more parsimonious models. In contrast, KCDE always performs a thorough leave-one-out cross-validation strategy to select the bandwidths. Therefore, KCDEs show a slightly more stable estimate for $N = 250$. With N increasing, however, GLaMs converge much faster and outperform KCDEs for $N \geqslant 500$ both in terms of the mean and median of the errors. For $N \geqslant 1{,}000$, the average performance of GLaM is even better than the best KCDE model among the 50 repetitions.

In this example of moderate dimensionality, building a GP with sequential design is surprisingly time-consuming, especially for large experimental designs. This is probably due to the sequential design of experiments, which adds new points one by one and updates the surrogate after each enrichment. The associated simulations were performed on the ETH Euler cluster, and the average CPU time varied from 463 seconds for $N = 250$ to over 9 days for $N = 4{,}000$ to build a single GP. For KCDE, it took about 20 CPU seconds for $N = 250$ up to 30 minutes for $N = 4{,}000$ on a standard laptop. In comparison, constructing a GLaM is always on the order of seconds: around 8 seconds for both $N = 250$ and $N = 4{,}000$ on a standard laptop.

5.5.3 Effect of replications

As pointed in Remark 5.1, the proposed method can also work with a data set containing replicates. The latter are simply treated as separate points in the ED. In this section, we analyze the effect of replications using the previous two analytical examples. To this end, we generate data by replicating $R \in \{5; 10; 25; 50\}$ for each set of input parameters in the ED. We keep the total number of simulations the same as nonreplicated cases by reducing the size of the ED accordingly. For instance, a data set of total $N = 1{,}000$ model evaluations with 10 replications consists of 100 different sets of input parameters, each of which is simulated 10 times.

For quantitative comparisons, we investigate a convergence study similar to Sections 5.5.1 and 5.5.2: the total number of runs N varies in $\{250; 500; 1{,}000; 2{,}000; 4{,}000\}$, and each scenario is repeated 50 times.

Figures 5.10 and 5.11 summarize the error defined in Eq. (5.21) averaged over the 50 repetitions for each $R \in \{5; 10; 25; 50\}$. In the first example, replications do not have a strong effect for $R \in \{5; 10; 25\}$. This is because the expansions for $\boldsymbol{\lambda}(\boldsymbol{x})$ contain only a few terms. Therefore, as long as we have enough ED points, exploring the input space and performing replications bring similar improvements to the surrogate accuracy. However, a large number of replications, i.e., $R = 50$, gives too few ED points for small values of N, which yields GLaMs of poor performance.

In the second example, we observe a clear negative effect of replications: for the same total amount of model runs, the surrogate quality deteriorates when increasing the number of replications / decreasing the size of the

experimental design.

In summary, homogeneous replications (i.e., those with the same number of replicates for each point of the experimental design) do not necessarily bring additional accuracy and may even lead to a "waste" of computational budget for the proposed GLaM method. Nevertheless, this does not imply that replications are always useless. On the one hand, for methods that explore the usage of replications, there is a trade-off between replications and exploration (Binois et al., 2019). On the other hand, an adaptive selection of different numbers of replications for each point in the experimental design could possibly improve the performance of the proposed method. However, unlike the heteroskedastic GP, GLaM not only estimates the mean and the variance but also produces the whole PDF. As a result, sequential design strategies for building GLaMs remain to be developed in future study and are outside the scope of the paper.

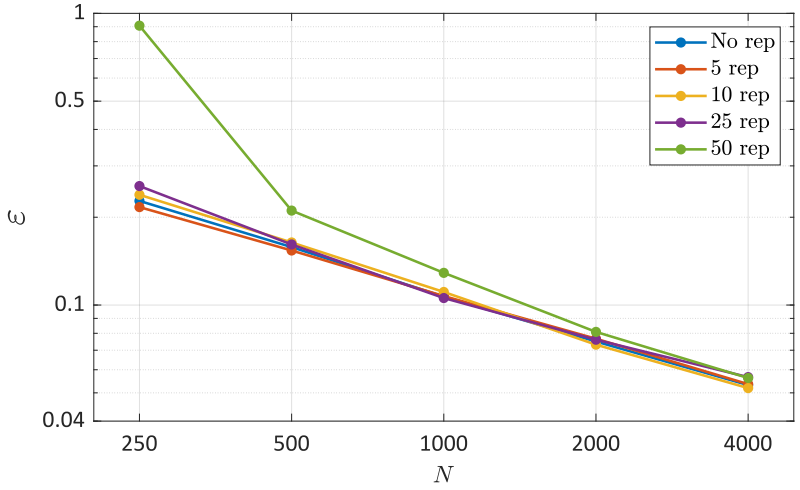


Figure 5.10: Example 1 — Comparison of the GLaMs built on data with different number of replications. The curves corresponds to the mean error over the 50 repetitions.

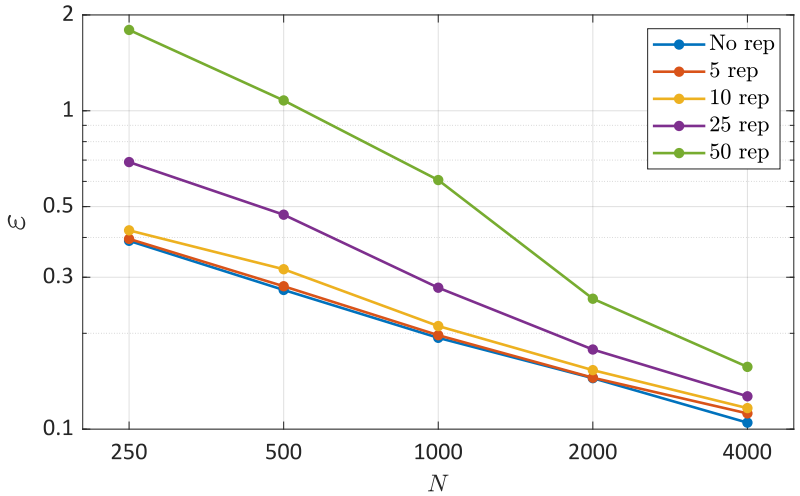


Figure 5.11: Example 2 — Comparison of the GLaMs built on data with different number of replications. The curves corresponds to the mean error over the 50 repetitions.

5.5.4 Example 3: Asian options

In this third example, we apply the proposed method to a financial case study, namely an *Asian option* (Kemna and Vorst, 1990). Such an option, a.k.a. average value option, is a derivative contract, the payoff of which is contingent on the average price of the underlying asset over a certain fixed time period. Due to the path-dependent nature, an Asian option has complex behavior, and its valuation is not straightforward, as opposed to European options.

Recall the Black–Scholes model defined in Eq. (5.27) that represents the evolution of a stock price $S_t(\boldsymbol{x})$. Instead of relying on the stock price on the maturity date $t = T$, the payoff of an Asian call option reads

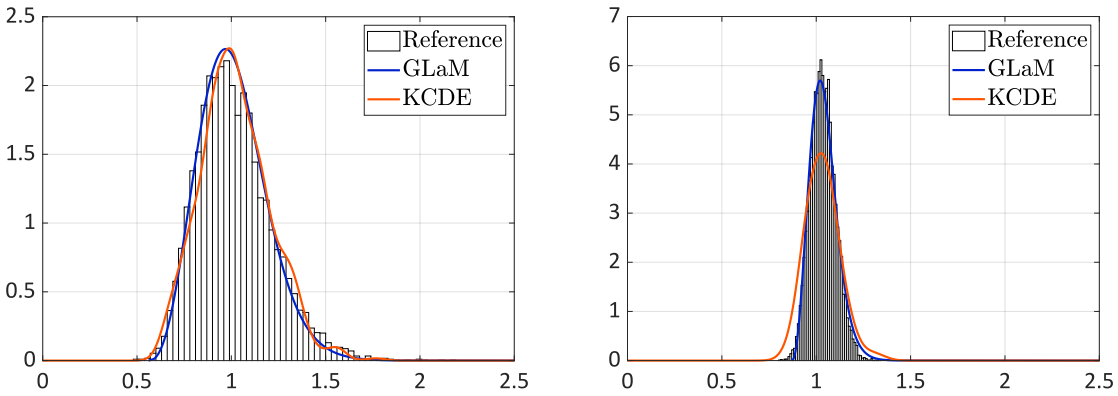
$$C(\boldsymbol{x}) = \max\left\{A_T(\boldsymbol{x}) - K, 0\right\}, \text{ with } A_t(\boldsymbol{x}) = \frac{1}{t}\int_0^t S_u(\boldsymbol{x})\mathrm{d}u. \tag{5.31}$$

where $A_t(\boldsymbol{x})$ is called the *continuous average process*, and K denotes the *strike price*. Because $A_T(\boldsymbol{x})$ plays an important role in the Asian option modeling Eq. (5.31), the PDF of $A_T(\boldsymbol{x})$ is of interest in this case study. As in Section 5.5.1, we set $T = 1$, which corresponds to a one-year inspection period. We choose $X_1 \sim \mathcal{U}(0, 0.1)$ and $X_2 \sim \mathcal{U}(0.1, 0.4)$ for the two input random variables. Unlike $S_1(\boldsymbol{x})$, the distribution of $A_1(\boldsymbol{x})$ cannot be derived analytically. It is necessary to simulate the trajectory of $S_t(\boldsymbol{x})$ to compute $A_1(\boldsymbol{x})$. Based on the Markovian and lognormal properties of $S_t(\boldsymbol{x})$, we apply the following recursive equations for the path simulation with a time step $\Delta t = 0.001$:

$$\begin{aligned} S_0(\boldsymbol{x}) &= 1, \\ S_{t+\Delta t}(\boldsymbol{x}) \mid S_t(\boldsymbol{x}) &\sim \mathcal{LN}\left(\log\left(S_t(\boldsymbol{x})\right) + \left(x_1 - \frac{x_2^2}{2}\right)\Delta t, x_2\sqrt{\Delta t}\right). \end{aligned}$$

Finally, the continuous average defined in Eq. (5.31) is approximated by the arithmetic mean, that is,

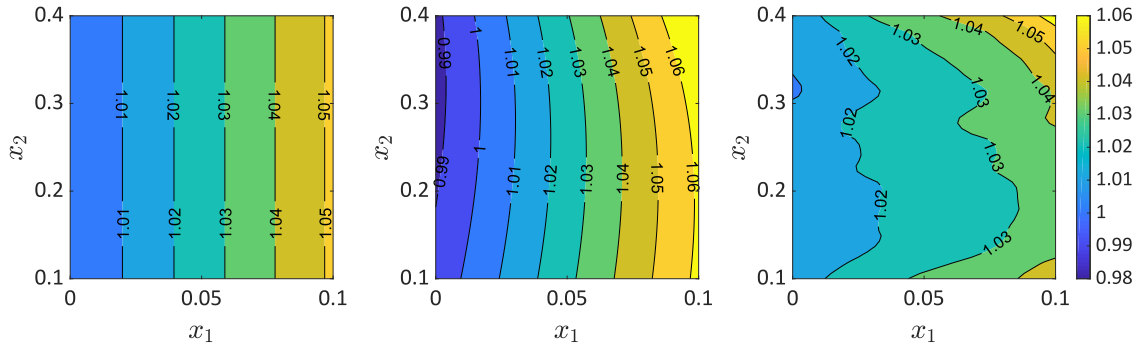
$$A_1(\boldsymbol{x}) = \frac{\sum_{k=1}^{1{,}000} S_{k\Delta t}(\boldsymbol{x})}{1{,}000}$$



(a) PDF for $\boldsymbol{x} = (0.03, 0.33)^\intercal$ (b) PDF for $\boldsymbol{x} = (0.07, 0.11)^\intercal$

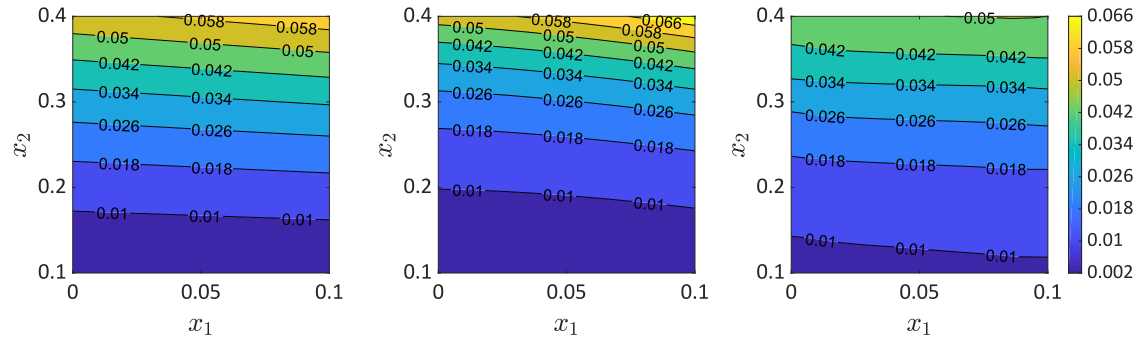
Figure 5.12: Asian option — Comparisons of the emulated PDF, $N = 500$

Figure 5.12 shows two response PDFs predicted by the two surrogate models constructed on an experimen-



(a) Reference (b) GLaM (c) KCDE

Figure 5.13: Asian option — Comparisons of the mean function estimation, $N = 500$.



(a) Reference (b) GLaM (c) KCDE

Figure 5.14: Asian option — Comparisons of the variance function estimation, $N = 500$.

tal design of $N = 500$. The reference histograms are calculated from 10^4 repeated runs of the simulator for each set of input parameters. We observe that the KCDE exhibits slight fluctuations at the right tail for high volatility (in Figure 5.12a) and does not well approximate the bulk of the response distribution for low volatility (in Figure 5.12b). In comparison, the GLaM can well represent the PDF shape in both cases and also more accurately approximates the tails. Figures 5.13 and 5.14 shows the mean and variance function, where the reference values can be obtained by applying Itô's calculus. For the experimental design of $N = 500$, the GLaM more accurately predicts the two functions. Finally, quantitative comparisons in Figure 5.15 confirm the superiority of GLaMs to KCDEs: GLaMs yield smaller average error for all $N \in \{250; 500; 1{,}000; 2{,}000; 4{,}000\}$ and demonstrate a better convergence rate. Moreover, for large experimental designs ($N \geqslant 2{,}000$), the average error of GLaMs is nearly half of that of KCDEs. The oracle Gaussian approximation in this case study has a similar error to GLaMs built on 1,000 model runs. For $N \geqslant 2{,}000$, GLaMs fitted from data are much more accurate than the best possible Gaussian-type mean-variance model.

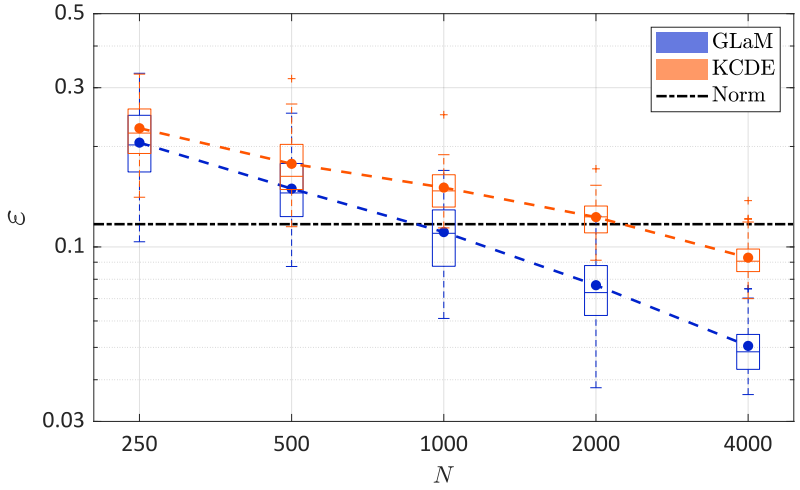


Figure 5.15: Asian option, average process $A_1(\boldsymbol{x})$ at $T = 1$ year — Comparison of the convergence of GLaMs and KCDEs in terms of the normalized Wasserstein distance as a function of the size of the experimental design. The dashed lines denote the average value over 50 repetitions of the full analysis. The black dash-dotted line represents the error of the model assuming that the response distribution is normal with the true mean and variance

As a second quantity of interest, we consider the expected payoff $\mu_C(\boldsymbol{x}) = \mathbb{E}[C(\boldsymbol{x})]$. This quantity not only is important for making investment decisions but also has a very similar form to the option price (Kemna and Vorst, 1990). The definition Eq. (5.31) implies that the payoff $C(\boldsymbol{x})$ is a mixed random variable, which has a probability mass at 0 and a continuous PDF on the positive line depending on the strike price K. In the following analysis, K is set to 1.

For GLaMs, $\mu_C(\boldsymbol{x})$ can be calculated by

$$\mu_C(\boldsymbol{x}) = \left(\lambda_1 - \frac{1}{\lambda_2\lambda_3} + \frac{1}{\lambda_2\lambda_4} - K\right)(1 - u_K) + \frac{1}{\lambda_2}\left(\frac{1 - u_K^{\lambda_3+1}}{\lambda_3\,(\lambda_3+1)} - \frac{(1-u_K)^{\lambda_4+1}}{\lambda_4\,(\lambda_4+1)}\right) \tag{5.32}$$

where λ's are the distribution parameters at $\boldsymbol{x}$, and u_K is the solution of the nonlinear equation

$$Q(u_K; \boldsymbol{\lambda}) = K. \tag{5.33}$$

with Q being the quantile function defined in Eq. (5.2).

Figure 5.16 shows the convergence of estimations of $\mu_C(\boldsymbol{x})$ in terms of the error defined in Eq. (5.26). The difference between the performance of GLaMs and KCDEs is not as significant as for the distribution estimation of $A_1(\boldsymbol{x})$ in Figure 5.15. For relatively small data sets, namely $N \leqslant 500$, both models work poorly: they are only able to explain on average no more than 70% of the variance of $\mu_C(\boldsymbol{X})$. In addition, GLaMs demonstrate a higher variability of the errors. For larger experimental designs $N \geqslant 2{,}000$, however, the performance of GLaMs improves significantly more than that of KCDEs. For $N = 4{,}000$, the average error of GLaMs is twice smaller than that of KCDEs, and the smallest error achieved by GLaMs is one order of magnitude smaller than the best KCDE.

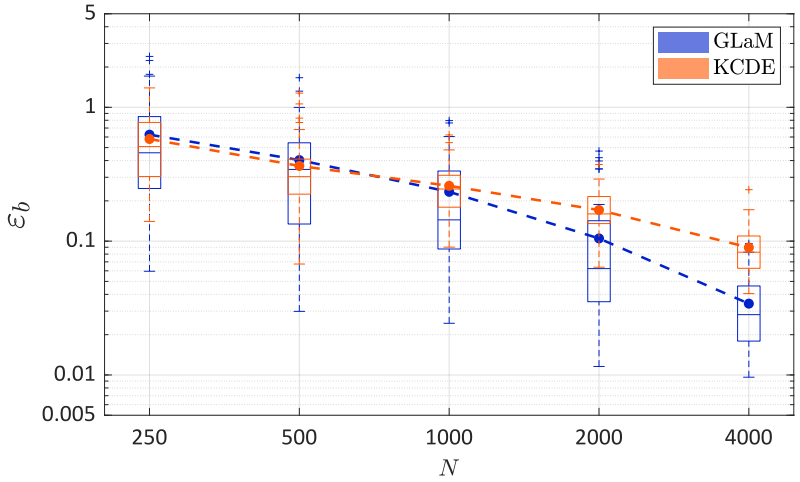


Figure 5.16: Asian option, expected payoff estimations — Comparison of the convergence of GLaMs and KCDEs in terms of the normalized mean squared error as a function of the size of the experimental design. The dashed lines denote the average value over 50 repetitions of the full analysis.

5.5.5 Example 4: Stochastic SIR model

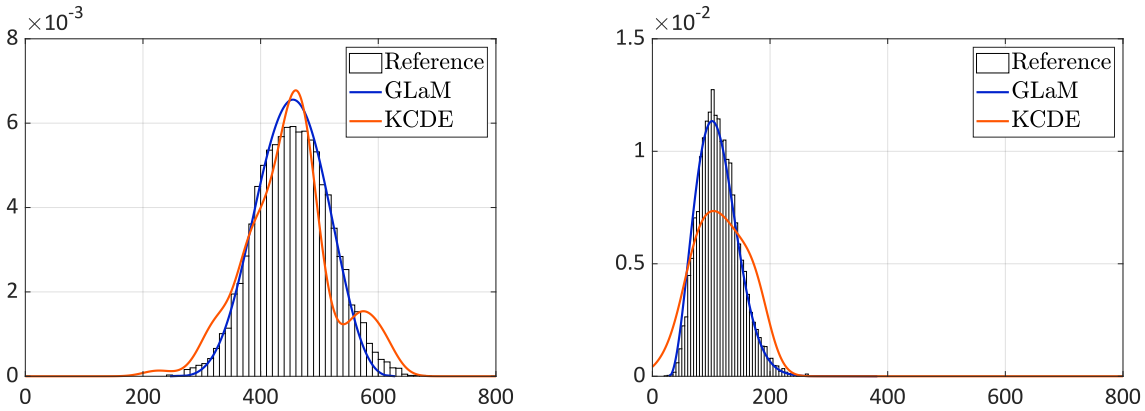
In this fourth example, we apply the proposed method to a *stochastic susceptible-infected-recovered* (SIR) model in epidemiology (Britton, 2010). This model simulates the spread of an infectious disease, which can help find appropriate epidemiological interventions to minimize social and ethical impacts during the outbreak.

According to the standard SIR model, at time t a population of size P_t contains three groups of individuals: susceptible, infected, and recovered, the counts of which are denoted by S_t, I_t, and R_t, respectively. These three quantities fully characterize a population configuration at time t. Among the three groups, only susceptible individuals can get infected due to close contact with infected individuals, whereas an infected person can recover and becomes immune to future infections. We consider a fixed population without newborns and deaths, i.e., the total population size is constant, $P_t = P$. As a result, S_t, I_t, and R_t satisfy the constraint $S_t + I_t + R_t = P$, and only the time evolution of (S_t, I_t) is necessary to characterize the spread of a disease.

To account for random recoveries and interactions among individuals, stochastic SIR models are usually preferred to represent the epidemic evolution. Without going into details, the model dynamics is briefly summarized as follows. The pair (I_t, S_t) evolves as a continuous-time Markov process following mutual transition rates β and γ, which denote the contact rate and recovery rate, respectively. The epidemic stops at time $t = T$

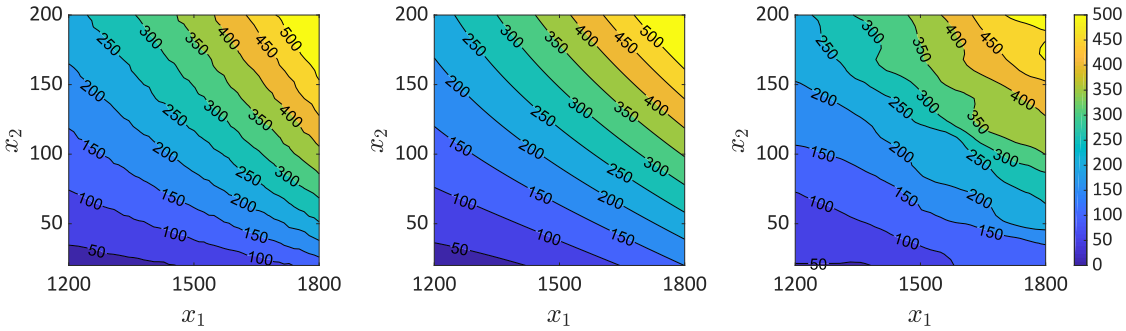
where $I_T = 0$, indicating that no further infections can occur. The evolution process is simulated by the *Gillespie algorithm* (Gillespie, 1977). The reader is referred to Britton (2010) for a more detailed presentation of stochastic SIR models.

In this case study, we set the total population equal to $P = 2{,}000$ and $\beta = \gamma = 0.5$ as in Binois et al. (2018). The initial configuration $\boldsymbol{x} = (S_0, I_0)$ is the vector of input parameters. To account for different scenarios, the input variables $\boldsymbol{X}$ are modeled as $X_1 \sim \mathcal{U}(1200, 1800)$ (initial number of susceptible individuals) and $X_2 \sim \mathcal{U}(20, 200)$ (initial number of infected individuals). The QoI is the total number of newly infected individuals during the outbreak, i.e., $Y(\boldsymbol{x}) = S_T - S_0$.



(a) PDF for $\boldsymbol{x} = (1714, 165)^\intercal$ (b) PDF for $\boldsymbol{x} = (1364, 61)^\intercal$

Figure 5.17: SIR model — Comparisons of the emulated PDF, $N = 500$

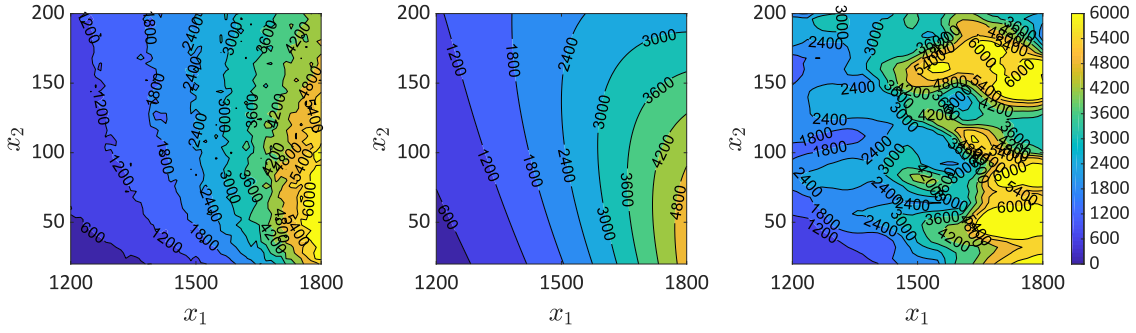


(a) Reference (b) GLaM (c) KCDE

Figure 5.18: SIR model — Comparisons of the mean function estimation in the plan $N = 500$.

Figure 5.17 compares two response PDFs estimated by a GLaM and by a KCDE for two sets of initial configurations, using an experimental design of size $N = 500$. The reference histograms are obtained by 10^4 repeated model runs for each $\boldsymbol{x}$. We observe that the PDF shape varies: it changes from symmetric to slightly right-skewed distributions depending on the input variables. The GLaM is able to accurately capture this shape variation, while KCDE exhibits relatively poor shape representations.

Figures 5.18 and 5.19 illustrate the mean and variance function. Because the analytical results are unknown for this simulator, we use 1,000 replications to estimate these quantities for plotting. We observe that both functions vary nonlinearly in the input space. Compared with the KCDE, the GLaM is able to capture the trend of the two functions and provides more accurate estimates. More detailed comparisons of the surrogate models



(a) Reference (b) GLaM (c) KCDE

Figure 5.19: SIR model — Comparisons of the variance function estimation, $N = 500$.

are shown in Figure 5.20. The error of the oracle Gaussian approximation is quite small. This implies that the response distribution for most of the input parameters in the input space is close to a Gaussian distribution. Nevertheless, GLaMs built on $N = 4{,}000$ model runs still demonstrate better average behavior. For all sizes of experimental design, GLaMs clearly outperform KCDEs. For $N \geqslant 500$, the biggest error of GLaMs is smaller than the smallest error of KCDEs among the 50 repetitions. Finally, to achieve the same accuracy as GLaMs, KCDEs require around 7 times more model runs.

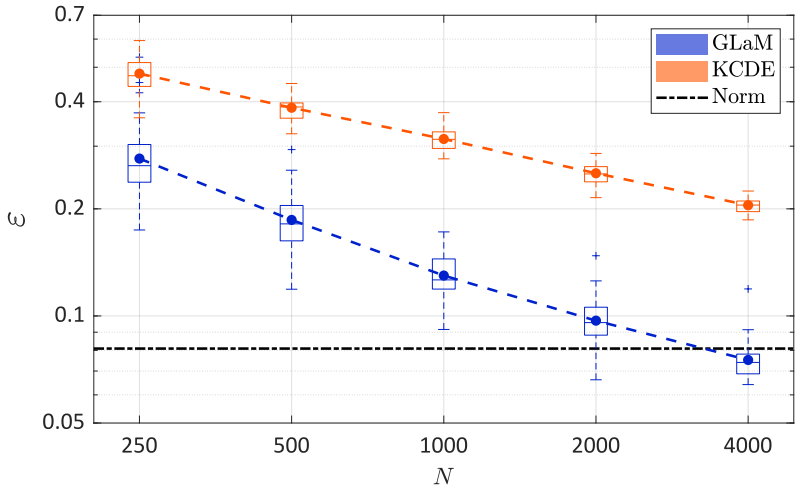


Figure 5.20: SIR model — Comparison of the convergence between GLaMs and KCDEs in terms of the normalized Wasserstein distance as a function of the size of the experimental design. The dashed line denotes the average value over 50 repetitions of the full analysis. The black dash-dotted line represents the error of the model assuming that the response distribution is normal with the true mean and variance

In epidemiological management, the expected value $\mu(\boldsymbol{x}) = \mathbb{E}[Y(\boldsymbol{x})]$ is crucial for decision making (Merl et al., 2009). Therefore, we investigate the accuracy of $\mu(\boldsymbol{x})$ estimations, and the results are in Figure 5.21. First, both GLaM and KCDE can explain more than 90% of the variance in $\mu(\boldsymbol{X})$ for $N = 250$, which implies an overall accurate approximation to the mean function. With increasing N, GLaM shows a more rapid decay of the error. Furthermore, GLaMs built on $N = 1{,}000$ have a similar (or even slightly better) performance to KCDEs with $N = 4{,}000$.

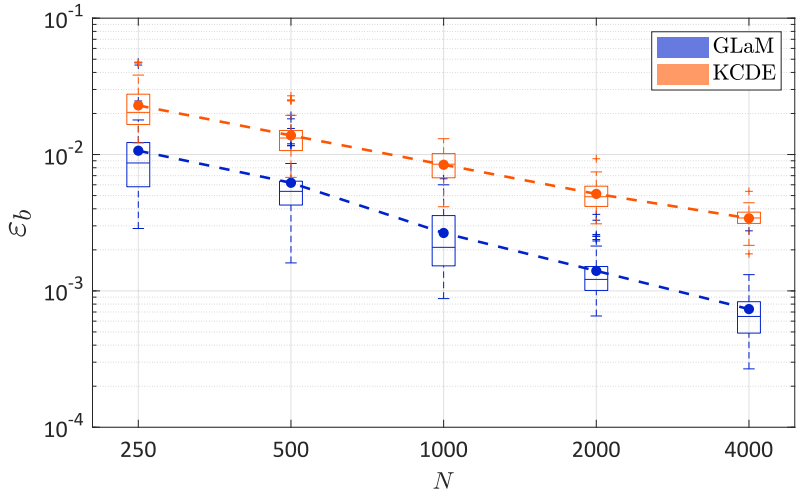


Figure 5.21: SIR model, mean value estimations — Comparison of the convergence between GLaMs and KCDEs in terms of the normalized mean-squared error as a function of the size of the experimental design. The dashed line denotes the average value over 50 repetitions of the full analysis.

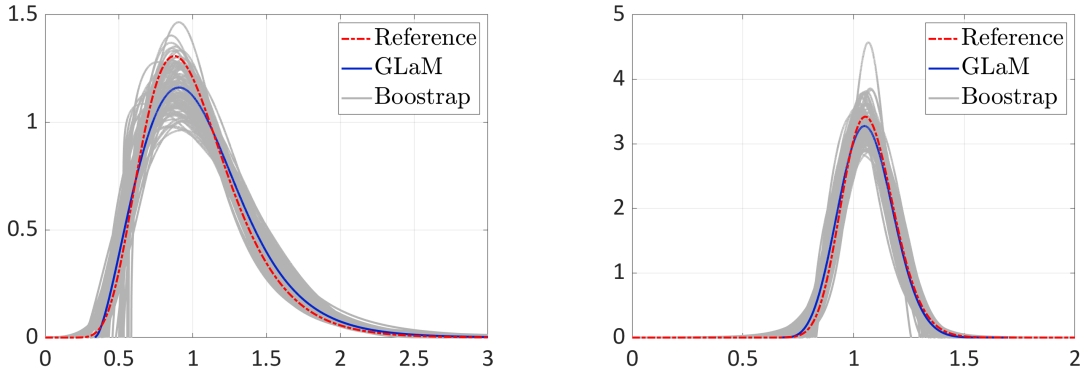
5.6 Conclusions

This paper presents an efficient and accurate nonintrusive surrogate modeling method for stochastic simulators that does not require replicated runs of the latter. We follow the setting of Zhu and Sudret (2020), where the generalized lambda distribution is used to flexibly approximate the response probability density function. The distribution parameters, as functions of the input variables, are approximated by polynomial chaos expansions. In this paper, however, we do not require replicated runs of the stochastic simulator, which provides a more general and versatile approach. We propose the maximum conditional likelihood estimator to construct such a model for given basis functions. This estimation method is shown to be consistent and applicable to data with or without replications. In addition, we modify the feasible generalized least-squares algorithm to select suitable truncation schemes for the distribution parameters, which also provides a good starting point for the subsequent optimization of the likelihood function.

The performance of the new method is illustrated on analytical examples and case studies in mathematical finance and epidemics. The results show that with a reasonable number of model runs, the developed algorithm can produce surrogate models that accurately approximate the response probability density function and capture the shape variations of the latter with $\boldsymbol{x}$. Considering the normalized Wasserstein distance as an error metric, generalized lambda models always show a better convergence rate than the nonparametric kernel conditional density estimator with adaptive bandwidth selections (from the package `np` in R). Furthermore, the proposed method generally yields more reliable estimates of certain important quantities.

Quantifying the uncertainty of surrogate models that emulate the entire response distribution of a stochastic simulator remains to be developed in future work, especially when no or only a few replications are available. One possibility is to use cross-validation to calculate the expected loss. However, when the log-likelihood is used as the loss function such as Eq. (5.17), the resulting score is not intuitive and is difficult to interpret. Alternatively, with a given basis for $\boldsymbol{\lambda}(\boldsymbol{x})$ in GLaMs, one can use bootstrap (Efron, 1982) to assess the uncertainty in the estimation of the coefficients. Figure 5.22 illustrates the PDF predictions of 100 bootstrapping GLaMs of

a data set with $N = 500$ of Example 1. Note that the associated theoretical aspects remain to be developed: it is necessary to prove the *bootstrap consistency*, which is usually achieved by showing the asymptotic normality of the estimator. As a result, the asymptotic properties of the maximum likelihood estimator in Eq. (5.17) need to be further investigated.



(a) PDF for $\boldsymbol{x} = (0.03, 0.33)^{\intercal}$ (b) PDF for $\boldsymbol{x} = (0.07, 0.11)^{\intercal}$

Figure 5.22: Example 1 — Uncertainty on the PDF predicted by GLaM for two values of the input parameters, using an experimental design of $N = 500$. The blue line is the PDF predicted by GLaM from the 500 data points. The grey lines correspond to 100 PDFs generated by GLaM using bootstrapped experimental designs.

Possible interesting applications of the proposed method to be investigated in future studies include reliability analysis and sensitivity analysis (Zhu and Sudret, 2021). To improve the performance of the generalized lambda surrogate model for small data sets, we plan to develop algorithms that select only important basis functions based on appropriate model selection criteria. Finally, since the generalized lambda distribution cannot represent multimodal distributions, potential extensions to mixtures of generalized lambda distributions may provide a more flexible surrogate for simulators with multimodal response distribution (Fadikar et al., 2018).

Acknowledgments

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524), whose support is gratefully acknowledged.

5.A Appendix

5.A.1 Consistency of the maximum likelihood estimator

In this section, we prove the consistency of the maximum likelihood estimator, as described in Theorem 5.1. For the ease of derivation, we introduce the following notation:

$$q_{\boldsymbol{c}}(\boldsymbol{x}, y) = f_{Y|\boldsymbol{X}}\left(y \middle| \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{c})\right), \quad p_{\boldsymbol{c}}(\boldsymbol{x}, y) = f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) = f_{\boldsymbol{X}}(\boldsymbol{x}) q_{\boldsymbol{c}}(\boldsymbol{x}, y),$$

where $q_{\boldsymbol{c}}$ denotes the conditional PDF with model parameters $\boldsymbol{c}$, and $p_{\boldsymbol{c}}$ corresponds to the associated joint PDF. Under this setting, we assume that the true distribution q_0 belongs to the family for a particular set of coefficients $\boldsymbol{c}_0$, i.e., $q_0 = q_{\boldsymbol{c}_0}$ and $p_0 = p_{\boldsymbol{c}_0}$. We denote the probability measure of the probability space of $(\boldsymbol{X}, Y)$ by P_0 and the Lebesgue measure by μ.

The maximum likelihood estimation defined in Eq. (5.16) belongs to the generalized method of moments (GMM; Hansen, 1982) for which we define the *loss function* by

$$\ell_{\boldsymbol{c}}(\boldsymbol{x}, y) = -\log\left(q_{\boldsymbol{c}}(\boldsymbol{x}, y)\right) \mathbb{1}_{q_0(\boldsymbol{x},y)>0}(\boldsymbol{x}, y). \tag{5.34}$$

It holds that

$$\boldsymbol{c}_0 = \arg\min_{\boldsymbol{c}} l(\boldsymbol{c}), \text{ where } l(\boldsymbol{c}) = \mathbb{E}\left[\ell_{\boldsymbol{c}}(\boldsymbol{X}, Y)\right].$$

The maximum likelihood estimator is then defined by

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} l_n(\boldsymbol{c}), \text{ where } l_n(\boldsymbol{c}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\boldsymbol{c}}\left(\boldsymbol{X}^{(i)}, Y^{(i)}\right),$$

where l_n is the empirical version of l.

To prove the consistency of a GMM estimator, the *uniform law of large numbers* is usually used. In the case of a maximum likelihood estimator for the generalized lambda model, classical methods (Newey and McFadden, 1994) to prove the uniform law of large numbers cannot be applied directly, due to the fact that the support of $q_{\boldsymbol{c}}$ can depend on the model parameters $\boldsymbol{c}$, as shown in Eq. (5.4). To circumvent this problem, we use the techniques suggested by van de Geer (2000) for the proof.

Lemma 5.1. *Under the conditions described in Theorem 5.1, we have the following:*

(i) Boundedness: $\sup_{\boldsymbol{c}\in\mathcal{C}} q_{\boldsymbol{c}}(\boldsymbol{x}, y) < +\infty$.

(ii) Continuity: $\forall\, \tilde{\boldsymbol{c}} \in \mathcal{C}$, *the map* $\boldsymbol{c} \mapsto q_{\boldsymbol{c}}$ *is continuous at* $\tilde{\boldsymbol{c}}$ *for* μ*-almost all* $(\boldsymbol{x}, y) \in \mathcal{D}_{\boldsymbol{x}} \times \mathbb{R}$.

Proof. (i) As the conditions of Theorem 5.1 indicate that $\mathcal{D}_{\boldsymbol{X}}$ and $\mathcal{C}$ are compact, the two sets are bounded according to the *Heine–Borel theorem*. Hence, the value of $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{c})$ is also bounded. We denote respectively $\{\overline{C}_i, i = 1, \ldots, 4\}$ and $\{\underline{C}_i, i = 1, \ldots, 4\}$ as the upper and lower bounds for each component of $\boldsymbol{\lambda}$:

$$\underline{C}_i \leqslant \lambda_i \leqslant \overline{C}_i \quad \forall i = 1, \ldots, 4. \tag{5.35}$$

In addition, Eq. (5.15) guarantees that $\lambda_2^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{c})$ is bounded away from 0, i.e., $\underline{C}_2 > 0$. Consider now Eq. (5.3) to evaluate the PDF of GLDs. If u in Eq. (5.3) does not exist in $[0, 1]$, $q_{\boldsymbol{c}} = 0$ and thus bounded. For $u \in [0, 1]$, we have

$$\frac{\lambda_2}{u^{\lambda_3-1} + (1-u)^{\lambda_4-1}} \leqslant \frac{\overline{C}_2}{u^{\overline{k}} + (1-u)^{\overline{k}}}, \tag{5.36}$$

where

$$\overline{k} = \max\left\{\overline{C}_3 - 1, \overline{C}_4 - 1\right\}.$$

Define the function $m(u) = u^{\overline{k}} + (1-u)^{\overline{k}}$, which corresponds to the denominator of Eq. (5.36). For $\overline{k} = 0$ and 1, $m(u)$ is a constant function equal to 2 and 1, respectively. If $k \neq 0, 1$, the derivative $m'(u) =$

$\overline{k}\left(u^{\overline{k}-1}-(1-u)^{\overline{k}-1}\right)$ is equal to 0 only at $u=0.5$ in $[0,1]$. As a result, $\min m(u)=\min\{m(0),m(0.5),m(1)\}$. For $\overline{k}<0$, $\min m(u)=m(0.5)=2^{1-\overline{k}}$. While for $\overline{k}>0$, $\min m(u)=\min\{m(0),m(0.5),m(1)\}=\min\left\{1,2^{1-\overline{k}}\right\}$. Hence, we have $\min m(u)\geqslant\min\left\{1,2^{1-\overline{k}}\right\}=C_m$. Taking this property into account, Eq. (5.36) becomes

$$\frac{\lambda_2}{u^{\lambda_3-1}+(1-u)^{\lambda_4-1}}\leqslant\frac{\overline{C}_2}{C_m}=C_q. \tag{5.37}$$

Therefore, $\sup_{\boldsymbol{c}\in\mathcal{C}}q_{\boldsymbol{c}}(\boldsymbol{x},y)\leqslant C_q$.

(ii) Next, we prove the continuity. For any $\tilde{\boldsymbol{c}}\in\mathcal{C}$, we classify the points $(\boldsymbol{x},y)\in\mathcal{D}_{\boldsymbol{x}}\times\mathbb{R}$ into three groups based on their corresponding latent variable $\tilde{u}$: (1) $\tilde{u}\in(0,1)$, (2) $\tilde{u}$ does not exist within $[0,1]$, and (3) $\tilde{u}=0$ or 1.

For $(\boldsymbol{x},y)$ in the first class, y is an interior point of the support of the conditional distribution $q_{\tilde{\boldsymbol{c}}}(\boldsymbol{x},\cdot)$. Thereby, the following equation holds:

$$y=Q(\tilde{u};\tilde{\boldsymbol{\lambda}})=\tilde{\lambda}_1+\frac{1}{\tilde{\lambda}_2}\left(\frac{\tilde{u}^{\tilde{\lambda}_3}-1}{\tilde{\lambda}_3}-\frac{(1-\tilde{u})^{\tilde{\lambda}_4}-1}{\tilde{\lambda}_4}\right), \tag{5.38}$$

where the distribution parameters $\tilde{\boldsymbol{\lambda}}$ are obtained by evaluating $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\tilde{\boldsymbol{c}})$. The partial derivatives of $Q(u;\boldsymbol{\lambda})$ with respect to all the relevant parameters are

$$\frac{\partial Q}{\partial u}=\frac{1}{\lambda_2}\left(u^{\lambda_3-1}+(1-u)^{\lambda_4-1}\right), \tag{5.39}$$

$$\frac{\partial Q}{\partial\lambda_1}=1, \tag{5.40}$$

$$\frac{\partial Q}{\partial\lambda_2}=-\frac{1}{\lambda_2^2}\left(\frac{u^{\lambda_3}-1}{\lambda_3}-\frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right), \tag{5.41}$$

$$\frac{\partial Q}{\partial\lambda_3}=\frac{1}{\lambda_2\lambda_3^2}\left(u^{\lambda_3}\ln(u)\lambda_3-\left(u^{\lambda_3}-1\right)\right), \tag{5.42}$$

$$\frac{\partial Q}{\partial\lambda_4}=\frac{1}{\lambda_2\lambda_4^2}\left(\left((1-u)^{\lambda_4}-1\right)-(1-u)^{\lambda_4}\ln(1-u)\lambda_4\right). \tag{5.43}$$

It can be easily observed that Eq. (5.39) and Eq. (5.40) are continuous functions of $u\in(0,1)$ and $\boldsymbol{\lambda}$. Although Eq. (5.41) is undefined for $\lambda_3=0$ and $\lambda_4=0$, the limit exists according to *l'Hôpital's rule*. The same holds for Eq. (5.42) and Eq. (5.43). As a result, we can extend Eqs. (5.41) to (5.43) by continuity, and thus they become continuous functions of $u\in(0,1)$ and $\boldsymbol{\lambda}$. Therefore, $Q(u,\boldsymbol{\lambda})$ is continuously differentiable. In addition, Eq. (5.39) is bounded away from 0. These two properties allow one to apply the *implicit function theorem*, and thus u is a continuous function of $\boldsymbol{\lambda}$ in a neighborhood of $\tilde{\boldsymbol{\lambda}}$, which implies that u is continuous at $\tilde{\boldsymbol{\lambda}}$. According to Eq. (5.3), the PDF is a continuous function of both u and $\boldsymbol{\lambda}$. Hence, using the continuity shown before, $f_Y(y;\boldsymbol{\lambda})$ is continuous at $\tilde{\boldsymbol{\lambda}}$. Furthermore, $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c})$ are C^∞ functions of $\boldsymbol{c}$, and thus $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c})$ is continuous at $\tilde{\boldsymbol{c}}$. Combining both the continuity of $f_Y(y;\boldsymbol{\lambda})$ and $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c})$, we have that $q_{\boldsymbol{c}}(\boldsymbol{x},y)$ is continuous at $\tilde{\boldsymbol{c}}$ for the point $(\boldsymbol{x},y)$.

Now consider a point $(\boldsymbol{x},y)$ in the second class, which implies that y is outside the support of $q_{\tilde{\boldsymbol{c}}}(\boldsymbol{x},\cdot)$, say, y is smaller than the lower bound of the support of $q_{\tilde{\boldsymbol{c}}}(\boldsymbol{x},\cdot)$. In this case, $q_{\tilde{\boldsymbol{c}}}(\boldsymbol{x},y)=0$. According to Eq. (5.4), if the lower bound is finite, it is a continuous function of $\boldsymbol{\lambda}$ and thus continuous at $\tilde{\boldsymbol{c}}$. As a result, for

$\boldsymbol{c}$ within a certain neighborhood of $\tilde{\boldsymbol{c}}$, the lower bound is larger than y, which implies $q_{\boldsymbol{x}}(\boldsymbol{x}, y) = 0$ for $\boldsymbol{c}$ in this neighborhood. Thereby, $q_{\boldsymbol{c}}(\boldsymbol{x}, y)$ is continuous at $\tilde{\boldsymbol{c}}$. Analogous reasoning holds for the case where y is bigger than the upper bound of the support.

The last class corresponds to the case where y is located on the endpoint of the support of $q_{\tilde{\boldsymbol{c}}}(\boldsymbol{x}, \cdot)$. By taking $\tilde{u} = 0$ and 1 in Eq. (5.38) or considering directly Eq. (5.4), we obtain two associated deterministic functions between $\boldsymbol{x}$ and y. As a result, points of the third class can be represented by two curves in $\mathcal{D}_x \times \mathbb{R}$, whose Lebesgue measure is zero. This closes the proof of continuity. □

Lemma 5.2. *The class $\mathcal{G}$ defined below satisfies the uniform strong law of large numbers:*

$$\mathcal{G} = \left\{ g_{\boldsymbol{c}} = \log\left(\frac{q_{\boldsymbol{c}} + q_0}{2q_0}\right) \mathbb{1}_{q_0>0} : \boldsymbol{c} \in \mathcal{C} \right\}. \tag{5.44}$$

Proof. According to the continuity property in Lemma 5.1, it is obvious that for all $\tilde{\boldsymbol{c}} \in \mathcal{C}$, the map $\boldsymbol{c} \mapsto g_{\boldsymbol{c}}$ is continuous at $\tilde{\boldsymbol{c}}$ for μ-almost all $(\boldsymbol{x}, y) \in \mathcal{D} \times \mathbb{R}$. By assumption, the probability measure P_0 is absolutely continuous with respect to μ, and thus $g_{\boldsymbol{c}}$ is continuous for P_0-almost all $(\boldsymbol{x}, y) \in \mathcal{D} \times \mathbb{R}$.

Define G as the envelope function of the class $\mathcal{G}$, i.e., $G(\boldsymbol{x}, y) = \sup_{\boldsymbol{c}\in\mathcal{C}} |g_{\boldsymbol{c}}(\boldsymbol{x}, y)|$. Let us prove that $G \in L_1(P_0)$, where $L_1(P_0)$ denotes the set of absolutely integrable functions with respect to P_0.

Taking the boundedness property in Lemma 5.1 into account, we obtain

$$g_{\boldsymbol{c}}(\boldsymbol{x}, y) \leqslant \log\left(\frac{2C_q}{q_0(\boldsymbol{x}, y)}\right) = \log(2C_q) - \log(q_0(\boldsymbol{x}, y)). \tag{5.45}$$

Obviously, $g_{\boldsymbol{c}}(\boldsymbol{x}, y) \geqslant -\log(2)$. Therefore,

$$\begin{aligned} |g_{\boldsymbol{c}}(\boldsymbol{x}, y)| &\leqslant \max\left\{\log(2), |\log(2C_q)| + |\log(q_0(\boldsymbol{x}, y))|\right\} \\ &\leqslant \log(2) + |\log(C_q)| + |\log(q_0(\boldsymbol{x}, y))|. \end{aligned} \tag{5.46}$$

Because the inequality is independent of $\boldsymbol{c}$, we have

$$\begin{aligned} G(\boldsymbol{x}, y) &\leqslant \log(2) + |\log(C_q)| + |\log(q_0(\boldsymbol{x}, y))|, \\ \mathbb{E}\left[G(\boldsymbol{X}, Y)\right] &\leqslant \log(2) + |\log(C_q)| + \mathbb{E}\left[|\log(q_0(\boldsymbol{X}, Y)|\right]. \end{aligned} \tag{5.47}$$

Now consider the last term in Eq. (5.47):

$$\begin{aligned} \mathbb{E}\left[|\log(q_0(\boldsymbol{X}, Y)|\right] &= \int_{\mathcal{D}_{\boldsymbol{x}}\times\mathbb{R}} |\log\left(q_0(\boldsymbol{x}, y)\right)| \, p_0(\boldsymbol{x}, y) \mathrm{d}\boldsymbol{x}\mathrm{d}y \\ &= \int_{\mathcal{D}_{\boldsymbol{x}}} \left(\int_{\mathbb{R}} |\log\left(q_0(\boldsymbol{x}, y)\right)| \, q_0(\boldsymbol{x}, y) \mathrm{d}y \right) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{5.48}$$

Through a change of variables, the integral within the parenthesis of Eq. (5.48) can be calculated as

$$B(\boldsymbol{x}) = \int_{\mathbb{R}} |\log\left(q_0(\boldsymbol{x}, y)\right)| \, q_0(\boldsymbol{x}, y) \mathrm{d}y = \int_0^1 \left| \log\left(\frac{\lambda_2}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}} \right) \right| \mathrm{d}u, \tag{5.49}$$

where $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x}; \boldsymbol{c}_0)$. According to Eq. (5.35), we have

$$\begin{aligned} B(\boldsymbol{x}) &\leqslant \int_0^1 |\log(\lambda_2)| + \left|\log\left(u^{\lambda_3-1} + (1-u)^{\lambda_4-1}\right)\right| \mathrm{d}u \\ &\leqslant k_2 + \int_0^1 \max\left\{\left|\log\left(u^{\underline{k}} + (1-u)^{\underline{k}}\right)\right|, \left|\log\left(u^{\overline{k}} + (1-u)^{\overline{k}}\right)\right|\right\} \mathrm{d}u, \end{aligned} \tag{5.50}$$

where

$$k_2 = \max\left\{\left|\log\left(\overline{C}_2\right)\right|, \left|\log\left(\underline{C}_2\right)\right|\right\}, \ \underline{k} = \min\left\{\underline{C}_3 - 1, \underline{C}_4 - 1\right\}, \ \overline{k} = \max\left\{\overline{C}_3 - 1, \overline{C}_4 - 1\right\}.$$

Using the symmetry of the integrand, we get

$$\begin{aligned} B(\boldsymbol{x}) &\leqslant k_2 + 2 \cdot \max\left\{\int_0^{\frac{1}{2}} \left|\log\left(u^{\underline{k}} + (1-u)^{\underline{k}}\right)\right| \mathrm{d}u, \int_0^{\frac{1}{2}} \left|\log\left(u^{\overline{k}} + (1-u)^{\overline{k}}\right)\right| \mathrm{d}u\right\} \\ &\leqslant k_2 + 2 \cdot \left(\int_0^{\frac{1}{2}} \left|\log\left(u^{\underline{k}} + (1-u)^{\underline{k}}\right)\right| \mathrm{d}u + \int_0^{\frac{1}{2}} \left|\log\left(u^{\overline{k}} + (1-u)^{\overline{k}}\right)\right| \mathrm{d}u\right). \end{aligned} \tag{5.51}$$

Without loss of generality, we now study the property of the integral

$$\int_0^{\frac{1}{2}} \left|\log\left(u^k + (1-u)^k\right)\right| \mathrm{d}u. \tag{5.52}$$

For $k = 0$, Eq. (5.52) is equal to $\frac{1}{2}\log(2)$. For $k > 0$, we have $u^k \leqslant (1-u)^k$, and thus

$$\begin{aligned} \int_0^{\frac{1}{2}} \left|\log\left(u^k + (1-u)^k\right)\right| \mathrm{d}u \leqslant \int_0^{\frac{1}{2}} \left|\log\left(2(1-u)^k\right)\right| \mathrm{d}u &\leqslant \frac{1}{2}\log(2) - \int_0^{\frac{1}{2}} k\log(1-u)\mathrm{d}u \\ &= \frac{1}{2}\log(2) + \frac{k}{2}\left(1 - \log(2)\right). \end{aligned} \tag{5.53}$$

Through similar calculation, for $k < 0$, we have

$$\begin{aligned} \int_0^{\frac{1}{2}} \left|\log\left(u^k + (1-u)^k\right)\right| \mathrm{d}u \leqslant \int_0^{\frac{1}{2}} \left|\log\left(2u^k\right)\right| \mathrm{d}u &\leqslant \frac{1}{2}\log(2) + \int_0^{\frac{1}{2}} k\log(u)\mathrm{d}u \\ &= \frac{1}{2}\log(2) + \frac{-k}{2}\left(\log(2) + 1\right). \end{aligned} \tag{5.54}$$

As a result, Eq. (5.52) is finite. More precisely,

$$\int_0^{\frac{1}{2}} \left|\log\left(u^k + (1-u)^k\right)\right| \mathrm{d}u \leqslant \frac{1}{2}\log(2) + \frac{|k|}{2}\left(\log(2) + 1\right). \tag{5.55}$$

Eq. (5.55) implies

$$B(\boldsymbol{x}) \leqslant k_2 + \log(2) + \left(|\underline{k}| + |\overline{k}|\right)\left(\log(2) + 1\right) = C_B. \tag{5.56}$$

By inserting Eq. (5.56) into Eq. (5.48), we obtain

$$\mathbb{E}\left[|\log(q_0(\boldsymbol{X}, Y)|\right] \leqslant C_B. \tag{5.57}$$

Then, according to Eq. (5.47), the envelope function G fulfills

$$\begin{aligned}\mathbb{E}\left[G(\boldsymbol{X},Y)\right] &\leqslant \log(2) + \left|\log(C_q)\right| + \mathbb{E}\left[\left|\log\left(q_0(\boldsymbol{X},Y)\right)\right|\right] \\ &= \log(2) + \left|\log(C_q)\right| + C_B < +\infty.\end{aligned} \tag{5.58}$$

Since G is always positive according to its definition, Eq. (5.58) means $G \in L_1(P_0)$. The continuity and the property of the envelope function G shown above allow applying (van de Geer, 2000, Lemma 3.10), which guarantees that $\mathcal{G}$ satisfies the uniform weak law of large numbers:

$$\sup_{\boldsymbol{c}\in\mathcal{C}}\left(\frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{c}}\left(\boldsymbol{X}^{(i)},Y^{(i)}\right) - \mathbb{E}\left[g_{\boldsymbol{c}}\left(\boldsymbol{X},Y\right)\right]\right) \xrightarrow[n\to+\infty]{P} 0. \tag{5.59}$$

Finally, (Talagrand, 1987, Theorem 22) extends the convergence to *almost surely*, which is the uniform strong law of large numbers. □

Now, we have all the ingredients to prove Theorem 5.1.

Proof. Following (van de Geer, 2000, Lemma 4.1, 4.2), it can be easily shown that

$$0 \leqslant \int_{\mathcal{D}_{\boldsymbol{x}}} h^2\left(q_{\hat{\boldsymbol{c}}}, q_0 \mid \boldsymbol{x}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \leqslant 8\left(\sum_{i=1}^{N} g_{\hat{\boldsymbol{c}}}\left(\boldsymbol{X}^{(i)},Y^{(i)}\right) - \mathbb{E}\left[g_{\hat{\boldsymbol{c}}}\left(\boldsymbol{X},Y\right)\right]\right), \tag{5.60}$$

where the Hellinger distance is given by

$$h^2\left(q_{\hat{\boldsymbol{c}}}, q_0 \mid \boldsymbol{x}\right) = \frac{1}{2}\int_{\mathbb{R}}\left(\sqrt{q_{\hat{\boldsymbol{c}}}(\boldsymbol{x},y)} - \sqrt{q_0(\boldsymbol{x},y)}\right)^2 \mathrm{d}y.$$

According to Lemma 5.2, Eq. (5.60) implies

$$\int_{\mathcal{D}_{\boldsymbol{x}}} h^2\left(q_{\hat{\boldsymbol{c}}}, q_0 \mid \boldsymbol{x}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \xrightarrow{\text{a.s.}} 0, \tag{5.61}$$

which is called the *Hellinger consistency*.

We define the function

$$R(\boldsymbol{c}) = \int_{\mathcal{D}_{\boldsymbol{x}}} h^2\left(q_{\boldsymbol{c}}, q_0 \mid \boldsymbol{x}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{5.62}$$

According to Lemma 5.1, $\forall\tilde{\boldsymbol{c}} \in \mathcal{C}$, the map $\boldsymbol{c} \mapsto \left(\sqrt{q_{\boldsymbol{c}}} - \sqrt{q_0}\right)^2$ is continuous at $\tilde{\boldsymbol{c}}$ for all $\boldsymbol{x} \in \mathcal{D}_x$ and almost all $y \in \mathbb{R}$. Since $\left(\sqrt{q_{\boldsymbol{c}}} - \sqrt{q_0}\right)^2 \leqslant q_{\boldsymbol{c}} + q_0$, and $\int_{\mathbb{R}}\left(q_{\boldsymbol{c}} + q_0\right)\mathrm{d}y = 2 < +\infty$, the map $\boldsymbol{c} \mapsto h^2\left(q_{\boldsymbol{c}}, q_0 \mid \boldsymbol{x}\right)$ is continuous for all $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{x}}$, which is guaranteed by the *generalized Lebesgue dominated convergence theorem*. Similarly, the map $\boldsymbol{c} \mapsto R(\boldsymbol{c})$ is also continuous.

Without going into lengthy discussions, it can be shown that the GLD is *not identifiable* only for $\lambda_3 = \lambda_4 = 1$ and $\lambda_3 = \lambda_4 = 2$. In other words, by excluding two points in the $\lambda_3 - \lambda_4$ plane, different values of $\boldsymbol{\lambda}$ lead to different distributions. Note that the two exceptions are the only two cases where the corresponding distributions are uniform distributions. As a result, the last condition in Theorem 5.1 excludes the nonidentifiable cases. Furthermore, $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c})$ are polynomials in $\boldsymbol{x}$ and linear in $\boldsymbol{c}$. Therefore, for $\boldsymbol{c} \neq \tilde{\boldsymbol{c}}$, $\boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c})$ and $\boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x};\tilde{\boldsymbol{c}}\right)$ are not identical for μ-almost all $\boldsymbol{x} \in \mathbb{R}^M$, and thus for $P_{\boldsymbol{X}}$-almost all $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. Hence, there

exists a set $\Omega_{\boldsymbol{x}}$ with $P_{\boldsymbol{X}}(\Omega_{\boldsymbol{x}}) > 0$ such that as long as $\boldsymbol{c} \neq \boldsymbol{c}_0$, $h\left(q_{\boldsymbol{c}}, q_0 \mid \boldsymbol{x}\right) > 0 \; \forall \boldsymbol{x} \in \Omega_{\boldsymbol{x}}$, which implies the uniqueness. Finally, combining Eq. (5.61) with the continuity and uniqueness of $R(\boldsymbol{c})$, we have $\hat{\boldsymbol{c}} \xrightarrow{\text{a.s.}} \boldsymbol{c}_0$. □

References

Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators—application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67–77.

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58:371–382.

Arnold, D. V. and Hansen, N. (2012). A (1+1)-CMA-ES for constrained optimisation. In Soule, T. and Moore, J. H., editors, *Proceedings of the Genetic and Evolutionary Computation Conference 2012 (GECCO 2012) (Philadelphia, PA)*, pages 297–304.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions—application to computational electromagnetic dosimetry. *International Journal for Uncertainty Quantification*, 9:351–363.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27:808–821.

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2019). Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61:7–23.

Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Britton, T. (2010). Stochastic epidemic models: A survey. *Mathematical Biosciences*, 225:24–35.

Browne, T., Iooss, B., Le Gratiet, L., Lonchampt, J., and Rémy, E. (2016). Stochastic simulators based optimization by Gaussian process metamodels—application to maintenance investments planning issues. *Quality and Reliability Engineering International*, 32(6):2067–2080.

Davidian, M. and Carroll, R. J. (1987). Variance function estimation. *Journal of the American Statistical Association*, 82:1079–1091.

Efromovich, S. (2010). Dimension reduction and adaptation in conditional density estimation. *Journal of the American Statistical Association*, 105:761–774.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Fadikar, A., Higdon, D., Chen, J., Lewis, B., Venkatramanan, S., and Marathe, M. (2018). Calibrating a stochastic, agent-based model using quantile-based emulation. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1685–1706.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Communications in Statistics - Theory and Methods*, 17:3547–3567.

Gautschi, W. (2004). *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81:2340–2361.

Goldberg, P. W., Williams, C. K. I., and Bishop, C. M. (1997). Regression with input-dependent noise: A Gaussian process treatment. In *Proceedings of the 10th International Conference on Advances in Neural Information Processing Systems (NIPS 1997), Colorado, USA*, pages 493–499.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99:1015–1026.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica*, 50:1029–1054.

Harvey, A. C. (1976). Estimating regression models with multiplicative heteroscedasticity. *Econometrica*, 44:461–465.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27:1–32.

Jakeman, J. D., Franzelin, F., Natayan, A., Eldred, M., and Plfüger, D. (2019). Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351:643–666.

Jimenez, M. N., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 5:378–402.

Karian, Z. A. and Dudewicz, E. J. (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press.

Kemna, A. G. Z. and Vorst, A. C. F. (1990). A pricing method for options based on average asset values. *Journal of Banking & Finance*, 14:113–129.

Marelli, S. and Sudret, B. (2019). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-104.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22:833–847.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ.

Merl, D., Johnson, L. R., Gramacy, R. B., and Mangel, M. (2009). A statistical framework for the adaptive management of epidemiological interventions. *PLoS ONE*, 4:e5089.

Moustapha, M., Lataniotis, C., Wiederkehr, P., Wagner, P.-R., Wicaksono, D., Marelli, S., and Sudret, B. (2019). UQLib User Manual. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-201.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:116–155.

Murcia, J. P., Réthoré, P. E., Dimitrov, N., Natarajan, A., Sørensen, J. D., Graf, P., and Kim, T. (2018). Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. *Renewable Energy*, 119:910–922.

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function. *Biometrika*, 74(2):221–232.

Newey, W. K. and McFadden, D. (1994). Chapter 36 large sample estimation and hypothesis testing. In *Handbook of Econometrics*, volume 4, pages 2111–2245. Elsevier.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Reddy, K. and Clinton, V. (2016). Simulating stock prices using geometric Brownian motion: Evidence from Australian companies. *Australasian Accounting, Business and Finance Journal*, 10(3):23–47.

Sadler, W. A. and Smith, M. H. (1985). Estimation of the response error relationship in immunoassay. *Clinical Chemistry*, 31:1802–1805.

Shreve, S. (2004). *Stochastic Calculus for Finance II*. Springer, New York.

Soize, C. and Ghanem, R. G. (2004). Physical systems with random uncertainties: Chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26(2):395–410.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637.

Talagrand, M. (1987). The Glivenko-Cantelli problem. *Annals of Probability*, 15:837–870.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, Cambridge, New York.

van de Geer, S. (2000). *Empirical Processes in M-Estimation*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

Villani, C. (2009). *Optimal Transport, Old and New*. Springer, Berlin.

Wooldridge, J. M. (2013). *Introductory Econometrics: A Modern Approach*. Cengage Learning, 5th edition.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815.

6

Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models

This chapter is a post-print of

> Zhu, X. and Sudret, B. (2021). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815. DOI:10.1016/j.ress.2021.107815.

differing from the published paper only in terms of layout and formatting.

Author contributions. **X. Zhu**: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - Original Draft, Visualization. **B. Sudret**: Supervision, Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition.

Abstract

Global sensitivity analysis aims at quantifying the impact of input variability onto the variation of the response of a computational model. It has been widely applied to deterministic simulators, for which a set of input parameters has a unique corresponding output value. Stochastic simulators, however, have intrinsic randomness due to their use of (pseudo)random numbers, so they give different results when run twice with the same input parameters but non-common random numbers. Due to this random nature, conventional Sobol' indices, used in global sensitivity analysis, can be extended to stochastic simulators in different ways. In this paper, we discuss three possible extensions and focus on those that depend only on the statistical dependence between input and output. This choice ignores the detailed data generating process involving the internal randomness, and can thus be applied to a wider class of problems. We propose to use the generalized lambda model to emulate the response distribution of stochastic simulators. Such a surrogate can be constructed without the need for replications. The proposed method is applied to three examples including two case studies in finance and

epidemiology. The results confirm the convergence of the approach for estimating the sensitivity indices even with the presence of strong heteroskedasticity and small signal-to-noise ratio.

6.1 Introduction

Computational models, a.k.a. simulators, have been extensively used to represent physical phenomena and engineering systems. They can help assess the reliability, control the risk and optimize the behavior of complex systems early at the design stage. Conventional simulators are usually deterministic, in the sense that repeated model evaluations with the same input parameters yield the same value of the output. In contrast, several runs of a stochastic simulator for a given set of input parameters provide different results. More precisely, the output of a stochastic simulator is a random variable following an unknown probability distribution. Hence, each model evaluation with the same input values generates a realization of the random variable. Mathematically, a stochastic simulator can be defined by

$$\begin{aligned} \mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\to \mathbb{R}, \\ (\boldsymbol{x}, \omega) &\mapsto \mathcal{M}_s(\boldsymbol{x}, \boldsymbol{Z}(\omega)), \end{aligned} \tag{6.1}$$

where $\boldsymbol{x}$ is the input vector that belongs to the input space $\mathcal{D}_{\boldsymbol{X}}$, and Ω denotes the probability space that represents the stochasticity. The intrinsic randomness is due to the fact that some *latent variables* $\boldsymbol{Z}(\omega)$ inside the model are not explicitly considered as a part of the input variables: given a fixed input value $\boldsymbol{x}_0$, the output of the simulator is a random variable.

In this respect, we can consider a stochastic simulator as a random field indexed by the parameters $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$ (Azzi et al., 2019). For a given realization of the latent variables $\boldsymbol{z}_0$, the simulator becomes a deterministic function of $\boldsymbol{x}$. This is realized in practice by initializing the random seed to the same value before running the simulator for different $\boldsymbol{x}$'s, a trick known as *common random numbers*. The (classical) functions $\boldsymbol{x} \mapsto \mathcal{M}_s(\boldsymbol{x}, \boldsymbol{z}_0)$ will be called *trajectories* in this paper. One particular trajectory corresponds to one particular value $\boldsymbol{z}_0$.

In contrast, for a given $\boldsymbol{x}_0 \in \mathcal{D}_{\boldsymbol{X}}$, the output of the stochastic simulator is a random variable. Its distribution can be obtained by repeatedly running the simulator with $\boldsymbol{x}_0$, yet different realizations of the latent variables called *replications*.

Stochastic simulators are ubiquitous in modern engineering, finance and medical sciences. Typical examples include stochastic differential equations (e.g., financial models [McNeil et al., 2005]) and agent-based models (e.g., epidemiological models [Britton, 2010]). To a certain extent, physical experiments can also be considered as stochastic models, because we may not be able to measure and consider all the relevant variables that can uniquely determine the experimental conditions.

In practice, the input variables may be affected by uncertainty due to noisy measurements, expert judgment or lack of knowledge. Therefore, they are modeled as random variables and grouped into a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_M)$, which is characterized by a joint distribution $f_{\boldsymbol{X}}$. Quantification of the contribution of input variability to the output uncertainty is a major task in sensitivity analysis (Saltelli et al., 2000). It allows us to identify the most important set of input variables that dominate the output variability and also to figure

out non-influential variables. This information provides more insights into the simulator and can be further used for model calibrations and decision making (de Rocquigny et al., 2008).

A large number of methods have been successfully developed to perform sensitivity analysis in the context of deterministic simulators (Saltelli et al., 2000; Helton et al., 2006; Borgonovo, 2017). Among others, the variance-based sensitivity analysis, also referred to as Sobol' indices (Sobol', 1993), is one of the most popular approaches, which relies on the analysis of variance. Several extensions of Sobol' indices to stochastic simulators can be found in the literature, depending on the treatment of the intrinsic randomness. It is worth emphasizing that the overall uncertainty now consists of two parts, namely the inherent stochasticity in the latent variables and the uncertainty in the input parameters $\boldsymbol{X}$. Iooss and Ribatet (2009) include the latent variables as a part of the input, which results in a natural extension of the classical Sobol' indices to stochastic simulators. Hart et al. (2016) and Jimenez et al. (2017) define the Sobol' indices as functions of the latent variables which, as a consequence, become random variables whose statistical properties can be studied. Recently, Azzi et al. (2020) propose to represent the intrinsic randomness by the entropy of the response distribution and to calculate the classical Sobol' indices on the latter. All in all, relatively little attention has been devoted to sensitivity analysis for stochastic simulators.

Sensitivity analysis usually requires a large number of model evaluations for different realizations of the input vector. Due to the intrinsic randomness of stochastic simulators, an additional layer of stochasticity comes on top of the input uncertainty, which requires repeated runs with the same input parameters to fully characterize the model response. As a consequence, such analyses become intractable when the simulator is expensive to evaluate. To alleviate the computational burden, surrogate models can be constructed to mimic the original numerical model at a smaller computational cost. Large efforts have been dedicated to emulating the mean and variance function of stochastic simulators (Dacidian and Carroll, 1987; Marrel et al., 2012; Binois et al., 2018). These two functions provide only the first two moments of the response distribution and are mostly used to estimate the Sobol' indices proposed in Iooss and Ribatet (2009). In recent papers (Zhu and Sudret, 2020, 2021), we developed a novel surrogate model, called *generalized lambda model* (GLaM), to emulate the whole response distribution of stochastic simulators. This model uses generalized lambda distributions (GLDs; Karian and Dudewicz, 2000) to flexibly approximate the response distribution, while the distribution parameters cast as functions of the inputs are approximated through polynomial chaos expansions (PCEs; Ghanem and Spanos, 2003).

Based on these premises, the goal of this paper is to establish a clear framework to carry out global sensitivity analysis for stochastic simulators, and to propose efficient computational approaches based on GLaM stochastic emulators. Therefore, the original contributions of this paper are two-fold. On the one hand, we give a thorough review of the current development of global sensitivity analysis for stochastic simulators. We point out the nature and the properties of different extensions of Sobol' indices, which provides a general guideline to their usage. On the other hand, we present a unified framework based on generalized lambda models to calculate a whole variety of global sensitivity indices using this single surrogate.

The paper is organized as follows. First, we review three extensions of Sobol' indices to stochastic simulators in Section 6.2. In Section 6.3, we present the framework of GLaMs (Zhu and Sudret, 2020). There, we recap the fitting procedure proposed in Zhu and Sudret (2021), where it is emphasized that there is no need for replicated runs. Then, we discuss the use of GLaMs for estimating different types of Sobol' indices. In Section 6.4, we illustrate the performance of GLaMs on three examples. While the first example is analytical, the second and

third ones are realistic case studies in finance and epidemiology, respectively. Finally, we summarize the main findings of the paper and provide an outlook for future research in Section 6.5.

6.2 Global sensitivity analysis of stochastic simulators

6.2.1 Sobol' indices

Variance-based sensitivity analysis has been extensively studied and successfully developed in the context of deterministic simulators. For a deterministic model $\mathcal{M}_d$, Sobol' indices quantify the contribution of each input variable $\{X_i, i = 1, \dots, M\}$, or combination thereof, to the variance of the model output $Y = \mathcal{M}_d(\boldsymbol{X})$.

In this paper, we assume that X_i's are mutually independent. Let us split the input vector into two subsets $\boldsymbol{X} = (\boldsymbol{X}_{\mathbf{u}}, \boldsymbol{X}_{\sim\mathbf{u}})$, where $\mathbf{u} \subset \{1, \dots, M\}$ and $\sim \mathbf{u}$ is the complement of $\mathbf{u}$, i.e., $\sim \mathbf{u} = \{1, \dots, M\} \setminus \mathbf{u}$. From the total variance theorem, the variance of the output can be decomposed as

$$\operatorname{Var}[Y] = \mathbb{E}\left[\operatorname{Var}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]\right] + \operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]\right]. \tag{6.2}$$

The first-order and total Sobol' indices introduced by Sobol' (1993) and Homma and Saltelli (1996) for the subset of input variables $\boldsymbol{X}_{\mathbf{u}}$ are defined by

$$S_{\mathbf{u}} \overset{\text{def}}{=} \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]\right]}{\operatorname{Var}[Y]}, \quad S_{T_{\mathbf{u}}} \overset{\text{def}}{=} 1 - \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\sim\mathbf{u}}\right]\right]}{\operatorname{Var}[Y]} = 1 - S_{\sim\mathbf{u}}. \tag{6.3}$$

Higher-order Sobol' indices can be defined with the help of $S_{\mathbf{u}}$. For example, the second-order or two-factor interaction Sobol' index of X_1 and X_2 is given by

$$S_{1,2} \overset{\text{def}}{=} S_{\{1,2\}} - S_1 - S_2, \tag{6.4}$$

where we denote $S_{\{i\}}$ by S_i for the sake of simplicity.

In the context of stochastic simulators defined in Eq. (6.1), the input variables alone do not determine the value of the output. Iooss and Ribatet (2009) extend $\boldsymbol{X}$ by adding the internal source of randomness represented by latent variables $\boldsymbol{Z}$, which turns the stochastic simulator into a deterministic one. In this case, all the input variables are gathered in $(\boldsymbol{X}_{\mathbf{u}}, \boldsymbol{X}_{\sim\mathbf{u}}, \boldsymbol{Z})$, and thus the Sobol' indices in Eq. (6.3) can be naturally extended to

$$S_{\mathbf{u}} \overset{\text{def}}{=} \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]\right]}{\operatorname{Var}[Y]}, \quad S_{T_{\mathbf{u}}} \overset{\text{def}}{=} 1 - \frac{\operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\sim\mathbf{u}}, \boldsymbol{Z}\right]\right]}{\operatorname{Var}[Y]}. \tag{6.5}$$

Note that $S_{\mathbf{u}}$ has the same expression as in the case of deterministic simulators, but $S_{T_{\mathbf{u}}}$ contains the additional variables $\boldsymbol{Z}$ (Marrel et al., 2012). Similarly, higher-order Sobol' indices corresponding to interactions among components of $\boldsymbol{X}$ are defined in the same way as deterministic simulators, whereas interactions between components of $\boldsymbol{X}$ and $\boldsymbol{Z}$ involve $\boldsymbol{Z}$ in their definition. Since this is a direct extension, the Sobol' indices defined in Eq. (6.5) are referred to as *classical Sobol' indices* in the sequel.

Another way to extend Sobol' indices to stochastic simulators is to first eliminate the internal randomness by representing the response random variable $Y(\boldsymbol{x})$ by some summarizing statistical quantity, called here quantity of interest (QoI) denoted by $\mathrm{QoI}(\boldsymbol{x})$, such as the mean value $m(\boldsymbol{x})$, variance $v(\boldsymbol{x})$ (Iooss and Ribatet, 2009), α-

quantile $q_\alpha(\boldsymbol{x})$ (Browne et al., 2016) and differential entropy $h(\boldsymbol{x})$ (Azzi et al., 2020). As a result, the stochastic simulator is reduced to a deterministic function $\mathrm{QoI}(\boldsymbol{x})$, and we can calculate the associated *QoI-based Sobol' indices* as follows:

$$S_{\mathbf{u}}^{\mathrm{QoI}} \overset{\text{def}}{=} \frac{\mathrm{Var}\left[\mathbb{E}\left[\mathrm{QoI}(\boldsymbol{X})|\boldsymbol{X}_{\mathbf{u}}\right]\right]}{\mathrm{Var}\left[\mathrm{QoI}(\boldsymbol{X})\right]}, \quad S_{T_{\mathbf{u}}}^{\mathrm{QoI}} \overset{\text{def}}{=} 1 - \frac{\mathrm{Var}\left[\mathbb{E}\left[\mathrm{QoI}(\boldsymbol{X})|\boldsymbol{X}_{\sim\mathbf{u}}\right]\right]}{\mathrm{Var}\left[\mathrm{QoI}(\boldsymbol{X})\right]}. \tag{6.6}$$

A third extension is defined by considering a stochastic simulator as a random field. For a fixed internal randomness $\boldsymbol{Z}(\omega) = \boldsymbol{z}$, the stochastic simulator is a deterministic function of the input variables, which corresponds to a trajectory. Hence, the associated Sobol' indices are well-defined. Yet, they are random variables because of their dependence on $\boldsymbol{Z}$ (Hart et al., 2016), which results in the *trajectory-based Sobol' indices*:

$$S_{\mathbf{u}}^{\mathrm{traj}}(\boldsymbol{Z}) \overset{\text{def}}{=} \frac{\mathrm{Var}_{\boldsymbol{X}_{\mathbf{u}}}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}, \boldsymbol{Z}\right]\right]}{\mathrm{Var}\left[Y \mid \boldsymbol{Z}\right]}, \quad S_{T_{\mathbf{u}}}^{\mathrm{traj}}(\boldsymbol{Z}) \overset{\text{def}}{=} 1 - \frac{\mathrm{Var}_{\boldsymbol{X}_{\sim\mathbf{u}}}\left[Y \mid \boldsymbol{X}_{\sim\mathbf{u}}, \boldsymbol{Z}\right]}{\mathrm{Var}\left[Y \mid \boldsymbol{Z}\right]}, \tag{6.7}$$

where the indices of the variance operators correspond to those variables to which these operators apply.

6.2.2 Discussion

The three types of Sobol' indices introduced above have different nature and focus. The classical Sobol' indices defined in Eq. (6.5) treat the latent variables $\boldsymbol{Z}$ as a set of separate input variables. As a result, indices of this type treat $\boldsymbol{Z}$ in the same way as $\boldsymbol{X}$. The first-order index $S_{\mathbf{u}}$ indicates how much the output variance can be reduced (in expectation) if we can fix the value of $\boldsymbol{X}_{\mathbf{u}}$. Besides, the classical Sobol' indices can also quantify the influence of the intrinsic randomness as well as its interactions with input variables.

The QoI-based Sobol' indices defined in Eq. (6.6) help study a specific statistical quantity of the model response, which is a deterministic function of the inputs. Using a summary quantity to represent the random output might lead to a loss of information (Hart et al., 2016), unless this quantity itself is of interest. For example, we may want to find the variable(s) that has the largest effect(s) on the 95% quantile (i.e., $q_\alpha(\boldsymbol{X})$ for $\alpha = 0.95$) of the model response. However, the importance (ranking) of the inputs X_i's can be quite different depending on the choice of the QoI.

Unlike the previous two types of indices, trajectory-based Sobol' indices presented in Eq. (6.7) are random variables. This is because the latent variables $\boldsymbol{Z}$ and the input parameters $\boldsymbol{X}$ are treated differently: conditioned on a given $\boldsymbol{Z} = \boldsymbol{z}_0$, the stochastic simulator reduces to a deterministic function of $\boldsymbol{X}$, and we can calculate the associated (classical) Sobol' indices. To evaluate the probability distribution of these trajectory-based Sobol' indices requires that the same random seeds can be explicitly fixed in the simulator when running it for different values of $\boldsymbol{x}$. In a sense, trajectory-based Sobol' indices emphasize the variation of the trajectory of stochastic simulators. They typically show the importance of each input variable in terms of its contribution to the variability of trajectories.

For stochastic simulators, the question "which input variable has the strongest effect?" is rather vague and cannot be answered by a single type of index. The analyst should properly pose the problem and select the appropriate sensitivity measures. If one aims at reducing the variance of the model output, the classical Sobol' indices are of interest. If one is interested in some summary QoIs (which is often the case for applications, e.g., quantiles in reliability analysis), the QoI-based indices are more appropriate. Finally, if one can control

the internal randomness and is interested in the variability of the model output for fixed intrinsic stochasticity, trajectory-based indices should be selected.

To further illustrate this discussion, let's consider the stochastic simulator

$$Y(\boldsymbol{X}) = \mathcal{M}_s(\boldsymbol{X}, Z) = X_1 + X_2 \cdot Z$$

where X_1, X_2 and Z are independent random variables following standard normal distributions. The classical first-order Sobol' indices are $S_1 = 0.5$ and $S_2 = 0$. Therefore, if we want to primarily reduce the variance of Y, X_1 should be investigated. For the response mean function $m(\boldsymbol{X}) = X_1$, we have $S_1^m = 1$ and $S_2^m = 0$, which indicates that X_1 contributes fully to the variation of the mean function. In contrast, for the variance function $v(\boldsymbol{X}) = X_2^2$, $S_1^v = 0$ and $S_2^v = 1$ reveals a different order. Regarding the trajectory-based indices, we have $S_1^{\text{traj}}(Z) = \frac{1}{1+Z^2}$ and $S_2^{\text{traj}}(Z) = \frac{Z^2}{1+Z^2}$ which are two random variables (due to the randomness in Z). The probability distribution of the two variables characterize how the randomness in $\boldsymbol{X}$ affects the function $\mathcal{M}_s(\boldsymbol{X}, z)$ with z being fixed as a single realization of Z. In summary, even for this simple toy example, the various indices provide different conclusions.

It is worth remarking that the classical Sobol' indices $S_{\mathbf{u}}$ in Eq. (6.5) share some common properties with the mean-based Sobol' indices in Eq. (6.6), when we consider the mean function $\text{QoI}(\boldsymbol{x}) \overset{\text{def}}{=} m(\boldsymbol{x}) = \mathbb{E}\left[Y \mid \boldsymbol{X} = \boldsymbol{x}\right]$:

$$S_{\mathbf{u}}^m = \frac{\text{Var}\left[\mathbb{E}\left[m(\boldsymbol{X})|\boldsymbol{X}_{\mathbf{u}}\right]\right]}{\text{Var}\left[m(\boldsymbol{X})\right]}. \tag{6.8}$$

According to the law of total expectation, we have

$$\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right] = \mathbb{E}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}\right] \mid \boldsymbol{X}_{\mathbf{u}}\right] = \mathbb{E}\left[m(\boldsymbol{X}) \mid \boldsymbol{X}_{\mathbf{u}}\right]. \tag{6.9}$$

As a result, $S_{\mathbf{u}}$ can be rewritten as

$$S_{\mathbf{u}} = \frac{\text{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{u}}\right]\right]}{\text{Var}\left[Y\right]} = \frac{\text{Var}\left[\mathbb{E}\left[m(\boldsymbol{X}) \mid \boldsymbol{X}_{\mathbf{u}}\right]\right]}{\text{Var}\left[Y\right]}, \tag{6.10}$$

This implies that both classical Sobol' indices $S_{\mathbf{u}}$ and mean-based Sobol' indices $S_{\mathbf{u}}^m$ provide the same ranking, as the numerators of Equations (6.8) and (6.10) are identical. However, it is worth emphasizing that $S_{\mathbf{u}}$ is not equal to $S_{\mathbf{u}}^m$ and they are measuring different quantities, since the denominator of Eq. (6.10) is $\text{Var}\left[Y\right]$ but that of Eq. (6.8) is $\text{Var}\left[m(\boldsymbol{X})\right]$.

As a summary for all three extensions, a stochastic simulator is essentially transformed into a reduced deterministic model at a certain stage. The classical Sobol' indices include the latent variables as a part of the inputs. The QoI-based Sobol' indices rely on a deterministic representation. The trajectory-based indices are random variables whose statistical properties can be studied at the cost of repeating a standard Sobol' analysis for different realizations of the latent variables separately. As a result, the three types of sensitivity indices can be estimated by modifying only slightly the standard methods based on Monte Carlo simulation developed for deterministic simulators (Saltelli et al., 2000).

The classical Sobol' indices involving $\boldsymbol{Z}$ (e.g., $S_{\boldsymbol{Z}}$, $S_{T_{\mathbf{u}}}$ in Eq. (6.5)) and the trajectory-based Sobol' indices require controlling the latent variables $\boldsymbol{Z}$. In practical computations, this is achieved by fixing the *random seed* in the computational model (Marrel et al., 2012; Hart et al., 2016). However, for certain types of stochastic

simulators, or when the data are generated by physical experiments, it may be difficult to control or even identify $\boldsymbol{Z}$. For the sake of general applicability, we focus in this paper only on Sobol' indices that can be estimated by manipulating $\boldsymbol{X}$, that is, the QoI-based Sobol' indices and, to some extent, the classical Sobol' indices.

Using Monte Carlo simulations to estimate these indices requires evaluating the simulator for various realizations of the input vector. In addition, it is generally necessary to evaluate the function $\mathrm{QoI}(\boldsymbol{x})$ for calculating the associated QoI-based Sobol' indices. However, this function is not directly accessible due to the intrinsic randomness. Therefore this function is usually estimated by using replications; i.e., for each realization $\boldsymbol{x}$, the simulator is run many times, and $\mathrm{QoI}(\boldsymbol{x})$ is estimated from the output samples. Both factors call for a large number of model runs, which becomes impracticable for costly models. Therefore, the use of surrogate models is unavoidable.

In the sequel, we present the generalized lambda model as a stochastic surrogate. Such a model emulates the response distribution conditioned on $\boldsymbol{X} = \boldsymbol{x}$, which fully characterizes the statistical dependence between the inputs and output. Therefore, it can be used to estimate the considered Sobol' indices.

6.3 Generalized lambda models

Generalized lambda models consist of mainly two parts: the generalized lambda distribution and polynomial chaos expansions. In this section, we briefly recap these two elements and present an algorithm to construct such a model without the need for replicated runs of the simulator. Then, we discuss how to estimate the sensitivity indices from the surrogate.

6.3.1 Generalized lambda distributions

The generalized lambda distribution is a flexible distribution family, which is designed to approximate many common distributions (Karian and Dudewicz, 2000), e.g., normal, lognormal, Weibull and generalized extreme value distributions. A GLD is defined by its *quantile function* $Q(u)$ with $u \in [0,1]$, that is, the inverse of the cumulative distribution function $Q(u) = F^{-1}(u)$. In this paper, we consider the GLD of the Freimer–Kollia–Mudholkar–Lin (FKML) family (Freimer et al., 1988) with four parameters, whose quantile function is defined by

$$Q(u;\boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right), \tag{6.11}$$

where λ_1 is the location parameter, λ_2 is the scaling parameter, and λ_3 and λ_4 are the shape parameters. λ_2 is required to be positive to produce valid quantile functions (i.e., Q being non-decreasing on $[0,1]$). Based on the quantile function defined in Eq. (6.11), the probability density function (PDF) of a random variable Y following a GLD is given by

$$f_Y(y;\boldsymbol{\lambda}) = \frac{f_U(u)}{Q'(u;\boldsymbol{\lambda})} = \frac{\lambda_2}{u^{\lambda_3-1}+(1-u)^{\lambda_4-1}}\mathbb{1}_{[0,1]}(u), \text{ with } u = Q^{-1}(y;\boldsymbol{\lambda}), \tag{6.12}$$

where $\mathbb{1}_{[0,1]}$ is the indicator function. A closed-form expression of Q^{-1} is not available for arbitrary values of λ_3 and λ_4. Therefore, evaluating the PDF for a given y usually requires solving the nonlinear equation (6.12) numerically.

GLDs cover a wide range of shapes determined by λ_3 and λ_4 (Zhu and Sudret, 2020). For instance, $\lambda_3 = \lambda_4$ produces symmetric PDFs, and $\lambda_3 < \lambda_4$ ($\lambda_3 > \lambda_4$) results in left-skewed (resp. right-skewed) distributions. Moreover, λ_3 and λ_4 are closely linked to the support and the tail behaviors of the corresponding PDF. More precisely, λ_3 and λ_4 control the left and right tail, respectively. Whereas $\lambda_3 > 0$ implies that the PDF support is left-bounded, $\lambda_3 \leqslant 0$ implies that the distribution has a lower infinite support. Similarly, $\lambda_4 > 0$ implies that the PDF support is right-bounded, whereas it is $+\infty$ for $\lambda_4 \leqslant 0$. In addition, for $\lambda_3 < 0$ ($\lambda_4 < 0$), the left (resp. right) tail decays asymptotically as a power law. Hence, GLDs can also provide fat-tailed distributions. The reader is referred to Zhu and Sudret (2020) and Freimer et al. (1988) for a longer presentation of GLDs.

6.3.2 Polynomial chaos expansions

Consider a deterministic model $\mathcal{M}_d$ that maps a set of input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_M) \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$ to the output $y \in \mathbb{R}$. Under the assumption that $Y = \mathcal{M}_d(\boldsymbol{X})$ has finite variance, $\mathcal{M}_d$ belongs to the Hilbert space $\mathcal{H}$ of square-integrable functions with respect to the inner product $\langle u, v \rangle_{\mathcal{H}} = \mathbb{E}\left[u(\boldsymbol{X})v(\boldsymbol{X})\right] = \int_{\mathcal{D}_{\boldsymbol{X}}} u(\boldsymbol{x})v(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$. If the joint distribution $f_{\boldsymbol{X}}$ satisfies certain conditions (Ernst et al., 2012), the simulator $\mathcal{M}_d$ admits a spectral representation in terms of orthogonal polynomials:

$$Y = \mathcal{M}_d(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{6.13}$$

where $\psi_{\boldsymbol{\alpha}}$ is a multivariate polynomial basis function indexed by $\boldsymbol{\alpha} \in \mathbb{N}^M$, and $c_{\boldsymbol{\alpha}}$ denotes the associated coefficient. The orthogonal basis can be obtained by using tensor products of univariate polynomials, each of which is orthogonal with respect to the probability measure $f_{X_i}(x_i)\mathrm{d}x_i$ of the i-th variable X_i:

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j). \tag{6.14}$$

Details about the construction of this generalized polynomial chaos expansion can be found in Xiu and Karniadakis (2002) and Sudret (2015).

The PCE defined in Eq. (6.13) contains an infinite sum of terms. However, in practice, it is only feasible to use a finite series as an approximation. To this end, truncation schemes are adopted to select a set of basis functions defined by a finite subset $\mathcal{A} \subset \mathbb{N}^M$ of multi-indices. A typical scheme is the hyperbolic (a.k.a. q-norm) truncation scheme (Blatman and Sudret, 2010) given by

$$\mathcal{A}^{p,q,M} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M : \|\boldsymbol{\alpha}\|_q \overset{\text{def}}{=} \left(\sum_{i=1}^{M} |\alpha_i|^q \right)^{\frac{1}{q}} \leqslant p \right\}, \tag{6.15}$$

where p is the maximum degree of polynomials, and $q \leqslant 1$ defines the quasi-norm $\|\cdot\|_q$. Note that with $q = 1$, we obtain the full basis of total degree less than p, which corresponds to the *standard truncation scheme*.

6.3.3 Formulation of generalized lambda models

Because of the flexibility of GLDs, we assume that the response random variable $Y(\boldsymbol{x})$ of a stochastic simulator for a given input vector $\boldsymbol{x}$ can be well approximated by a GLD. Hence, the associated distribution parameters $\boldsymbol{\lambda}$ are functions of the input variables:

$$Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{6.16}$$

Under appropriate conditions discussed in Section 6.3.2, each component of $\boldsymbol{\lambda}(\boldsymbol{x})$ can be represented by a series of orthogonal polynomials. Because $\lambda_2(\boldsymbol{x})$ is required to be positive (see Section 6.3.1), the associated polynomial chaos representation is built on the natural logarithmic transform $\log\left(\lambda_2(\boldsymbol{x})\right)$. This results in the following approximations:

$$\lambda_l\left(\boldsymbol{x}\right) \approx \lambda_s^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_l} c_{l,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad l = 1, 3, 4, \tag{6.17}$$

$$\lambda_2\left(\boldsymbol{x}\right) \approx \lambda_2^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} c_{2,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right), \tag{6.18}$$

where $\mathcal{A}_l$ $(l = 1, 2, 3, 4)$ is a finite set of selected basis functions for λ_l, and $c_{l,\boldsymbol{\alpha}}$'s are the coefficients. For the purpose of clarity, we explicitly express $\boldsymbol{c}$ in the PC approximations $\lambda_l^{\mathrm{PC}}\left(\boldsymbol{x}; \boldsymbol{c}\right)$ to emphasize that $\boldsymbol{c}$ are the model parameters yet to be estimated from data.

6.3.4 GLaM constructions

We assume that our costly stochastic simulator is evaluated once for each point $\boldsymbol{x}^{(i)}$ of the experimental design $\mathcal{X}$, and the associated model response $y^{(i)}$ is collected in $\mathcal{Y}$:

$$\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}, \quad \mathcal{Y} = \left\{\mathcal{M}_s\left(\boldsymbol{x}^{(1)}, \omega^{(1)}\right), \ldots, \mathcal{M}_s\left(\boldsymbol{x}^{(N)}, \omega^{(N)}\right)\right\} \tag{6.19}$$

As already mentioned (and as emphasized by the notation $\omega^{(i)}$), no replications are required, and we do not control the random seed. To construct a GLaM from the available data $(\mathcal{X}, \mathcal{Y})$, both the truncated sets $\boldsymbol{\mathcal{A}}$ of basis functions and the coefficients $\boldsymbol{c}$ shall be determined. In this section, we summarize the method proposed in Zhu and Sudret (2021), which is designed to achieve both purposes without the need for replications.

Sometimes prior knowledge is available to set the basis functions. For example, when working with a standard linear regression problem, the data is supposed to be generated by

$$Y = \beta_0 + \boldsymbol{X}\boldsymbol{\beta} + \epsilon, \tag{6.20}$$

where ϵ has mean zero and is independent of $\boldsymbol{X}$. This case can be treated within the GLaM framework as follows: $\mathcal{A}_1$ contains the constant and linear term, and $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$ contain only a constant term. Note that such a GLaM allows estimating the distribution of ϵ, which is not required to be normal, whereas the usual linear regression framework assumes normally distributed ϵ.

However, in general there is no prior knowledge that would help select $\boldsymbol{\mathcal{A}}$. Thus, we make the following assumptions to find appropriate hyperbolic truncation schemes defined in Eq. (6.15) for each $\lambda_i, i = 1, \ldots, 4$:

(A1) The response distribution of $Y(\boldsymbol{x})$ can be well-approximated by a generalized lambda distribution;

(A2) The shape of this distribution smoothly varies as a function of $\boldsymbol{x}$, so that the parameters $\lambda_i(\boldsymbol{x})$ can be well approximated by a low-order PCE

Because the shape of a GLD is controlled by λ_3 and λ_4, the associated hyperbolic truncation schemes $\mathcal{A}^{p,q,M}$ can be set with a small value of p, say $p = 1$.

Moreover, the parameters $\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$ mainly affect the variation of the mean $m(\boldsymbol{x})$ and of the variance $v(\boldsymbol{x})$ as a function of the input $\boldsymbol{x}$, respectively. As a result, they may require possibly larger degree p. To this end, we modify the feasible generalized least-squares (FGLS; Wooldridge, 2013) to find suitable truncation schemes for the mean and variance function modeled as

$$m(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} c_{m,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad v(\boldsymbol{x}) = \exp\left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_v} c_{v,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \right).$$

Basically, FGLS iterates between a weight least-square problem (WLS) to fit the mean function, and an ordinary least-square (OLS) analysis to estimate the variance function.

The details of the modified FGLS are presented in Algorithm 6.1. In this algorithm, the inputs $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ stand for the set of candidate degrees and q-norms that are tested to expand $\lambda_1(\boldsymbol{x})$, respectively. The same notation apply to $\boldsymbol{p}_2$ and $\boldsymbol{q}_2$ for $\lambda_2(\boldsymbol{x})$. Indeed, because of the low cost of least-square analysis, various combinations of p and q are tested for both $\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$.

More precisely, AOLS denotes *adaptive ordinary least-squares* with degree and q-norm adaptivity (Marelli and Sudret, 2019; Blatman and Sudret, 2011). This algorithm first builds a series of PCEs, each of which is obtained by applying ordinary least-squares with a truncation scheme $\mathcal{A}^{p,q,M}$ defined by a combination of $p \in \boldsymbol{p}$ and $q \in \boldsymbol{q}$. Then, it selects the PCE, therefore the associated truncation scheme, with the smallest *leave-one-out* errors (see Blatman and Sudret, 2011 for details). WLS denotes the use of weighted least-squares, which takes the estimated variance $\hat{\boldsymbol{v}}$ as weight to re-estimate $\boldsymbol{c}_m$. In this procedure, the truncation set $\mathcal{A}_m$ for $m(\boldsymbol{x})$ is selected only once (before the loop), whereas a set of truncation schemes $\{\mathcal{A}_v^i : i = 1, \ldots, N_{\mathrm{FGLS}}\}$ is obtained.

We finally select the one with the smallest leave-one-out error as the final truncated set $\mathcal{A}_v$ for $v(\boldsymbol{x})$. The number of iterations N_{FGLS} is defined by the user, typically $N_{\mathrm{FGLS}} = 5$–10. After applying Algorithm 6.1, we set $\mathcal{A}_1 = \mathcal{A}_m$ and $\mathcal{A}_2 = \mathcal{A}_v$.

Once the basis functions are selected, we use the maximum (conditional) likelihood estimator to estimate $\boldsymbol{c}$:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c} \in \mathcal{C}} \mathsf{L}(\boldsymbol{c}), \tag{6.21}$$

where $\mathsf{L}(\boldsymbol{c})$ is the conditional negative log-likelihood

$$\mathsf{L}(\boldsymbol{c}) = \sum_{i=1}^{N} -\log\left(f^{\mathrm{GLD}}\left(y^{(i)}; \boldsymbol{\lambda}^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{c} \right) \right) \right), \tag{6.22}$$

with f^{GLD} being the probability density function of the GLD defined in Eq. (6.12).

Algorithm 6.1 Modified feasible generalized least-squares

Input: $(\mathcal{X}, \mathcal{Y})$, $\boldsymbol{p}_1$, $\boldsymbol{q}_1$, $\boldsymbol{p}_2$, $\boldsymbol{q}_2$
Output: truncated sets for the mean and variance function–$\mathcal{A}_m$ and $\mathcal{A}_v$
$\mathcal{A}_m,\ \hat{\boldsymbol{c}}_m \leftarrow \mathrm{AOLS}\left(\mathcal{X}, \mathcal{Y}, \boldsymbol{p}_1, \boldsymbol{q}_1\right)$
for $i \leftarrow 1, \ldots, N_{\mathrm{FGLS}}$ **do**
  $\hat{\boldsymbol{m}} \leftarrow \sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} c_{m,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\mathcal{X})$
  $\tilde{\boldsymbol{r}} \leftarrow 2 \log\left(|\mathcal{Y} - \hat{\boldsymbol{m}}|\right)$
  $\mathcal{A}_v^i,\ \hat{\boldsymbol{c}}_v,\ \varepsilon_{\mathrm{LOO}}^i \leftarrow \mathrm{AOLS}\left(\mathcal{X}, \tilde{\boldsymbol{r}}, \boldsymbol{p}_2, \boldsymbol{q}_2\right)$
  $\hat{\boldsymbol{v}} \leftarrow \exp\left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_v} c_{v,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\mathcal{X})\right)$
  $\hat{\boldsymbol{c}}_m \leftarrow \mathrm{WLS}\left(\mathcal{X}, \mathcal{Y}, \mathcal{A}_m, \hat{\boldsymbol{v}}\right)$
end for
$i^* = \arg\min\left\{\varepsilon_{\mathrm{LOO}}^i : i = 1, \ldots, N_{\mathrm{FGLS}}\right\}$ and $\mathcal{A}_v \leftarrow \mathcal{A}_v^{i^*}$

The advantages of the proposed estimator are twofold. On the one hand, the simulator is required to be evaluated only once (but not limited to one) on each point of the experimental design. Thereby, replications are not necessary (yet possible), and the method is versatile in this respect. On the other hand, if the underlying computational model can be exactly represented by a GLaM for a specific choice of $\boldsymbol{c}$, the maximum likelihood estimator is *consistent* (see proof in Zhu and Sudret, 2021).

In practice, the evaluation of $\mathsf{L}(\boldsymbol{c})$ is not straightforward because the PDF of generalized lambda distributions does not have an explicit form: it is necessary to solve nonlinear equations as shown in Eq. (6.12). Nevertheless, the nonlinear function $Q(u; \boldsymbol{\lambda})$ is monotonic and defined on $[0, 1]$. Therefore, we proposed using the bisection method (Burden et al., 2015) to efficiently solve the nonlinear equations.

6.3.5 Sensitivity analysis with GLaMs

6.3.5.1 Introduction

The various Sobol' indices introduced in Eq. (6.5) and (6.6) can be estimated by sampling from the conditional distribution $Y \mid \boldsymbol{X}_{\mathbf{u}}$. Because of the specific format of the GLaM definition, such a sampling can be easily performed.

The generalized lambda distribution parameterizes the quantile function $Q(u; \boldsymbol{\lambda})$ (see Eq. (6.11)), which can be seen as the inverse probability integral transform. In other words, the random variable $Q(U; \boldsymbol{\lambda})$ with $U \sim \mathcal{U}(0, 1)$ follows $\mathrm{GLD}(\boldsymbol{\lambda})$. As a result, sampling from a GLD is straightforward. We define the function $Q_{\mathrm{GLaM}} : (u, \boldsymbol{x}) \in [0, 1] \times \mathcal{D}_{\boldsymbol{X}} \mapsto \mathbb{R}$ by

$$Q_{\mathrm{GLaM}}(u; \boldsymbol{x}) = Q(u; \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x})) = \lambda_1^{\mathrm{PC}}(\boldsymbol{x}) + \frac{1}{\lambda_2^{\mathrm{PC}}(\boldsymbol{x})} \left(\frac{u^{\lambda_3^{\mathrm{PC}}(\boldsymbol{x})} - 1}{\lambda_3^{\mathrm{PC}}(\boldsymbol{x})} - \frac{(1-u)^{\lambda_4^{\mathrm{PC}}(\boldsymbol{x})} - 1}{\lambda_4^{\mathrm{PC}}(\boldsymbol{x})} \right). \tag{6.23}$$

$Q_{\mathrm{GLaM}}(U; \boldsymbol{x})$ is a so-called *GLaM stochastic emulator* where $U \sim \mathcal{U}(0, 1)$ serves as a latent variable that introduces the internal source of randomness. Precisely, $Q_{\mathrm{GLaM}}(U; \boldsymbol{x})$ is a random variable following the surrogate response PDF for $\boldsymbol{X} = \boldsymbol{x}$, and $Q_{\mathrm{GLaM}}(u; \boldsymbol{x})$ provides its corresponding u-quantile. In other words, GLaM is

a simple stochastic surrogate model with only one latent variable (namely, U); this surrogate behaves similarly to the original stochastic simulator in terms of the response distribution for any $\boldsymbol{x}$.

Eq. (6.23) emulates the conditional quantile function $Q_{Y|\boldsymbol{X}}(u;\boldsymbol{x})$ of the original model, so calculating Sobol' indices $S_{\mathbf{u}}$ of the deterministic function $Q_{\mathrm{GLaM}}(u;\boldsymbol{x})$ can directly provide the classical Sobol' indices defined in Eq. (6.5). Note that Q_{GLaM} also allows us to calculate classical Sobol' indices involving U, e.g., S_U. However, since the surrogate approximates only the response distribution but cannot produce the trajectories, these Sobol' indices are not representative of those of the original model, e.g., $S_{\boldsymbol{Z}}$ that requires estimating $\mathrm{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{Z}\right]\right]$ (according to Eq. (6.5)), where the inner expectation $\mathbb{E}\left[Y \mid \boldsymbol{z}\right]$ is taken over a trajectory.

For QoI-based Sobol' indices in Eq. (6.6), if the quantity of interest $q_{\mathrm{GLaM}}(\boldsymbol{x})$ can be directly calculated from generalized lambda distributions, we can just treat it as a classical surrogate model of $\mathrm{QoI}(\boldsymbol{x})$. This is the case for the mean $m(\boldsymbol{x})$ and the variance $v(\boldsymbol{x})$ (see Section 6.a.1 for details). In addition, if $\mathrm{QoI}(\boldsymbol{x})$ is a u-quantile of the response distribution, Eq. (6.23) is used directly.

Finally, if it is impossible to evaluate analytically $q_{\mathrm{GLaM}}(\boldsymbol{x})$, we generate a large sample set from Eq. (6.23) by sampling $U \sim \mathcal{U}(0,1)$, and then use the sample statistic $\hat{q}_{\mathrm{GLaM}}(\boldsymbol{x})$ as a surrogate model for $\mathrm{QoI}(\boldsymbol{x})$.

6.3.5.2 Monte Carlo estimates

Because both $Q_{\mathrm{GLaM}}(u;\boldsymbol{x})$ and $q_{\mathrm{GLaM}}(\boldsymbol{x})$ are deterministic, we can use methods based on Monte Carlo simulations (Homma and Saltelli, 1996) to estimate the considered Sobol' indices. Here, we illustrate the estimator suggested by Janon et al. (2014) for classical Sobol' indices estimations.

We first define two random variables $Y = Q_{\mathrm{GLaM}}(U;\boldsymbol{X}_{\mathbf{u}},\boldsymbol{X}_{\sim\mathbf{u}})$ and $Y_{\mathbf{u}} = Q_{\mathrm{GLaM}}(\tilde{U};\boldsymbol{X}_{\mathbf{u}},\tilde{\boldsymbol{X}}_{\sim\mathbf{u}})$, where $\tilde{U}$ and $\tilde{\boldsymbol{X}}_{\sim\mathbf{u}}$ are independent copies of U and $\boldsymbol{X}_{\sim\mathbf{u}}$. This indicates that $Y_{\mathbf{u}}$ is correlated to Y by using the same set of random variables $\boldsymbol{X}_{\mathbf{u}}$ as argument. In addition, Y and $Y_{\mathbf{u}}$ follow the same distribution, and thus they share the same moments, e.g., $\mathbb{E}\left[Y\right] = \mathbb{E}\left[Y_{\mathbf{u}}\right], \mathbb{E}\left[Y^2\right] = \mathbb{E}\left[Y_{\mathbf{u}}^2\right]$.

Following Janon et al. (2014), $S_{\mathbf{u}}$ defined in Eq. (6.5) can be re-written as:

$$S_{\mathbf{u}} = \frac{\mathrm{Cov}\left[Y, Y_{\mathbf{u}}\right]}{\mathrm{Var}\left[Y\right]} = \frac{\mathbb{E}\left[Y\,Y_{\mathbf{u}}\right] - \left(\mathbb{E}\left[Y\right]\right)^2}{\mathbb{E}\left[Y^2\right] - \left(\mathbb{E}\left[Y\right]\right)^2} = \frac{\mathbb{E}\left[Y\,Y_{\mathbf{u}}\right] - \left(\frac{1}{2}\mathbb{E}\left[Y + Y_{\mathbf{u}}\right]\right)^2}{\frac{1}{2}\mathbb{E}\left[Y^2 + Y_{\mathbf{u}}^2\right] - \left(\frac{1}{2}\mathbb{E}\left[Y + Y_{\mathbf{u}}\right]\right)^2}. \tag{6.24}$$

We generate N_{MC} realizations of Y and $Y_{\mathbf{u}}$ by sampling (independently) $\boldsymbol{X}_{\mathbf{u}}$, $\boldsymbol{X}_{\sim\mathbf{u}}$, U, $\tilde{\boldsymbol{X}}_{\sim\mathbf{u}}$, and $\tilde{U}$. The expectations in Eq. (6.24) can be estimated by sample statistics, which leads to

$$\hat{S}_{\mathbf{u}} = \frac{\frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}} y^{(i)}\,y_{\mathbf{u}}^{(i)} - \left(\frac{1}{2N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}\left(y^{(i)} + y_{\mathbf{u}}^{(i)}\right)\right)^2}{\frac{1}{2N}\sum_{i=1}^{N_{\mathrm{MC}}}\left(\left(y^{(i)}\right)^2 + \left(y_{\mathbf{u}}^{(i)}\right)^2\right) - \left(\frac{1}{2N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}} y^{(i)} + y_{\mathbf{u}}^{(i)}\right)^2}, \tag{6.25}$$

where $y^{(i)} = Q_{\mathrm{GLaM}}\left(u^{(i)};\boldsymbol{x}_{\mathbf{u}},\boldsymbol{x}_{\sim\mathbf{u}}\right)$ and $y_{\mathbf{u}}^{(i)} = Q_{\mathrm{GLaM}}\left(\tilde{u}^{(i)};\boldsymbol{x}_{\mathbf{u}},\tilde{\boldsymbol{x}}_{\sim\mathbf{u}}\right)$ are the i-th realizations of Y and $Y_{\mathbf{u}}$, respectively.

For QoI-based Sobol' indices defined in Eq. (6.6), we follow the same procedure by replacing Q_{GLaM} by q_{GLaM}.

6.3.5.3 PCE-based estimates

As discussed in Section 6.3.5.1, estimating the considered two types of Sobol' indices of a GLaM surrogate model is reduced to studying two deterministic functions $Q_{\mathrm{GLaM}}(u;\boldsymbol{x})$ and $q_{\mathrm{GLaM}}(\boldsymbol{x})$. According to the definition in Section 6.2.1, $\boldsymbol{X}$ has mutually independent components, which are also independent of $U \sim \mathcal{U}(0,1)$. Both functions can be represented by polynomial chaos expansions (see Section 6.3.2):

$$\begin{aligned} Q_{\mathrm{GLaM}}(u;\boldsymbol{x}) &\approx Q^{\mathrm{PC}}_{\mathrm{GLaM}}(u;\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}^Q} c^Q_{\boldsymbol{\alpha}}\psi^Q_{\boldsymbol{\alpha}}(u,\boldsymbol{x}), \\ q_{\mathrm{GLaM}}(\boldsymbol{x}) &\approx q^{\mathrm{PC}}_{\mathrm{GLaM}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}^q} c^q_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \end{aligned} \tag{6.26}$$

where $\mathcal{A}^Q \subset \mathbb{N}^{M+1}$ and $\mathcal{A}^q \subset \mathbb{N}^M$ are the truncated sets defining the basis functions $\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$'s and $\psi^Q_{\boldsymbol{\alpha}}(u,\boldsymbol{x})$'s, respectively, as discussed in Eq. (6.14). Note that each multi-index in $\mathcal{A}^Q$ has a dimension $M+1$ because of the additional variable u, and the univariate basis functions of u in $\psi^Q_{\boldsymbol{\alpha}}(u,\boldsymbol{x})$ are Legendre polynomials (Xiu and Karniadakis, 2002). The advantage of using a PCE surrogate is that its Sobol' indices (of any order) can be analytically calculated by post-processing its coefficients (Sudret, 2008).

Several methods have been developed to construct PCEs for deterministic functions with given basis functions, such as the projection method (Ghanem and Spanos, 2003) and ordinary least-squares (Berveiller et al., 2006). To both determine the truncated set and estimate the associated coefficients, we opt for the hybrid-LAR algorithm (Blatman and Sudret, 2011). This method selects the most important basis functions among a candidate set, before ordinary least-squares is used to compute the coefficients. The selection procedure of the algorithm is based on *least angle regression* (LAR; Efron et al., 2004).

Practically, we first generate N_{PC} samples by sampling $\boldsymbol{X}$ and U. They are used to evaluate the target function $Q_{\mathrm{GLaM}}(u;\boldsymbol{x})$ or $q_{\mathrm{GLaM}}(\boldsymbol{x})$ to obtain the associated model responses. Then, we apply the hybrid-LAR algorithm with the generated data to construct the PCE surrogate. Finally, the Sobol' indices are calculated by post-processing the PC coefficients.

In the following examples, we use the PCE-based estimates for the Sobol' indices of the GLaM surrogate model, instead of performing Monte Carlo simulations, as the accuracy of the former turned out to be extremely good.

6.4 Examples

In this section, we illustrate the performance of GLaMs for global sensitivity analysis on an analytical example and two case studies. We focus on the classical first-order Sobol' indices and QoI-based total Sobol' indices. The choice of the QoI depends on the focus of the example. To characterize the examples, we define the *signal-to-noise ratio* of a stochastic simulator by

$$\mathrm{SNR} = \frac{\mathrm{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}\right]\right]}{\mathbb{E}\left[\mathrm{Var}\left[Y \mid \boldsymbol{X}\right]\right]} = \frac{\mathrm{Var}\left[m(\boldsymbol{X})\right]}{\mathrm{Var}\left[Y\right] - \mathrm{Var}\left[m(\boldsymbol{X})\right]}. \tag{6.27}$$

This quantity gives the ratio between the variance of Y explained by the mean function $m(\boldsymbol{x})$ and the remaining variance.

We use Latin hypercube sampling (McKay et al., 1979) to generate the experimental design. The stochastic simulator is evaluated only once on each combination of input parameters. The associated output values are used to construct surrogates with the proposed estimation procedure introduced in Section 6.3.4.

To assess the overall surrogate quality, we define the error measure

$$\varepsilon_Q \overset{\text{def}}{=} \frac{\mathbb{E}\left[\left(Q_{Y|\boldsymbol{X}}(U;\boldsymbol{X}) - Q_{\text{GLaM}}(U;\boldsymbol{X})\right)^2\right]}{\text{Var}\left[Q_{Y|\boldsymbol{X}}(U;\boldsymbol{X})\right]} = \frac{\mathbb{E}\left[\left(Q_{Y|\boldsymbol{X}}(U;\boldsymbol{X}) - Q_{\text{GLaM}}(U;\boldsymbol{X})\right)^2\right]}{\text{Var}\left[Y\right]} \tag{6.28}$$

where $Q_{Y|\boldsymbol{X}}(u;\boldsymbol{x})$ is the conditional quantile function of the model, $Q_{\text{GLaM}}(u;\boldsymbol{x})$ is that of the GLaM following the definition in Eq. (6.23), and $U \sim \mathcal{U}(0,1)$. This error has a form similar to the Wasserstein distance between probability measures (Villani, 2009). In addition, we also define an error measure to assess the accuracy of estimating the quantity of interest $\text{QoI}(\boldsymbol{x})$ whose approximation by GLaM is denoted by $q_{\text{GLaM}}(\boldsymbol{x})$

$$\varepsilon_q \overset{\text{def}}{=} \frac{\mathbb{E}\left[\left(\text{QoI}(\boldsymbol{X}) - q_{\text{GLaM}}(\boldsymbol{X})\right)^2\right]}{\text{Var}\left[\text{QoI}(\boldsymbol{X})\right]}. \tag{6.29}$$

The expectations in Eq. (6.28) and Eq. (6.29) are estimated by averaging the error over a test set $\mathcal{X}_{\text{test}}$ of size 10^5.

Experimental designs of various size N are investigated to study the convergence of the proposed method. For each size, 50 independent realizations of these experimental designs are carried out to account for statistical uncertainty in the random design. As a consequence, estimates for each scenario are represented by box plots.

6.4.1 A three-dimensional toy example

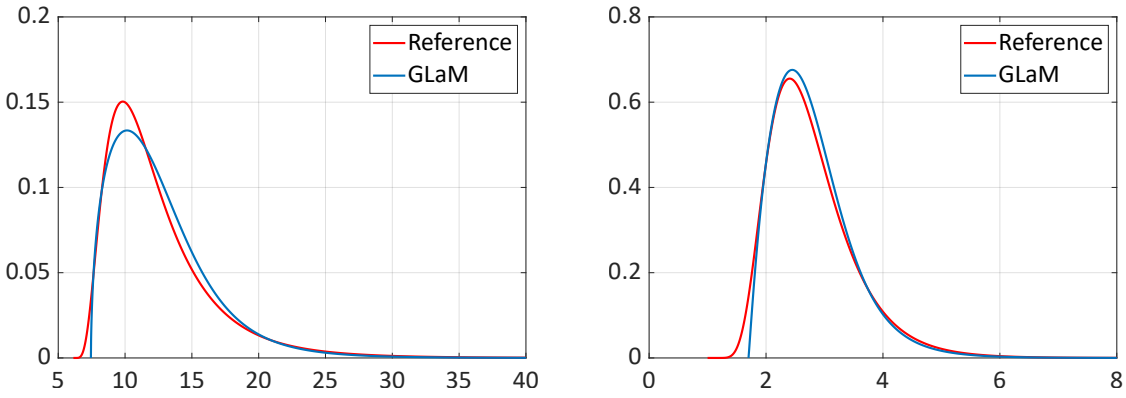
The first example is defined as follows:

$$Y(\boldsymbol{x},\omega) = \sin(x_1) + 7\sin^2(x_2) + \exp\left(\frac{x_1}{\pi} + x_3\, Z(\omega)\right) \tag{6.30}$$

where $X_1, X_2 \sim \mathcal{U}(0,2\pi)$, $X_3 \sim \mathcal{U}(0.25, 0.75)$ are independent input variables, and $Z \sim \mathcal{N}(0,1)$ denotes the latent variable that introduces the intrinsic randomness. The response distribution is a shifted lognormal distribution: the shift is equal to $\sin(x_1) + 7\sin^2(x_2)$ and the lognormal distribution is parameterized by $\mathcal{LN}\left(\frac{x_1}{\pi}, x_3\right)$. As a result, this stochastic simulator has a nonlinear location function and a strong heteroskedastic effect: the variance varies between 0.069 and 72.35. Besides, this example has a mild signal-to-noise ratio $\text{SNR} = 1.4$. This implies that the input variables can explain around 58% of the total variance of Y (i.e., $S_{\{1,2,3\}} = 0.58$).

Figure 6.1 compares the PDFs predicted by a GLaM built on an experimental design of $N = 1{,}000$ with the reference response PDFs of the simulator. The results show that the developed algorithm correctly identifies the shape of the underlying shifted lognormal distribution. Moreover, the PDF supports and tails are also accurately approximated.

We consider the differential entropy $h(\boldsymbol{x})$ (Azzi et al., 2020) as the QoI in this example. Because the analytical response distribution and entropy are known, we investigate the convergence of GLaM in terms of the conditional quantile function estimation Eq. (6.29) and the entropy estimation Eq. (6.28). The size of experimental design varies among $N \in \{250; 500; 1{,}000; 2{,}000; 4{,}000\}$. Note that the entropy of a GLD does not



(a) PDF for $\boldsymbol{x} = (5\pi/3, \pi/2, 0.6)^{\intercal}$ (b) PDF for $\boldsymbol{x} = (\pi/3, \pi, 0.4)^{\intercal}$

Figure 6.1: Toy example — Emulated response PDFs, $N = 1{,}000$

.

have a closed form. Therefore, we use 10^4 Monte Carlo samples to estimate this quantity of a GLaM for each $\boldsymbol{x}$ in the test set.

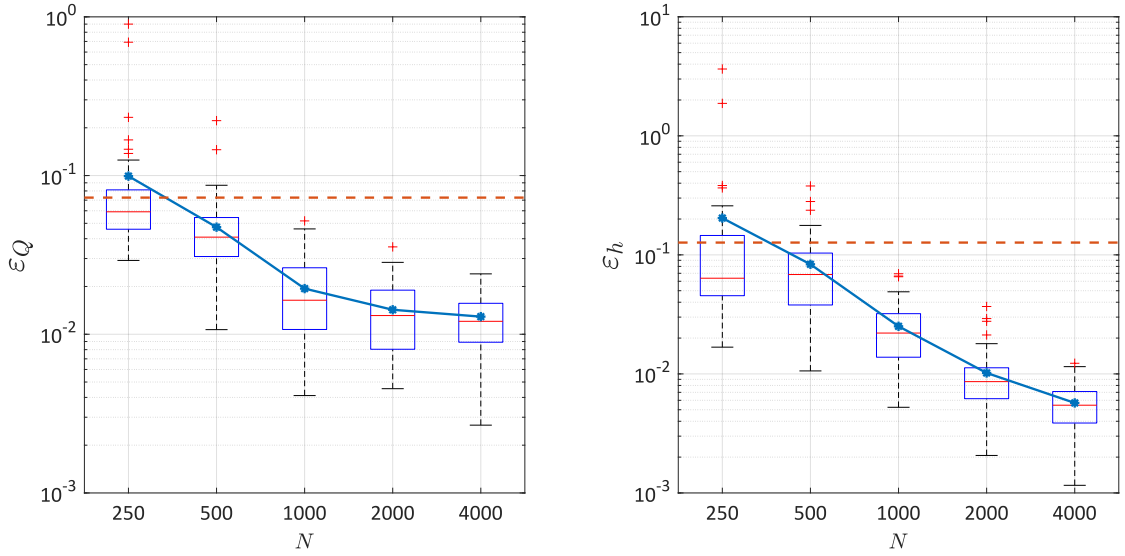
In addition, we consider another model where we approximate the response distribution with a normal distribution. The mean and variance (as functions of $\boldsymbol{x}$) for such an approximation are chosen as the true mean and variance of the original. In other words, this model represents the "oracle" of Gaussian-type mean-variance models.

The results are summarized in Figure 6.2. The proposed method exhibits a clear convergence with respect to N for both $Q(u; \boldsymbol{x})$ and $h(\boldsymbol{x})$ estimations. We observe in Figure 6.2a that the decay of ε_Q has two regimes separated by $N = 1{,}000$. For a small N, the error coming from the use of finite samples dominates the estimation accuracy. When we consider a large data set, the error mainly comes from the model misspecification, because the stochastic simulator cannot be exactly represented by a GLaM. This phenomenon is not significant for the entropy estimation, which demonstrates a relatively consistent decay.

The accuracy of the oracle normal approximation is reported with red dash lines in Figure 6.2. The error shown is only due to model misspecifications (because the true response distribution is not Gaussian) since we use the underlying true mean and variance. For both measures, the medians of the errors of GLaMs built on $N = 250$ model runs are smaller than those of the normal approximation. For $N \geqslant 1{,}000$, the GLaM clearly outperforms the oracle of Gaussian-type mean-variance models. This example illustrates the limits of such Gaussian-type models in practice.

Finally, the errors of GLaMs are below 0.05 for $N \geqslant 1{,}000$ indicating that the surrogate is able to explain over 95% of the variance of the target functions.

For sensitivity analysis, we focus on the classical first-order and the entropy-based total Sobol' indices. Figure 6.3 and Figure 6.4 show the convergence of GLaMs for estimating these quantities of each input variable. The reference values are derived from Eq. (6.30). As shown by the two figures, this toy example is designed to have X_2 as the most important variable according to the classical first-order Sobol' indices, which also indicates that X_2 contributes the most to the variance of the mean function $m(\boldsymbol{X})$. In contrast, it has zero effect to the entropy. In comparison, X_1 is the dominant variable for the variation of the entropy $h(\boldsymbol{X})$. Because X_3 mainly controls the shape of the response distribution (especially the right tail), it has a minor first-order effect to the mean function, which leads to a very small value of S_3. In contrast, the entropy depends on the distribution



(a) Conditional quantile function estimation

(b) Entropy estimation

Figure 6.2: Toy example — Convergence study. The blue lines denote the errors averaged over 50 repetitions of the full analysis. The red dash lines are the corresponding errors of the model assuming that the response distribution is normal with the true mean and variance

shape, and thus $S^h_{T_3}$ is not negligible. The results reveal that GLaMs capture this characteristic and yield accurate estimates for both classical Sobol' indices and entropy-based Sobol' indices.

Similar to Figure 6.2, we also reported the sensitivity indices calculated by using Gaussian approximations with the true mean and variance. Because the classical first-order indices depend only on the mean and variance functions, the oracle Gaussian model gives the exact values. Therefore, we showed only the results for the entropy-based Sobol' indices in Figure 6.4. It is clear that the Gaussian approximation with the true mean and variance demonstrates a significant bias. In contrast, GLaMs show nearly no bias and approximate much more accurately the reference values.

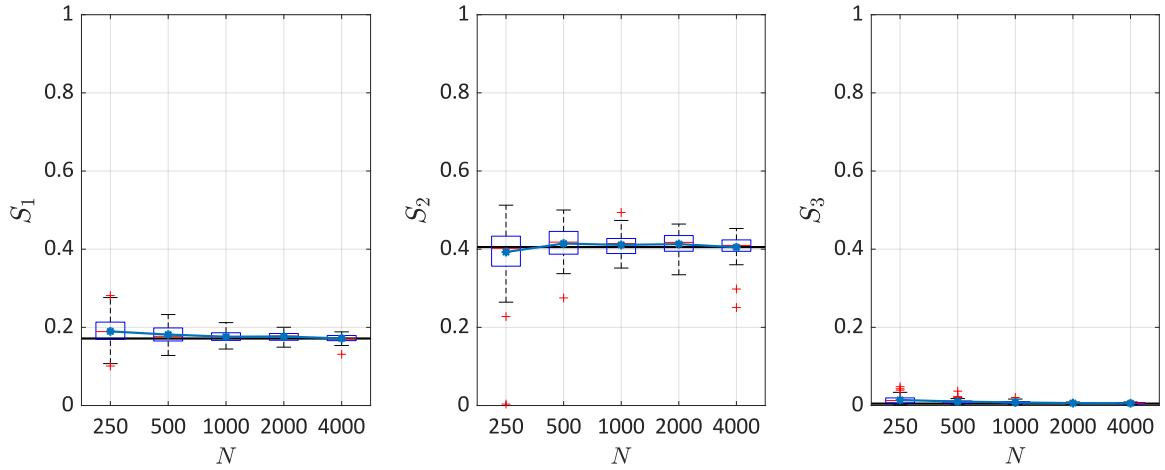


Figure 6.3: Estimation of the classical first-order Sobol' indices. The black lines are the reference values, and the blue lines denote the average values of 50 repetitions of the full analysis.

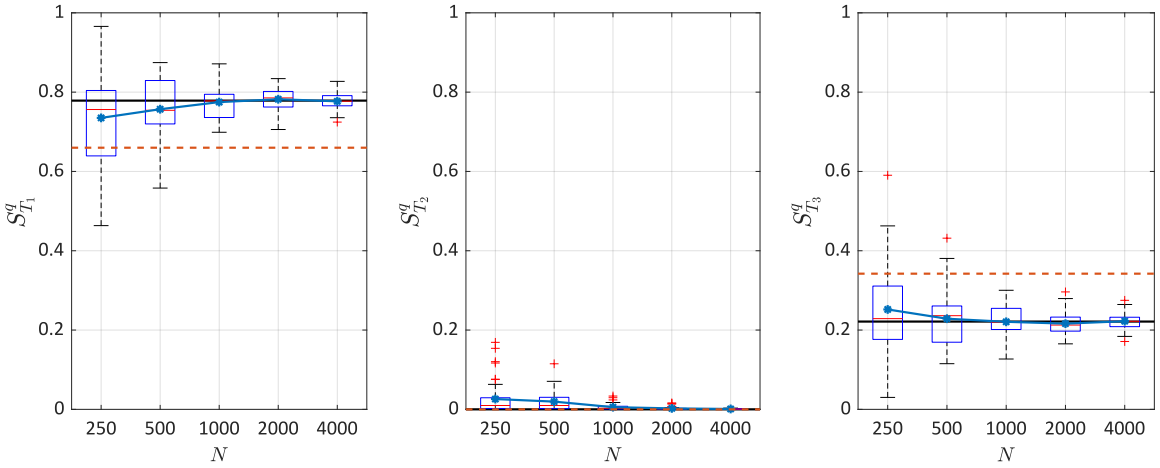


Figure 6.4: Estimation of the entropy-based total Sobol' indices. The black lines are the reference values, and the blue lines denote the average values of 50 repetitions of the full analysis. The red dash lines correspond to the indices calculated from the normal approximation using the true mean and variance

6.4.2 Heston model

In this example, we perform the global sensitivity analysis for a Heston model used in mathematical finance (Heston, 1993). The Heston model describes the evolution of a stock price Y_t. It is an extension of the geometric Brownian motion by modeling the volatility as a stochastic process v_t, instead of considering it as constant. Hence, the Heston model is a stochastic volatility model and consists of two coupled stochastic differential equations:

$$\begin{aligned} \mathrm{d}Y_t &= \mu Y_t \mathrm{d}t + \sqrt{v_t} Y_t \mathrm{d}W_t^1, \\ \mathrm{d}v_t &= \kappa(\theta - v_t)\mathrm{d}t + \sigma\sqrt{v_t}\mathrm{d}W_t^2, \end{aligned} \tag{6.31}$$

with

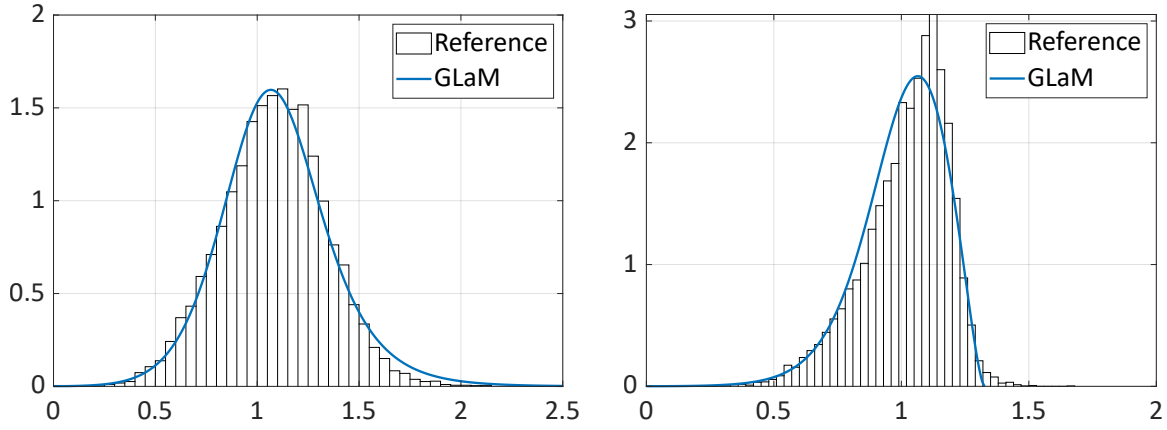
$$\mathbb{E}\left[\mathrm{d}W_t^1 \, \mathrm{d}W_t^2\right] = \rho \mathrm{d}t, \tag{6.32}$$

where W_t^1 and W_t^2 are two Wiener processes with correlation coefficient ρ, which introduce the intrinsic randomness of the stochastic model. The model parameters $\boldsymbol{x} = (\mu, \kappa, \theta, \sigma, \rho, v_0)^\intercal$ are summarized in Table 6.1. The range of the last five input variables are selected based on the parameters calibrated from real data (S&P 500 and Eurostoxx 50) (Rouah, 2013). The range of the first variable μ is set to $[0, 0.1]$ to take the uncertainty of the expected return rate into account. Without loss of generality, we set $Y_0 = 1$.

Table 6.1: Parameters of the Heston model

Variable	Description	Distribution
μ	Expected return rate	$\mathcal{U}(0, 0.1)$
κ	Mean reversion speed of the volatility	$\mathcal{U}(0.3, 2)$
θ	Long term mean of the volatility	$\mathcal{U}(0.02, 0.07)$
σ	Volatility of the volatility	$\mathcal{U}(0.2, 0.4)$
ρ	Correlation coefficient between $\mathrm{d}W_t^1$ and $\mathrm{d}W_t^2$	$\mathcal{U}(-1, -0.5)$
v_0	Volatility at time 0	$\mathcal{U}(0.02, 0.07)$

In this example, we are interested in the stock price after one year, i.e., $Y_t(\boldsymbol{x})$ with $t = 1$. A closed form solution to Eq. (6.31) is generally not available. To get samples of $Y_1(\boldsymbol{x})$, we simulate the entire time evolution of Y_t and v_t for a given $\boldsymbol{x}$ using Euler integration scheme with $\Delta t = 0.001$ over the time interval $[0, 1]$. Note that when simulating the bivariate process (Y_t, v_t), a problem may happen: since v_t follows a Cox–Ingersoll–Ross process (Rouah, 2013), the simulation scheme can generate negative values for v_t. To overcome the problem, we apply the full truncation scheme, which replaces the update of v_t by $\max(v_t, 0)$ (Rouah, 2013).



(a) PDF for $\boldsymbol{x} = (0.08, 1.6, 0.06, 0.35, -0.6, 0.06)^\intercal$ (b) PDF for $\boldsymbol{x} = (0.02, 0.5, 0.03, 0.25, -0.8, 0.03)^\intercal$

Figure 6.5: Heston model — Emulated response PDFs, $N = 2{,}000$

Figure 6.5 shows two response PDFs predicted by a surrogate built upon $N = 2{,}000$ model runs. The reference histograms are obtained from 10^4 repeated model runs with the same input parameters. We observe that the variance of the response distribution is not constant, e.g., 0.065 and 0.027 for the two illustrated PDFs. Moreover, the PDF shape varies: it changes from symmetric to left-skewed distributions depending on the model parameters. This would be difficult to approximate with a simple distribution family such as normal or lognormal. In contrast, GLaMs are able to accurately capture this shape variation, because of the flexibility of generalized lambda distributions.

Even though a closed form distribution of $Y_1(\boldsymbol{x})$ does not exist, the mean function $m(\boldsymbol{x}) = \mathbb{E}[Y_1(\boldsymbol{x})]$ can be derived analytically:

$$m(\boldsymbol{x}) = \exp(x_1) = \exp(\mu). \tag{6.33}$$

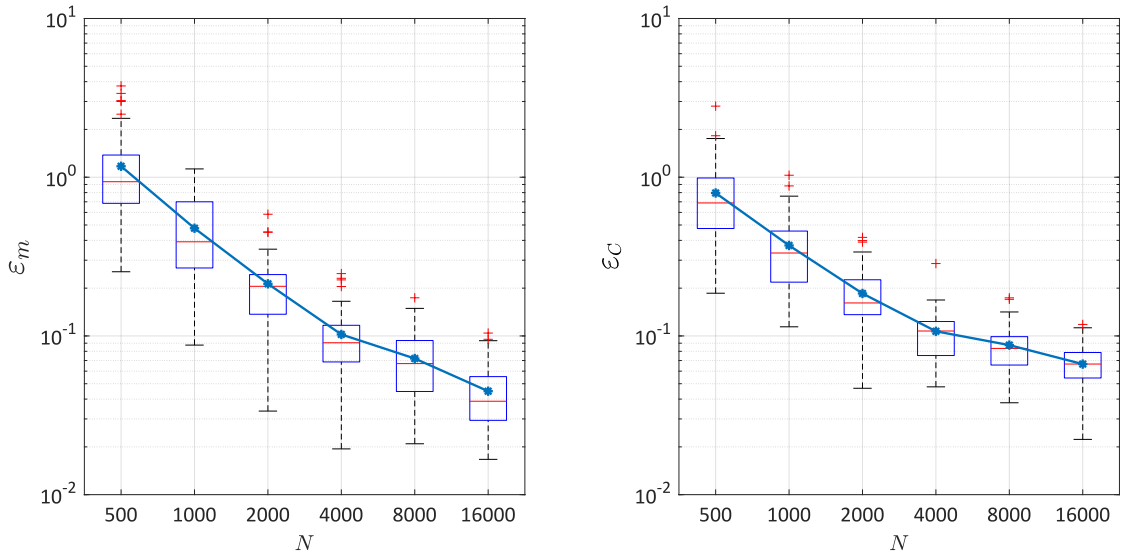
As a result, we use ε_m defined in Eq. (6.29) with $\mathrm{QoI}(\boldsymbol{x}) = m(\boldsymbol{x})$ to assess the convergence of the surrogate. In addition, we also consider the expected payoff of an European call option. The payoff $C(\boldsymbol{x})$ and the expected payoff $m_C(\boldsymbol{x})$ of an European call option are defined by

$$\begin{aligned} C(\boldsymbol{x}) &= \max\{0, Y_1(\boldsymbol{x}) - K\}, \\ m_C(\boldsymbol{x}) &= \mathbb{E}[C(\boldsymbol{x})], \end{aligned} \tag{6.34}$$

where K is the *strike price* and set to 1 in the following analysis. In finance, m_C not only is important for making investment decisions but also has a very similar form to the option price (Shreve, 2004). For the Heston model, numerical methods based on the Fourier transform have been developed to calculate the expected payoff without the need for Monte Carlo simulations (Heston, 1993). For the GLaM surrogate, this quantity can also

be calculated numerically (see Section 6.a.1). As a second performance index, we compute the associated error denoted by ε_C (Eq. (6.29)) for the convergence study.

Figure 6.6 shows box plots of the errors ε_m and ε_c for $N \in \{500; 1{,}000; 2{,}000; 4{,}000; 8{,}000; 16{,}000\}$. Both ε_m and ε_c are relatively large for $N \leqslant 2{,}000$. This is mainly due to the fact that the variability of the model response is dominated by the intrinsic randomness: the model parameters $\boldsymbol{X}$ altogether are only able to explain about 2% of the variance of the output (i.e., $S_{\{1,\dots,6\}} = 0.02$). In other words, the stochastic simulator has a very small signal-to-noise ratio $\mathrm{SNR} = 0.02/(1-0.02) \approx 0.02$. Since GLDs are flexible, a few data scattered in a moderately high dimensional space may not provide enough information of the response distribution variation. We observe that for $N \leqslant 1{,}000$, the selection procedure proposed in Algorithm 6.1 can choose $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$ being only constant. Such a model is too simple and thus fails to capture the variations of the scalar quantities. Consequently, it is necessary to have enough data to achieve an accurate estimate: when increasing the size of N of the LHS design, we observe a clear decay of the errors.



(a) Mean estimation

(b) Expected payoff estimation

Figure 6.6: Heston model — Convergence study. The blue lines denote the errors averaged over 50 repetitions of the full analysis.

We now study the convergence for the Sobol' indices estimations. According to Eq. (6.33), the mean function depends only on the first input variable X_1, which contributes little (2%) to the total variance of $Y_1(\boldsymbol{X})$. This implies that the classical Sobol' indices are not informative (they are either 0 or very close to 0). However, we cannot ignore the variability of the input variables because the response distribution demonstrates a clear dependence on the input parameters, as shown in Figure 6.5. Therefore, we focus on the accuracy of the expected-payoff-based total Sobol' indices, denoted by $S^C_{T_{\mathbf{u}}}$.

As a second quantity of interest, we also calculate the total Sobol' indices associated to the 95%-superquantile, referred to as $S^{\mathrm{sq}}_{T_{\mathbf{u}}}$. Superquantiles are known as the *conditional value-at-risk*, which is an important risk measure in finance (Acerbi, 2002). The α-superquantile of a random variable Y is defined by

$$\mathrm{sq}_\alpha = \mathbb{E}\left[Y \mid Y \geqslant q_\alpha\right], \tag{6.35}$$

where q_α is the α-quantile of Y. This quantity corresponds to the conditional expectation of Y being larger than its α-quantile. For the Heston model, this quantity does not have an analytical closed form, whereas sq_α of a GLD can be derived analytically (see Section 6.a.1).

We use 10^5 Monte Carlo samples to evaluate (numerically) the function $m_C(\boldsymbol{x})$ to obtain a reference value for each $S^C_{T_\mathbf{u}}$. To calculate the Sobol' indices associated to the 95%-superquantile $\mathrm{sq}_{95}(\boldsymbol{X})$, it is necessary to evaluate the function $\mathrm{sq}_{95}(\boldsymbol{x})$. Because it cannot analytically derived for the Heston model, we use 10^4 replications to calculate the sample 95%-superquantile $\hat{\mathrm{sq}}_{95}(\boldsymbol{x})$ as an estimate for $\mathrm{sq}_{95}(\boldsymbol{x})$. Then, we treat it as a deterministic function and use 10^4 samples to estimate each Sobol' index. This indicates that a total number of 7×10^8 model runs are performed to obtain the six reference 95%-superquantile-based total Sobol' indices. Because only 10^4 samples are used to estimate each $S^{\mathrm{sq}}_{T_\mathbf{u}}$, we use bootstraps (Efron, 1979) to calculate the 95% confidence interval to account for the uncertainty of the Monte Carlo simulation.

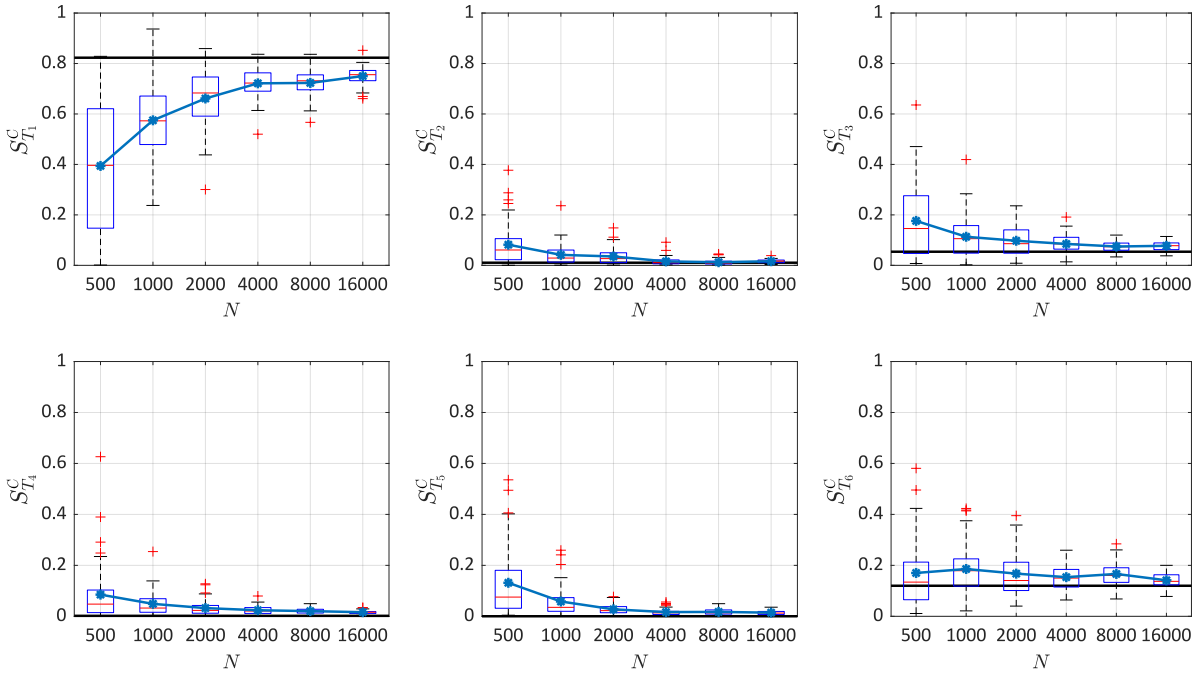


Figure 6.7: Estimation of the expected-payoff-based total Sobol' indices. The black lines are the reference values, and the blue lines denote the average values of 50 repetitions.

Figures 6.7 and 6.8 confirm and quantify the convergence of GLaMs to estimate $S^C_{T_\mathbf{u}}$ and $S^{\mathrm{sq}}_{T_\mathbf{u}}$. For the expected payoff $m_C(\boldsymbol{x})$, the first variable μ is the most important. The estimation of its total effect converges from below the reference value, and we observe a bias in the estimate. Nevertheless, with N large enough ($\geqslant 4{,}000$), the GLaM can always correctly identify its importance (the bias is 0.072 for $\geqslant 8{,}000$ and 0.055 for $\geqslant 16{,}000$), and each classical first-order Sobol' index of the other five variables converge to the reference line. The 95%-superquantile suggests a different ranking: μ, θ, ρ and v_0 (corresponding to the first, third, fifth and sixth input variable, respectively) have similar total effects, which are superior to those of κ and σ (i.e., the second and fourth input variables). In addition, none of the input variables has nearly 0 total effect. The GLaM surrogate model accurately reproduces the phenomena. Moreover, the estimates generally vary around the reference values, and larger N results in narrower spread of the box plots.

As a conclusion, GLaM surrogates allow us to represent accurately the QoI of the Heston model and carry

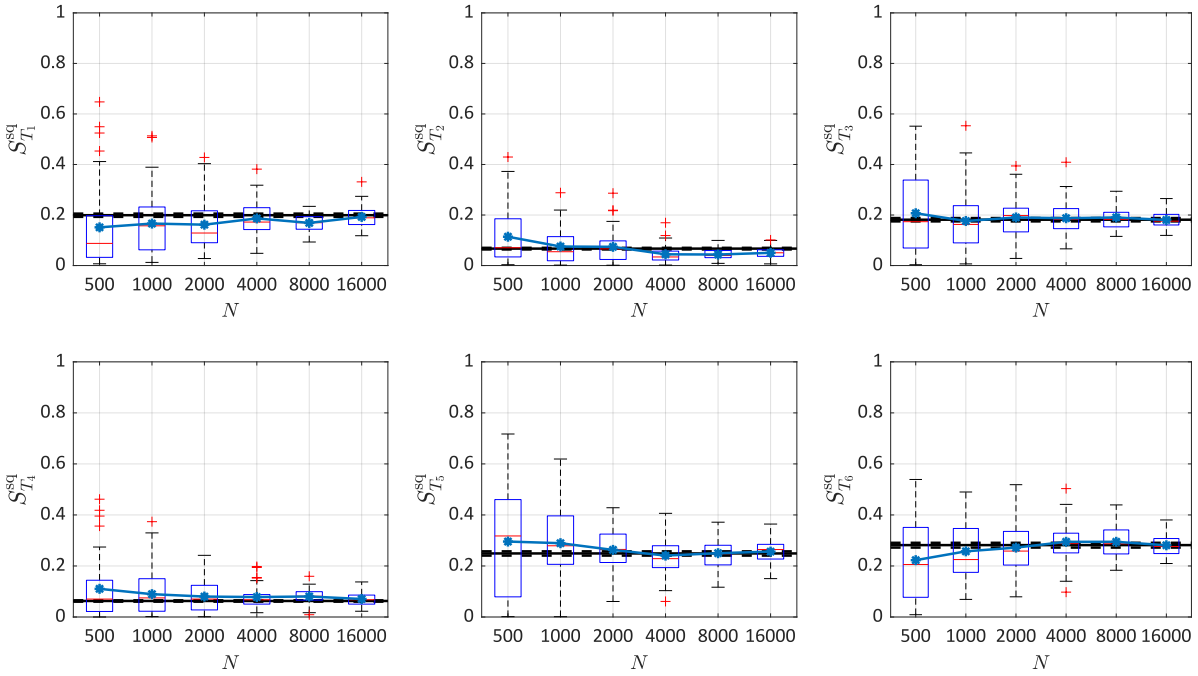


Figure 6.8: Estimation of the 95%-superquantile-based total Sobol' indices. The blue lines denote the average value of 50 repetitions of the full analysis. The black lines are the reference values, and the dashed lines correspond to the 95% confidence intervals.

out a detailed sensitivity analysis at the cost of $\mathcal{O}(10^4)$ runs of the stochastic simulator. Note that in this example the leave-one-out errors of the polynomial chaos expansions built on the GLaM surrogates are of the order of $o(10^{-4})$, which justifies the use of PCE-based Sobol' indices.

6.4.3 Stochastic SIR model

In this example, we apply the proposed method to a *stochastic Susceptible–Infected–Recovered* (SIR) model in epidemiology (Britton, 2010). This model simulates the spread of an infectious disease, which can help conduct appropriate epidemiological intervention to minimize the social and ethical impacts during the outbreak.

In a SIR model, a population of size P_t at time t can be partitioned into three groups: susceptible, infected and recovered during the outbreak of an epidemic. Susceptible individuals are those who can get infected by contacting an infectious person. Infected individuals are suffering from the disease and are contagious. They can recover (therefore classified as recovered) and become immune to future infections. The number of individuals within each group is denoted by E_t, I_t and R_t, respectively. Without differentiating individuals, these three quantities characterize the configuration of the population at a given time t. Hence, their evolutions represent the spread of the epidemic. In this study, we consider a fixed population without newborns and deaths, i.e., the total population size is constant, $P_t = P$. As a result, E_t, I_t and R_t satisfy the constraint $E_t + I_t + R_t = P$, and thus only the time evolution of E_t and I_t is necessary to represent the disease evolution.

Without going into detailed assumptions of the model, we illustrate the system dynamics in Figure 6.9, where the black icons represent susceptible individuals, the red icons indicate infected persons, and the blue icons are those recovered. Suppose that at time t the population has the configuration (E_t, I_t) (top left figure of Figure 6.9). Infected individuals can meet susceptible individuals, or they may receive essential treatments

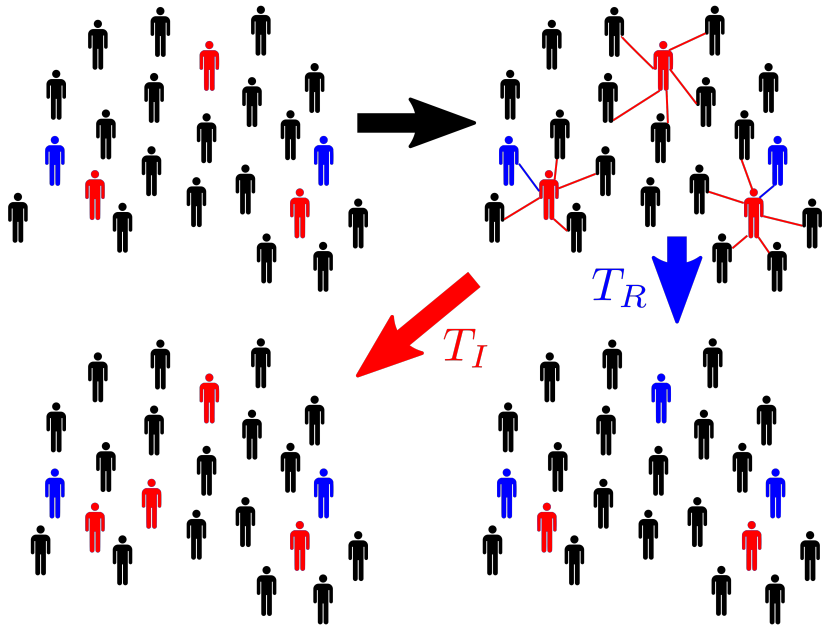


Figure 6.9: Dynamics of the stochastic SIR model: black icons denote susceptible individuals, red icons represent infected individuals, and blue icons are those recovered.

and recover from the disease. Hence, the next configuration has two possibilities: (1) C_I, where one susceptible individual is infected; (2) C_R, where one infected person recovers. The population state evolving either to C_I or C_R depends on two random variables, T_I and T_R, which denote the respective time to move to the associated candidate configuration. Both random variables follow an exponential distribution, yet with different parameters:

$$T_I \sim \text{Exp}(\lambda_I), \quad \lambda_I = \beta \frac{E_t I_t}{P}, \tag{6.36}$$

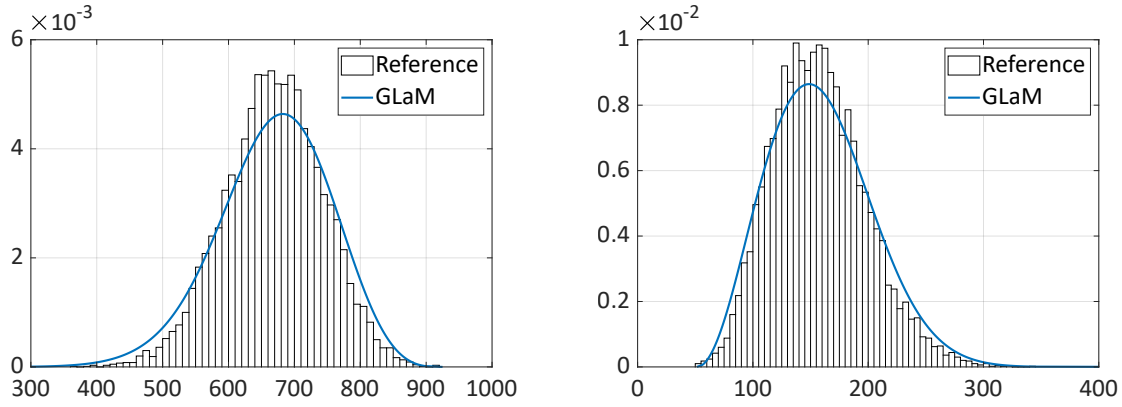
$$T_R \sim \text{Exp}(\lambda_R), \quad \lambda_R = \gamma I_t, \tag{6.37}$$

where β indicates the contact rate of an infected individual, and γ is the recovery rate. If $T_I > T_R$, C_I becomes the next configuration at $t + T_I$ with $S_{t+T_I} = E_t - 1$ and $I_{t+T_I} = I_t + 1$, and vice versa. This update step iterates until time T when $I_T = 0$. Because the population size is finite and the recovered individuals will not get infected again, the total number of updates is finite ($\leqslant P$). This number is not a constant due to the updating process, indicating that the amount of latent variables of this simulator is also random. Note that the evolution procedure described here corresponds to the *Gillespie algorithm* (Gillespie, 1977).

In this case study, we set $P = 2{,}000$. $\boldsymbol{x} = (E_0, I_0, \beta, \gamma)$ is the vector of input parameters. To account for different scenarios, the input variables $\boldsymbol{X}$ are modeled as $X_1 \sim \mathcal{U}(1{,}600\,,\, 1{,}800)$, $X_2 \sim \mathcal{U}(20, 200)$ and $X_3, X_4 \sim \mathcal{U}(0.5, 0.7)$. The uncertainty in the first two variables can be interpreted as lack of knowledge of the initial condition. While the last two variables are affected by possible interventions, such as social distancing measures that can reduce the contact rate β and increasing medical resources that improves the recovery rate γ. We are interested in the total number of newly infected individuals during the outbreak, i.e., $E_T - E_0$. The signal-to-noise ratio of this stochastic model is estimated to be SNR ≈ 6.7, which is relatively large.

Figure 6.10 shows the response PDF estimation of the surrogate model built on an experimental design of

$N = 1{,}000$. The reference histograms are calculated from 10^4 repeated model runs with the same input values. We observe that the response distribution changes from right-skewed to left-skewed distributions (so we can also find symmetric distributions in between), which is correctly represented by the surrogate. In addition, GLaMs also accurately approximate the bulk and the support of the response PDF.



(a) PDF for $\boldsymbol{x} = (1750, 150, 0.55, 0.65)^{\mathsf{T}}$ (b) PDF for $\boldsymbol{x} = (1650, 80, 0.65, 0.55)^{\mathsf{T}}$

Figure 6.10: Stochastic SIR model — Emulated response PDFs, $N = 1{,}000$

In this example, we investigate the convergence of GLaMs for estimating the classical first-order Sobol' indices and the standard-deviation-based total Sobol' indices, denoted by $S^{\sigma}_{T_u}$. To calculate the reference values, we use 10^5 Monte Carlo samples for each classical Sobol' index. Regarding the standard-deviation-based Sobol' indices, we calculate the sample standard deviation $\hat{\sigma}(\boldsymbol{x})$ based on 10^4 replications. Then, we apply Monte Carlo simulations with 10^4 samples to estimate the associated Sobol' indices. The total cost to get reference values is thus equal to 5×10^8. As in the previous example, we use bootstraps to calculate the 95% confidence intervals.

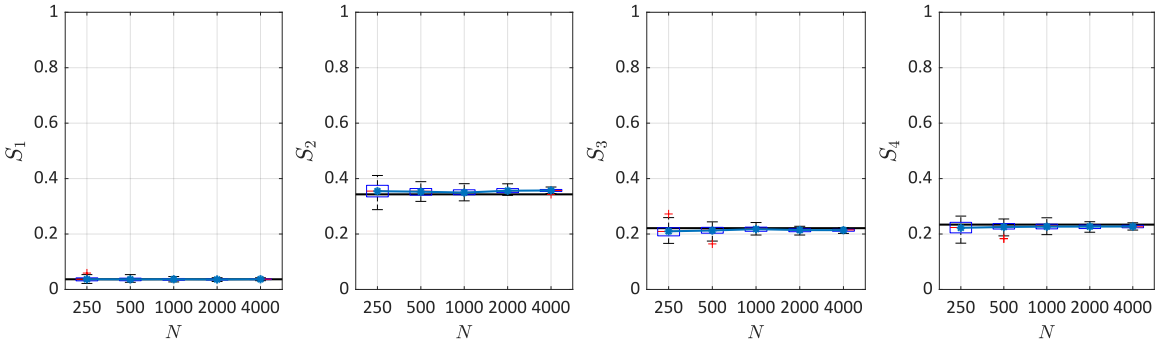


Figure 6.11: Estimation of the classical first-order Sobol' indices. The black lines are the reference values, and the blue lines denote the average values of 50 repetitions of the full analysis.

Figures 6.11 and 6.12 show the results of the convergence study. In terms of the classical Sobol' indices, the GLaM yields accurate estimates even when $N = 250$: the box plots scatter around the reference values with a small variability. Among the four input variables, the second one I_0 that corresponds to the number of infected individuals at time 0 is the most important. It is followed by the contact rate and the recovery rate, which show similar first-order effect. As a result, performing medical test to better determine I_0 would be the most effective way to reduce the variance of the output. In contrast, Figure 6.12 suggests that controlling the contact rate and recovery rate would be the best measure to reduce the variation of $\sigma(\boldsymbol{X})$. For estimating the associated Sobol'

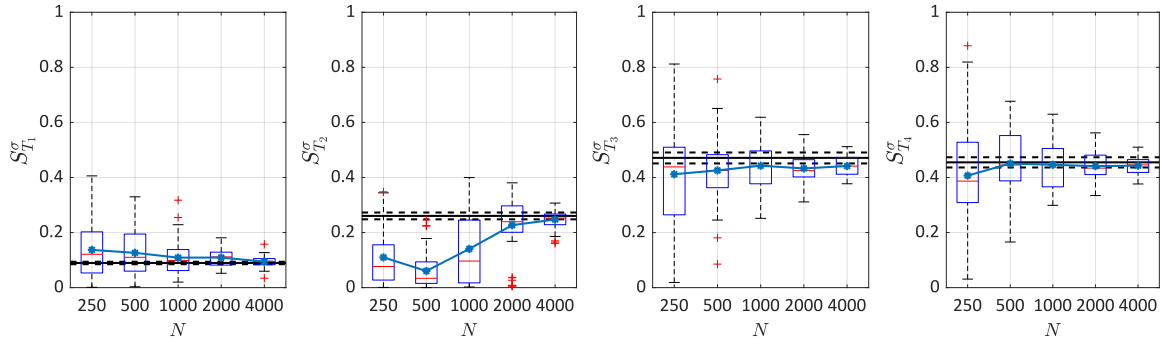


Figure 6.12: Estimation of the standard-deviation-based total Sobol' indices. The blue lines denote the average values of 50 repetitions of the full analysis. The black lines are the reference values, and the dashed lines correspond to the 95% confidence intervals.

indices, the GLaM converges within the 95% confidence intervals of the Monte Carlo estimates, and the spread of the box plots decreases significantly with N increasing.

Finally, we remark that when higher-order Sobol' indices are of interest, Monte Carlo simulations require additional runs of the original model. For example, 4×10^8 more model evaluations should be performed to obtain the reference values for the standard-deviation-based second-order Sobol' indices. This results in a large amount of total model runs, which become impracticable even with cheap models. In contrast, GLaM surrogates can be used without additional cost: the PCE-based method presented in Section 6.3.5.3 provides analytical higher-order indices by post-processing the PC coefficients. In this example, the leave-one-out errors of PCE built on the GLaM surrogates are of the order $\mathcal{O}(10^{-3})$, which justifies the use of PCE estimates. Based on the surrogate model, we observe that the largest standard-deviation-based second-order Sobol' indices, which translate parameters interactions, are I_0 and β, I_0 and γ. Both have a value 0.09, while the other second-order interactions are very small. As illustrated in Figure 6.12, I_0 has a total effect $S^{\sigma}_{T_2} = 0.26$. Moreover, it has a relatively small first-order effect $S^{\sigma}_2 = 0.05$. This implies that I_0 mainly affects the variance of $\sigma(\boldsymbol{X})$ through its interactions with β and γ.

6.5 Conclusions

In this paper, we discuss the nature and focus of three extensions of Sobol' indices to stochastic simulators: classical Sobol' indices, QoI-based Sobol' indices and trajectory-based Sobol' indices. The first two types are of interest because of their versatility and applicability to a broad class of problems. We propose to use the generalized lambda model as a stochastic emulator to estimate the considered indices. This surrogate model aims at emulating the entire response distribution, instead of focusing only on some scalar statistical quantities, e.g., mean and variance. More precisely, it relies on using the four-parameter generalized lambda distribution to approximate the response distribution. The associated distribution parameters as functions of the input are represented by polynomial chaos expansions. Such a surrogate can be constructed without the need for replications, and thus it is not restricted to a special data structure.

Because of the special formulation of GLaM, the considered sensitivity indices can be estimated by directly working with deterministic functions. This allows applying the methods developed for deterministic simula-

tors, namely estimators based on Monte Carlo simulations and on polynomial chaos expansions. In this paper, we suggest the latter to post-process the surrogate model to achieve high computational efficiency.

The performance of the proposed method for estimating various Sobol' indices is illustrated on three examples with different signal-to-noise ratios. The toy example is designed to have a strong heteroskedastic effect. It shows the general convergent behavior of GLaMs for approximating the conditional quantile functions and estimating the entropy of the response distributions. The second example is a Heston model from mathematical Finance. This case study has a very small signal-to-noise ratio and demonstrates a shape variation of the response PDF. The surrogate generally yields accurate estimate of the Sobol' indices associated to the expected payoff and the 95%-superquantile. The last example is a stochastic SIR model in epidemiology, in which GLaMs exhibit robust estimates of the classical Sobol' indices and the standard-deviation-based Sobol' indices. All three examples have a different ranking of the input variables depending on the type of Sobol' indices, which is correctly captured by GLaMs when comparing to reference values obtained by extremely costly Monte Carlo simulations. Fairly accurate results are obtained at a cost of $\mathcal{O}\left(10^4\right)$ runs of the simulator compared to reference values based on $\mathcal{O}\left(10^8\right)$ runs by a brute force approach.

In future work, we plan to develop algorithms to improve GLaMs for small data sets. Besides, we will investigate GLaMs for estimating distribution-based sensitivity indices (Borgonovo, 2007; Huoh, 2013). The estimation of these indices usually requires a large number of model runs to infer the conditional PDF, which can be easily obtained from GLaMs. In addition, appropriate contrast measures between distributions, such as the Wasserstein metric, can be developed for sensitivity analysis in the context of stochastic simulators. Finally, developing sensitivity indices for stochastic simulators with dependent input variables will allow engineers to tackle a broader group of problems.

Acknowledgements

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524), the support of which is gratefully acknowledged.

6.a Appendix

6.a.1 Some properties of GLDs

The mean and variance of a GLD can be calculated by

$$m = \lambda_1 - \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3+1} - \frac{1}{\lambda_4(\boldsymbol{x})+1}\right), \tag{6.38}$$

$$v = \frac{(d_2 - d_1^2)}{\lambda_2^2}, \tag{6.39}$$

where $\{d_k : k = 1, 2\}$ are defined by

$$\begin{aligned} d_1 &= \frac{1}{\lambda_3} \mathrm{B}(\lambda_3 + 1, 1) - \frac{1}{\lambda_4} \mathrm{B}(1, \lambda_4 + 1), \\ d_2 &= \frac{1}{\lambda_3^2} \mathrm{B}(2\lambda_3 + 1, 1) - \frac{2}{\lambda_3 \lambda_4} \mathrm{B}(\lambda_3 + 1, \lambda_4 + 1) + \frac{1}{\lambda_4^2} \mathrm{B}(1, 2\lambda_4 + 1), \end{aligned} \tag{6.40}$$

with B denoting the beta function.

The expected payoff defined in Eq. (6.41) of a GLD with the strike price K is given by

$$\begin{aligned} m_C &\stackrel{\text{def}}{=} \mathbb{E}\left[\max\left\{Y - K, 0\right\}\right] \\ &= \left(\lambda_1 - \frac{1}{\lambda_2 \lambda_3} + \frac{1}{\lambda_2 \lambda_4} - K\right)(1 - u_K) + \frac{1}{\lambda_2}\left(\frac{1 - u_K^{\lambda_3 + 1}}{\lambda_3\ (\lambda_3 + 1)} - \frac{(1 - u_K)^{\lambda_4 + 1}}{\lambda_4\ (\lambda_4 + 1)}\right). \end{aligned} \tag{6.41}$$

where u_K is the solution of the nonlinear equation:

$$Q(u_K; \boldsymbol{\lambda}) = K. \tag{6.42}$$

The α-superquantile sq_α defined in Eq. (6.35) of a GLD has a closed-form:

$$\begin{aligned} \mathrm{sq}_\alpha &\stackrel{\text{def}}{=} \mathbb{E}\left[Y \mid Y > q_\alpha\right] \\ &= \lambda_1 - \frac{1}{\lambda_2 \lambda_3} + \frac{1}{\lambda_2 \lambda_4} + \frac{1}{(1 - \alpha)\lambda_2}\left(\frac{1 - \alpha^{\lambda_3 + 1}}{\lambda_3\ (\lambda_3 + 1)} - \frac{\alpha^{\lambda_4 + 1}}{\lambda_4\ (\lambda_4 + 1)}\right). \end{aligned} \tag{6.43}$$

References

Acerbi, C. (2002). Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking and Finance*, 26:1505–1518.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions—application to computational electromagnetic dosimetry. *International Journal for Uncertainty Quantification*, 9:351–363.

Azzi, S., Sudret, B., and Wiart, J. (2020). Sensitivity analysis for stochastic simulators using differential entropy. *International Journal for Uncertainty Quantification*, 10:25–33.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15(1–3):81–92.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27:808–821.

Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Borgonovo, E. (2007). A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92:771–784.

Borgonovo, E. (2017). *Sensitivity analysis – An Introduction for the Management Scientist*. Springer.

Britton, T. (2010). Stochastic epidemic models: A survey. *Mathematical Biosciences*, 225:24–35.

Browne, T., Iooss, B., Le Gratiet, L., Lonchampt, J., and Rémy, E. (2016). Stochastic simulators based optimization by Gaussian process metamodels—application to maintenance investments planning issues. *Quality and Reliability Engineering International*, 32(6):2067–2080.

Burden, R. L., Faires, J. D., and Burden, A. M. (2015). *Numerical analysis*. Cengage Learning.

Dacidian, M. and Carroll, R. J. (1987). Variance function estimation. *Journal of the American Statistical Association*, 400:1079–1091.

de Rocquigny, E., Devictor, N., and Tatantola, S. (2008). *Uncertainty in Industrial Practice: A guide to Quantitative Uncertainty Management*. John Wiley & Sons, New York.

Efron, B. (1979). Bootstrap methods: another look at the Jackknife. *The Annals of Statistics*, 7(1):1–26.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32:407–499.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Communications in Statistics - Theory and Methods*, 17:3547–3567.

Ghanem, R. G. and Spanos, P. (2003). *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, Mineola, 2nd edition.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81:2340–2361.

Hart, J. L., Alexanderian, A., and Gremaud, P. A. (2016). Efficient computation of Sobol' indices for stochastic models. *SIAM Journal on Scientific Computing*, 39:1514–1530.

Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B. (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10):1175–1209.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6:327–343.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of non linear models. *Reliability Engineering & System Safety*, 52:1–17.

Huoh, Y. (2013). *Sensitivity Analysis of Stochastic Simulators with Information Theory*. PhD thesis, University of California, Berkeley.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94:1194–1204.

Janon, A., Klein, T., Lagnoux, A., Nodet, M., and Prieur, C. (2014). Asymptotic normality and efficiency of two Sobol' index estimators. *ESAIM: Probability and Statistics*, 18:342–364.

Jimenez, M. N., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 5:378–402.

Karian, Z. A. and Dudewicz, E. J. (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press.

Marelli, S. and Sudret, B. (2019). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-104.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22:833–847.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ.

Rouah, F. D. (2013). *The Heston Model in Matlab and C#*. John Wiley & Sons, New Jersey.

Saltelli, A., Chan, K., and Scott, E. M., editors (2000). *Sensitivity analysis*. John Wiley & Sons.

Shreve, S. (2004). *Stochastic Calculus for Finance II*. Springer, New York.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical and Computer Modelling*, 1:407–414.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93:964–979.

Sudret, B. (2015). Polynomial chaos expansions and stochastic finite element methods. In Phoon, K.-K. and Ching, J., editors, *Risk and Realiability in Geotechnical Engineering*, Risk and Reliability in Geotechnical Engineering, chapter 6, pages 265–300. Taylor and Francis.

Villani, C. (2009). *Optimal Transport, Old and New*. Springer, Berlin.

Wooldridge, J. M. (2013). *Introductory Econometrics: A Modern Approach*. Cengage Learning, 5th edition.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

7

Stochastic polynomial chaos expansions to emulate stochastic simulators

This chapter is a post-print of

> Zhu, X. and Sudret, B. (2023). Stochastic polynomial chaos expansions to emulate stochastic simulators, *International Journal for Uncertainty Quantification*, 13:31–52. DOI:10.1615/Int.J.UncertaintyQuantification.2022042912. [1]

differing from the published paper only in terms of layout and formatting.

Author contributions. **X. Zhu**: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - Original Draft, Visualization. **B. Sudret**: Supervision, Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition.

Abstract

In the context of uncertainty quantification, computational models are required to be repeatedly evaluated. This task is intractable for costly numerical models. Such a problem turns out to be even more severe for stochastic simulators, the output of which is a random variable for a given set of input parameters. To alleviate the computational burden, surrogate models are usually constructed and evaluated instead. However, due to the random nature of the model response, classical surrogate models cannot be applied directly to the emulation of stochastic simulators. To efficiently represent the probability distribution of the model output for any given input values, we develop a new stochastic surrogate model called *stochastic polynomial chaos expansions*. To this aim, we introduce a latent variable and an additional noise variable, on top of the well-defined input variables, to reproduce the stochasticity. As a result, for a given set of input parameters, the model output is given by a

[1] First published in International Journal of Uncertainty Quantification in Volume 13, Issue 2, 2023, published by Begell House, Inc. Copyright © by Begell House, Inc.

function of the latent variable with an additive noise, thus a random variable. As the latent variable is purely artificial and does not have physical meanings, conventional methods (pseudo-spectral projections, collocation, regression, etc.) cannot be used to build such a model. In this paper, we propose an adaptive algorithm which does not require repeated runs of the simulator for the same input parameters. The performance of the proposed method is compared with the generalized lambda model and a state-of-the-art kernel estimator on two case studies in mathematical finance and epidemiology and on an analytical example whose response distribution is bimodal. The results show that the proposed method is able to accurately represent general response distributions, i.e., not only normal or unimodal ones. In terms of accuracy, it generally outperforms both the generalized lambda model and the kernel density estimator.

7.1 Introduction

In modern engineering, computational models, a.k.a. simulators, are commonly used to simulate different operational scenarios of complex systems *in silico*. These models help engineers assess the reliability, control the risk, and optimize the system components in the design phase. Conventional simulators are usually deterministic: a given set of input parameters has a unique corresponding model response. In other words, repeated model evaluations with the same input values will always give identical results. In contrast, stochastic simulators return different outcomes of the model response when run twice with the same input parameters.

Stochastic simulators are widely used in engineering and applied science. The intrinsic stochasticity typically represents some uncontrollable effect in the system (McNeil et al., 2005; Britton, 2010). For example, in mathematical finance, Brownian motions are commonly introduced to represent stochastic effects and volatility of the stock market (McNeil et al., 2005). In epidemic simulations, additional random variables on top of the well-defined characteristic values of the population are used to simulate the stochastic spread of a disease (Britton, 2010).

Mathematically, a stochastic simulator can be viewed as a function

$$\begin{aligned} \mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\to \mathbb{R}, \\ (\boldsymbol{x}, \omega) &\mapsto \mathcal{M}_s(\boldsymbol{x}, \omega), \end{aligned} \tag{7.1}$$

where $\mathcal{D}_{\boldsymbol{X}}$ is the domain of the input parameters, and Ω denotes the probability space that represents the internal stochasticity. The latter is due to some latent random variables $\boldsymbol{\Xi}(\omega)$ which are not explicitly considered as a part of the input variables. The stochastic simulator can then be considered as a deterministic function of the input vector $\boldsymbol{x}$ and the latent variables $\boldsymbol{\Xi}$. However, it is assumed that one can only control $\boldsymbol{x}$ but not $\boldsymbol{\Xi}$ when evaluating the model. Hence, when the value of $\boldsymbol{x}$ is fixed but $\boldsymbol{\Xi}$ is generated randomly following the underlying probability distribution, the output remains random.

In practice, each model evaluation for a fixed vector of input parameters $\boldsymbol{x}_0$ uses a particular realization of the latent variables, i.e., a particular $\omega_0 \in \Omega$ that is usually controlled by the random seed. Thus, it provides only one realization of the output random variable. In order to fully characterize the associated distribution of $\mathcal{M}_s(\boldsymbol{x}_0, \cdot)$, it is necessary to repeatedly run the stochastic simulator with the same input parameters $\boldsymbol{x}_0$. The various output values obtained by this procedure are called *replications* in the sequel.

In the context of uncertainty quantification or optimization, various input values should be investigated. To this aim, multiple runs of the simulator are needed for many different inputs and for many replications. This becomes impracticable for high-fidelity costly numerical models. In this context, surrogate models have received tremendous attention in the past two decades. A surrogate model is a proxy of the original model constructed from a limited number of model runs. However, standard surrogate models such as polynomial chaos expansions (Ghanem and Spanos, 2003) and Gaussian processes (Rasmussen and Williams, 2006) that have been successfully developed for deterministic simulators are not directly applicable to emulating stochastic simulators due to the random nature of the latter.

In the past decade, large efforts have been dedicated to estimating some summary quantities of the response distribution which are deterministic functions of the input.

For the mean and variance of the response distribution, Ankenman et al. (2010) proposed using replications to estimate the mean and variance for various input values. The mean function is represented by a Gaussian process, for which the variance estimated from the replications is cast as a heteroskedastic effect. Marrel et al. (2012) modeled both the mean and variance by Gaussian processes. The estimation procedure is similar to the feasible generalized least-squares (Wooldridge, 2013) that consists in alternating between fitting the mean from the data and the variance from the residuals. This approach does not require replications. Binois et al. (2018) proposed jointly optimizing the likelihood to represent the mean and variance by Gaussian processes, which is mainly designed for data with replications.

To estimate the quantiles of the response distribution, Koenker and Bassett (1978) proposed optimizing the *check function*, which established the quantile regression method. Plumlee and Tuo (2014) suggested estimating the quantiles by performing replications and building a Gaussian process from the estimated quantiles. The reader is referred to Torossian et al. (2020) for a detailed review.

The methods listed above produce only targeted summary quantities. However, far less literature has been devoted to the emulation of the entire probability distribution function of the response random variable for a given input. Three types of methods can be found in the literature.

Moutoussamy et al. (2015) proposed using replications to characterize the response distribution for different input values. Then, the fitted distributions (based on replications) for the discrete input values can be extended to the entire input space by parametric or nonparametric techniques. Since this approach capitalizes on replications for local inference, it is necessary to generate many replications to obtain an accurate surrogate (Zhu and Sudret, 2020), i.e., in the order of $10^3 - 10^4$ (Moutoussamy et al., 2015).

In the second approach, a stochastic simulator is considered as a random field indexed by the input variables (Azzi et al., 2019; Lüthen et al., 2022b). When fixing the internal stochasticity ω in Eq. (7.1), the stochastic simulator is a mere deterministic function of $\boldsymbol{x}$, called *a trajectory*. This function can be emulated by standard surrogate methods. Collecting different trajectories, one can approximate the underlying random field using Karhunen–Loève expansions. Therefore, it is necessary to fix the internal randomness to apply this approach, which is practically achieved by controlling the random seed.

The third type of methods is referred to as the statistical approach and does not require replications or manipulating the random seed. If the response distribution belongs to the exponential family, generalized linear models (McCullagh and Nelder, 1989) and generalized additive models (Hastie and Tibshirani, 1990) can be efficiently applied. For arbitrary types of response distributions, nonparametric estimators developed in statistics can be applied, namely kernel density estimators (Fan and Gijbels, 1996; Hall et al., 2004) and projection esti-

mators (Efromovich, 2010). However, nonparametric estimators are known to suffer from the *curse of dimensionality*, which indicates that the necessary amount of data increases drastically with increasing input dimensionality. To balance between very restrictive parametric assumptions and nonparametric approaches, Zhu and Sudret (2021a,b) proposed using generalized lambda distributions to approximate the response distributions. The four distribution parameters are seen as functions of the input and further represented by polynomial chaos expansions. The main limitation of this approach is that it cannot produce multimodal distributions, however.

In this paper, we develop an original approach that directly emulates the functional representation in Eq. (7.1). More precisely, we extend the classical polynomial chaos expansions to emulating stochastic simulators. We introduce a latent variable and a noise variable to reproduce the random behavior of the model output. We develop an adaptive method to construct such a surrogate model. This novel stochastic surrogate is parametric and shown to be not limited to unimodal distributions.

The remainder of the paper is organized as follows. In Section 7.2, we first review the standard polynomial chaos representations. In Section 7.3, we present a novel formulation named *stochastic polynomial chaos expansions* which is meant for stochastic simulators. In Section 7.4, we present the algorithms to adaptively build such a surrogate from data without the need for replications. We illustrate the performance of the proposed method on a complex analytical example and on case studies from mathematical finance and epidemiology in Section 7.5. Finally, we conclude the main findings of the paper and provide outlooks for future research in Section 7.6.

7.2 Reminder on polynomial chaos expansions

Polynomial chaos expansions (PCEs) have been widely used in the last two decades to emulate the response of deterministic simulators in many fields of applied science and engineering. Consider a deterministic model $\mathcal{M}_d$ which is a function that maps the input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)^\intercal \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$ to the scalar output $y = \mathcal{M}_d(\boldsymbol{x}) \in \mathbb{R}$. In the context of uncertainty quantification, the input vector $\boldsymbol{x}$ is affected by uncertainties and thus modeled by a random vector $\boldsymbol{X}$ with prescribed joint probability density function (PDF) denoted by $f_{\boldsymbol{X}}$. In the sequel, we focus on the case where the input parameters are independent for simplicity. Therefore, the joint PDF is expressed by

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{j=1}^{M} f_{X_j}(x_j), \tag{7.2}$$

where f_{X_j} is the marginal PDF of the input random variable X_j. Note that in the case where the input vector $\boldsymbol{X}$ has dependent components, it is always possible to transform them into independent ones using the Nataf or Rosenblatt transform (Nataf, 1962; Rosenblatt, 1952; Blatman and Sudret, 2010).

Because of the randomness in the input, the model response $Y = \mathcal{M}_d(\boldsymbol{X})$ becomes a random variable. Provided that Y has a finite variance, i.e., $\mathrm{Var}[Y] < +\infty$, the function $\mathcal{M}_d$ belongs to the Hilbert space $\mathcal{H}$ of square-integrable functions with respect to the inner product

$$\langle u, v \rangle_{\mathcal{H}} \overset{\text{def}}{=} \mathbb{E}[u(\boldsymbol{X})v(\boldsymbol{X})] = \int_{\mathcal{D}_{\boldsymbol{X}}} u(\boldsymbol{x})v(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{7.3}$$

Under certain conditions on the joint PDF $f_{\boldsymbol{X}}$ (Ernst et al., 2012), the Hilbert space $\mathcal{H}$ possesses a polynomial

basis. As a result, $\mathcal{M}_d$ can be represented by an orthogonal series expansion

$$\mathcal{M}_d(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{7.4}$$

where $c_{\boldsymbol{\alpha}}$ is the coefficient associated with the basis function $\psi_{\boldsymbol{\alpha}}$ that is defined by the multi-index $\boldsymbol{\alpha}$. More precisely, the multivariate basis function $\psi_{\boldsymbol{\alpha}}$ is given by a tensor product of univariante polynomials

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j), \tag{7.5}$$

where α_j indicates the degree of $\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ in its j-th component x_j, and $\left\{\phi_k^{(j)} : k \in \mathbb{N}\right\}$ is the orthogonal polynomial basis with respect to the marginal distribution f_{X_j} of X_j, which satisfies

$$\mathbb{E}\left[\phi_k^{(j)}(X_j)\, \phi_l^{(j)}(X_j)\right] = \delta_{kl}. \tag{7.6}$$

In the equation above, the Kronecker symbol δ_{kl} is such that $\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise.

Following Eq. (7.5), the multivariate polynomial basis is defined from univariate orthogonal polynomials that depend on the corresponding marginal distribution. For uniform, normal, gamma and beta distributions, the associated orthogonal polynomial families are known analytically (Xiu and Karniadakis, 2002). For arbitrary marginal distributions, such a basis can be iteratively computed by the *Stieltjes procedure* (Gautschi, 2004).

The spectral representation in Eq. (7.4) involves an infinite sum of terms. In practice, the series needs to be truncated to a finite sum. The standard truncation scheme is defined by selecting all the polynomials whose total degree is small than a given value p, i.e., $\mathcal{A}^{p,M} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M, \sum_{j=1}^{M} \alpha_j \leqslant p\right\}$. However, this will provide a large number of terms for big values of p and M. A more flexible scheme is the hyperbolic (q-norm) truncation scheme (Blatman and Sudret, 2011):

$$\mathcal{A}^{p,q,M} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M, \|\boldsymbol{\alpha}\|_q \leqslant p\right\}, \tag{7.7}$$

where p is the maximum polynomial degree, and $q \in (0, 1]$ defines the quasi-norm $\|\boldsymbol{\alpha}\|_q = \left(\sum_{j=1}^{M} |\alpha_j|^q\right)^{1/q}$. This truncation scheme allows excluding high-order interactions among the input variables but keeps univariate effects up to degree p. Note that with $q = 1$, we recover the full basis of total degree less than p.

To estimate the coefficients in Eq. (7.4), one popular approach relies on minimizing the mean-squared error between the model response and the surrogate model. The basic method applies ordinary least-squares (OLS) with a given set of basis (e.g., defined by a truncation scheme; Berveiller et al., 2006). In this approach, the model is evaluated on a number of points called the *experimental design* $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$. The associated model responses are gathered into $\boldsymbol{y} = \left\{y^{(1)}, \ldots, y^{(N)}\right\}$ with $y^{(i)} = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right)$. The basis functions (and thus the coefficients) can be arranged by ordering the multi-indices $\{\boldsymbol{\alpha}_j\}_{j=1}^{P}$. The regression matrix $\boldsymbol{\Psi}$ is defined by $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}\left(\boldsymbol{x}^{(i)}\right)$. By minimizing the mean-squared error between the original model and the surrogate on the experimental design, the OLS estimator is given by

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}\|_2^2 \tag{7.8}$$

With increasing polynomial degree or input dimension, the number of coefficients increases drastically. As a consequence, a large number of models runs are necessary to guarantee a good accuracy, which becomes intractable for costly simulators. To solve this problem, Blatman and Sudret (2011), Doostan and Owhadi (2011), and Babacan et al. (2010) developed methods to build sparse PCEs by only selecting the most influential polynomials. The reader is referred to the review papers by Lüthen et al. (2021, 2022a) for more details.

7.3 Stochastic polynomial chaos expansions

7.3.1 Introduction

Let us now come back to stochastic simulators. It would be desirable to have a spectral expansion such as Eq. (7.4) for stochastic simulators. Indeed, the standard PCE has numerous features such as close-to-zero-cost model evaluations, and clear interpretation of the coefficients in terms of sensitivity analysis (Sudret, 2008). However, because the spectral expansion in Eq. (7.4) is a deterministic function of the input parameters, it cannot be directly used to emulate stochastic simulators.

Considering the randomness in the input variables, the output of a stochastic simulator is a random variable. The randomness of the latter comes from both the intrinsic stochasticity and the uncertain inputs. When fixing the input parameters, the model response remains random. For the purpose of clarity, we denote by $Y_{\boldsymbol{x}}$ the random model response for the input parameters $\boldsymbol{x}$ and by Y the model output containing all the uncertainties: following Eq. (7.1), we have

$$Y_{\boldsymbol{x}} \overset{\text{def}}{=} \mathcal{M}_s(\boldsymbol{x}, \omega), \quad Y \overset{\text{def}}{=} \mathcal{M}_s(\boldsymbol{X}(\omega), \omega). \tag{7.9}$$

From a probabilistic perspective, $Y_{\boldsymbol{x}}$ is equivalent to the conditional random variable $Y \mid \boldsymbol{X} = \boldsymbol{x}$. Let $F_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x})$ denote the associated cumulative distribution function (CDF). By using the probability integral transform, we can transform *any* continuous random variable Z to the desired distribution, that is

$$Y_{\boldsymbol{x}} \overset{\text{d}}{=} F^{-1}_{Y|\boldsymbol{X}}\left(F_Z(Z) \mid \boldsymbol{x}\right), \tag{7.10}$$

where F_Z is the CDF of Z. The equality in Eq. (7.10) is to be understood *in distribution*, meaning that two random variables on the left- and right-hand side follow the same distribution. In Eq. (7.10), the right-hand side is a deterministic function of both $\boldsymbol{x}$ and z. As a result, assuming that Y has a finite variance, we can represent this function using a PCE in the $(\boldsymbol{X}, Z)$ space, that is,

$$F^{-1}_{Y|\boldsymbol{X}}\left(F_Z(Z) \mid \boldsymbol{X}\right) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^{M+1}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}, Z). \tag{7.11}$$

For a given vector of input parameters $\boldsymbol{x}$, the expansion is a function of the artificial latent variable Z, thus a random variable

$$Y_{\boldsymbol{x}} \overset{\text{d}}{=} \sum_{\boldsymbol{\alpha} \in \mathbb{N}^{M+1}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z). \tag{7.12}$$

Then, we apply a truncation scheme $\mathcal{A}$ (e.g., Eq. (7.7)) to reduce Eq. (7.12) to a finite sum

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{\approx} \tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z \right). \tag{7.13}$$

Even though Eq. (7.13) is derived from Eq. (7.11), it is more general. Eq. (7.10) offers one way to represent the response distribution by a transform of a latent variable. But many other transforms can achieve the same goal. For example, using $Z \sim \mathcal{N}(0,1)$, both $\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})Z$ and $\mu(\boldsymbol{x}) - \sigma(\boldsymbol{x})Z$ can represent the stochastic simulator defined by $Y_{\boldsymbol{x}} \sim \mathcal{N}(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$. Because we are interested in the response distribution, Eq. (7.13) only requires that the polynomial transform of the latent variable produces a distribution that is close to the response distribution, but the transform does not need to follow Eq. (7.11) exactly. Note that the latent variable Z is only introduced to reproduce the stochasticity, but it does not allow us to represent the detailed data generating process of the simulator though. In other words, the PCE in Eq. (7.13) cannot emulate the response for a particular replication, yet it provides a representation of the distribution of $Y_{\boldsymbol{x}}$.

7.3.2 Potential issues with the formulation in Eq. (7.13)

Building a PCE by least-squares as presented in Section 7.2 requires evaluating the deterministic function to surrogate, which, in the case of stochastic simulators, is the left-hand side of Eq. (7.11). However, it is practically impossible to evaluate such a function, as the response distribution $F^{-1}_{Y|\boldsymbol{X}}$ is unknown. One common way to fit the latent variable model defined in Eq. (7.13) is maximum likelihood estimation (Everitt, 1984; Desceliers et al., 2006). In this section, we show some potential problems associated with a standard use of this method for building Eq. (7.13), which calls for a novel fitting algorithm.

According to the definition in Eq. (7.13), $\tilde{Y}_{\boldsymbol{x}}$ is a function of Z. Denote $f_Z(z)$ the PDF of Z and $\mathcal{D}_Z$ the support of Z. Based on a change of variable (Jacod and Protter, 2004), we can obtain the PDF of $\tilde{Y}_{\boldsymbol{x}}$, which is denoted by $f_{\tilde{Y}_{\boldsymbol{x}}}(y; \boldsymbol{x}, \boldsymbol{c})$. As a result, the (conditional) likelihood function of the coefficients $\boldsymbol{c}$ for a data point $(\boldsymbol{x}, y)$ is given by

$$l(\boldsymbol{c}; \boldsymbol{x}, y) = f_{\tilde{Y}_{\boldsymbol{x}}}(y; \boldsymbol{x}, \boldsymbol{c}). \tag{7.14}$$

Now, let us consider an experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \right\}$. The stochastic simulator is assumed to be evaluated *once* for each point $\boldsymbol{x}^{(i)}$, yielding $\boldsymbol{y} = \left\{ y^{(1)}, \ldots, y^{(N)} \right\}$ with $y^{(i)} = \mathcal{M}_s \left(\boldsymbol{x}^{(i)}, \omega^{(i)} \right)$. Note that here we do not control the random seed, so the model outcomes for different values of $\boldsymbol{x}$ are *independent*. Thus, the likelihood function can be computed by the product of $l \left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)} \right)$ over the N data points. As a result, the maximum likelihood estimator is given by

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i=1}^{N} \log l \left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)} \right). \tag{7.15}$$

Eq. (7.15) commonly serves as a basic approach for fitting parametric statistical models (including stochastic surrogates; McCullagh and Nelder, 1989; Hastie et al., 2001; Zhu and Sudret, 2021a). However, the likelihood function of the latent PCE defined in Eq. (7.13) is unbounded and can reach $+\infty$, making the maximization problem Eq. (7.15) ill-posed.

To illustrate the issue, let us consider a simple stochastic simulator without input variables, which gives a

realization of Y upon each model evaluation. Hence, the surrogate in Eq. (7.13) contains only the latent variable Z, that is, $\tilde{Y} = g(Z) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(Z)$. For simplicity, let $g(z)$ be a second-degree polynomial expressed by monomials $g(z) = a_1 z^2 + a_2 z + a_3$. Note that there is a one-to-one mapping between monomials and full polynomial chaos basis, so one can map $\boldsymbol{a} = (a_1, a_2, a_3)^\intercal$ to $\boldsymbol{c}$ through a change of basis. Using a change of variable (Jacod and Protter, 2004), the PDF of $\tilde{Y}$ is

$$f_{\tilde{Y}}(y) = \frac{f_Z(z)}{|g'(z)|}\mathbb{1}_{g(z)}(y), \tag{7.16}$$

where $\mathbb{1}$ is the indicator function, and g' denotes the derivative of g. For a given y_0, certain choices of $\boldsymbol{a}$ can make any given z_0 with $f_Z(z_0) \neq 0$ satisfy $g(z_0) = y_0$ and $g'(z_0) = 0$:

$$\begin{cases} g(z_0) = y_0 \\ g'(z_0) = 0 \end{cases} \Rightarrow \begin{cases} a_1 z_0^2 + a_2 z_0 + a_3 - y_0 = 0 \\ 2a_1 z_0 + a_2 = 0 \end{cases} \Rightarrow \begin{cases} -z_0^2\, a_1^2 + a_3 - y_0 = 0 \\ a_2 = -2 z_0\, a_1 \end{cases}. \tag{7.17}$$

The system of equations in Eq. (7.17) is underdetermined for $\boldsymbol{a}$. Therefore, there are infinite combinations of the coefficients $\boldsymbol{a}$, and therefore of $\boldsymbol{c}$, such that the denominator of Eq. (7.16) is zero and the numerator is non-zero, which gives $f_{\tilde{Y}}(y_0) = +\infty$. Consequently, the maximum likelihood estimation will always produce a certain vector $\boldsymbol{c}$ that makes the likelihood reach $+\infty$.

As a conclusion, the surrogate ansatz of Eq. (7.13) can produce non-smooth conditional PDFs with singularity points where $f_{\tilde{Y}_{\boldsymbol{x}}}$ tends to infinity. Consequently, the standard maximum likelihood estimation would fail.

7.3.3 Formulation of stochastic polynomial chaos expansions

In the previous section, we discussed some potential problems of the model defined in Eq. (7.13). To regularize the optimization problem in Eq. (7.15) and smooth out the produced PDFs, we introduce an additive noise variable ϵ, and define the stochastic surrogate as follows:

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{\approx} \tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z) + \epsilon, \tag{7.18}$$

where ϵ is a centered Gaussian random variable with standard deviation σ, i.e., $\epsilon \sim \mathcal{N}(0, \sigma^2)$. With this new formulation, the response PDF of the stochastic surrogate is a convolution of that of the PCE and the Gaussian PDF of ϵ. Let $G_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)$. The PDF of $\tilde{Y}_{\boldsymbol{x}} = G_{\boldsymbol{x}} + \epsilon$ reads

$$f_{\tilde{Y}_{\boldsymbol{x}}}(y) = (f_{G_{\boldsymbol{x}}} * f_\epsilon)(y) = \int_{-\infty}^{+\infty} f_{G_{\boldsymbol{x}}}(y-t) f_\epsilon(t)\mathrm{d}t. \tag{7.19}$$

Using Hölder's inequality, the above integral is bounded from above by

$$\|f_{G_{\boldsymbol{x}}}\|_1 \, \|f_\epsilon\|_\infty = \|f_\epsilon\|_\infty = \frac{1}{\sigma\sqrt{2\pi}}, \tag{7.20}$$

meaning that the PDF of $\tilde{Y}_{\boldsymbol{x}}$ and the associated likelihood function are bounded.

To illustrate the role of the additive noise variable in Eq. (7.18), let us consider a random variable Y with bimodal distribution to be represented by

$$Y \overset{\mathrm{d}}{\approx} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(Z) + \epsilon, \tag{7.21}$$

where the latent variable Z follows a standard normal distribution and $\epsilon \sim \mathcal{N}(0, \sigma)$. In the case $\sigma = 0$ (the noise term vanishes), we build the model by applying a standard algorithm such as least-angle regression (LAR; Blatman and Sudret, 2011) to the probability integral transform $F_Y^{-1}(F_Z(Z))$. When the regularization term ϵ is added, maximum likelihood estimation can be used (see Section 7.4.1 for details) to construct the surrogate.

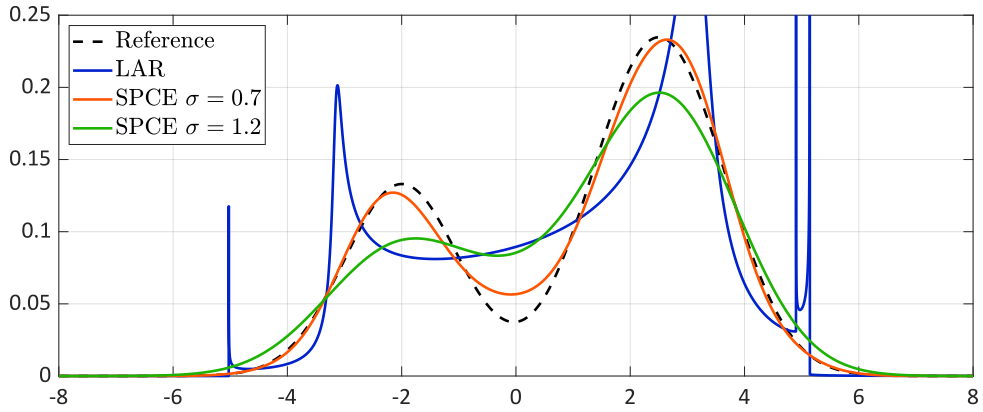


Figure 7.1: Emulating a bimodal distribution. The blue line corresponds to the result of using LAR to represent directly the probability integral transform (without regularization term). The red and green lines are the results of maximum likelihood estimation for two different values of σ.

Figure 7.1 shows the original (reference) PDF, and the ones obtained by LAR ($\sigma = 0$) and by the stochastic PCE for two different values of σ. It is observed that the PDF obtained by LAR has singularity points, which confirms the analysis in Section 7.3.2, whereas the proposed noise term regularizes the PDFs. Moreover, LAR is applied directly to the probability integral transform which in practice is unknown. In contrast, the maximum likelihood estimation does not require knowing the values of Z (in this example, only the realizations of Y are used). Finally, the value of σ affects the accuracy of the model. Hence, σ is an additional parameter of the model that must also be fitted to the data to get the optimal approximation. The fitting procedure is detailed in the next section.

7.4 Fitting the stochastic polynomial chaos expansion

To construct a stochastic PCE defined in Eq. (7.18), one needs to estimate both the coefficients $\boldsymbol{c}$ and the standard deviation σ of the noise variable. In this section, we present a method to calibrate these parameters from data without replications. Moreover, we propose an algorithm that adaptively selects an appropriate distribution for the latent variable Z and truncation scheme $\mathcal{A}$.

7.4.1 Maximum likelihood estimation

Let us assume for a moment that the standard deviation σ of the noise variable is given (the estimation of σ will be investigated separately in Section 7.4.4). From Eq. (7.18), we see that our surrogate response $\tilde{Y}_{\boldsymbol{x}}$ is the sum of a polynomial function of $(\boldsymbol{x}, z)$ and the noise variable ϵ. Therefore, its PDF can be computed by

$$\begin{aligned} f_{\tilde{Y}_{\boldsymbol{x}}}(y) &= \int_{\mathcal{D}_Z} f_{\tilde{Y}_{\boldsymbol{x}}|Z}(y \mid z) f_Z(z) \mathrm{d}z \\ &= \int_{\mathcal{D}_Z} \frac{1}{\sigma} \varphi \left(\frac{y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)}{\sigma} \right) f_Z(z) \, \mathrm{d}z, \end{aligned} \tag{7.22}$$

since $\tilde{Y}_{\boldsymbol{x}} \mid Z = z$ is a Gaussian random variable with mean value $\sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)$ and variance σ^2 according to Eq. (7.18). In this equation, φ stands for the standard normal PDF. Therefore, for a given data point $(\boldsymbol{x}, y)$, the likelihood of the parameters $\boldsymbol{c}$ conditioned on σ reads

$$l(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) = \int_{\mathcal{D}_Z} \frac{1}{\sqrt{2\pi}\sigma} \exp \left(- \frac{\left(y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z) \right)^2}{2\sigma^2} \right) f_Z(z) \mathrm{d}z. \tag{7.23}$$

In practice, we can use numerical integration schemes, namely Gaussian quadrature (Golub and Welsch, 1969), to efficiently evaluate this one-dimensional integral, that is

$$l(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) \approx \tilde{l}(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) = \sum_{j=1}^{N_Q} \frac{1}{\sqrt{2\pi}\sigma} \exp \left(- \frac{\left(y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z_j) \right)^2}{2\sigma^2} \right) w_j, \tag{7.24}$$

where N_Q is the number of integration points, z_j is the j-th integration point, and w_j is the corresponding weight, both associated to the weight function f_Z. Based on Eq. (7.24) and the available data $(\mathcal{X}, \boldsymbol{y})$, the PCE coefficients $\boldsymbol{c}$ can be fitted using the maximum likelihood estimation (MLE)

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i}^{N} \log \left(\tilde{l} \left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)}, \sigma \right) \right). \tag{7.25}$$

The gradient of Eq. (7.24), and therefore of Eq. (7.25), can be derived analytically. Hence, we opt for the derivative-based BFGS quasi-Newton method (Fletcher, 1987) to solve this optimization problem.

7.4.2 Starting point for the optimization

The objective function to optimize in Eq. (7.25) is highly nonlinear. As a result, a good starting point is necessary to ensure convergence. According to the properties of the polynomial chaos basis functions, the mean function of a stochastic PCE can be expressed as

$$\tilde{m}(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{E}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \mathbb{E}_{Z,\epsilon} \left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z \right) + \epsilon \right] = \sum_{\boldsymbol{\alpha} \in \mathcal{A}, \alpha_z = 0} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{7.26}$$

where α_z is the degree of the univariate polynomial in Z. Eq. (7.26) contains all the terms without Z, as indicated by $\alpha_z = 0$. We define this set of multi-indices as

$$\mathcal{A}_m = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_z = 0\}. \tag{7.27}$$

Another surrogate $\hat{m}(\boldsymbol{x})$ of the mean function can be obtained by using standard (or sparse) regression to directly fit the following expansion:

$$m(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{E}[Y_{\boldsymbol{x}}] \approx \hat{m}(\boldsymbol{x}) \overset{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} c^m_{\boldsymbol{\alpha}} \psi(\boldsymbol{x}). \tag{7.28}$$

The obtained coefficients $\boldsymbol{c}^m$ are used as initial values for the coefficients $\{\boldsymbol{c}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}_m\}$ of the stochastic surrogate in the optimization procedure, i.e., $\boldsymbol{c}_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathcal{A}_m$.

For the other coefficients $\{c_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A} \backslash \mathcal{A}_m\}$, we randomly initialize their value.

7.4.3 Warm-start strategy

Because of the form of the likelihood Eq. (7.23), the gradient at the starting point can take extremely large values when σ is small. In this case, the optimization algorithm may become unstable and converge to an undesired local optimum. To guide the optimization, we propose a warm-start strategy summarized in Algorithm 7.1. We generate a decreasing sequence $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_{N_s}\}$ with $\sigma_{N_s} = \sigma$ (the target value). In this paper, we choose the maximum value σ_1 of the sequence as the square root of the *leave-one-out error* ε_{LOO} in the mean fitting procedure (see Section 7.a.1 for the explanation of this choice). Then, $\boldsymbol{\sigma}$ is generated equally-spaced in the log-space between $\sqrt{\varepsilon_{\text{LOO}}}$ and σ. Starting with σ_1 which is the largest element of $\boldsymbol{\sigma}$, we build a stochastic PCE based on Eq. (7.25) with the initial values defined above (the mean function estimation and random initialization). Then, the results are used as a starting point for the construction of the surrogate for σ_2. We repeat this procedure sequentially for each element in $\boldsymbol{\sigma}$ with each new starting point being the results of the previous optimization. Because the standard deviation decreases progressively to the target value and the starting point is updated accordingly, the associated gradient for each optimization prevents extremely big values.

Algorithm 7.1 Warm-start approach for estimating $\boldsymbol{c}$ with known σ

Input: $(\mathcal{X}, \boldsymbol{y}), \sigma, \mathcal{A}$

Output: Coefficients $\hat{\boldsymbol{c}}$

$\boldsymbol{c}^m, \varepsilon_{\text{LOO}} \leftarrow \text{OLS}(\mathcal{X}, \boldsymbol{y}, \mathcal{A}_m)$ % Estimation of the coefficients of the mean function

$c^0_{\boldsymbol{\alpha}} \leftarrow c^m_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathcal{A}_m$ and randomly initialize $\{c^0_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A} \backslash \mathcal{A}_m\}$

$\boldsymbol{\sigma}_{\log} \leftarrow \text{linspace}\left(\log\left(\sqrt{\varepsilon_{\text{LOO}}}\right), \log(\sigma), N_s\right)$

$\boldsymbol{\sigma} \leftarrow \exp\left(\boldsymbol{\sigma}_{\log}\right)$

for $i \leftarrow 1, \ldots, N_s$ **do**

  Solve Eq. (7.25) to compute $\boldsymbol{c}^i$ using $\boldsymbol{c}^{i-1}$ as initial values

end for

$\hat{\boldsymbol{c}} \leftarrow \boldsymbol{c}^{N_s}$

7.4.4 Cross-validation

As explained in Section 7.3.2, the hyperparameter σ cannot be jointly estimated together with the PCE coefficients $\boldsymbol{c}$ because the likelihood function can reach $+\infty$ for certain choices of $\boldsymbol{c}$ and $\sigma = 0$. As a result, σ should be tuned separately from the estimation of $\boldsymbol{c}$.

In this paper, we propose applying cross-validation (CV; Hastie et al., 2001) to selecting the optimal value of σ. More precisely, the data $(\mathcal{X}, \boldsymbol{y})$ are randomly partitioned into N_{cv} equal-sized groups $\{V_k : k = 1, \dots, N_{\text{cv}}\}$ (so-called N_{cv}-fold CV). For $k \in \{1, \dots, N_{\text{cv}}\}$, we pick the k-th group V_k as the validation set and the other $N_{\text{cv}} - 1$ folds denoted by $V_{\sim k}$ as the training set. The latter is used to build a stochastic PCE following Eq. (7.25) and Algorithm 7.1, which yields

$$\hat{\boldsymbol{c}}_k(\sigma) = \arg\max_{\boldsymbol{c}} \sum_{i \in V_{\sim k}} \log\left(\tilde{l}\left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)}, \sigma\right)\right). \tag{7.29}$$

Note that the coefficients depend on the value of σ, and thus we explicitly write them as functions of σ. The validation set V_k is then used to evaluate the *out-of-sample* performance:

$$\mathrm{l}_k(\sigma) = \sum_{i \in V_k} \log\left(\tilde{l}\left(\hat{\boldsymbol{c}}_k(\sigma); \boldsymbol{x}^{(i)}, y^{(i)}, \sigma\right)\right). \tag{7.30}$$

We repeat this procedure for each group of the partition $\{V_k : k = 1, \dots, N_{\text{cv}}\}$ and sum up the respective score to estimate the generalized performance, referred to as *CV score* in the sequel. Then, the optimal value of σ is selected as the one that maximizes this CV score:

$$\hat{\sigma} = \arg\max_{\sigma} \sum_{k=1}^{N_{\text{cv}}} \mathrm{l}_k(\sigma). \tag{7.31}$$

Because of the nested optimization in Eq. (7.29), the gradient of Eq. (7.31) is difficult to derive. In this paper, we apply the derivative-free Bayesian optimizer (Snoek et al., 2012) to solving Eq. (7.31) and search for σ within the range $[0.1, 1] \times \sqrt{\varepsilon_{\text{LOO}}}$. The upper bound of the interval is explained in Section 7.a.1. The lower bound is introduced to prevent numerical instabilities near $\sigma = 0$. According to our investigations, the optimal value $\hat{\sigma}$ is always within the proposed interval.

After solving Eq. (7.31), the selected $\hat{\sigma}$ is used in Eq. (7.25) with all the available data to build the final surrogate.

Large value of N_{cv} can lead to high computational cost, especially when N is big. In this paper, we choose $N_{\text{cv}} = 10$ for $N < 200$ (small data set), $N_{\text{cv}} = 5$ for $200 \leqslant N < 1{,}000$ (moderate data set) and $N_{\text{cv}} = 3$ for $N \geqslant 1{,}000$ (big data set).

7.4.5 Adaptivity

The method developed in Sections 7.4.1 and 7.4.4 allows us to build a stochastic PCE for a given distribution of the latent variable Z and truncated set $\mathcal{A}$ of polynomial chaos basis. In principle, one can choose any continuous probability distribution for the latent variable and a large truncated set. However, in practice, certain types of latent variables may require a lot of basis functions to approximate well the shape of the response dis-

tribution. This leads to many model parameters to estimate, which would cause overfitting when only a few data are available. In this section, we propose a procedure to iteratively find a suitable distribution for the latent variable Z and truncation scheme $\mathcal{A}$.

We consider N_z candidate distributions $\boldsymbol{D} = \{D_1, \dots, D_{N_z}\}$ for the latent variable, N_p degrees $\boldsymbol{p} = \{p_1, \dots, p_{N_p}\}$ and N_q q-norms $\boldsymbol{q} = \{p_1, \dots, p_{N_p}\}$ that are used to define the hyperbolic truncation scheme in Eq. (7.7). Both $\boldsymbol{p}$ and $\boldsymbol{q}$ are sorted in increasing order.

The adaptive procedure is shown in Algorithm 7.2 and described here. For each type of latent variable and truncation set $\mathcal{A} = \mathcal{A}^{p,q,M}$, we first apply the hybrid LAR algorithm developed by Blatman and Sudret (2011) to fitting the mean function $\hat{m}(\boldsymbol{x})$ as shown in Eq. (7.28). This algorithm only selects the most important basis among the candidate set $\mathcal{A}_m$ defined in Eq. (7.27). To reduce the total number of unknowns in the optimization Eq. (7.25), we exclude from $\mathcal{A}$ the basis functions in $\mathcal{A}_m$ that are not selected by hybrid LAR. In other words, we only estimate the coefficients associated with the basis functions that either have $\alpha_z \neq 0$ or are selected by the hybrid LAR when fitting the mean function $m(\boldsymbol{x})$. Then, we use the methods presented in Sections 7.4.1 and 7.4.4 to build a stochastic PCE for $\mathcal{A}$ and record the CV score. The latter is used for model comparisons, and the one with the best CV score is selected as the final surrogate.

Algorithm 7.2 Adaptive algorithm for building a stochastic PCE

Input: $(\mathcal{X}, \boldsymbol{y}), \boldsymbol{D}, \boldsymbol{p}, \boldsymbol{q}$

Output: $D_{opt}, \mathcal{A}_{opt}, \hat{\boldsymbol{c}}, \hat{\sigma}$

$\mathrm{l}_{opt} \leftarrow -\infty$
for $i_z \leftarrow 1, \dots, N_z$ **do**
  Set $Z \sim D_{i_z}$
  for $i_p \leftarrow 1, \dots, N_p$ **do**
    for $i_q \leftarrow 1, \dots, N_q$ **do**
      $\mathcal{A} \leftarrow \mathcal{A}^{p_{i_p}, q_{i_q}, M+1}$
      $\mathcal{A}_m \leftarrow \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathcal{A}, \alpha_z = 0\}$, $\mathcal{A}_c \leftarrow \mathcal{A} \backslash \mathcal{A}_m$
      $\mathcal{A}_n \leftarrow$ Hybrid-LAR $(\mathcal{X}, \boldsymbol{y}, \mathcal{A}_m)$ % Selection of the basis for $\hat{m}(\boldsymbol{x})$
      $\mathcal{A} \leftarrow \mathcal{A}_n \bigcup \mathcal{A}_c$
      Apply the algorithm presented in Sections 7.4.1 and 7.4.4 to build a stochastic PCE with $\mathcal{A}$, which gives $\boldsymbol{c}$, σ, and the CV score l_{i_p, i_q} associated with σ.
    end for
  end for
end for
Return the model with the maximum CV score

In order to avoid going through all the possible combinations, we propose a heuristic *early stopping criterion* for both degree and q-norm adaptivity. If two consecutive increases of q-norm cannot improve the CV score, the inner loop for q-norm adaptivity stops. Besides, if the best model (among all the q-norms) of a larger degree decreases the CV score, the algorithm stops exploring higher degrees. Note that the early stopping is only applied to p- and q-adaptivity, but all the candidate distributions are investigated.

In summary, we sketch the overall procedure (presented in Sections 7.4.1 to 7.4.5) to adaptively build a stochastic PCE from data in Figure 7.2.

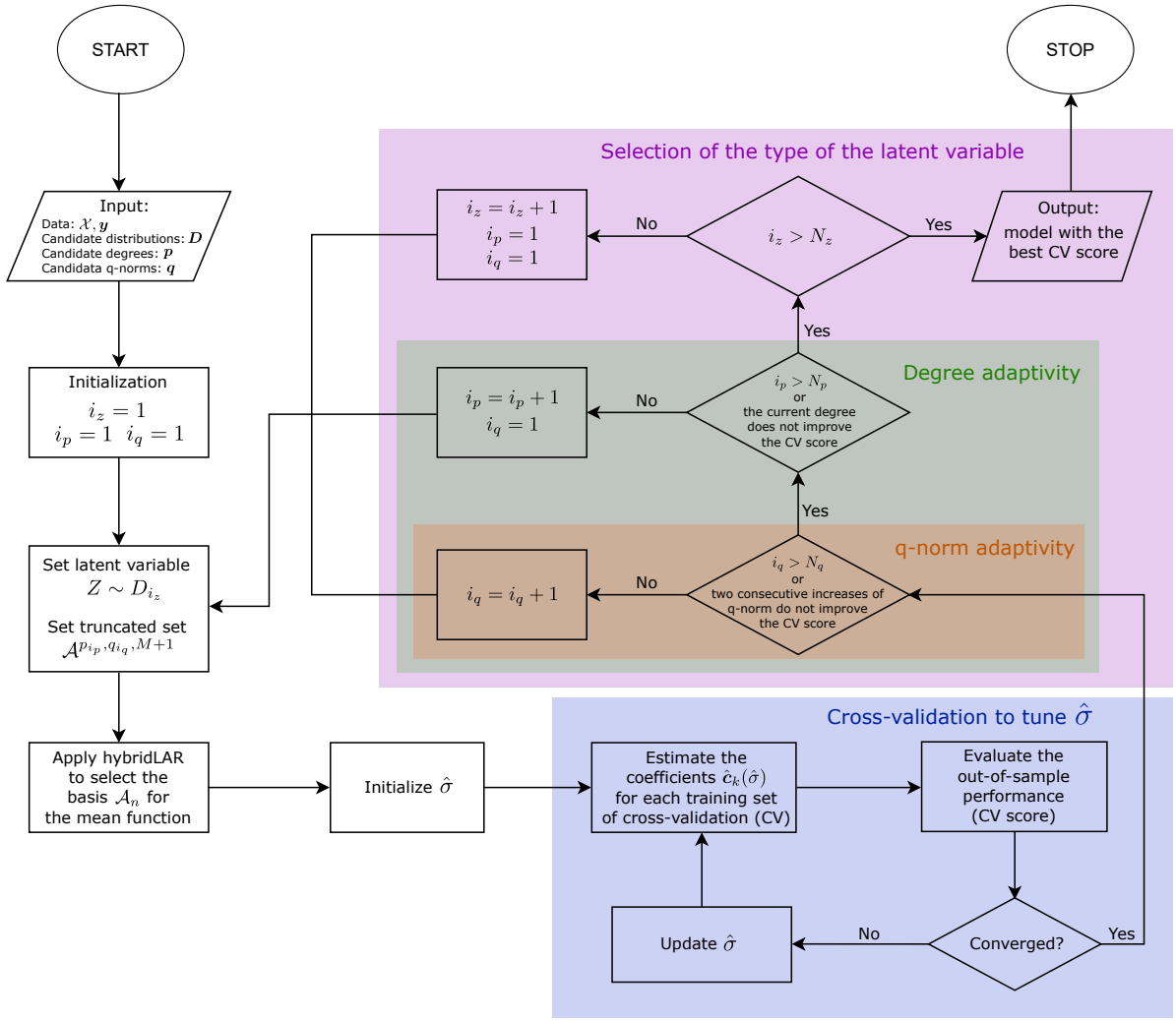


Figure 7.2: Flow chart of the procedure to adaptively build a stochastic PCE

In the application examples, we choose $N_Z = 2$ possible distributions for the latent variable Z, namely a standard normal distribution $\mathcal{N}(0,1)$ and a uniform distribution $\mathcal{U}(-1,1)$. The truncation parameters $\boldsymbol{p}$ and $\boldsymbol{q}$ may be selected according to the dimensionality M of the problem and the prior knowledge on the level of non-linearity. We typically use $\boldsymbol{p} = \{1,2,3,4,5\}$ and $\boldsymbol{q} = \{0.5, 0.75, 1\}$.

7.4.6 Post-processing of stochastic polynomial chaos expansions

In this section, we show how to post-process a stochastic PCE for various analyses. The very feature of this surrogate is that it provides a functional mapping between the input parameters $\boldsymbol{X}$, the latent variable Z, and the noise term ϵ:

$$\tilde{Y} \overset{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} (\boldsymbol{X}, Z) + \epsilon, \tag{7.32}$$

To generate realizations of $\tilde{Y}$, we simply sample $\boldsymbol{X}$, Z and ϵ following their distributions and then evaluate Eq. (7.32). To obtain samples of $\tilde{Y}_{\boldsymbol{x}}$ for a fixed $\boldsymbol{x}$ (e.g., to plot the conditional distribution), we follow the same procedure with fixed $\boldsymbol{X} = \boldsymbol{x}$. Moreover, Eq. (7.32) can be easily vectorized for efficient sampling.

By generating a large number of samples, one can display the distribution of $\tilde{Y}$ and $\tilde{Y}_{\boldsymbol{x}}$ using histograms or kernel density estimation. We can also use the quadrature version in Eq. (7.24) to get an explicit form of the conditional response distribution of $\tilde{Y}_{\boldsymbol{x}}$.

In addition, because the proposed surrogate model is derived based on PCE, it inherits all the good properties of PCE. In particular, some important quantities can be directly computed by post-processing the PCE coefficients $\boldsymbol{c}$ and the parameter σ, without the need for sampling. Indeed, the mean and variance of $\tilde{Y}$ are given by

$$\mathbb{E}\left[\tilde{Y}\right] = c_{\boldsymbol{0}}, \quad \mathrm{Var}\left[\tilde{Y}\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\boldsymbol{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2. \tag{7.33}$$

where $c_{\boldsymbol{0}}$ is the coefficient of the constant function.

As already shown in Eq. (7.26), for a given value of $\boldsymbol{x}$, the mean of the model response $\tilde{Y}_{\boldsymbol{x}}$ can be computed as

$$\mathbb{E}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_z=0} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{7.34}$$

Similarly, we can compute the variance as follows:

$$\mathrm{Var}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \mathrm{Var}_{Z,\epsilon}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x}, Z\right) + \epsilon\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}^2\psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}) + \sigma^2. \tag{7.35}$$

7.4.7 Global sensitivity analysis

In the context of global sensitivity analysis of stochastic simulators (Zhu and Sudret, 2021b), various types of Sobol' indices can also be computed analytically for the proposed surrogate model. The *classical Sobol' indices* are defined from the Sobol'-Hoeffding decomposition of the deterministic model given by the stochastic simulator with both the well-defined input variables $\boldsymbol{X}$ and its intrinsic stochasticity as explicit inputs ω, see Eq. (7.1). Since the surrogate model in Eq. (7.32) is also a deterministic function of $\boldsymbol{X}$ and the additional variables Z and ϵ, the Sobol' indices can be efficiently computed from the PCE coefficients, similarly to the classical PCE-based Sobol' indices (Sudret, 2008). For example, the first-order classical Sobol' index of the i-th input X_i is given by

$$S_i \overset{\text{def}}{=} \frac{\mathrm{Var}\left[\mathbb{E}\left[\tilde{Y} \mid X_i\right]\right]}{\mathrm{Var}\left[\tilde{Y}\right]} = \frac{\sum_{\boldsymbol{\alpha}\in\mathcal{A}_i} c_{\boldsymbol{\alpha}}^2}{\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\boldsymbol{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \tag{7.36}$$

where $\mathcal{A}_i \overset{\text{def}}{=} \{\boldsymbol{\alpha}\in\mathcal{A} : \alpha_i \neq 0,\ \alpha_j = 0\,, \forall j \neq i\}$. Similarly, one can also calculate higher-order and total Sobol' indices of the model Eq. (7.32). Let us split the input vector into two subsets $\boldsymbol{X} = (\boldsymbol{X}_{\mathbf{u}}, \boldsymbol{X}_{\sim\mathbf{u}})$, where $\mathbf{u} \subset \{1,\ldots,M\}$ and $\sim\mathbf{u}$ is the complement of $\mathbf{u}$, i.e., $\sim\mathbf{u} = \{1,\ldots,M\}\setminus\mathbf{u}$. The higher-order and total Sobol'

indices, denoted by $S_{\mathbf{u}}$ and S_{T_i}, respectively, are given by

$$S_{\mathbf{u}} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \quad S_{T_i} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_i\neq 0} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \tag{7.37}$$

where $\mathcal{A}_{\mathbf{u}} \overset{\text{def}}{=} \{\boldsymbol{\alpha}\in\mathcal{A} : \alpha_i \neq 0,\ \alpha_j = 0\,, \alpha_z = 0\,, \forall i \in \mathbf{u}, \forall j \in {\sim}\mathbf{u}\}$. However, as mentioned in Section 7.3, the surrogate model aims only at emulating the response distribution of the simulator instead of representing the detailed data generation process. Therefore, the indices involving the artificial variables introduced in the surrogate (i.e., Z and ϵ), e.g., the first-order Sobol' index for Z and the total Sobol' index for each component of $\boldsymbol{X}$, do not reveal the nature of the original model (Zhu and Sudret, 2021b).

The QoI-based Sobol' indices quantify the influence of the input variables on some quantity of interest of the random model response, e.g., mean, variance, and quantiles (Zhu and Sudret, 2021b). As the mean function in Eq. (7.26) is a PCE, the associated Sobol' indices can be computed in a straightforward way (Sudret, 2008). Similar to Eq. (7.36), the first-order index is given by

$$S_i^m \overset{\text{def}}{=} \frac{\operatorname{Var}\left[\mathbb{E}\left[\tilde{m}(\boldsymbol{X}) \mid X_i\right]\right]}{\operatorname{Var}\left[\tilde{m}(\boldsymbol{X})\right]} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_i} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}, \tag{7.38}$$

while higher-order and total Sobol' indices of the mean function read

$$S_{\mathbf{u}}^m = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}, \quad S_{T_i}^m = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_i\neq 0} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}. \tag{7.39}$$

In addition, the variance function in Eq. (7.35) is a polynomial. The associated Sobol' indices can be computed by building another PCE to represent Eq. (7.35) the without error.

7.5 Numerical examples

In this section, we validate the proposed method on several examples, namely case studies from mathematical finance and epidemiology and a complex analytical example with bimodal response distributions. To illustrate its performance, we compare the results obtained from the stochastic polynomial chaos expansion (SPCE) with two state-of-the-art models that are developed for emulating the response distribution of stochastic simulators. The first one is the generalized lambda model (GLaM). This surrogate uses the four-parameter generalized lambda distribution to approximate the response distribution of $Y_{\boldsymbol{x}}$ for any $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. The distribution parameters, as functions of the inputs, are represented by PCEs (see details in Zhu and Sudret, 2020, 2021a). The second model is based on kernel conditional density estimator (KCDE; Hayfield and Racine, 2008). This method uses kernel density estimation to fit the joint distribution $\hat{f}_{\boldsymbol{X},Y}(\boldsymbol{x}, y)$ and the marginal distribution

$\hat{f}_{\boldsymbol{X}}(\boldsymbol{x})$. The response distribution is then estimated by

$$f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{\hat{f}_{\boldsymbol{X},Y}(\boldsymbol{x}, y)}{\hat{f}_{\boldsymbol{X}}(\boldsymbol{x})} = \frac{\sum_{i=1}^{N} \frac{1}{h_y} K_Y\left(\frac{y-y^{(i)}}{h_y}\right) \prod_{j=1}^{M} \frac{1}{h_j} K_j\left(\frac{x_j - x_j^{(i)}}{h_j}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{M} \frac{1}{h_j} K_j\left(\frac{x_j - x_j^{(i)}}{h_j}\right)}, \tag{7.40}$$

where K_y and K_j's are the kernels for Y and X_j's, and h_y and h_j's are the associated bandwidths which are hyperparameters selected by a thorough leave-one-out cross-validation (Hall et al., 2004).

Finally, we also consider a model where we represent the response with a normal distribution. The associated mean and variance as functions of the input $\boldsymbol{x}$ are set to the *true* values obtained from the simulator. Therefore, the accuracy of such an approximation measures how close the response distribution is to the normal distribution. Moreover, this model represents the "oracle" of Gaussian-type mean-variance models, such as the ones presented in Marrel et al. (2012) and Binois et al. (2018).

To quantitatively compare the various surrogates, we define an error metric between the simulator and the emulator by

$$\varepsilon = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(Y_{\boldsymbol{X}}, \tilde{Y}_{\boldsymbol{X}}\right)\right]}{\mathrm{Var}\left[Y\right]}, \tag{7.41}$$

where $Y_{\boldsymbol{x}}$ is the model response, $\tilde{Y}_{\boldsymbol{x}}$ denotes that of the surrogate (with the same input parameters as $Y_{\boldsymbol{x}}$), and Y is the model output aggregating all the uncertainties from both the input and the intrinsic stochasticity. d_{WS} is the *Wasserstein distance of order two* (Villani, 2009) between the two probability distributions defined by

$$d_{\mathrm{WS}}^2\left(Y_1, Y_2\right) \overset{\mathrm{def}}{=} \|Q_1 - Q_2\|_2^2 = \int_0^1 \left(Q_1(u) - Q_2(u)\right)^2 \mathrm{d}u, \tag{7.42}$$

where Q_1 and Q_2 are the quantile functions of random variables Y_1 and Y_2, respectively. The error metric ε in Eq. (7.41) is unitless and invariant to shift and scale, i.e.,

$$\frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(aY_{\boldsymbol{X}} + b, a\tilde{Y}_{\boldsymbol{X}} + b\right)\right]}{\mathrm{Var}\left[aY + b\right]} = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(Y_{\boldsymbol{X}}, \tilde{Y}_{\boldsymbol{X}}\right)\right]}{\mathrm{Var}\left[Y\right]}. \tag{7.43}$$

To evaluate the numerator in Eq. (7.41), we generate a test set $\mathcal{X}_{\mathrm{test}}$ of size $N_{\mathrm{test}} = 1{,}000$ from the input distribution of $\boldsymbol{X}$. The Wasserstein distance is calculated for each point $\boldsymbol{x} \in \mathcal{X}_{\mathrm{test}}$ and then averaged over N_{test}.

We use Latin hypercube sampling (LHS; McKay et al., 1979) to generate the experimental design and the test set. The stochastic simulator is evaluated only once for each set of input parameters, i.e., we do not use replications. To study the convergence property of the surrogates, experimental designs of various sizes are investigated. Each scenario is run 20 times with independent experimental designs to account for the statistical uncertainty in the LHS design and also in the internal stochasticity of the simulator. As a result, error estimates for each size of experimental design are represented by box plots constructed from the 20 repetitions of the full analysis.

7.5.1 Geometric Brownian motion

In the first example, we consider the *Black-Scholes* model that is popular in mathematical finance (McNeil et al., 2005)

$$\mathrm{d}S_t = x_1 S_t \,\mathrm{d}t + x_2 S_t \,\mathrm{d}W_t. \tag{7.44}$$

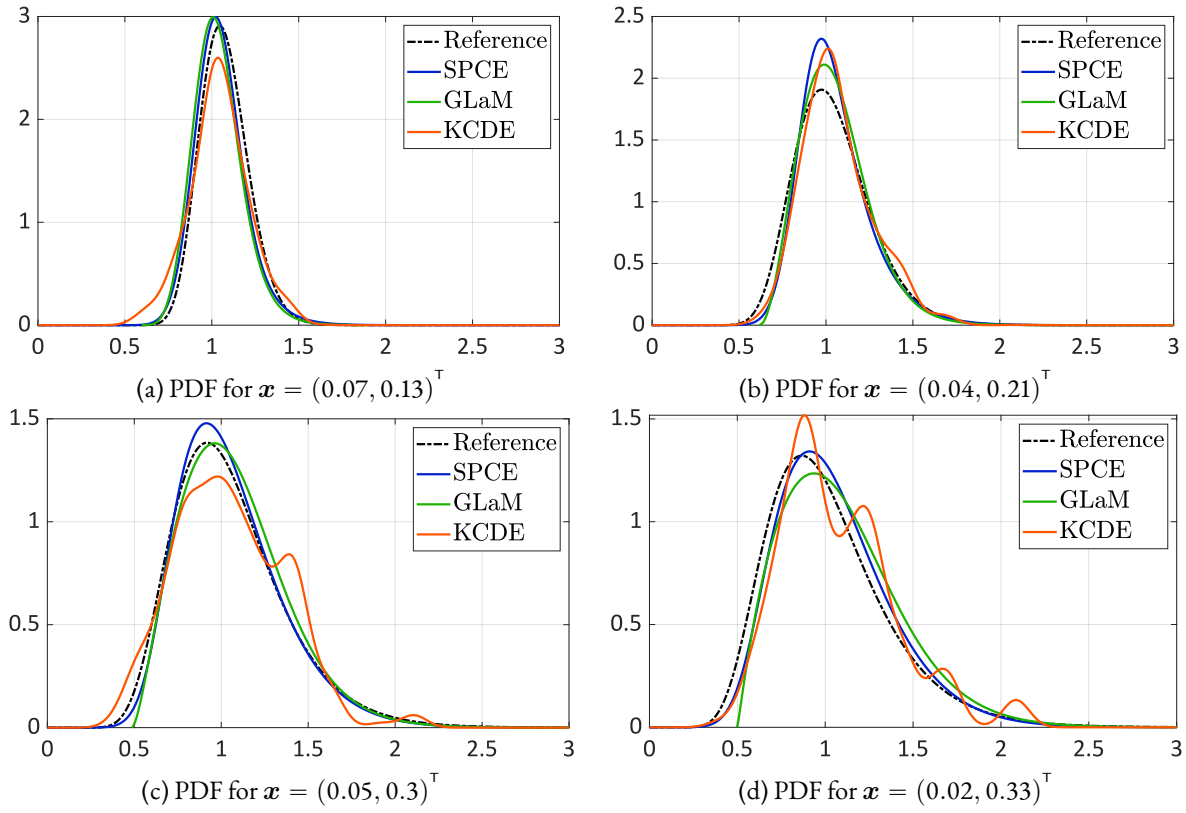
Eq. (7.44) is a stochastic differential equation used to model the evolution of a stock price S_t. Here, $\boldsymbol{x} = (x_1, x_2)^\intercal$ are the input variables that describe the expected return rate and the volatility of the stock, respectively. W_t is a Wiener process that represents the stochastic behavior of the market. Without loss of generality, we set the initial condition to $S_0 = 1$.

The simulator is stochastic: for a given $\boldsymbol{x}$, the stock price S_t is a stochastic process, where the stochasticity comes from W_t. In this example, we are interested in $Y_{\boldsymbol{x}} = S_1$, which corresponds to the stock value at $t = 1$ year. We set $X_1 \sim \mathcal{U}(0, 0.1)$ and $X_2 \sim \mathcal{U}(0.1, 0.4)$ to represent the uncertainty in the return rate and the volatility, where the ranges are selected based on real data (Reddy and Clinton, 2016).

The solution to Eq. (7.44) can be derived using Itô calculus (Shreve, 2004): $Y_{\boldsymbol{x}}$ follows a lognormal distribution defined by

$$Y_{\boldsymbol{x}} \sim \mathcal{LN}\left(x_1 - \frac{x_2^2}{2}, x_2\right). \tag{7.45}$$

As the distribution of $Y_{\boldsymbol{x}}$ is known analytically in this simple example, we can sample directly from the response distribution to get the model output instead of simulating the whole path of S_t.



(a) PDF for $\boldsymbol{x} = (0.07, 0.13)^\intercal$ (b) PDF for $\boldsymbol{x} = (0.04, 0.21)^\intercal$

(c) PDF for $\boldsymbol{x} = (0.05, 0.3)^\intercal$ (d) PDF for $\boldsymbol{x} = (0.02, 0.33)^\intercal$

Figure 7.3: Geometric Brownian motion — Comparisons of the emulated PDFs, $N = 400$.

Figure 7.3 illustrates four response PDFs predicted by the considered surrogates built on an experimental

design of size $N = 400$. We observe that with 400 model runs, both SPCE and GLaM accurately represent the variation of the response PDF. Moreover, SPCE better represents the left tail in Fig. 7.3d. In contrast, KCDE can well approximate the response PDF for low volatility (in Fig. 7.3a) but exhibits unrealistic oscillations in the case of high volatility.

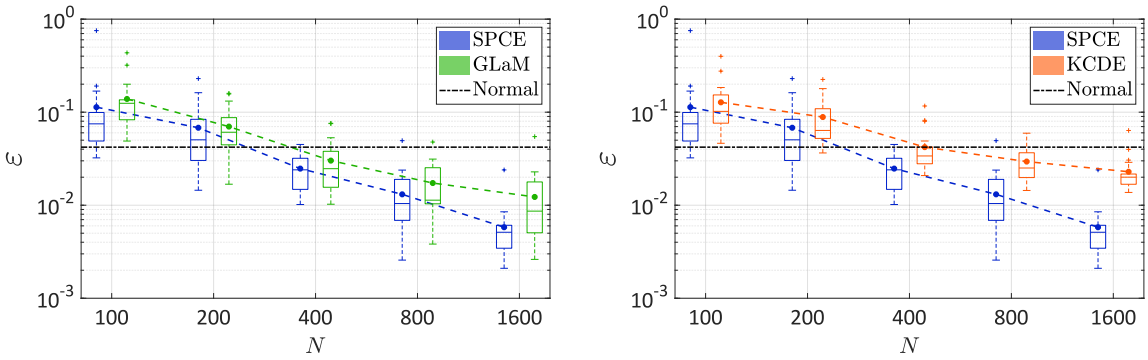


Figure 7.4: Geometric Brownian motion — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results. The black dash-dotted line represents the error of the model assuming that the response distribution is normal and using the true mean and variance.

For convergence studies, we vary the size of the experimental design $N \in \{100; 200; 400; 800; 1{,}600\}$ and plot the error ε defined in Eq. (7.41) with respect to N in Fig. 7.4. In order to show more details, each subfigure in Fig. 7.4 compares SPCE with one competitor. We observe that the average error of KCDE built on $N = 400$ model runs is similar to the best normal approximation, whereas both SPCE and GLaM provide smaller errors. Compared with KCDE and GLaM, the average performance of SPCE is always the best for all sizes of experimental design. For large N, namely $N = 1{,}600$, the average error of SPCE is less than half of that of KCDE, and the spread of the error is narrower than that obtained by GLaM.

7.5.2 Stochastic SIR model

The second example is the stochastic *Susceptible-Infected-Recovered* (SIR) model frequently used in epidemiology (Britton, 2010). This model simulates the outbreak of an infectious disease which spreads out through stochastic contacts between infected and susceptible individuals. The simulator is a compartmental state-space model. More precisely, a population of P individuals at time t is partitioned into three groups: (1) *susceptible individuals* who have not caught the disease and may be infected by close contact with infectious patients; (2) *infected individuals* who are contaminated and infectious; (3) *recovery individuals* who have recovered from the disease and are immune to future infections. The count of each group is denoted by S_t, I_t, and R_t, respectively. Because no newborn or death is considered, the three quantities satisfy $E_t + I_t + R_t = P$. As a result, any two out of the three counts, e.g., E_t and I_t, can characterize the configuration of the population of size P at time t.

Figure 7.5 illustrates the dynamics of the model, where the black icons stand for susceptible individuals, the red icons correspond to infected persons, and the blue icons are the ones who have recovered. At time t, the state of the population is given by (S_t, I_t) (the top left panel of Fig. 7.5). The next configuration depends on two transition channels: infection and recovery. The first channel evolves the system to C_I where one susceptible individual is infected (the bottom left panel of Fig. 7.5). The recovery channel proceeds to C_R where

one infected person recovers (the bottom right panel of Fig. 7.5). Whether the system evolves to the candidate state C_I or C_R depends on two random variables, T_I and T_R which are the respective transition time of each channel. Both T_I and T_R follow an exponential distribution, yet with different parameters:

$$\begin{aligned} T_I &\sim \text{Exp}(\lambda_I), \quad \lambda_I = \beta \frac{S_t I_t}{P}, \\ T_R &\sim \text{Exp}(\lambda_R), \quad \lambda_R = \gamma I_t, \end{aligned} \tag{7.46}$$

where β is the contact rate of an infected individual, and γ is the recovery rate. The next configuration of the population is the one that comes first, i.e., for $T_R < T_I$, the system evolves to C_R at $t+T_R$ with $S_{t+T_R} = E_t - 1$ and $I_{t+T_I} = I_t + 1$, and vice versa. We iterates this updating procedure until the time T where $I_T = 0$ corresponding to no remaining infected individual: no infection or recovery can happen, and the outbreak stops. Since the population size is constant and recovered individuals will not be infected again, the outbreak will stop at finite time, i.e., $T < +\infty$. The simulation process described here corresponds to the *Gillespie algorithm* (Gillespie, 1977).

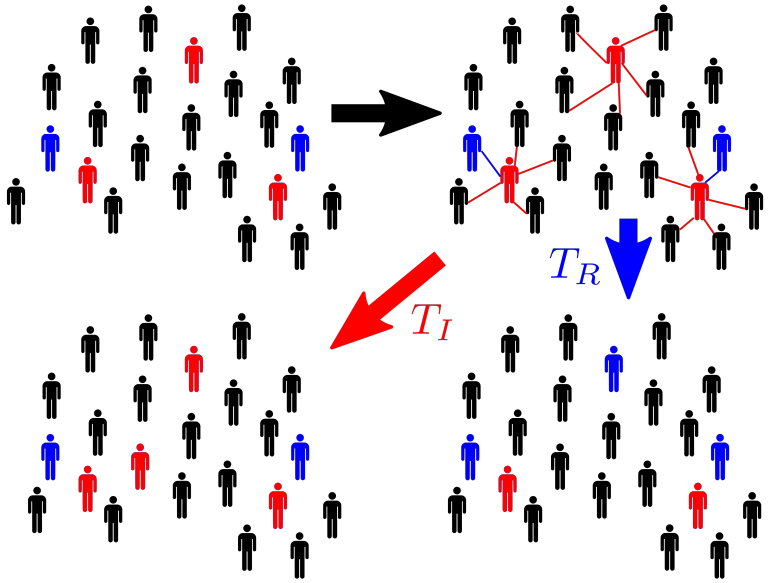
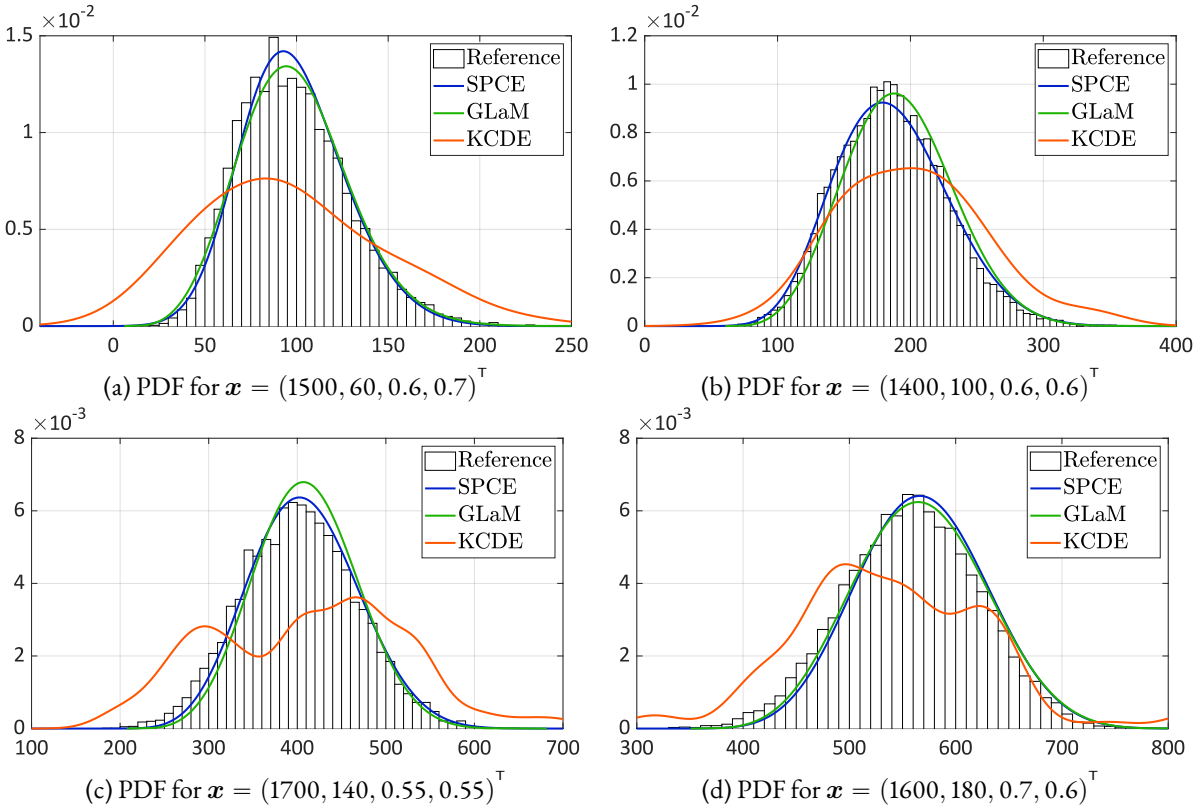


Figure 7.5: Dynamics of the stochastic SIR model: black icons stand for susceptible individuals, red icons represent infected individuals, and blue icons are the ones that have recovered.

The input variables of the simulator are the initial conditions S_0 and I_0 and the transitive rates β and γ. We are interested in the total number of newly infected individuals during the outbreak without counting the initial infections, which is an important quantity in epidemics management (Binois et al., 2018). This can be calculated by the difference between the number of susceptibles at time 0 and T, i.e., $Y = S_0 - S_T$. Because each updating step in Eq. (7.46) depends on two latent variables T_I and T_R, the simulator is stochastic. Moreover, the total number of latent variables is also random.

In this case study, we set $P = 2{,}000$. To account for different scenarios, the input variables $\boldsymbol{X} = \{S_0, I_0, \beta, \gamma\}$ are modeled as $S_0 \sim \mathcal{U}(1{,}200\ ,\ 1{,}800)$, $I_0 \sim \mathcal{U}(20, 200)$, and $\beta, \gamma \sim \mathcal{U}(0.5, 0.75)$. The uncertainty in the first two variables is due to the lack of knowledge of the initial condition. The two transitive rates β, γ are affected by possible interventions such as quarantine and increase of medical resources.



(a) PDF for $\boldsymbol{x} = (1500, 60, 0.6, 0.7)^\intercal$

(b) PDF for $\boldsymbol{x} = (1400, 100, 0.6, 0.6)^\intercal$

(c) PDF for $\boldsymbol{x} = (1700, 140, 0.55, 0.55)^\intercal$

(d) PDF for $\boldsymbol{x} = (1600, 180, 0.7, 0.6)^\intercal$

Figure 7.6: Stochastic SIR — Comparisons of the emulated PDFs, $N = 1{,}600$.

Figure 7.6 illustrates the response PDF for four different sets of input parameters. Because of the transition process in Eq. (7.46), no analytical closed-form distribution of $Y_{\boldsymbol{x}}$ can be derived. Therefore, we use 10^4 replications for each input values to obtain the reference histograms. The surrogate models are trained on an experimental design of size $N = 1{,}600$ (without any replications). We observe that the four PDFs are unimodal. The reference histogram in Fig. 7.6a is slightly right-skewed, while the others in Fig. 7.6 are symmetric. SPCE and GLaM produce similar predictions of the PDF which are very close to the reference histograms. In comparison, KCDE overestimates the spread of the distributions in. Moreover, the KCDE prediction has non-negligible probability for unrealistic negative values in Fig. 7.6a. Besides, it exhibits relatively poor shape representations with spurious wiggles in Fig. 7.6c and Fig. 7.6d.

Figure 7.7 compares the performance of the surrogates built on various sizes of experimental design $N \in \{200; 400; 800; 1{,}600; 3{,}200\}$. To evaluate the error defined in Eq. (7.41), the reference distribution for each $\boldsymbol{x}$ is given by the empirical distribution of 10^4 replications. The oracle normal approximation gives an error of 6×10^{-4} which is smaller than any of the surrogates in consideration. Note that this model is not built on the training data but using the mean and variance from the 10^4 replications for each test point. This implies that the response distribution is close to normal. We do not include this error in Fig. 7.7 to not loose detailed comparisons of the surrogate models. Figure 7.7 reveals a poor performance of KCDE in this case study. This is because the example is four-dimensional, and KCDE is a kernel-based method which is known to suffer from the *curse of dimensionality*. In contrast, SPCE and GLaM are flexible parametric models, and both provide a much smaller error than KCDE for all values of N. Compared with GLaM, SPCE yields a similar spread of the error but demonstrates better average performance for $N \geqslant 400$.

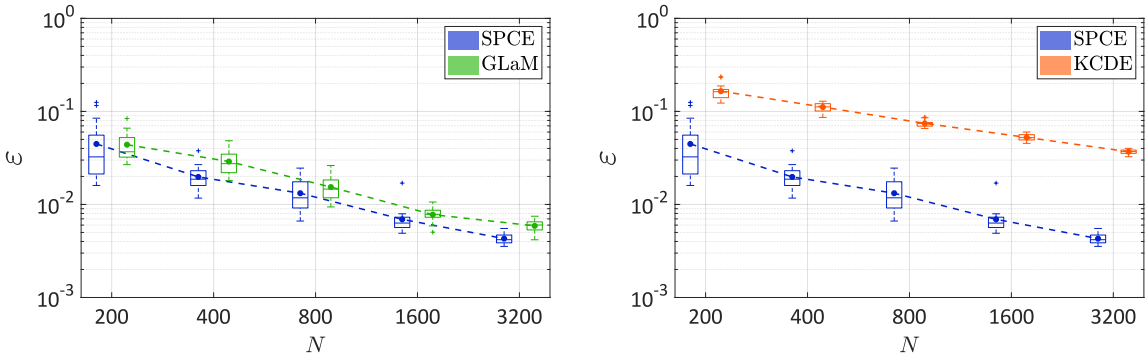


Figure 7.7: Stochastic SIR — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results. The Gaussian model that assumes the response distribution being normal with the mean and variance estimated from 10^4 replications yields an error of 6×10^{-4}, which is not plotted in the figure.

7.5.3 Bimodal analytical example

The response distributions of the previous two examples are unimodal. In the last example, we consider a complex analytical example to test the flexibility of the stochastic polynomial chaos expansion. For this purpose, we directly define the response distribution to approximate as

$$f_{Y|X}(y \mid x) = 0.5\,\varphi\left(1.25\,y - \left(5\sin^2(\pi \cdot x) + 5x - 2.5\right)\right) + 0.75\,\varphi\left(1.25\,y - \left(5\sin^2(\pi \cdot x) - 5x + 2.5\right)\right) \tag{7.47}$$

where φ stands for the standard normal PDF. This response PDF is a mixture of two Gaussian PDFs with weights 0.6 and 0.8. The mean function of each component distribution depends on the input variable x. Let $X \sim \mathcal{U}(0,1)$. With different realization of X, the two components change their location accordingly. Figure 7.8 illustrates a data set generated by $N = 800$ model runs and the mean function of each component of Eq. (7.47) which varies nonlinearly with respect to the input. It is clear that the resulting conditional distribution is bimodal for small ($x \lesssim 0.2$) and large values of x ($x \gtrsim 0.8$), whereas it is unimodal in between.

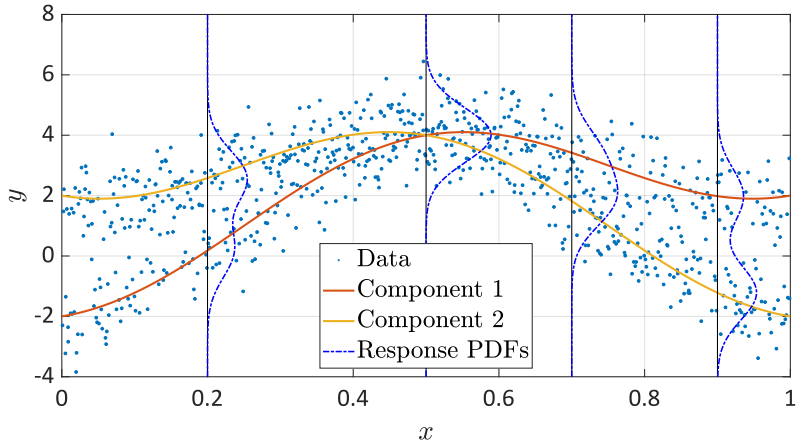
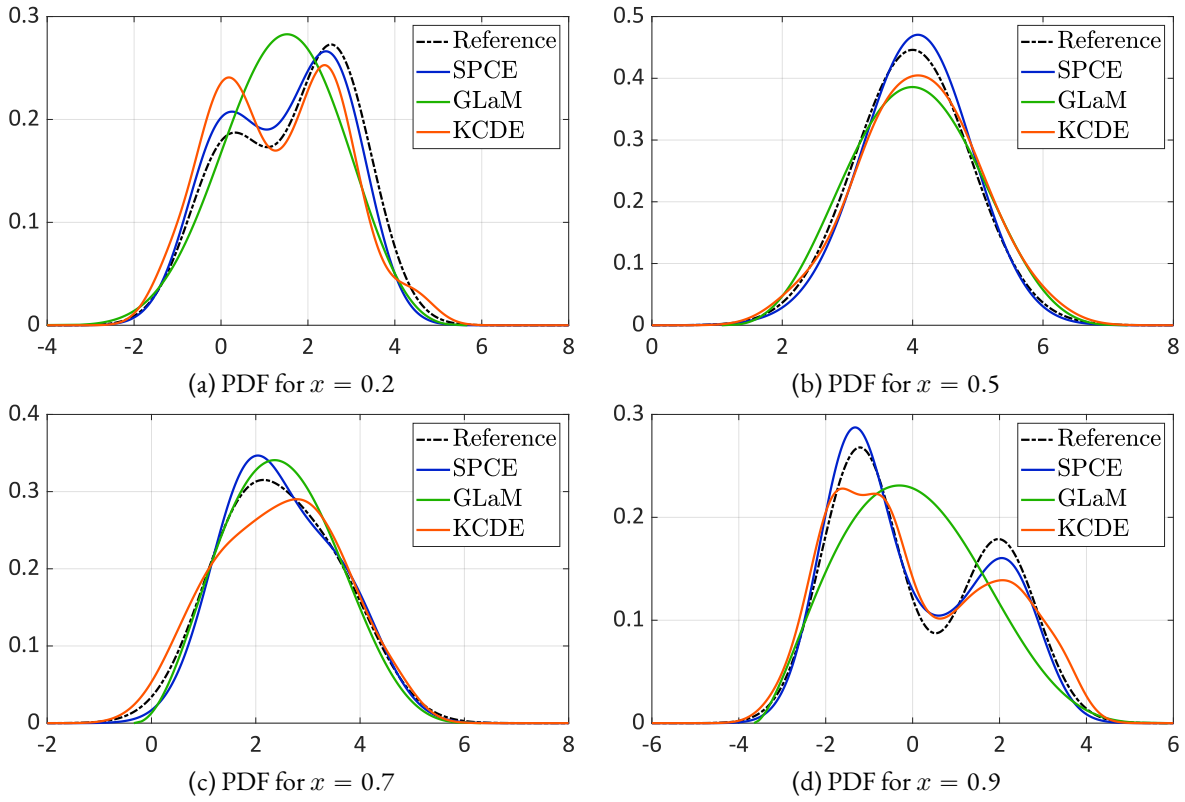


Figure 7.8: Bimodal analytical example — Illustration of the model with an experimental design of $N = 800$

Figure 7.9 compares the response PDF estimated by the surrogates built on the experimental design of Fig. 7.8 ($N = 800$) for four different values of x. We observe that small values of x yield a bimodal distribution

with the higher mode on the right. With x increasing, the two modes merge and form a unimodal distribution at $x = 0.5$. Then, the two modes separate again, which leads to bimodal distributions with the higher mode on the left. This shape variation can also be observed from Fig. 7.8.

As opposed to the previous two examples, GLaM cannot represent this evolution, since generalized lambda distributions cannot produce multimodal distributions. In contrast, SPCE and KCDE capture well the bimodality and also the shape variation. Moreover, in Fig. 7.9c the higher mode is moving to the left, which is a feature not exhibited by KCDE but correctly captured by SPCE.



(a) PDF for $x = 0.2$ (b) PDF for $x = 0.5$

(c) PDF for $x = 0.7$ (d) PDF for $x = 0.9$

Figure 7.9: Bimodal analytical example — Comparisons of the emulated PDFs, $N = 800$.

Quantitative comparisons for $N \in \{100; 200; 400; 800; 1{,}600\}$ in Fig. 7.10 confirm our observation in Fig. 7.9. Because of the bimodality, GLaM provides the least accurate approximation. When increasing N, it converges slowly to the same error as the best normal approximation which is clearly outperformed by the best two surrogates: SPCE and KCDE for $N \geqslant 800$. Both SPCE and KCDE show a consistent decay of the error. Only when a few samples $N = 100$ are available does KCDE provide stabler estimates (the spread of the error is small) and better average performance. For $N \geqslant 200$, SPCE yields more accurate results and exhibits an overall faster rate of convergence. In summary, this example demonstrates that SPCE can represent bimodal distributions with a high accuracy.

7.6 Conclusions

In this paper, we present a novel surrogate model called stochastic polynomial chaos expansions (SPCE) to emulate the response distribution of stochastic simulators. This surrogate is an extension of the classical polynomial

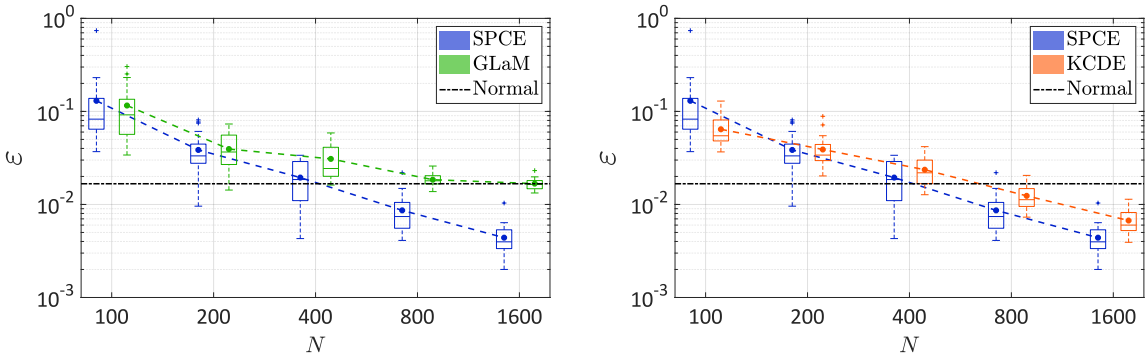


Figure 7.10: Bimodal analytical example — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis. The black dash-dotted line represents the error of the model assuming that the response distribution is normal with the true mean and variance.

chaos expansions developed for deterministic simulators. In order to represent the intrinsic stochasticity of the simulator, we combine a latent variable with the well-defined inputs to form a polynomial chaos representation. In addition, we introduce an additive Gaussian noise as a regularizer. We propose using the maximum likelihood estimation for calibrating the coefficients $\boldsymbol{c}$ of the polynomial basis. The standard deviation σ of the noise variable is a hyperparameter that regularizes the optimization problem for the polynomial coefficients $\boldsymbol{c}$ and is tuned by cross-validation to avoid overfitting. The cross-validation score is also used as a model selection criterion to choose an appropriate truncation scheme for the polynomial chaos expansion in an adaptive manner, and the most suitable distribution for the latent variable. As seen from the presentation and the application examples, the proposed method does not require replications.

The performance of the developed method is illustrated on examples from mathematical finance and epidemiology and on an analytical example showcasing a bimodal response distribution. The results show that SPCE is able to well approximate various response distributions whether unimodal or not, with a reasonable number of model runs.

Using an appropriate error measure defined in Eq. (7.41), SPCE is compared with the generalized lambda model (GLaM) and one state-of-the-art kernel conditional density estimator (KCDE). In the first two examples where the response distribution is unimodal, SPCE noticeably outperforms KCDE and provides slightly more accurate results than GLaM which is known for its flexibility for representing unimodal distributions. In the last example featuring bimodal distributions which cannot be well approximated by generalized lambda distributions, SPCE can still capture the complex shape variation and yields smaller errors than KCDE. All in all, SPCE generally performs as the best against the various competitors considered in this study.

Applications of the proposed method to complex engineering problems, such as wind turbine design (Abdallah et al., 2019) and structural dynamics (Mai et al., 2017), should be considered in future investigations. Statistical properties (e.g., consistency and asymptotics) of the maximum likelihood estimation used in SPCE remains to be studied. This will allow for assessing the uncertainty in the estimation procedure.

Finally, the proposed approach has been validated so far only for problems with small to moderate dimentionality. To improve the efficiency and performance of SPCE in high dimensions, models that have a general sparse structure (not only regarding the mean function) are currently under investigations.

Acknowledgments

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524), whose support is gratefully acknowledged.

7.a Appendix

7.a.1 Upper bound

In this section, we demonstrate that the leave-one-out error obtained from fitting the mean function Eq. (7.28) provides an upper bound for σ^2.

Taking the expectation of Eq. (7.35) with respect to $\boldsymbol{X}$, it holds

$$\mathbb{E}\left[\mathrm{Var}\left[\tilde{Y} \mid \boldsymbol{X}\right]\right] = \mathbb{E}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{X}) + \sigma^2\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 + \sigma^2. \tag{7.48}$$

The leave-one-out error $\varepsilon_{\mathrm{LOO}}$ in the mean-fitting process is an estimate of $\mathbb{E}\left[\left(\hat{m}(\boldsymbol{X}) - Y_{\boldsymbol{X}}\right)^2\right]$ (James et al., 2014). The latter can be decomposed as

$$\begin{aligned}\mathbb{E}\left[\left(\hat{m}(\boldsymbol{X}) - Y_{\boldsymbol{X}}\right)^2\right] &= \mathbb{E}\left[\left(\hat{m}(\boldsymbol{X}) - m(\boldsymbol{X}) + m(\boldsymbol{X}) - Y_{\boldsymbol{X}}\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{m}(\boldsymbol{X}) - m(\boldsymbol{X})\right)^2\right] + \mathbb{E}\left[\mathrm{Var}\left[Y \mid \boldsymbol{X}\right]\right]\end{aligned}. \tag{7.49}$$

Aiming at approximating $Y_{\boldsymbol{x}}$ with $\tilde{Y}_{\boldsymbol{x}}$, we have $\mathbb{E}\left[\mathrm{Var}\left[Y \mid \boldsymbol{X}\right]\right] \approx \mathbb{E}\left[\mathrm{Var}\left[\tilde{Y} \mid \boldsymbol{X}\right]\right]$. Hence, $\varepsilon_{\mathrm{LOO}}$ provides an upper bound for Eq. (7.48) and therefore for σ^2.

References

Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators—application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67–77.

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58:371–382.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions—application to computational electromagnetic dosimetry. *International Journal for Uncertainty Quantification*, 9:351–363.

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2010). Bayesian compressive sensing using Laplace priors. *IEEE Transactions on Image Processing*, 19(1):53–63.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15(1–3):81–92.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27:808–821.

Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Britton, T. (2010). Stochastic epidemic models: A survey. *Mathematical Biosciences*, 225:24–35.

Desceliers, C., Ghanem, R. G., and Soize, C. (2006). Maximum likelihood estimation of stochastic chaos representations from experimental data. *International Journal for Numerical Methods in Engineering*, 66:978–1001.

Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034.

Efromovich, S. (2010). Dimension reduction and adaptation in conditional density estimation. *Journal of the American Statistical Association*, 105:761–774.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Everitt, B. S. (1984). *An Introduction to Latent Variables Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition.

Gautschi, W. (2004). *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press.

Ghanem, R. G. and Spanos, P. (2003). *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, Mineola, 2nd edition.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81:2340–2361.

Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99:1015–1026.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer.

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27:1–32.

Jacod, J. and Protter, P. (2004). *Probability Essentials*. Springer, 2nd edition.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46:33–50.

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022a). A benchmark of basis-adaptive sparse polynomial chaos expansions for engineering regression problems. *International Journal for Uncertainty Quantification*, 12:49–74.

Lüthen, N., Marelli, S., and Sudret, B. (2022b). A spectral surrogate model for stochastic simulators computed from trajectory samples. *Submitted*.

Mai, C. V., Konakli, K., and Sudret, B. (2017). Seismic fragility curves for structures using non-parametric representations. *Frontiers of Structural and Civil Engineering*, 11(2):169–186.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22:833–847.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:116–155.

Nataf, A. (1962). Détermination des distributions dont les marges sont données. *Comptes Rendus de l'Academie des Sciences*, 225:42–43.

Plumlee, M. and Tuo, R. (2014). Building accurate emulators for stochastic simulations via quantile Kriging. *Technometrics*, 56:466–473.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Reddy, K. and Clinton, V. (2016). Simulating stock prices using geometric Brownian motion: Evidence from Australian companies. *Australasian Accounting, Business and Finance Journal*, 10(3):23–47.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23:470–472.

Shreve, S. (2004). *Stochastic Calculus for Finance II*. Springer, New York.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NIPS 2012), Colorado, USA*, pages 2951–2959. Curran Associates, Inc.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93:964–979.

Torossian, L., Picheny, V., Faivre, R., and Garivier, A. (2020). A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety*, 201.

Villani, C. (2009). *Optimal Transport, Old and New*. Springer, Berlin.

Wooldridge, J. M. (2013). *Introductory Econometrics: A Modern Approach*. Cengage Learning, 5th edition.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021a). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

Zhu, X. and Sudret, B. (2021b). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815.

8

Seismic fragility analysis using stochastic polynomial chaos expansions

This chapter is a pre-print of the submitted version of

> Zhu, X., Broccardo, M., and Sudret, B. (2023). Seismic fragility analysis using stochastic polynomial chaos expansions.*Probabilistic Engineering Mechanics*, 72:103413. DOI:10.1016/j.probengmech.2023.103413.

A few typos are corrected in the text, and some figures are improved for a better graphical presentation.

Author contributions. **X. Zhu**: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - Original Draft, Visualization. **M. Broccardo**: Conceptualization, Methodology, Software, Writing - Original Draft. **B. Sudret**: Supervision, Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition.

Abstract

Within the performance-based earthquake engineering (PBEE) framework, the fragility model plays a pivotal role. Such a model represents the probability that the engineering demand parameter (EDP) exceeds a certain safety threshold given a set of selected intensity measures (IMs) that characterize the earthquake load. The state-of-the art methods for fragility computation rely on full non-linear time-history analyses. Within this perimeter, there are two main approaches: the first relies on the selection and scaling of recorded ground motions; the second, based on random vibration theory, characterizes the seismic input with a parametric stochastic ground motion model (SGMM). The latter case has the great advantage that the problem of seismic risk analysis is framed as a forward uncertainty quantification problem. However, running classical full-scale Monte Carlo simulations is intractable because of the prohibitive computational cost of typical finite element models. Therefore, it is of great interest to define fragility models that link an EDP of interest with the SGMM parameters

— which are regarded as IMs in this context. The computation of such fragility models is a challenge on its own and, despite few recent studies, there is still an important research gap in this domain. This comes with no surprise as classical surrogate modeling techniques cannot be applied due to the stochastic nature of SGMM. This study tackles this computational challenge by using *stochastic polynomial chaos expansions* to represent the statistical dependence of EDP on IMs. More precisely, this surrogate model estimates the full conditional probability distribution of EDP conditioned on IMs. We compare the proposed approach with some state-of-the-art methods in two case studies. The numerical results show that the new method outperforms its competitors in estimating both the conditional distribution and the fragility functions.

8.1 Introduction

The PEER[1] performance-based earthquake engineering (PBEE) framework introduced two decades ago (Cornell and Krawinkler, 2000) represents the state-of-the-art approach to seismic risk assessment. The framework builds on the total probability theorem by convolving the output of probabilistic seismic hazard analysis (PSHA; Cornell, 1968) with fragility, damage, and loss models. The output of the PSHA analysis is given by the so-named hazard curves, which are rates of occurrence of a given *intensity measure* (IM, e.g., peak ground acceleration, spectral acceleration, etc.) or a vector of IMs. The damage of a structure is typically characterized by the *engineering demand parameter* (EDP) which represents the structural response (e.g., the maximum interstory drift for a multistory building, the maximum base shear, etc.).

A critical component of the framework is represented by the statistical relationship between IMs and EDP. This relationship, named fragility model, is a function of the IMs and computes the EDP exceeding probability (e.g., EDP exceeds a certain threshold) conditioned on the corresponding value of IMs. As an important part of PBEE, fragility models have become a rich field of research with two major lines of investigation. The first line is based on the selection and scaling of recorded ground motions and (non-)linear time history analysis. An incomplete list of studies following this line of research includes Vamvatsikos and Cornell (2002), Baker and Cornell (2006), Luco and Bazzurro (2007), and Kiani and Khanmohammadi (2015).

The second line of research builds on stochastic ground motion models (SGMM; Rezaeian and Der Kiureghian, 2008, 2010), and (non-)linear time history analysis. An SGMM typically combines a set of engineering-meaningful parameters, referred to as SGMM parameters in the sequel, with a set of hidden aleatory variables (e.g., white noise) to generate synthetic ground motions. The available records are considered as realizations of the SGMM and used to calibrate the SGMM parameters. The latter are modeled as random variables to account for epistemic uncertainties due to limited data. In this setting, the SGMM parameters are statistically related to the earthquake and site characteristics (e.g., magnitude, faulting mechanism, source-to-site distance, and the site shear-wave velocity) via predictive equations. In essence, these are classical ground motion predictive equations (GMPEs; Cornell, 1968) with the IMs being the SGMM parameters.

Following this line, a fragility model becomes the statistical relationship between the SGMM parameters and the EDP. These models, when developed, allow for a rapid seismic risk assessment by computing directly or via (inexpensive) simulations of the convolutions of the PEER-PBEE framework. Within this perimeter, therefore, the development of efficient algorithms for fragility computation is paramount. While several studies

[1] Pacific Earthquake Engineering Center

use a SGMM for seismic risk assessment (an incomplete list includes Taflanidis and Beck, 2009; Gidaris et al., 2015; Mai et al., 2017; Smerzini and Pitilakis, 2018; Ghosh and Chakraborty, 2020; Ghosh et al., 2021), to the best of our knowledge, fragility models as a function of the SGMM parameters have been explicitly introduced only recently (Abbiati et al., 2021).

In this context, however, there is a research gap in the development of efficient algorithms that allow a feasible computation of these special fragility models. This paper aims to fill this gap by using the stochastic polynomial chaos expansion (SPCE; Zhu and Sudret, 2023), which we show to be the most computationally efficient option up to date. As such, this paper focuses only on the fragility model computation without employing the full seismic risk analysis.

A great advantage of the simulation-based approach is that the problem of seismic risk analysis can be framed as a forward uncertainty quantification problem (Abbiati et al., 2021). In fact, by combining the SGMM with the dynamical analysis of structures, one obtains a simulator that maps a set of ground motion parameters to the associated EDP. More specifically, this is a *stochastic simulator* (Zhu and Sudret, 2020; Abbiati et al., 2021), i.e., several runs with the same ground motion parameters produce different values of the EDP, due to the aleatory hidden variables in the generation of ground motions. Therefore, one can run multiple simulations for given values of IMs without introducing bias. Moreover, this allows for coupling the seismic hazard model and the fragility function without going through intermediate variables.

When working with ground motion parameters, replication-based methods have been proposed so far in the literature (Gidaris et al., 2015; Abbiati et al., 2021). In this framework, one fixes the SGMM parameters, and the hazard model produces a set of consistent earthquake loads for dynamical analysis of the structure. This procedure is called *replication*, as we evaluate repeatedly the simulator for the same values of the input. The associated EDP values are realizations of the structural response conditioned on the given SGMM parameters. Therefore, they can be used to estimate the underlying conditional distribution. This procedure is repeated for different SGMM parameters, and the fragility function can be estimated from the conditional distribution. Because many replications (e.g., 100) are necessary to characterize the conditional distribution, this approach requires a large number of model runs (as shown in Abbiati et al., 2021).

To alleviate the computational cost, in this paper, we explore the methods that *do not* rely on replications (Cornell et al., 2002; Shinozuka et al., 2000; Mai et al., 2017). Since the SGMM parameters are vector-valued IMs, some methods developed for fragility analysis with a single IM can be extended and applied. Cornell et al. (2002) proposed the so-called cloud analysis, which is a linear model in the log-scale with a homoscedastic Gaussian noise. This parametric model relies on rather restrictive assumptions (log-linearity and homoscedasticity).

Alternatively, fragility models can be computed in a classification framework (Shinozuka et al., 2000; Baker, 2015). This method only works with binary damage variables (whether the structure fails or not) and does not make use of the precise value of the EDP, which leads to a certain loss of information. More recently, nonparametric models, namely kernel smoothing, have been proposed in the literature (Noh et al., 2015; Mai et al., 2017). However, it is well-known that nonparametric models suffer from the curse of dimensionality (Tsybakov, 2009): the model accuracy decreases drastically with increasing input dimensionality (in our case, the number of IMs).

In this paper, to better balance the model flexibility and limited number of simulations, we propose applying the newly developed *stochastic polynomial chaos expansion* (SPCE) technique (Zhu and Sudret, 2023). This model introduces an artificial latent variable and a noise variable to represent the random nature of the stochastic

simulation. More precisely, it expresses the EDP as a function of the IMs and the latent variable plus the additive noise. Therefore, this model can tackle a full representation of conditional distributions. It follows that natural byproducts of the analysis are the *classical* fragility models. In fact, one can naturally develop statistical relations between classical IMs and the selected EDP. In this case, the classical IMs are available as statistics of the synthetic ground motions[2], and the fragility models can be used in the original PEER-PBEE framework directly.

The paper is organized as follows. In Section 8.2, we outline the stochastic simulator approach; then, we recap the extension of classical methods developed to multiple intensity measures. In Section 8.3, we summarize the main ingredients of the stochastic polynomial chaos expansion. In Section 8.4, we use a synthetic ground motion model and two computational examples to illustrate the performance of the proposed method. Finally, we conclude with the main finding of the study and give an outlook for future research in Section 8.6.

8.2 Stochastic simulator approach for fragility analysis

8.2.1 The stochastic simulator approach

This paper follows the line of research that uses an SGMM to characterize seismic excitation. Using the representation introduced in Abbiati et al. (2021), the stochastic ground motion can be expressed as follows

$$A(t) = \mathcal{M}_a(t, \boldsymbol{\Xi}|\boldsymbol{X}), \tag{8.1}$$

where $\mathcal{M}_a$ represents the synthesis formula of a parametric SGMM, $\boldsymbol{\Xi}$ is a Gaussian vector (with i.i.d. standard normal random variables) representing the aleatory variability of the process, and $\boldsymbol{X}$ is a random vector collecting the parameters of the model and the associated epistemic variability. The SGMM parameters are selected to be engineering meaningful (Rezaeian and Der Kiureghian, 2008; Broccardo and Dabaghi, 2017); therefore, in this framework, $\boldsymbol{X}$ can be regarded as a vector of IMs. In the PEER-PBEE framework, $\boldsymbol{X}$ is statistically related to the earthquake and site characteristics via predictive equations. However, this study focuses only on the fragility model computation and, therefore, for simplicity, we use a marginal joint probability distribution of $\boldsymbol{X}$ fitted to a specific seismic catalog (see Section 8.4.1 for further details).

Let Y denote the EDP (e.g., maximum interstory drift) of a structural system of interest computed as $Y = \mathcal{M}_d(A(t)|\boldsymbol{x}_d)$, where $\mathcal{M}_d$ is an expensive-to-evaluate deterministic solver[3] with $\boldsymbol{x}_d$ being a set of deterministic parameters (e.g., a finite element model with deterministic masses, damping, and constitutive models). It follows that Y can be expressed as

$$Y = \mathcal{M}_d(\mathcal{M}_a(t, \boldsymbol{\Xi}|\boldsymbol{X})|\boldsymbol{x}_d) = \mathcal{M}_s(\boldsymbol{\Xi}|\boldsymbol{X}), \tag{8.2}$$

where $\mathcal{M}_s \overset{\text{def}}{=} \mathcal{M}_d \circ \mathcal{M}_a$ is a stochastic simulator since for $\boldsymbol{X} = \boldsymbol{x}$ the response Y is still stochastic (due to the aleatory variability encoded in $\boldsymbol{\Xi}$). Provided with this framework, the objective of this study is to use a stochastic

[2] In this case, one has to verify that the rate of exceedance of the classical IMs emerging from the SGMM is compatible to the ones derived by PSHA analysis (Rezaeian and Der Kiureghian, 2010)

[3] In Abbiati et al. (2021), the solver is also assumed to be stochastic to accommodate random fields. In this paper, we choose the more restrictive deterministic solver as it is the most typical case in earthquake engineering

surrogate model, namely the SPCE, to develop fragility models.

8.2.2 Fragility analysis

In PBEE, seismic loads are typically characterized by a selected set of IMs. An incomplete list of conventional IMs includes peak ground acceleration, spectral acceleration, peak ground velocity, and Arias intensity (Mackie and Stojadinović, 2003). In general, an IM can represent any "optimal" feature of the seismic load. According to Mackie and Stojadinović (2003), optimal is defined as being practical, sufficient, effective, and efficient (see Mackie and Stojadinović, 2003 for further details). To improve the power of the prediction and reduce the variability among ground motions, one can combine several IMs for fragility analysis (Baker and Cornell, 2005; Seyedi et al., 2010; Modica and Stafford, 2014).

In the SGMM context, a natural choice for the IMs is the set of SGMM parameters. This allows applying directly the PBEE-PEER framework by convolving the predictive equations (which extend the classical GMPEs) with these fragility models based on the SGMM parameters (Abbiati et al., 2021). In this study, we pursue this philosophy by proposing SPCE as a computational method that outperforms the current state of the art. In particular, this section first introduces the general concept of fragility models; second, it reviews a series of computational methods which can be used directly in this context and that we will use to compare the proposed SPCE approach.

The structural performance is usually defined by the event that the EDP exceeds a certain threshold δ_0, which represents a predefined damage level. A fragility model expresses the exceeding probability as a function of IMs, that is,

$$p_f(\boldsymbol{x}) = \mathbb{P}\left(Y > \delta_0 \mid \boldsymbol{X} = \boldsymbol{x}\right) = 1 - F_{Y|\boldsymbol{X}}(\delta_0 \mid \boldsymbol{x}). \tag{8.3}$$

Using the probability distribution characterizing the SGMM parameters, we generate N samples grouped into $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\}$. Unlike Gidaris et al. (2015) and Abbiati et al. (2021), where $\mathcal{O}(10^2)$ replications are used, we do not consider replications in this paper to drastically reduce the overall number of simulations. This is feasible because of the features of the SPCE approach introduced by Zhu and Sudret (2023), which is recapped in Section 8.3. Therefore, for each set of the ground motion parameters $\boldsymbol{x}^{(i)}$, we generate one synthetic ground motion and then compute the associated EDP $y^{(i)}$ which is collected in $\mathcal{Y} = \left\{y^{(1)}, \dots, y^{(N)}\right\}$.

In the sequel, we introduce a series of classical fragility model computation methods which can be used directly in this context. Moreover, we use these benchmark methods to compare the proposed SPCE approach.

One of the most popular methods for fragility analysis is the linear model (i.e., cloud analysis; Cornell et al., 2002; Modica and Stafford, 2014), where the logarithm of EDP is expressed as a linear function of the logarithm of the IMs with an independent additive Gaussian noise, i.e.,

$$\log(Y) = \sum_{i=1}^{M} \beta_0 + \sum_{j=1} \beta_j \log(x_j) + e, \tag{8.4}$$

where $e \sim \mathcal{N}(0, \sigma^2)$. The model parameters $\boldsymbol{\beta}$ and σ can be estimated using standard ordinary least-squares. Eq. (8.4) gives directly the conditional probability density function (PDF), and the fragility function is calcu-

lated as

$$p_f(\boldsymbol{x}) = 1 - \Phi\left(\frac{\delta_0 - \beta_0 - \sum_{j=1}^{M} \beta_j \ln(x_j)}{\sigma}\right), \tag{8.5}$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution.

Probit regression is another classical method used to estimate directly fragility functions (Shinozuka et al., 2000; Baker, 2015). In this context, fragility models are interpreted as a soft classifier. In the earthquake engineering community, the CDF of a lognormal distribution is typically selected as classifier. Although this method is usually used for a single IM, it can be extended directly to the case of multiple IMs, that is,

$$p_f(\boldsymbol{x}) = \Phi\left(\beta_0 + \sum_{j=1} \beta_j \ln(x_j)\right). \tag{8.6}$$

The model parameters $\boldsymbol{\beta}$ are estimated by maximum likelihood estimation. In this classification framework, the threshed δ_0 is used to directly classify the samples of the outcomes (e.g., $\{\text{not fail}\} \stackrel{\text{def}}{=} \{EDP < \delta_0\}$, $\{\text{fail}\} \stackrel{\text{def}}{=} \{EDP \geqslant \delta_0\}$), and the precise value of the EDP is ignored. Therefore, δ_0 is a property of the classifier; in other words, when the value of δ_0 varies, it is necessary to build a new model.

In recent years, nonparametric methods for fragility model computations have gained momentum (Noh et al., 2015; Mai et al., 2017), given their inherent flexibility. Recall the definition of the conditional distribution

$$f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{f_{Y,\boldsymbol{X}}(y, \boldsymbol{x})}{f_{\boldsymbol{X}}(\boldsymbol{x})}. \tag{8.7}$$

Without introducing restrictive assumptions, the distributions $f_{Y,\boldsymbol{X}}$ and $f_{\boldsymbol{X}}$ can be estimated using nonparametric estimators, namely kernel smoothing, which then provides an estimate of the conditional distribution. In this approach, the bandwidths are hyper-parameters to be defined. Noh et al. (2015) proposed selecting the bandwidths by engineering judgments and prior information. Mai et al. (2017) applied the method developed in Duong and Hazelton (2005) to estimate separately $f_{Y,\boldsymbol{X}}$ and $f_{\boldsymbol{X}}$. However, this does not yield a valid conditional distribution (the integral over y is unequal to 1). In this paper, we consider a more advanced nonparametric method developed by Li et al. (2013) that is typically designed for estimating the conditional CDF, as the latter is directly related to the exceeding probability. Following Mai et al. (2017), the kernel estimator is applied to the logarithmic transform of the data to guarantee the positiveness of the EDP and the IMs.

8.3 Stochastic polynomial chaos expansion

The methods reviewed in the previous section have their limitations: the linear model relies on very restrictive assumptions, the probit model does not make full use of the available information, and the kernel estimator suffers from the curse of dimensionality (Tsybakov, 2009). To achieve better accuracy with a limited number of simulations, we propose using the stochastic polynomial chaos expansion (SPCE) approach recently proposed in Zhu and Sudret (2023) to estimate the probability distribution of the EDP, Y, conditioned on the IMs, $\boldsymbol{X} = \boldsymbol{x}$. The conditional random variable is denoted by $Y_{\boldsymbol{x}}$. In this section, we recap the principle of the standard polynomial chaos expansion (PCE) and its extension to SPCE.

PCE is a surrogate model that has been widely applied to emulate deterministic simulators in the context

of uncertainty quantification. Considering the uncertain input variables $\boldsymbol{X}$, this surrogate represents a deterministic model $\mathcal{M}_d : \boldsymbol{x} \mapsto \mathcal{M}_d(\boldsymbol{x})$ by a series of polynomial expansions, that is,

$$\mathcal{M}_d(\boldsymbol{X}) \approx \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{8.8}$$

where $\psi_{\boldsymbol{\alpha}}$ is the basis function defined by the multi-index $\boldsymbol{\alpha}$, $c_{\boldsymbol{\alpha}}$ is the associated coefficient, and $\mathcal{A}$ is the truncated set of multi-indices that define the basis functions used in the expansion.

For $\boldsymbol{X}$ with independent components, the basis function is given by a product of univariate polynomials:

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j), \tag{8.9}$$

where M is the dimension of $\boldsymbol{X}$, i.e., the number of input parameters, α_j is the polynomial degree in x_j, and $\left\{\phi_k^{(j)} : k \in \mathbb{N}\right\}$ is the orthogonal polynomial basis with respect to the marginal distribution f_{X_j}, which satisfies

$$\mathbb{E}\left[\phi_k^{(j)}(X_j)\, \phi_l^{(j)}(X_j)\right] = \begin{cases} 1 & \text{if } l = k, \\ 0 & \text{otherwise.} \end{cases} \tag{8.10}$$

For uniform, normal, gamma, and beta distributions, the associated univariate orthogonal polynomials are well known as Legendre, Hermite, Laguerre, and Jacobi polynomials (Xiu and Karniadakis, 2002).

When $\boldsymbol{X}$ has dependent components, the tensor product in Eq. (8.9) generally does not produce an orthogonal basis. To circumvent this problem, one common way is to transform $\boldsymbol{X}$ into an auxiliary vector $\boldsymbol{H} = \mathcal{T}(\boldsymbol{X})$ with independent components (e.g., a standard normal vector) using the Nataf or Rosenblatt transform (Torre et al., 2019). The polynomial basis is then defined with respect to the auxiliary variables

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(h_j). \tag{8.11}$$

where $\boldsymbol{h} = \mathcal{T}(\boldsymbol{x})$, and $\left\{\phi_k^{(j)} : k \in \mathbb{N}\right\}$ is defined by the marginal distribution of H_j.

Let us introduce now the stochastic extension of PCE. Eq. (8.8) is a deterministic function of the input variables $\boldsymbol{x}$. To represent the stochastic behavior in the earthquake simulation, we include an artificial latent variable Z in the expansion and an additive noise variable ϵ which results in the SPCE (Zhu and Sudret, 2023):

$$\log\left(Y_{\boldsymbol{x}}\right) \overset{\mathrm{d}}{\approx} \log\left(\tilde{Y}_{\boldsymbol{x}}\right) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x}, Z\right) + \epsilon, \tag{8.12}$$

where the expansion is expressed on the logarithmic transform of $Y_{\boldsymbol{x}}$ to ensure the EDP is positive (this transform is also applied by Gidaris et al., 2015). The noise variable ϵ is a centered Gaussian random variable with standard deviation σ, i.e., $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Here, we aim at approximating the distribution of the EDP $Y_{\boldsymbol{x}}$ for any $\boldsymbol{x}$. As a result, we use the notation $\overset{\mathrm{d}}{\approx}$ to denote *approximation in distribution*. The artificial latent variable Z in Eq. (8.12) is only introduced to reproduce the stochasticity, and it is *not* related to the high-dimensional hidden random vector $\boldsymbol{\Xi}$ in the stochastic ground motion model of Eq. (8.2). In this paper, we select a standard Gaussian latent variable$Z \sim$

$\mathcal{N}(0,1)$. With this choice, if only linear terms are considered in Eq. (8.12), the SPCE is equivalent to the linear model in Eq. (8.4).

To build such a model, we need to determine the coefficients $\boldsymbol{c}$ of the expansion and the standard deviation σ of the noise term.For a data point $(\boldsymbol{x}, y)$ the conditional likelihood can be expressed as (see details in Zhu and Sudret, 2023)

$$l(\boldsymbol{c}, \sigma; \boldsymbol{x}, y) = \frac{1}{y} \int_{\mathcal{D}_Z} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\log(y) - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)\right)^2}{2\sigma^2} \right) f_Z(z) \mathrm{d}z. \tag{8.13}$$

In practice, we can apply the Gaussian quadrature (Golub and Welsch, 1969) with respect to the weight function f_Z to efficiently evaluate the one-dimensional integral, that is

$$\begin{aligned} l(\boldsymbol{c}, \sigma; \boldsymbol{x}, y) &\approx \tilde{l}(\boldsymbol{c}, \sigma; \boldsymbol{x}, y) \\ &= \frac{1}{y} \sum_{j=1}^{N_Q} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\log(y) - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z_j)\right)^2}{2\sigma^2} \right) w_j, \end{aligned} \tag{8.14}$$

where N_Q is the number of integration points, z_j is the j-th integration point, and w_j is the associated weight. Based on Eq. (8.14) and the available data $(\mathcal{X}, \boldsymbol{y})$, we calibrate the coefficients $\boldsymbol{c}$ by maximum likelihood estimation (MLE)

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i}^{N} \log\left(\tilde{l}\left(\boldsymbol{c}, \sigma; \boldsymbol{x}^{(i)}, y^{(i)}\right)\right). \tag{8.15}$$

The standard deviation σ cannot be fitted jointly with $\boldsymbol{c}$ because the likelihood in Eq. (8.13) is unbounded for $\sigma = 0$ (see Zhu and Sudret, 2023 for a detailed discussion). Therefore, σ is a hyper-parameter, and we use cross-validation with the out-of-sample likelihood as the performance metric to select an optimal value for σ. In addition, the cross-validation score is also useful for determining an appropriate truncated set $\mathcal{A}$.

After constructing the model, one can efficiently generate new samples of $\tilde{Y}_{\boldsymbol{x}}$ by fixing the value of $\boldsymbol{x}$ and sampling (Z, ϵ) to evaluate Eq. (8.12). Therefore, probabilistic quantities of $\tilde{Y}_{\boldsymbol{x}}$ (e.g., mean, variance, quantiles, and exceeding probabilities Eq. (8.3)) can be estimated by large-scale Monte Carlo simulations. Similarly, jointly sampling $(\boldsymbol{X}, Z, \epsilon)$ produces samples of $\tilde{Y}$ which can be used to study the properties of the emulated EDP.

8.4 Numerical examples

In this section, we compare SPCE with the methods reviewed in Section 8.2.2, namely the linear model (LM; Cornell et al., 2002), the kernel conditional distribution estimator (KCDE), and the classical classification-based fragility model (i.e., the probit model; Shinozuka et al., 2000), on two numerical examples. For the KCDE, we apply the kernel estimator developed for conditional CDF estimation (Li et al., 2013) which is available in the package `np` (Hayfield and Racine, 2008) implemented in R. To quantitatively assess the performance, we report the convergence of the models for the estimation of the conditional distribution and the fragility function.

When comparing the distribution estimation, we consider only LM, SPCE, and KCDE, as the probit model directly estimates the fragility function without providing the conditional distribution. Since LM, SPCE, and

KCDE are all applied to the logarithmic transform of the EDP, we examine the estimation accuracy of the conditional distribution of the transformed quantity. In this respect, we use the normalized Wasserstein distance (Zhu and Sudret, 2023) as the error metric which reads

$$\varepsilon = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(\log\left(Y_{\boldsymbol{X}}\right), \log\left(\tilde{Y}_{\boldsymbol{X}}\right)\right)\right]}{\mathrm{Var}\left[\log(Y)\right]}, \tag{8.16}$$

where $Y_{\boldsymbol{x}}$ is the EDP obtained from the stochastic simulation, $\tilde{Y}_{\boldsymbol{x}}$ is that of the surrogate model, and d_{WS} is the *Wasserstein distance of order two* (Villani, 2009) between two probability measures. For continuous random variables Y_1 and Y_2 with quantile functions (i.e., inverse CDF) Q_1 and Q_2, this distance can be computed by

$$d_{\mathrm{WS}}^2\left(Y_1, Y_2\right) = \|Q_1 - Q_2\|_2^2 = \int_0^1 \left(Q_1(u) - Q_2(u)\right)^2 \mathrm{d}u, \tag{8.17}$$

For the fragility model in Eq. (8.3) which is a deterministic function of $\boldsymbol{x}$, we use the relative mean-squared error to assess the global approximation accuracy

$$\varepsilon_p \overset{\text{def}}{=} \frac{\mathbb{E}\left[\left(p_f(\boldsymbol{X}) - \tilde{p}_f(\boldsymbol{X})\right)^2\right]}{\mathrm{Var}\left[p_f(\boldsymbol{X})\right]}, \tag{8.18}$$

where p_f is the fragility function of the simulator, and $\tilde{p}_f$ denotes that of the surrogate.

8.4.1 Stochastic ground motion model

This section briefly describes the simplified SGMM model used in our analysis. It is out of the scope of the current study to develop predictive equations that link the SGMM parameters to the earthquake site and source characteristics. Specifically, we employ a site-based SGMM defined in the frequency domain (Broccardo and Dabaghi, 2017; Vlachos et al., 2016). The model is the spectral representation of the original time-domain model implemented in Rezaeian and Der Kiureghian (2010). It targets broad-band excitations, which are typically associated with far-field ground motions.

In detail, the SGMM is completely characterized by an evolutionary power spectral density (EPSD; Priestley, 1965). Like its original time-domain counterpart, this representation allows separating the temporal and spectral components of the process (Broccardo and Dabaghi, 2017; Vlachos et al., 2016). In this study, without losing generality, we neglect the non-stationary spectral characteristics of the ground motion. In fact, within a good engineering approximation, the frequency content and the bandwidth of the strong ground motion phase can be assumed constant for broad-band excitations. Moreover, it is assumed that severe structural damage occurs during the strong motion phase.

Finally, the spectral content of the process is represented by a normalized stationary Kanai-Tajimi power spectral density (KT-PSD), which is a function of two parameters: the main frequency, ω_g, and the bandwidth, ζ_g. The normalized KT-PSD produces a stationary process with unit variance so that the intensity of the ground motion is completely controlled by a time-modulating function. We use a gamma modulating function (Rezaeian and Der Kiureghian, 2010; Broccardo and Dabaghi, 2017), which is completely defined by the expected Arias intensity I_a, the time at which 45% of the expected Arias intensity is reached, t_{mid}, and the effective dura-

tion of the motion, D_{5-95}. Finally, the complete SGMM EPSD is given by modulating the normalized KT-PSD with the time modulating function. Moreover, to ensure zero residual velocity and displacement, we apply a high-pass filter using the evolutionary theory of Priestley (see Broccardo and Dabaghi, 2017 for a detailed description). To summarize, the SGMM model parameters are $\boldsymbol{x} = [I_a, t_{\text{mid}}, D_{5-95}, \omega_g, \zeta_g]$.

Next, we fit the SGMM model to a catalog of recorded far-field ground motions from the PEER NGA-West2 database (the same used in Broccardo and Dabaghi, 2019). The catalog includes 71 ground motions recorded at a range of distances (10-90 km) and site conditions from reverse earthquakes with a magnitude between 6 and 7.6. The two horizontal components of each record are rotated into the major and intermediate principal directions (Rezaeian and Der Kiureghian, 2010). In this study, we used only the major component (i.e., we used 71 time series). The fitting procedure for the frequency content of the ground motion is described in detail in Broccardo and Dabaghi (2017). However, in Broccardo and Dabaghi (2017), ω_g and ζ_g are a time-varying function, while in this study ω_g corresponds to the main frequency of the ground motions at t_{mid} (which is considered the strong phase of the ground motion). Moreover, we fix ζ_g to a constant value of 0.9, which was a good approximation for the selected broad-band excitations[4]. The approach to estimates the parameters $I_a, t_{\text{mid}}, D_{5-95}$ follows Rezaeian and Der Kiureghian (2010). In this respect, the free SGMM parameters are random variables (i.e., $\boldsymbol{x}$ becomes a random vector, $\boldsymbol{X}$) to account for the epistemic uncertainty related to the chosen data set.

Provided with the 71 estimates of the free parameters, we fit a joint-probability model based on log-normal marginal distributions and a Gaussian copula (i.e., a joint log-normal distribution). Consequently, the models also account for the dependence structure among the parameters. The joint-probability model parameters are reported in Table 8.1. Finally, the simulation of the synthetic time series follows a two-step simulation (which is typical in a stochastic simulator setting). First, the SGMM parameters are sampled from the joint log-normal distribution. Second, using the synthesis formula of the frequency domain representation of a stochastic process (Shinozuka and Deodatis, 1991), the time series are generated by filtering white noise Gaussian vectors with the EPSD and the high-pass filter. Therefore, for a given set of model parameters $\boldsymbol{X} = \boldsymbol{x}$, multiple time series can be generated. Consequently, the EDP of interest is a random variable even when $\boldsymbol{X} = \boldsymbol{x}$.

Table 8.1: Ground motion parameters, g is the gravitational constant expressed in [m/s^2]

Name	Distribution
I_a [g$^2\cdot$ s]	$\mathcal{LN}\left(-4.61, 1.45^2\right)$
t_{mid} [s]	$\mathcal{LN}\left(2.55, 0.90^2\right)$
D_{5-95} [s]	$\mathcal{LN}\left(2.67, 0.53^2\right)$
ω_g [rad/s]	$\mathcal{LN}\left(1.42, 0.59^2\right)$
Correlation matrix	$R = \begin{pmatrix} 1 & 0.015 & -0.23 & -0.13 \\ 0.015 & 1 & 0.68 & -0.36 \\ -0.23 & 0.68 & 1 & -0.11 \\ -0.13 & -0.36 & -0.11 & 1 \end{pmatrix}$

[4]We found that the EDP response was not sensitive to large values of ζ_g. Therefore, we used a point approximation and reduced the parameter space. Note that this approximation does not limit the generality of the SPCE approach for fragility model computation.

8.4.2 Toy example

In this example, we introduce the properties of a three-story shear frame idealized as a three-degree of freedom system. We are interested in the dynamic response of the system subjected to the ground motions generated according to Section 8.4.1. The interstory behavior is inelastic, with a force-interstory-drift relationship based on a Bouc-Wen hysteretic model (Wen, 1976). Specifically, the i-th interstory restoring force is written as

$$q_i(v_i(t), \dot{v}_i(t)) = k_i \left[\alpha v_i(t) + (1-\alpha) z(t)\right], \tag{8.19}$$

where $v_i(t)$ denotes the interstory drift, α is a parameter that controls the degree of inelasticity (i.e., $\alpha = 1$ corresponds to the linear case), k_i is the initial elastic interstory stiffness, and $z(t)$ is the hysteretic response governed by the following law

$$\dot{z}(t) = -\gamma \left|\dot{v}(t)\right| \left|z(t)\right|^{n-1} - \eta \left|z(t)\right|^{n} \dot{v}_i(t) + A\dot{v}(t), \tag{8.20}$$

where γ, n, A and η are the model parameters. The values of structural properties, including the local masses m_i and damping c_i, and model parameters are reported in Table 8.2. The story yield displacement, δ_y, is set to 0.01 m and the post-hardening stiffness is set at 10% of the elastic stiffness k_i for all the three stories. The EDP of interest is the maximum interstory drift, i.e.,

$$Y = \max\left[\max_t[v_1(t)], \max_t[v_2(t)], \max_t[v_3(t)]\right]. \tag{8.21}$$

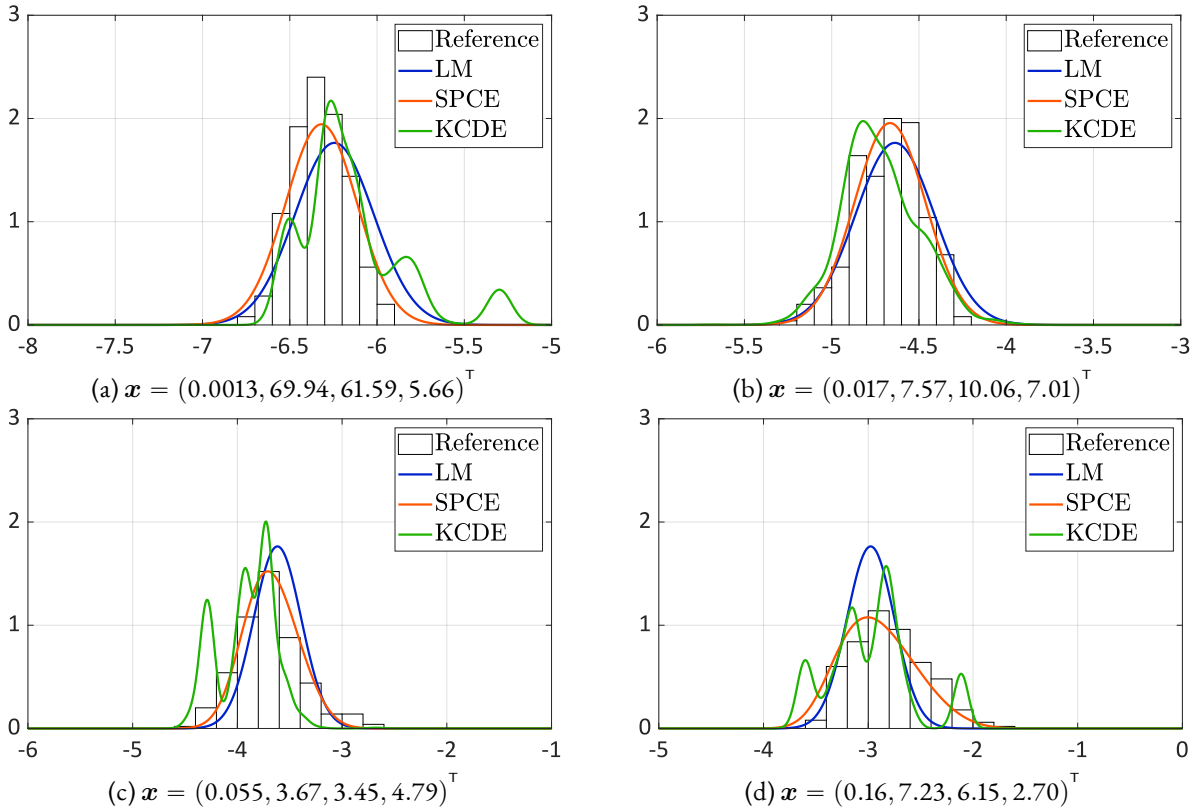
Table 8.2: Structural properties and Bouc-Wen parameters ($\delta_y = 0.01$ m).

	$m_i\ 10^6$[kg]	c_i [10^6[Ns/m]	$k_i\ 10^8$[N/m]	α	n	γ [1/m^n]	η [1/m^n]	A
Story 1	1	1.73	3.0	0.1	5	$1/(2\delta_y)^n$	$1/(2\delta_y)^n$	1
Story 2	1	1.73	2.4	0.1	5	$1/(2\delta_y)^n$	$1/(2\delta_y)^n$	1
Story 3	1	1.73	1.5	0.1	5	$1/(2\delta_y)^n$	$1/(2\delta_y)^n$	1

Figure 8.1 illustrates the conditional PDF of the maximum interstory drift for four different values of the ground motion variables. The models are constructed based on a total number of 1,000 simulations. The reference histograms are obtained by replicating the simulation 250 times for each set of ground motion parameters, i.e., we generated 250 ground motions for each $\boldsymbol{x}$ and computed the associated structural responses. The distributions are plotted on the logarithmic transform of $Y_{\boldsymbol{x}}$, which allows for verifying the assumptions of the linear model.

As shown in Figure 8.1, the linear model can represent the overall location and shape of the conditional distribution: the prediction of the mean values are close to the reference histograms that demonstrate normal-like shapes. Nevertheless, the linear model loses some details of the mean estimation in Figures 8.1a and 8.1c and cannot capture the heteroskedastic effect (varying variance). The PDF predictions of KCDE are quite poor, as it yields spurious oscillations in Figures 8.1a, 8.1c and 8.1d. This is because the bandwidth selection procedure (Hayfield and Racine, 2008) is designed for estimating the conditional CDF. Moreover, the conditional distribution estimation requires estimating the joint distribution of $(\boldsymbol{X}, Y)$ in Eq. (8.7) which is of dimension 5.

This is rather high for nonparametric estimators and leads to the observed poor predictions. In contrast, SPCE can accurately emulate the PDFs in terms of not only the location and the heteroskedastic effect but also the shape of the distributions: Figure 8.1d is slightly right-skewed which is well represented by SPCE.



(a) $\boldsymbol{x} = (0.0013, 69.94, 61.59, 5.66)^\intercal$ (b) $\boldsymbol{x} = (0.017, 7.57, 10.06, 7.01)^\intercal$

(c) $\boldsymbol{x} = (0.055, 3.67, 3.45, 4.79)^\intercal$ (d) $\boldsymbol{x} = (0.16, 7.23, 6.15, 2.70)^\intercal$

Figure 8.1: Example 1 — comparison of emulated PDFs of $\log(Y_{\boldsymbol{x}})$ for four different values of $\boldsymbol{x}$; the models are built on $N = 1{,}000$ simulations.

To study the convergence of the various methods, we generated a big data pool of size 10^5 (following the distribution of $\boldsymbol{X}$ described in Table 8.1). We randomly subsampled it to have samples of desired sizes $N \in \{250; 500; 1{,}000; 2{,}000; 4{,}000\}$ to train the models. Note that this mimics the procedure of random design of experiment. To account for the uncertainties in the estimation, we repeated the procedure 20 times for each sample size (i.e., we obtain 20 models constructed on independent subsamples for each N). To evaluate the error metrics defined in Eqs. (8.16) and (8.18), we generated a validation set of size 400. For each validation point, we used 250 replications to have a reference distribution (meaning a total number of 400×250 simulations for the validation set). The error estimates for each sample size are represented by box plots constructed from the 20 repetitions of the full analysis.

Figure 8.2 shows the results of the models for estimating the conditional distribution. For relatively small sample sizes $N \leqslant 500$, the linear model gives the best results. This is because the linear model is very simple, and its assumptions are relatively "suitable" for this example. More precisely, the error of a statistical model can be decomposed into bias and variance (James et al., 2014). In Figure 8.1, we observe that the conditional distribution is close to Gaussian, the mean function does not exhibit a strong nonlinearity, and the heteroskedastic effect is relatively weak. Therefore, the bias of the linear model is rather small. Because of its simplicity, the linear model has a small variance. As a result, when only a few data points are available, the linear model gives the best

results. However, with increasing sample size, the errors of the linear model run into a plateau. This is due to the irreducible bias (caused by the model misspecification). On the contrary, SPCE and KCDE are more flexible models that have smaller bias but bigger variance. Hence, both models exhibit a clear decay of the error. Due to its nonparametric feature, KCDE is merely comparable to the linear model for $N = 4{,}000$. When enough samples are available, i.e., $N \geqslant 1{,}000$, SPCE is the best model. Furthermore, the average error of SPCE is three times smaller than those of the linear model and kernel estimator for $N = 4{,}000$.

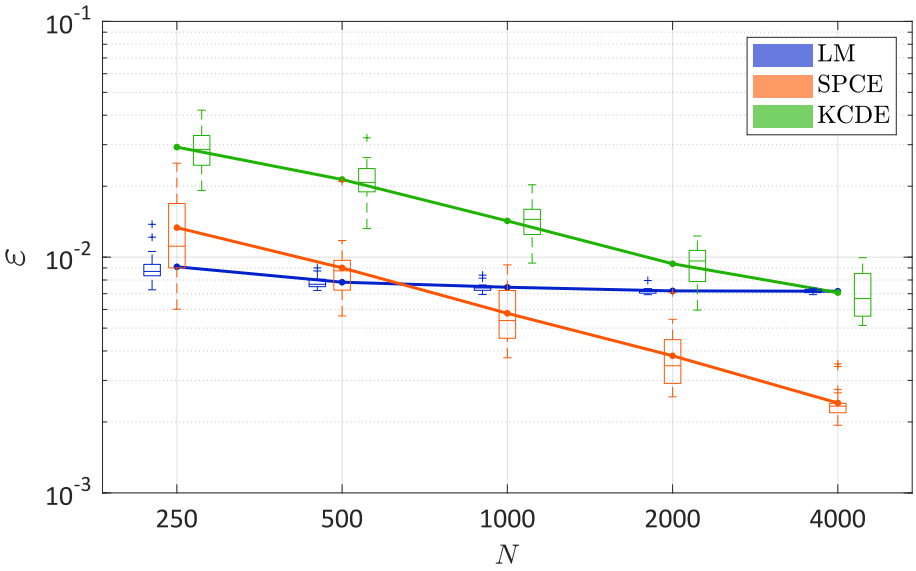
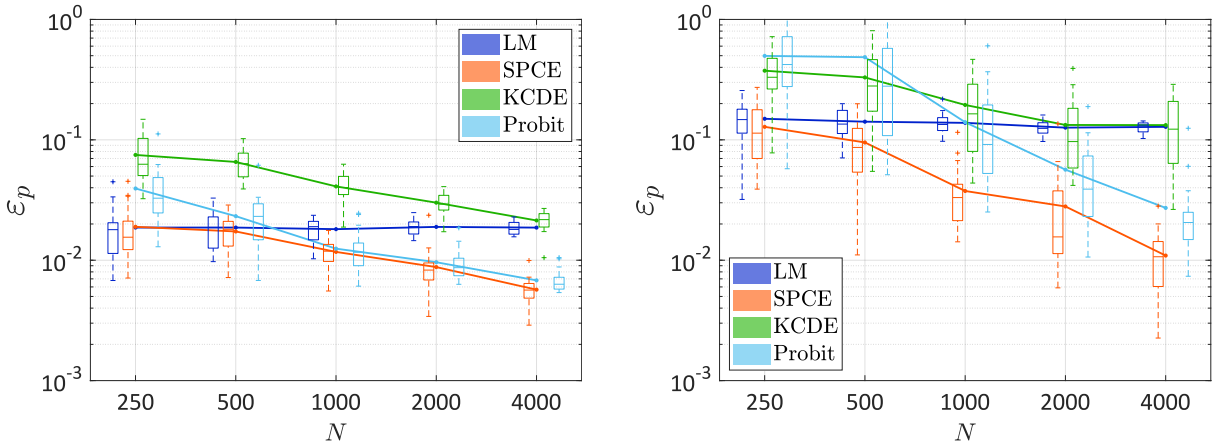


Figure 8.2: Example 1 — comparison of the convergence among the models in terms of the normalized Wasserstein distance. The lines correspond to the average values over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results.

When considering fragility functions, we select two thresholds $\delta_0 = 0.02$ m and $\delta_0 = 0.07$ m. The relative mean-squared errors for estimating the associated fragility functions are reported in Figure 8.3. In general, SPCE produces the best overall approximation to the fragility functions for all sample sizes. Similar to what we observed in Figure 8.2, the performance of the linear model barely improves with increasing N. For $\delta_0 = 0.07$ m, SPCE outperforms the linear model in the case of a few samples $N \leqslant 500$. This indicates that SPCE better approximates the tails. The probit model yields relatively large errors for training sets of sizes $N \leqslant 1{,}000$ in the estimation of the fragility function associated with $\delta_0 = 0.07$ m. This is because this model ignores the precise values of EDP and only works with the binary variable. For $\delta_0 = 0.07$ m, only a small fraction of samples (about 1.3%) exceed the threshold. Consequently, the probit model only produces reliable estimates for large N. Finally, KCDE performs quite poorly even though the associated bandwidth selection procedure is designed for CDF estimation.

8.4.3 Three-story frame

As a second example, we apply the methods to study the three-story steel frame modeled with the software OpenSees (Pacific Earthquake Engineering and Research Center, 2004). The geometry of the structure is shown in Figure 8.4a, and the story height and floor span are $H = 3$ m and $L = 5$ m, respectively. We choose the standard European IPE A 330 for the beams and HE 200 AA for the columns.



(a) Fragility function for $\delta_0 = 0.02$ m

(b) Fragility function for $\delta_0 = 0.07$ m

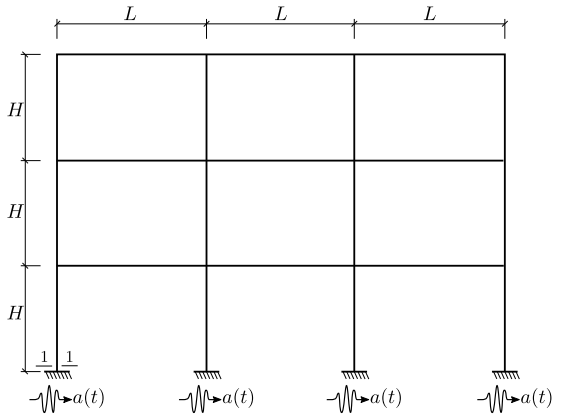
Figure 8.3: Example 1 — comparison of the convergence among the models in terms of the fragility functions. The lines correspond to the average values over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results.

The mechanical property of the steel follows the uniaxial Giuffre-Menegotto-Pinto model with isotropic strain hardening (material of type "Steel02" in OpenSees). More precisely, we set the Young's modulus to $E = 205{,}000$ MPa, the yield stress to $f_y = 235$ MPa, and the strain hardening ratio to $b = 0.01$ (the other parameters controlling the elastic-plastic transition are given by $R0 = 18$, $CR1 = 0.925$, and $CR2 = 0.15$). The load applied to the structure consists of dead load (weight of frame elements and supported floors) and live load, which results in a total distributed load on the beams equal to $q = 20$ kN/m (Mai et al., 2017).

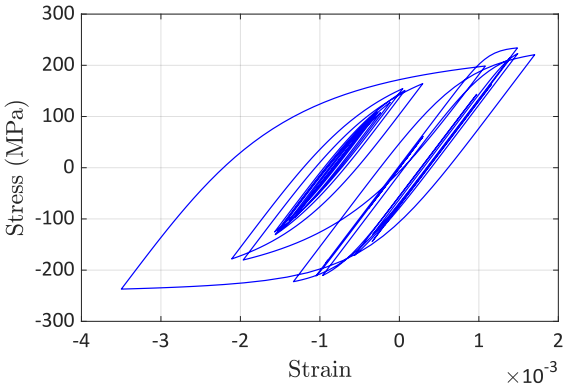
The structural components (beams and columns) are modeled by nonlinear beam elements based on the iterative force-based formulation. The element cross-sections are defined by a set of fiber sections, which allows modeling the plasticity over the cross-section. Figure 8.4b illustrates the stress-strain relation of the bottom left column for the frame under an example ground motion. The first two fundamental periods of the structure are 0.950 s and 0.317 s (from modal analysis), respectively. In this study, we are interested in the dynamic response of the system subjected to the ground motions generated according to Section 8.4.1. The EDP of interest is the maximum interstory drift ratio.

Figure 8.5 shows the prediction of the conditional PDFs for four different values of $\boldsymbol{x}$. The reference histogram of each $\boldsymbol{x}$ is calculated by performing 250 replications, and the surrogate models are built on 1,000 simulations. Similar to the first example Figure 8.1, we observe that the conditional distributions have bell shapes that are close to Gaussian distributions. The linear model can well approximate the location of the distributions, so the (log-)mean function does not demonstrate a strong non-linearity. The variance of the conditional distribution does not vary too much. The linear model shows a good overall approximation, but it fails to characterize the precise variation of the distribution. On the contrary, the kernel method is too flexible and completely mispredicts the shape of the distribution. In contrast, SPCE turns out to accurately represent not only the location and shape of the distribution but also the heteroskedastic effect.

For the convergence study, we followed the same procedure as Section 8.4.2. In this example, we generated a data pool of size 50,000. We randomly subsampled this data set to have the experimental design of sizes N $\{250; 500; 1{,}000; 2{,}000; 4{,}000\}$ to build the surrogate models. We repeated the analysis 20 times for each

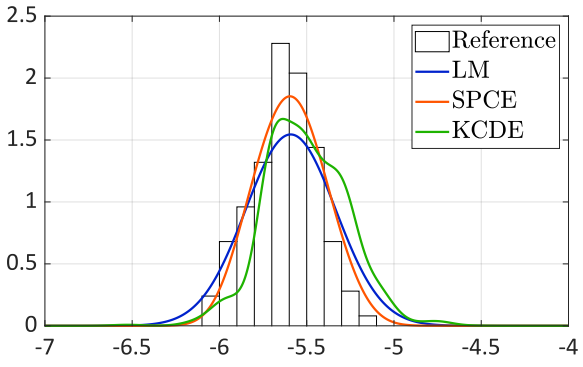


(a) Illustration of the frame structure

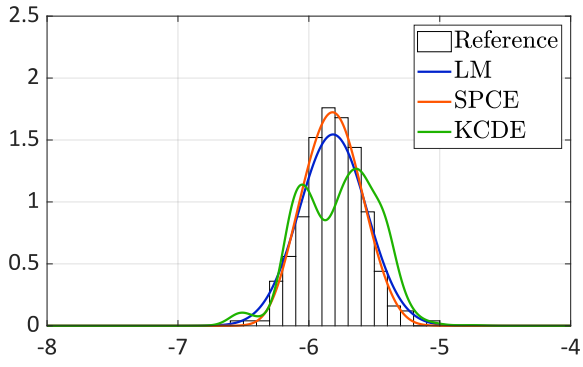


(b) Hysteric behavior of the steel material at section 1-1 for a ground motion

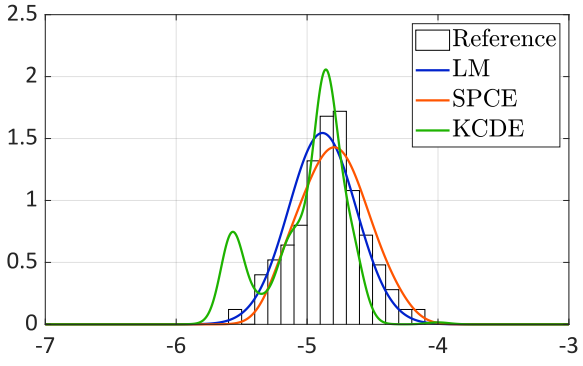
Figure 8.4: Example 2 — three-story steel frame.



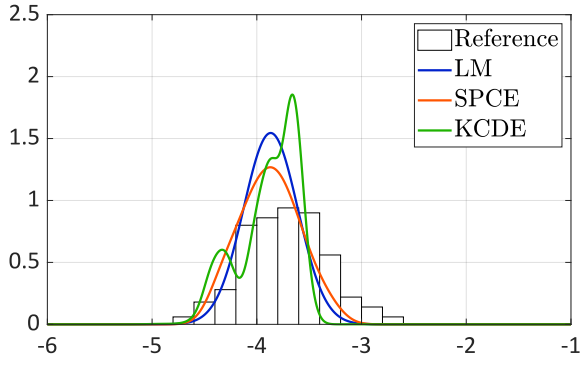
(a) $\boldsymbol{x} = (0.0098, 44.21, 47.08, 2.33)^{\intercal}$



(b) $\boldsymbol{x} = (0.0058, 31.36, 26.89, 2.69)^{\intercal}$



(c) $\boldsymbol{x} = (0.014, 15.15, 10.19, 1.24)^{\intercal}$



(d) $\boldsymbol{x} = (0.11, 12.85, 5.49, 1.76)^{\intercal}$

Figure 8.5: Example 2 — comparison of emulated PDFs of $\log(Y_{\boldsymbol{x}})$ for four different values of $\boldsymbol{x}$; the models are built on $N = 1{,}000$ simulations.

value of N to account for the uncertainties (due to the random ground motion parameters and the intrinsic stochasticity of the ground motion model). To evaluate the error defined in Eqs. (8.16) and (8.18), we created a validation set of size 200, and we performed 250 replications for each validation point to have a reference conditional distribution.

Figure 8.6 shows the error metric defined in Eq. (8.16). Similar to Figure 8.2, the linear model is superior to SPCE and KCDE when only $N = 250$ data points are used. With increasing N, its errors exhibit narrower spreads, but the average values do not decrease due to the bias resulted from the model simplicity. The kernel estimator exhibits a better convergence rate but performs poorly overall. SPCE has a similar performance to the linear model at $N = 500$ and surpasses the latter for $N \geqslant 1{,}000$. In addition, SPCE has a clear decay of the errors with a similar rate to KCDE. For $N = 4{,}000$, the average error of SPCE is less than half of those of the linear model and KCDE.

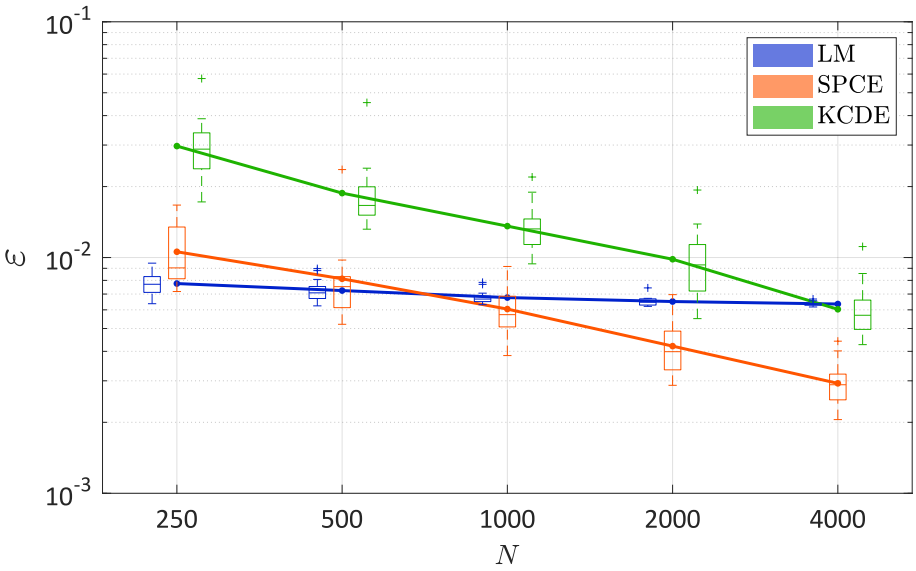
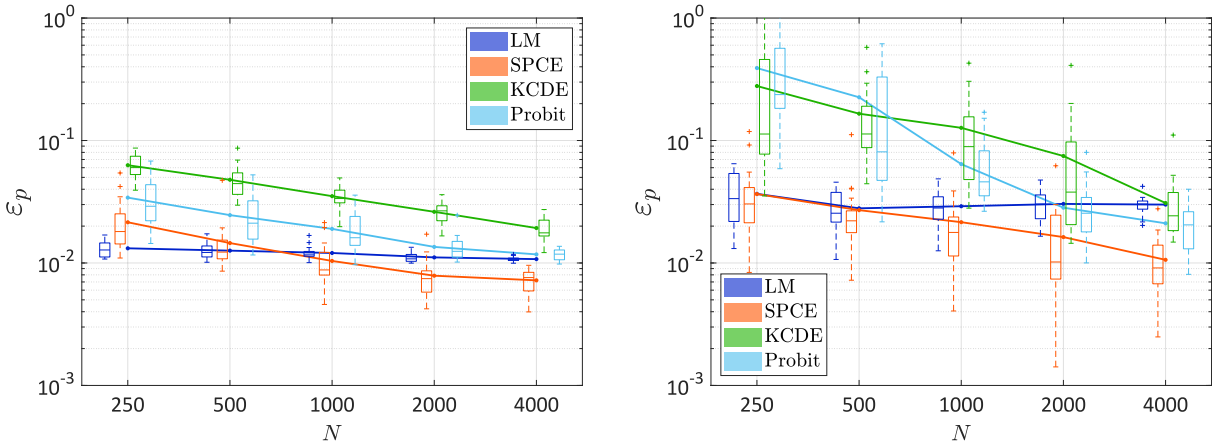


Figure 8.6: Example 2 — comparison of the convergence among the models in terms of the normalized Wasserstein distance. The lines correspond to the average values over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results.

For fragility function, we select two thresholds $\delta_0 = 0.7\%$ and $\delta_0 = 2.5\%$ which are typically used to characterize light and moderate damages for steel frames (Federal Emergency Management Agency, 2000). The relative mean-squared errors for estimating the associated two fragility functions are reported in Figure 8.7. For the small threshold of $\delta_0 = 0.7\%$, the results are similar to the distribution estimation in Figure 8.6. Specifically: first, the linear model yields the best estimates of the fragility function when small data sets of $N = 250$ are considered, but the errors get stagnant with more data; second, KCDE is too flexible to estimate robustly the fragility function due to its nonparametric feature; third, SPCE performs similarly to the linear model for $N = 500$ but outperforms all the other models for $N \geqslant 1{,}000$. Unlike the first example (Figure 8.3a), the errors of the probit model are not comparable to these of SPCE but between the linear model and KCDE. For the high threshold of $\delta_0 = 2.5\%$, SPCE is the best model for all values of N. The simplicity of the linear model leads to a significant irreducible bias. In contrast, SPCE, KCDE, and the probit model all demonstrate a clear decay of the errors. The kernel estimator has a large spread of errors but a slow convergence of the average value. The probit model performs poorly for $N \leqslant 500$ because the model ignores the precise values of the EDP and only

a few data points exceed the threshold (ca. 1.8% in the data set). In summary, SPCE generally provides more accurate estimates of the fragility functions than the other models.



(a) Fragility function for $\delta_0 = 0.7\%$

(b) Fragility function for $\delta_0 = 2.5\%$

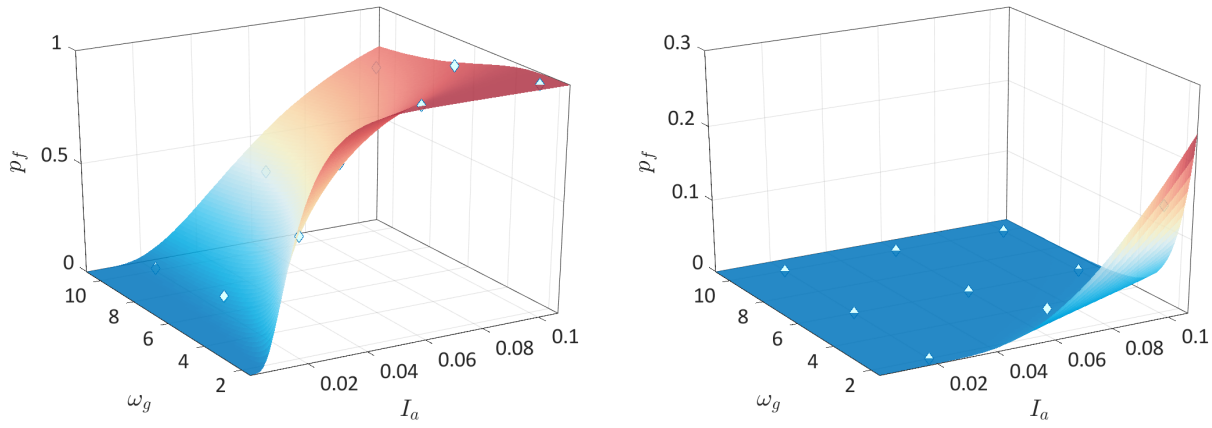
Figure 8.7: Example 2 — comparison of the convergence among the models in terms of the fragility function. The lines correspond to the average values over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results.

In this example, we plot the two fragility functions in the $I_a - \omega_g$ plan of an SPCE built upon 1,000 model evaluations in Figure 8.8. The plotted fragility models are obtained by averaging out the functions with respect to t_{mid} and D_{5-95}. Specifically, we obtain the "cross section" fragility model conditional to each $\{t_{\text{mid}}, D_{5-95}\}$ sample and, then, we compute the average fragility model.

We choose $I_a - \omega_g$ which are the most important parameters of the fragility functions according to a sensitivity analysis. This outcome is in line with the results reported in Abbiati et al. (2021). As a comparison, we run the simulator for a validation set of nine points obtained by the Cartesian product of $I_a \in \{0.02, 0.06, 0.1\}$ and $\omega_g \in \{2, 6, 10\}$. The reference failure probability associated with each validation point is computed by 250 replications (i.e., a total number of 2,250 simulations for validation). As seen in Figure 8.8, the diamonds representing the reference points lie fairly well on the estimated fragility surface. More precisely, the average absolute error of SPCE (averaged over the 9 validation points) is 2.7% for $\delta_0 = 0.7\%$ and 0.7% for $\delta_0 = 2.5\%$. In this case, we observe that the dominant variable is the Arias intensity I_a (also confirmed by the sensitivity analysis). This was expected, given the broad-band nature of the excitation, which "spread" the energy content among the full range of frequencies.

8.4.4 Discussion

The models considered in this paper were constructed on data without replications. For replication-based approaches (Gidaris et al., 2015; Abbiati et al., 2021), a typical number of $\mathcal{O}(10^2)$ replications are used. Following such a strategy, the amount of points exploring the input space would significantly reduce to only $\mathcal{O}(10)$ (as the total number of simulations varies in $\{250; 500; 1{,}000; 2{,}000; 4{,}000\}$). This does not allow for good coverage of the input space, especially when the failure occurs with a higher probability at the tail of the input distribution. Moreover, using replications in the estimation of conditional distributions of a parametric model is also not optimal, as shown in Zhu and Sudret (2021).



(a) Fragility function for $\delta_0 = 0.7\%$

(b) Fragility function for $\delta_0 = 2.5\%$

Figure 8.8: Example 2 — Fragility function in the $I_a - \omega_g$ plan of a SPCE built on 1,000 samples. The diamond points correspond to the reference value computed from 250 replications.

Our numerical results demonstrate that SPCE is accurate for estimating both the conditional distribution and the fragility functions for different thresholds. Therefore, SPCE provides a good balance between the model flexibility and limited data. The linear model performs usually well for small values of N but cannot correctly approximate fragility functions with large thresholds. Due to its restrictive assumptions, the linear model cannot be further improved by using more data. Surprisingly, the kernel estimator is almost always the worst model despite that the bandwidth selection procedure is designed for CDF estimation. The probit model directly estimates the fragility function and has a rather low accuracy compared to the other models.

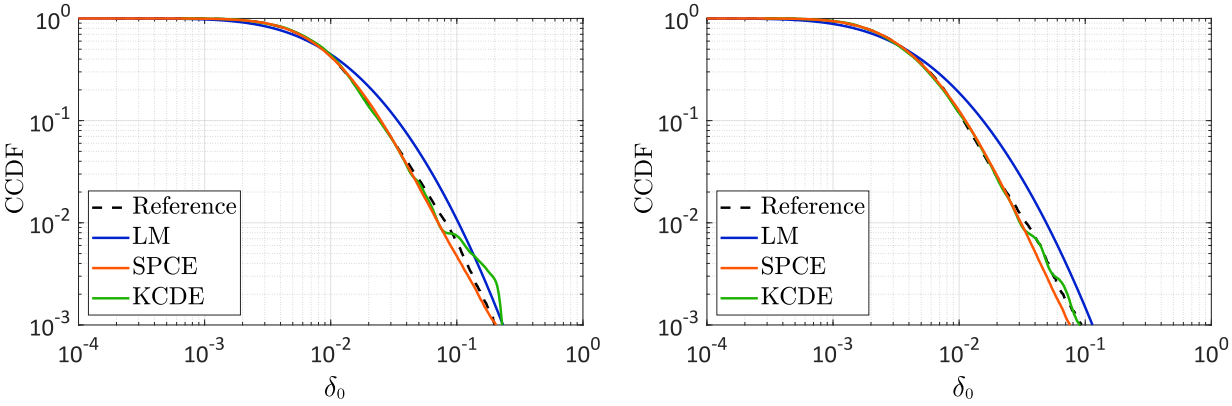
8.5 Additional post-processing

8.5.1 CCDF of the EDP

As the conditional distribution is available from SPCE, one can aggregate the uncertainties in $\boldsymbol{X}$ and evaluate the overall risks by uncertainty propagation. As an example, we can compute the complementary cumulative distribution function (CCDF) defined by $\mathbb{P}(Y \geqslant \delta)$ of the EDP by resampling Y from SPCE. This represents the unconditioned exceeding probability of the EDP as a function of δ_0.

In Figure 8.9, we plot the CCDFs of the two examples estimated by LM, SPCE, and KCDE, as the probit model does not allow resampling the EDP. The reference curves are the empirical CCDFs using all the available samples (10^5 for the first example and 50,000 for the second example). The surrogate models are built on 1,000 simulations.

We observe that the linear model exhibits a systematic gap at the tail: it overestimates the exceeding probabilities for relatively large values of δ_0, which cannot be reduced by increasing N. The CCDFs obtained from the kernel estimator are generally more accurate than the linear model but are unstable for big values of δ_0 in Figure 8.9a. SPCE achieves a high accuracy in Figure 8.9a but has a slight discrepancy at the tail in Figure 8.9b) which, according to the numerical investigation, can be efficiently reduced by using more data.



(a) Example 1 (3-DOF system) — CCDF of the maximum interstory drift

(b) Example 2 (OpenSees model) — CCDF of the maximum interstory drift ratio

Figure 8.9: Comparisons of the CCDF estimation (the models are built on $N = 1{,}000$).

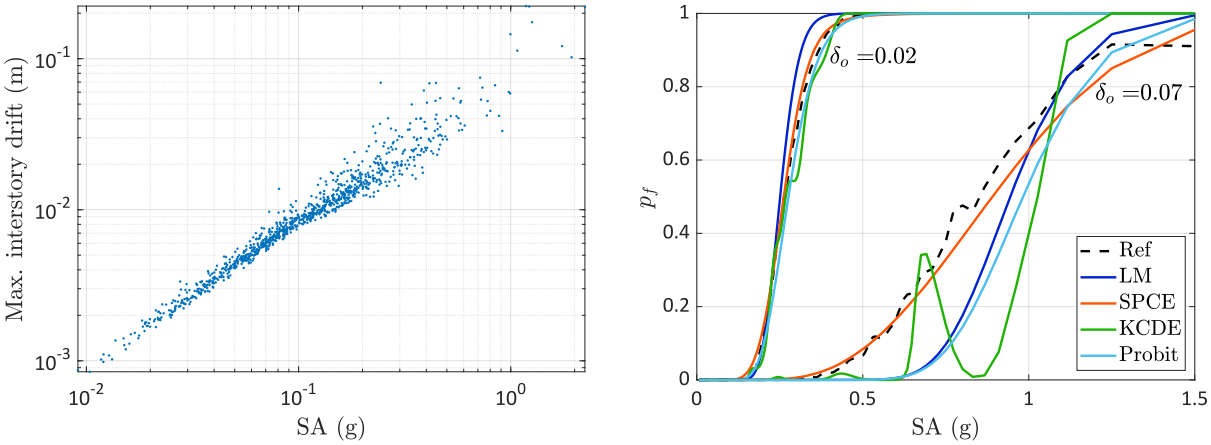
8.5.2 Classical fragility curves

With the data generated for estimating the distribution of EDP conditioned on the ground motion parameters, we can also compute the fragility curves with respect to a classical IM, such as peak ground acceleration (PGA) or spectral acceleration at the fundamental frequency (SA). More precisely, we first extract the values of the selected IM from the synthetic seismograms and then apply the proposed method to estimate conditional distributions which, by post-processing, gives the fragility curves. As an illustration, we choose a data set of size 1,000 to estimate the fragility curves for each of the examples in Sections 8.4.2 and 8.4.3.

For the first example (the 3-DOF system), we select the spectral acceleration (SA) as IM. More specifically, SA corresponds to the spectral acceleration for a single-degree-of-freedom system with a period equal to the fundamental period of the structural and viscous damping ratio equal to 2%. Figure 8.10a shows the scatter plot of the 1,000 data points. We observe that the data have a strong heteroskedasticity (in the log-log scale) reflecting a typical nonlinear structural behavior.

Figure 8.10b summarizes the fragility curves estimated by the different models (constructed on the data illustrated in Figure 8.10a) for $\delta_0 = 0.02$ m and $\delta_0 = 0.07$ m. The reference fragility curves are computed by applying the kernel estimator to all the available data (i.e., 10^5). Due to the heteroskedastic effect and the possibly non-Gaussian shape of the conditional distribution, the linear model has a significant gap to the reference, in particular for $\delta_0 = 0.07$. KCDE produces an irregular fragility curve for $\delta_0 = 0.07$. The reason is that most of the data are in the region where the intrinsic variability is not significant, which leads to a small value of the selected bandwidth. This results in a large variance of the estimation in the region where the data are sparse, as KCDE is a local estimator. The probit model is quite accurate for $\delta_0 = 0.02$, but it yields unstable estimate of the fragility curve for $\delta_0 = 0.07$. This is because only a few points (9 out of 1,000) lead to exceedance. Finally, SPCE built on the data in Figure 8.10b approximates the fragility curves with high accuracy.

For the second example (OpenSees model), we use the PGA as intensity measure. Figure 8.11a shows the scatter plot of a data set of size $N = 1{,}000$. We observe that the PGA is a less relevant IM than the SA as the relationship PGA-EDP shows a much larger variability. Therefore, the derived fragility curves are less informative than the previous ones. The PGA-EDP relationship is close to linear with a homoscedastic noise (in the log-log scale). Therefore, the linear model is able to approximate well the fragility curves with relatively small



(a) Training samples of size $N = 1{,}000$

(b) Comparison of fragility curve estimation

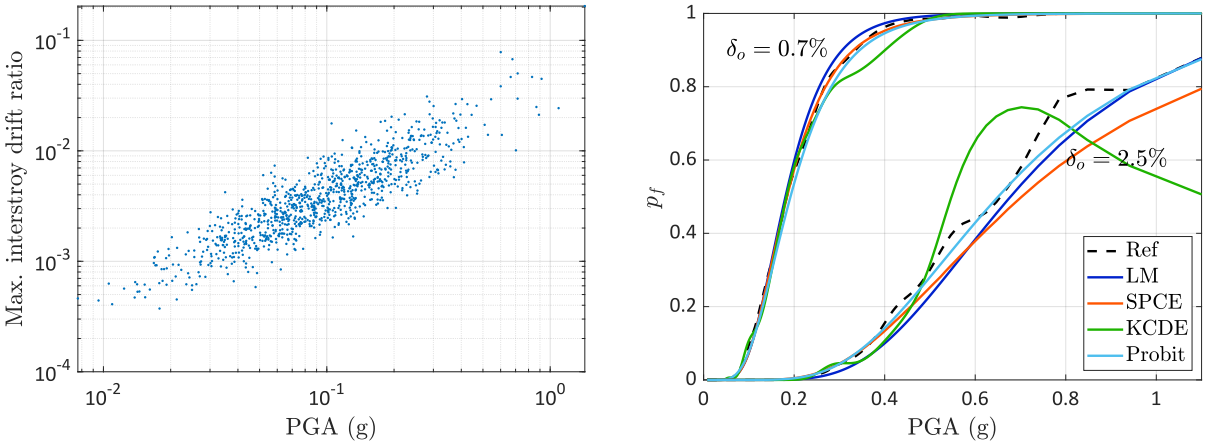
Figure 8.10: Example 1 — fragility curves using spectral acceleration as IM.

biases. Unlike Figure 8.10b, the kernel estimator yields smooth predictions, but the fragility curve shows a non-increasing behavior for $\delta_0 = 2.5\%$. The probit model and SPCE provide the most accurate estimate for the fragility curve of $\delta_0 = 0.7\%$. However, in this case, SPCE underestimates the exceeding probabilities associated with the threshold of $\delta_0 = 2.5\%$ for very large values of PGA. This is because most of the data are in the region where PGA is small and the structure does not fail with very high probabilities: the 95% and 99% quantiles of PGA are $0.351g$ and $0.605g$, and the associated reference exceeding probabilities are 0.079 and 0.4427, respectively. The SPCE is a flexible model developed to estimate the overall conditional distribution (with respect to the probability distribution of the IM), but not designed to fit directly the tail of the distribution. As a consequence, in specific cases, it may suffer of over-fitting and lack of robust extrapolation behavior for extreme quantiles. In this case, the problem is exacerbated by the relative large variability between PGA and EDP, which makes difficult the estimation of the tail of the distribution.

The lack of failure data for large damage thresholds is a well-known problem in fragility analysis. In the classical framework for fragility computation based on real ground motions, this problem is overcome by scaling the ground motions and fitting procedures based on censored data (Baker, 2015). In the context of stochastic simulation, scaling is not recommended (Grigoriu, 2011). A promising future research line is to develop an importance sampling scheme to simulate extreme events from the SGMM model and fill adaptively the EDP intervals of interest. Observe that the presented SPCE approach is orthogonal to this research line and can be easily adapted and applied once the adaptive importance density scheme is developed.

8.6 Conclusions

In this paper, we propose methods to efficiently perform fragility analysis based on artificial ground motions, following the recent development in Abbiati et al. (2021). We characterize the ground motion model by a few engineering-meaningful parameters that are calibrated from seismic records and modeled by random variables. Combing this model with the dynamical analysis of structures, we obtain a stochastic simulator: for a given set of ground motion parameters, the engineering demand parameter that characterizes the structural damage is random. Because of this non-deterministic relation, classical surrogate models cannot be used to represent the



(a) Training samples of size $N = 1{,}000$

(b) Comparison of fragility curve estimation

Figure 8.11: Example 2 — fragility curves using peak ground acceleration as IM.

simulator.

Some methods that have been developed for estimating classical fragility curves can be extended and applied by regarding the ground motion parameters as multiple intensity measures. To have a reliable model without introducing restrictive assumptions, we propose using the recently developed stochastic surrogate model called *stochastic polynomial chaos expansion* to emulate the conditional distribution. This model introduced an artificial latent variable and a noise variable to reproduce the stochastic behavior of the earthquake simulation.

The performance of the proposed method is illustrated by two numerical examples: a three-degree-of-freedom system and a 3-story steel frame (modeled in OpenSees). For the conditional distribution estimation, SPCE is compared with the linear model and a state-of-the-art kernel conditional distribution estimator. Using an appropriate error measure defined in Eq. (8.16) to assess the accuracy, we observe that the linear model reaches its performance limit for only $N = 250$ simulations because of its simplicity. The kernel estimator is too flexible to have a stable estimate as a consequence of its nonparametric feature. In contrast, SPCE demonstrates a steep decay of the errors and yields the best approximation for $N \geqslant 1{,}000$.

For the fragility function, we include the probit model in the comparison. The results show that SPCE prevails over the other models, especially for higher thresholds. In addition, SPCE can be used to propagate the uncertainties in the ground motion parameters to evaluate the overall risks. By resampling the model, SPCE can accurately estimate the complementary cumulative distribution function with limited data even at the tail. Furthermore, one can also apply the method to estimate the fragility curves with respect to classical intensity measures, i.e., PGA and SA.

SPCE can generally produce accurate estimates of the fragility curves. However, the data are mostly in the safe region for a high threshold because of the sampling procedure. Thus, extrapolating SPCE for extreme quantiles of the intensity measure with limited data is not reliable. To cope with small exceeding probabilities, adaptive design strategies remain to be explored. The simulation scheme will not simply sample the distribution of the ground motion parameters but adaptively select the samples in the region where the structure is prone to fail to improve the predictive quality of the surrogate model (Echard et al., 2011; Marelli and Sudret, 2018). This will also benefit the estimation of fragility curves with classical IM.

We underline that the proposed method can be extended by considering uncertain parameters in the struc-

tural properties to tackle a larger set of problems. It can also be applied to model other probabilistic components in PBEE such as relating decision variables (e.g., monetary loss) to structural damage and the damage state to EDP. With these models representing the conditional distributions, one can evaluate the exceeding probability function of the decision variables by resampling (similar to the calculation of the CCDF in Figure 8.9). Studies in this direction are currently under investigation.

Acknowledgment

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524), whose support is gratefully acknowledged.

References

Abbiati, G., Broccardo, M., Marelli, S., and Paolacci, F. (2021). Seismic fragility analysis based on artificial ground motions and surrogate modeling of validated structural simulators. *Earthquake Engineering & Structural Dynamics*, 9:2314–2333.

Baker, J. W. (2015). Efficient analytical fragility function fitting using dynamic structural analysis. *Earthquake Spectra*, 31(1):579–599.

Baker, J. W. and Cornell, C. A. (2005). A vector-valued ground motion intensity measure consisting of spectral acceleration and epsilon. *Earthquake Engineering & Structural Dynamics*, 34:1193–1217.

Baker, J. W. and Cornell, C. A. (2006). Spectral shape, epsilon and record selection. *Earthquake Engineering & Structural Dynamics*, 35:1077–1095.

Broccardo, M. and Dabaghi, M. (2017). A spectral-based stochastic ground motion model with a non-parametric time-modulating function. In *12th International Conference on Structural Safety and Reliability, Vienna, Austria*, volume 2017, pages 1–10.

Broccardo, M. and Dabaghi, M. (2019). Preliminary validation of a spectral-based stochastic ground motion model with a non-parametric time-modulating function. In *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13), Seoul, South Korea*.

Cornell, C. A. (1968). Engineering seismic risk analysis. *Bulletin of the Seismological Society of America*, 58(5):1583–1606.

Cornell, C. A., Jalayer, F., Hamburger, R. O., and Foutch, D. A. (2002). Probabilistic basis for 2000 SAC federal emergency management agency steel moment frame guidelines. *Earthquake Engineering & Structural Dynamics*, 128:526–533.

Cornell, C. A. and Krawinkler, H. (2000). Progress and challenges in seismic performance assessment. *PEER center news*, 3(2):1–3.

Duong, T. and Hazelton, M. L. (2005). Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32:485–506.

Echard, B., Gayton, N., and Lemaire, M. (2011). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*, 33(2):145–154.

Federal Emergency Management Agency (2000). Commentary for the seismic rehabilitation of buildings. Technical report.

Ghosh, S. and Chakraborty, S. (2020). Seismic fragility analysis of structures based on Bayesian linear regression demand models. *Probabilistic Engineering Mechanics*, 61:103081.

Ghosh, S., Roy, A., and Chakraborty, S. (2021). Kriging metamodeling-based Monte Carlo simulation for improved seismic fragility analysis of structures. *Journal of Earthquake Engineering*, 25:1316–1336.

Gidaris, I., Taflanidis, A. A., and Mavroeidis, G. P. (2015). Kriging metamodeling in seismic risk assessment based on stochastic ground motion models. *Earthquake Engineering & Structural Dynamics*, 44:2377–2399.

Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Grigoriu, M. (2011). To scale or not to scale seismic ground-acceleration records. *Journal of Engineering Mechanics*, 137(4):284–293.

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27:1–32.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Kiani, J. and Khanmohammadi, M. (2015). New approach for selection of real input ground motion records for incremental dynamic analysis (IDA). *J. Earthq. Eng.*, 19:592–623.

Li, Q., Lin, J., and Racine, J. S. (2013). Optimal bandwidth selection for nonparametric conditional distribution and quantile functions. *Journal of Business & Economic Statistics*, 31:57–65.

Luco, N. and Bazzurro, P. (2007). Does amplitude scaling of ground motion records result in biased nonlinear structural drift responses. *Earthquake Engineering & Structural Dynamics*, 36:1813–1835.

Mackie, K. and Stojadinović, B. (2003). *Seismic demands for performance-based design of bridges*. Pacific Earthquake Engineering Research Center Berkeley.

Mai, C. V., Konakli, K., and Sudret, B. (2017). Seismic fragility curves for structures using non-parametric representations. *Frontiers of Structural and Civil Engineering*, 11(2):169–186.

Marelli, S. and Sudret, B. (2018). An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*, 75:67–74.

Modica, A. and Stafford, P. J. (2014). Vector fragility surfaces for reinforced concrete frames in Europe. *Bulletin of Earthquake Engineering*, 12:1725–1753.

Noh, H. Y., Lallemant, D., and Kiremidjian, A. S. (2015). Development of empirical and analytical fragility functions using kernel smoothing methods. *Earthquake Engineering & Structural Dynamics*, 44:1163–1180.

Pacific Earthquake Engineering and Research Center (2004). *OpenSees: The Open System for Earthquake Engineering Simulation*.

Priestley, M. B. (1965). Evolutionary spectra and non-stationary processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):204–229.

Rezaeian, A. and Der Kiureghian, A. (2008). A stochastic ground motion model with separable temporal and spectral nonstationarities. *Earthquake Engineering & Structural Dynamics*, 37:1565–1584.

Rezaeian, A. and Der Kiureghian, A. (2010). Simulation of synthetic ground motions for specified earthquake and site characteristics. *Earthquake Engineering & Structural Dynamics*, 39:1155–1180.

Seyedi, D. M., Gehl, P., Douglas, J., Davenne, L., Mezher, N., and Ghavamian, S. (2010). Development of seismic fragility surfaces for reinforced concrete buildings by means of nonlinear time-history analysis. *Earthquake Engineering & Structural Dynamics*, 39:91–108.

Shinozuka, M. and Deodatis, G. (1991). Simulation of stochastic processes by spectral representation. *Applied Mechanics Reviews*, 44(4):191–204.

Shinozuka, M., Feng, M., Lee, J., and Naganuma, T. (2000). Statistical analysis of fragility curves. *Journal of Engineering Mechanics*, 126:1224–1231.

Smerzini, C. and Pitilakis, K. (2018). Seismic risk assessment at urban scale from 3D physics-based numerical modeling: the case of Thessaloniki. *Bulletin of Earthquake Engineering*, 16:2609–2631.

Taflanidis, A. A. and Beck, J. L. (2009). Life-cycle cost optimal design of passive dissipative devices. *Structural Safety*, 31(6):508–522.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, Cambridge, New York.

Vamvatsikos, D. and Cornell, C. (2002). Incremental dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 31:491–514.

Villani, C. (2009). *Optimal Transport, Old and New*. Springer, Berlin.

Vlachos, C., Papakonstantinou, K. G., and Deodatis, G. (2016). A multi-modal analytical non-stationary spectral model for characterization and stochastic simulation of earthquake ground motions. *Soil Dynamics and Earthquake Engineering*, 80:177–191.

Wen, Y.-K. (1976). Method for random vibration of hysteretic systems. *J. Eng. Mech.*, 102(2):249–263.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

Zhu, X. and Sudret, B. (2023). Stochastic polynomial chaos expansions to emulate stochastic simulators. *International Journal for Uncertainty Quantification*, 13:31–52.

Every new beginning comes from some other beginning's end.

Lucius Annaeus Seneca

9 Conclusions

9.1 Summary

In contrast to conventional deterministic computational models, stochastic simulators are affected by their intrinsic stochasticity, and their response remains a random variable even for a fully specified set of input parameters. To assess the probabilistic properties of the model response, repeated runs with the same set of input parameters are therefore necessary. In the context of uncertainty quantification or optimization, various input values should also be investigated. Both aspects call for a large number of model runs that can be intractable when dealing with computationally expensive simulators. A common solution to this problem is to construct surrogate models that approximate the original simulator but can be evaluated at a low cost. Due to the random nature of stochastic simulators, however, well-established deterministic surrogate models cannot be applied directly. To fill this gap, this manuscript aims at developing efficient and accurate surrogate models to emulate stochastic simulators. More precisely, we have focus on estimating the entire response probability distribution over the input domain.

9.1.1 Literature review

Compared to deterministic simulators, surrogate modeling of stochastic simulators is a much less mature research field. Most of the methods developed in this area are replication-based approaches, where the simulator is repeatedly evaluated for each point of the experimental design (ED). The probability distribution of the response is characterized by a small set of statistics of the available replications. The estimated quantities are considered noisy observations of the underlying functions, which can in turn be estimated by regressions using conventional deterministic surrogate modeling techniques. As a result, the model construction typically consists of two steps: local inference and regression.

Another class of methods views the stochastic simulator as a random field indexed by the input variables. It requires controlling the intrinsic stochasticity to evaluate trajectories. By collecting multiple trajectories, one can estimate the covariance function and build up a surrogate random field, e.g., via Karhunen–Loève expansions.

As a result, methods of this category not only estimate the response distribution but also capture the dependence structure of the stochastic simulator as a random field. Because it is necessary to fix the intrinsic stochasticity, this type of approach is not always applicable, especially when experimental data are involved.

Most importantly, we pointed out in Chapter 3 that working with data containing unknown sources of randomness is a rather classical task in statistical learning. The response distribution is mathematically a conditional distribution. Many statistical models have been developed to estimate some important quantities of the conditional distribution, such as mean, variance, and quantiles, as functions of the input. For estimating the entire conditional distribution, however, conventional models either assume a very restrictive parametric distribution family or use nonparametric models that require a large number of model runs for the inference.

9.1.2 Generalized lambda models (GLaMs)

The first model that we proposed in the manuscript is GLaM. This stochastic surrogate model uses the four-parameter generalized lambda distribution (GLD) to represent the response distribution. GLD is a very flexible family that can accurately approximate many commonly used parametric distributions and cover a large range of unimodal distributions. Under the assumption that the model response to any input value follows a GLD, its four distribution parameters are functions of the input variables. We proposed using polynomial chaos expansion (PCE) to represent these four functions. Therefore, to determine a GLaM, one should select the basis functions and estimate the associated PCE coefficients.

To construct such a surrogate, we developed a replication-based approach in Chapter 4. In this chapter, we tested both the method of moments and the maximum likelihood estimation (MLE) for local inference. The sparse regression algorithm called hybrid least-angle regression (LAR) was used in the second step for basis selection and coefficients estimation. We pointed out the inefficiency of such a sequential optimization strategy: the data are separated into two steps but never considered altogether. Consequently, replications are crucial and limit the accuracy of the regression step. To improve the model performance, we suggested an additional joint step leveraging all the data. Numerical examples from various fields including wind turbine simulations confirmed the superiority of the new method over the two-step replication-based approach.

Based on the idea of this joint framework, we proposed in Chapter 5 the use of MLE to estimate the coefficients of GLaM without going through replications. We proved the consistency of the estimation method, which provides it with some theoretical justifications. To select an appropriate set of basis functions, we applied the feasible generalized least-squares (FGLS) with hybrid-LAR method to alternatively fit the mean and variance functions. The associated basis functions are used for the location and scale parameters of the GLD. For the two shape parameters, we chose low-degree polynomials, which are suitable for problems where the shape of the response distribution does not vary in a significant nonlinear way. Numerical benchmarks have shown that without replications, the developed method generally outperforms the best Gaussian approximations and one of the state-of-the-art nonparametric kernel estimators.

9.1.3 Global sensitivity analysis

In Chapter 6, we investigated global sensitivity analysis for stochastic simulators. We first provided a thorough review of three possible extensions of Sobol' indices to stochastic models. First, by including the intrinsic

stochasticity as part of the input, stochastic models are turned into deterministic, and thus classical Sobol' indices in Section 2.4.1 are well defined. Second, a certain quantity of interest (QoI) of the response distribution can serve as the main performance indicator in practice. As it is a deterministic function of the input, the associated Sobol' indices can be used for such a specific focus. Third, by treating the stochastic simulator as a random field, one can evaluate the Sobol' indices for each trajectory. These indices are random variables with their randomness resulting from the intrinsic stochasticity of the simulator. We analyzed the nature and mathematical interpretations of the three proposed extensions and offered guidelines for their usage. Furthermore, we applied GLaM to help efficiently estimate the Sobol' indices that are related to the statistical dependence between the model input and output. The case studies illustrated the wide applicability and accuracy of the surrogate model.

9.1.4 Stochastic polynomial chaos expansions (SPCEs)

Limited by the capacity of GLD, GLaM cannot represent stochastic simulators whose response can follow a multimodal distribution. To gain more flexibility, we extended the classical PCE to a stochastic emulator in Chapter 7. In this model, we introduced an artificial latent variable to jointly form a PCE together with the input variables. To regularize the surrogate response distribution and the estimation of the coefficients, we enriched the extended PCE with an additional additive Gaussian noise. To construct such a stochastic emulator, we proposed using MLE to estimate the coefficients (the consistency of this estimation method is discussed in Section A.2). We considered the standard deviation of the noise term as a hyperparameter and tune its value by cross-validation (CV). Moreover, we developed an adaptive algorithm to select the truncation scheme of the PCE and the type of latent variable based on hybrid-LAR and CV, without the need for replications. The numerical results demonstrated that the novel surrogate could compete with GLaM for estimating unimodal response distributions. Additionally, for multimodal distributions where GLaM fails, SPCE can still produce robust approximations and outperform nonparametric kernel estimators.

In Chapter 8, we applied SPCE to seismic fragility analysis in earthquake engineering. Following a newly developed framework for simulation-based fragility analysis, we selected a set of physically meaningful parameters of the ground motion model as intensity measures. Because of the white noise used for generating coherent stochastic ground motions, the engineering demand parameter (EDP) that is related to the structural damage remains random in response to a given set of ground motion parameters. To quantitatively represent this stochastic behavior, we employed SPCE to estimate the response distribution. The numerical results showed that SPCE outperforms conventional methods in terms of estimating both the response distribution and fragility function: on the one hand, SPCE is much more versatile than linear models relying on restrictive Gaussian assumptions; on the other hand, SPCE is more efficient than the fully nonparametric kernel model.

9.2 Limitations and outlook

As this thesis is an early investigation in the field of stochastic emulations, some aspects and ideas remain to be explored. In this section, we discuss the limitations of the proposed methods and provide several paths and challenges for future research.

9.2.1 Theoretical development

Because the probability density function (PDF) support of a GLD depends on the distribution parameters, the negative log-likelihood function of GLaM is not bounded (from below). This makes it hard to study the statistical properties of MLE. In Chapter 5, we employed techniques from the empirical process theory to show the consistency of MLE. However, this is under a strong condition: the underlying model is assumed to be correctly represented by a GLaM for a certain choice of coefficients. As a result, model misspecifications have not been considered from the theoretical perspective, even though numerical results have illustrated the applicability of the method to problems that are not exact GLaM. Hence, it remains to study the limiting model to which MLE of GLaM would converge with increasing sample size under model misspecifications.

For SPCE, we showed in Section A.2 that MLE consistently minimizes the Kullback–Leibler divergence despite model misspecifications. As SPCE does not assume a specific type of response distribution, its capacity is yet to be studied in theory: from the intuition of its formulation in Eq. (7.11), SPCE seems to have a certain property of universal approximation, which is worth further investigation.

The asymptotic behavior of the MLE for both GLaM and SPCE is not yet fully understood. For GLaM, the difficulty comes again from the fact that the support of the response distribution depends on the unknown parameters. For SPCE, the main challenge lies in its non-identifiability (see Section A.2). Note that the asymptotic properties are not only theoretical aspects but offer a practical way to quantify the uncertainty in the model construction. Similarly, bootstrap consistency is another important statistical property to look into for MLE, as it guarantees the application of bootstrapping.[1] For non-asymptotic behavior in the case of finite samples, one may explore the recent development of conformal prediction (Vovk et al., 2005; Lei et al., 2018; Romano et al., 2019; Chernozhukov et al., 2021; Sesia and Romano, 2021) to establish a good prediction interval of the model response for any given input values regardless of model misspecifications.

9.2.2 Active learning

The current methods apply a "one-shot" strategy: the ED is generated at once for a given simulation budget (either with replications or not), and the surrogate model is built by applying the estimation methods to the available data. Therefore, the data generation process is independent of the model construction. To make maximum use of the computational resources, it is necessary to bridge these two components.

Active learning is a group of methods in machine learning aiming at finding an optimal ED to adaptively build the model (Settles, 2009). It consists in enriching sequentially the ED by querying the model response on new points selected based on optimizing a suitable learning function to efficiently improve the surrogate performance. For deterministic simulators, active learning has been widely applied in the field of reliability analysis (Echard et al., 2011; Marelli and Sudret, 2018; Wagner et al., 2022; see Moustapha et al., 2022 for a detailed review). For stochastic simulators, methods have been mainly developed for improving the mean function estimation using Gaussian processes (Binois et al., 2019). For the developed models, which estimate the entire response distribution, it is crucial to find an appropriate learning function to guide the sequential design of experiments. Besides, for seismic fragility analysis, engineers are mainly interested in the fragility function

[1] Typically, local asymptotic linearity seems to be necessary for consistency of bootstrap (Mammen, 1992), which also calls for investigating the asymptotic behavior of the estimators.

rather than in the response distribution. Therefore, designing a good learning function to improve the failure probability estimation from stochastic surrogate models would be particularly beneficial to applications in earthquake engineering.

9.2.3 High-dimensional problems

To select appropriate basis functions for GLaM, we apply the feasible generalized least-squares method with sparse regression methods for the location and scale parameters, and we prescribe low-degree polynomials for the two shape parameters. Such a choice relies on the empirical observation that the shape of the response distribution does typically not have a significant variation in practical applications. In this case, the location and scale parameters are closely related to the response mean and variance functions, respectively. This setting would be inappropriate when the shape of the response distribution varies in a strongly nonlinear way over the input domain. In addition, for high-dimensional problems, the sparse regression algorithms may select too many irrelevant basis functions, and even low-degree polynomials can contain a lot of unknown terms.

For SPCE, we use sparse regression methods to select the part of the basis functions that is related to the response mean function. The developed adaptive algorithm does not work on the level of individual basis functions but rather looks for a suitable truncation scheme. Similar to GLaM, a general sparse structure would be desirable for dealing with high-dimensional problems.

As a result, it is necessary to handle sparsity in a more systematic way for both GLaM and SPCE. The first option would be stepwise regression (Jennrich and Sampson, 1968). This approach consists of two principal steps: forward selection and backward elimination. The first step starts with only the constant function and progressively enriches the selected basis functions one by one according to a certain model selection criterion. The second step eliminates one by one the basis functions that are less relevant. To develop algorithms of this type for GLaM and SPCE, it is essential to design an efficient strategy to decide at each iteration step which basis function should be added or eliminated, as well as a model selection criterion to decide whether the proposed operation should be accepted.

A popular approach for tackling high-dimensional problems is to introduce a penalty function to the loss function, that is,

$$\hat{c}_N = \arg\min_{\boldsymbol{c}} L_N(\boldsymbol{c}) + \mathrm{pen}_{\boldsymbol{\theta}}(\boldsymbol{c}), \tag{9.1}$$

where $\mathrm{pen}_{\boldsymbol{\theta}}$ is a penalty function with hyperparameters $\boldsymbol{\theta}$. Multiple penalty functions have been proposed in statistical learning and can be potentially used to improve the developed models, such as ridge regularization (Hoerl and Kennard, 1970), least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), adaptive LASSO (Zou, 2006), elastic-net (Zou and Hastie, 2005), or smoothly clipped absolute deviation (Fan and Li, 2001). In this respect, suitable optimization algorithms like Zou and Li (2008) and Fan et al. (2018) remain to be explored to cope with the highly nonlinear likelihood function and possibly complex penalty function.

9.2.4 SPCE with a flexible latent variable

For SPCE developed in Chapter 7, the type of the latent variable is a hyperparameter, and we chose between uniform and normal distributions in that chapter. To automatically tune the latent variable and further enrich the capacity of SPCE under finite samples, we can choose a flexible probability distribution for the latent vari-

able, such as beta distribution or GLD. With this, the SPCE is expressed as

$$\tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\theta},\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z_{\boldsymbol{\theta}}\right) + \epsilon, \tag{9.2}$$

where $Z_{\boldsymbol{\theta}}$ denotes the latent variable with the distribution parameters $\boldsymbol{\theta}$. The latter can be estimated jointly with the coefficients of the SPCE, and thus no more CV is needed at this level. However, the orthonormal basis associated with $Z_{\boldsymbol{\theta}}$ depends also on $\boldsymbol{\theta}$. As a result, the gradient of the likelihood function with respect to $\boldsymbol{\theta}$ is yet to be investigated, and an appropriate method should be developed to solve the associated optimization problem.

9.2.5 Other loss functions

In this work, the negative log-likelihood function has been chosen as the loss function, and it corresponds to the Kullback–Leibler divergence between the surrogate model and the simulator. The reason for this choice is that it does not require replications in the estimation and the formulation of the associated optimization problem is straightforward. However, the loss function is non-convex and the optimization can be difficult to solve, as it is the case for GLaM in Chapter 4 and for tuning σ of SPCE in Chapter 7. Alternatively, one can use the integrated mean-squared error loss in Eq. (3.34) that is widely applied to nonparametric kernel estimators. Moreover, as we used the Wasserstein distance to assess the model performance, methods related to this error metric (Arjovsky et al., 2017; Genevay et al., 2018) may be more relevant.

For seismic fragility analysis, the fragility function is the most important quantity to estimate. Minimizing an error metric of the whole distribution is less sensitive to the exceeding probability of a specific damage-related threshold, which is usually at the tail of the distribution. Consequently, it might be unsuitable for this study. To this end, one can explore other kinds of loss functions or combine metrics from classification (e.g., area under the receiver operating characteristic curve [Fawcett, 2006]) to better estimate the fragility function.

9.2.6 Extension to vector-valued response

This thesis has focused on surrogate modeling of stochastic simulators with a scalar output. The developed methods are still applicable to computational models with multiple output variables if the statistical dependence of the response variables is not of interest or if they are conditionally independent given the input variables. In this case, we can emulate individually the response distribution of each component of the output vector. However, if the statistical dependence among the response variables is unknown and should be correctly represented (e.g., fragility analysis with multiple critical components [Nielson and DesRoches, 2007]), it is necessary to estimate the response joint distribution.

A first extension is to use the proposed method to represent the response distribution of each component of the response variables and to model separately their dependence. The latter can be typically tackled by copulas (Torre et al., 2019b). Within this perimeter, one needs to select and fit an appropriate copula. If a parametric copula is used, the copula parameters are generally functions of the input variables and need to be estimated from data (in the simplest case, one can assume that the copula parameters are constant).

A second option is to introduce multiple latent variables $\boldsymbol{Z}$ to SPCE. Hence, each output component is expressed as

$$Y_j \overset{\mathrm{d}}{\approx} \tilde{Y}_j = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} c_{j,\boldsymbol{\alpha}} \psi_{\boldsymbol{\theta},\boldsymbol{\alpha}} (\boldsymbol{x}, \boldsymbol{Z}) + \epsilon_j, \tag{9.3}$$

where the same set of latent variables is consistently used to model inherently the statistical dependence. Therefore, the joint response PDF is given by

$$f_{\tilde{\boldsymbol{Y}}|\boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{x}) = \int_{\mathcal{D}_{\boldsymbol{Z}}} \prod_j \frac{1}{\sigma_j} \varphi \left(\frac{y_j - \sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} c_{j,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{z})}{\sigma_j} \right) f_{\boldsymbol{Z}}(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}, \tag{9.4}$$

where φ is the PDF of the standard normal distribution. The evaluation of the joint PDF involves a multi-dimensional integral, which can become intractable for high-dimensional $\boldsymbol{Z}$. As Eq. (9.3) is very similar to conditional generative models reviewed in Section 3.1.5.4, techniques developed in this field can be helpful for both modeling and fitting.

A third way is to sequentially construct the surrogate model by treating iteratively a part of the output variables as input. More precisely, we can factorize the joint response PDF by

$$f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_j f_{Y_j|\boldsymbol{X},Y_1,\ldots,Y_{j-1}} (y_j \mid \boldsymbol{x}, y_1, \ldots, y_{j-1}) \,. \tag{9.5}$$

Therefore, we can first build a surrogate model for Y_1, i.e., $f_{Y_1|\boldsymbol{X}}(y_1 \mid \boldsymbol{x})$. Then, we combine Y_1 with $\boldsymbol{X}$ and build a surrogate model for Y_2, i.e., $f_{Y_2|\boldsymbol{X},Y_1}(y_2 \mid \boldsymbol{x}, y_1)$. We repeat sequentially this procedure for each component, and the surrogate model of the last component requires combining all the other response variables with $\boldsymbol{X}$. As this framework allows for reusing all the existing methods, approaches and possible improvements in this direction are worth further investigation.

9.3 Final conclusion

In the present manuscript, we have successfully extended the classical deterministic surrogate modeling tools to the field of stochastic simulators. More precisely, we have proposed two surrogate models, namely GLaM and SPCE, to emulate the entire response distribution of stochastic simulators. To construct them, we have developed adaptive methods that are applicable to data without replications. We demonstrated their effectiveness and efficiency on multiple examples and engineering case studies, where the proposed surrogate models exhibit greater performance compared to state-of-the-art models. All in all, the novel approaches have a lot of potential for real-world applications, and they will hopefully foster ideas for future research.

Appendices

A

Complementary discussions

In this chapter, we complement the main findings and developments by discussing some related topics. In Section A.1, we look into the role of replications in building surrogate models. In Section A.2, we show the consistency of maximum likelihood estimation (MLE) for stochastic polynomial chaos expansion (SPCE).

A.1 Replicate or not?

Most of the methods developed in the thesis aim at being independent of replications or tend to avoid them. In contrast, replications lie at the core of replication-based approaches (Ankenman et al., 2010; Plumlee and Tuo, 2014; Moutoussamy et al., 2015). Besides, some methods also explore replications to achieve a better efficiency (Binois et al., 2018, 2019). To look into the question of whether to replicate, we discuss in this section the role of replications for M-estimators (presented in Section 3.1), based on which the estimation methods in this manuscript are developed.

Let us consider a random experimental design (ED) $\mathcal{X}$ following the joint distribution of $\boldsymbol{X}$. For each point in $\mathcal{X}$, we repeatedly and independently evaluate the simulator R times. As a result, for a total budget of N model runs, the ED size is $|\mathcal{X}| = N/R$.

Following the discussions in Section 3.1.1, we choose an appropriate loss function ℓ based on the statistical assumption and the estimation target. The associated M-estimator is given by Chapters 4 and 5, that is,

$$L_{N,R}(\boldsymbol{c}) = \frac{1}{N} \sum_{i=1}^{N/R} \sum_{r=1}^{R} \ell\left(\boldsymbol{c}; \boldsymbol{X}^{(i)}, Y^{(i,r)}\right) \tag{A.1}$$

where we treat the data as random variables to study the statistical properties of the estimator. Here, $\boldsymbol{X}^{(i)}$ denotes the i-th point in $\mathcal{X}$, and $Y^{(i,r)}$ is the model response of the r-th replicated run for $\boldsymbol{X}^{(i)}$. By taking the expectation, we obtain

$$\mathbb{E}\left[L_{N,R}(\boldsymbol{c})\right] = \mathbb{E}\left[\ell\left(\boldsymbol{c}; \boldsymbol{X}^{(i)}, Y^{(i,r)}\right)\right] = \mathbb{E}\left[\ell\left(\boldsymbol{c}; \boldsymbol{X}, Y\right)\right] = L(\boldsymbol{c}). \tag{A.2}$$

According to the law of large numbers, $\mathbb{E}\left[L_{N,R}(\boldsymbol{c})\right]$ converges in expectation to $L(\boldsymbol{c})$ with $N/R \to +\infty$, independent of the size of R. This complies with the main idea of M-estimators in Section 3.1.2. The variance of Eq. (A.1) is calculated by

$$\begin{aligned}\operatorname{Var}\left[L_{N,R}(\boldsymbol{c})\right] =& \frac{1}{N^2}\sum_{i=1}^{N/R}\operatorname{Var}\left[\sum_{r=1}^{R}\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r)}\right)\right] \\ &+\frac{1}{N^2}\sum_{i_1\neq i_2}\operatorname{Cov}\left[\sum_{r=1}^{R}\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i_1)},Y^{(i_1,r)}\right),\sum_{r=1}^{R}\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i_2)},Y^{(i_2,r)}\right)\right].\end{aligned} \tag{A.3}$$

Because the samples of the ED are independently drawn, the covariance across different input samples is zero. As a result, Eq. (A.3) becomes

$$\begin{aligned}&\operatorname{Var}\left[L_{N,R}(\boldsymbol{c})\right] = \frac{1}{N^2}\sum_{i=1}^{N/R}\operatorname{Var}\left[\sum_{r=1}^{R}\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r)}\right)\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N/R}\left(\sum_{r=1}^{R}\operatorname{Var}\left[\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r)}\right)\right]+\sum_{r_1\neq r_2}\operatorname{Cov}\left[\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r_1)}\right),\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r_2)}\right)\right]\right).\end{aligned} \tag{A.4}$$

Under the independent sampling scheme, the variance terms in Eq. (A.4) can be simplified to

$$\operatorname{Var}\left[\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r)}\right)\right] = \operatorname{Var}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\right]. \tag{A.5}$$

Unlike Eq. (A.3), the covariance terms are not 0, as the same $\boldsymbol{X}$ is used in the loss function and for evaluating the stochastic simulator. Resorting to the same techniques used in the derivations of Eq. (2.80), we obtain

$$\operatorname{Cov}\left[\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r_1)}\right),\ell\left(\boldsymbol{c};\boldsymbol{X}^{(i)},Y^{(i,r_2)}\right)\right] = \operatorname{Var}\left[\mathbb{E}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\middle|\boldsymbol{X}\right]\right]. \tag{A.6}$$

By injecting Eq. (A.5) and Eq. (A.6) to Eq. (A.4), the variance of the empirical loss function is

$$\begin{aligned}\operatorname{Var}\left[L_{N,R}(\boldsymbol{c})\right] &= \frac{1}{N^2}\sum_{i=1}^{N/R}\left(R\operatorname{Var}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\right]+R(R-1)\operatorname{Var}\left[\mathbb{E}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\middle|\boldsymbol{X}\right]\right]\right) \\ &= \frac{1}{N}\operatorname{Var}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\right]+\frac{R-1}{N}\operatorname{Var}\left[\mathbb{E}\left[\ell\left(\boldsymbol{c};\boldsymbol{X},Y\right)\middle|\boldsymbol{X}\right]\right].\end{aligned} \tag{A.7}$$

As the empirical loss Eq. (A.1) is an unbiased and consistent estimator of $L(\boldsymbol{c})$, it is desirable to have a variance as small as possible, so that it converges as fast as possible to $L(\boldsymbol{c})$. To this end, $R = 1$, i.e., no replication, should be chosen to minimize the variance according to Eq. (A.7). Because both models developed in this thesis are constructed with MLE, which is an M-estimator, it is preferred not to have replications. Rigorously speaking, one should compute the variance of the M-estimator (assuming that it is consistent). However, the asymptotic properties of the M-estimators for the proposed models in this thesis are yet to be investigated. Hence, we cannot draw a definitive conclusion in theory but use Eq. (A.7) to shed light on this aspect, as the loss function plays a central role in M-estimators for both estimations and validations. Nevertheless, for the loss function ℓ being "regular" (e.g., van der Vaart, 1998, Section 5.3), it can be easily shown that the asymptotic

variance of the associated M-estimator $\hat{\boldsymbol{c}}_{N,R}$ is

$$\begin{aligned}\operatorname{Var}\left[\hat{\boldsymbol{c}}_{N,R}\right] = &\frac{1}{N}\mathbb{E}\left[\nabla^2\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right]^{-1}\mathbb{E}\left[\nabla\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\nabla\left(\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right)^{\intercal}\right]\mathbb{E}\left[\nabla^2\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right]^{-1}\\ &+\frac{R-1}{N}\mathbb{E}\left[\nabla^2\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right]^{-1}\mathbb{E}\left[\mathbb{E}\left[\nabla\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\mid\boldsymbol{X}\right]\mathbb{E}\left[\left(\nabla\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right)^{\intercal}\mid\boldsymbol{X}\right]\right]\mathbb{E}\left[\nabla^2\ell(\boldsymbol{c}_0;\boldsymbol{X},Y)\right]^{-1}.\end{aligned} \tag{A.8}$$

where $\boldsymbol{c}_0$ is the optimal set of coefficients that minimize the expected loss $L(\boldsymbol{c})$. Therefore, using no replication would result in a more efficient estimator. In addition, this aspect is illustrated empirically by the numerical results in Section 5.5.3.

The discussions above are limited to the strategy of performing the same number of replications for each point of the ED and to M-estimators with a joint loss function as in Eq. (A.1). In some cases, however, replications can be important and helpful. First, replication-based methods reviewed in Section 3.2 rely on information extracted from replications. The conventional two-step sequential fitting procedure calls for replications to ensure an accurate local inference (Moutoussamy et al., 2015). Second, for Gaussian process models, relatively big training samples would make the model construction and prediction very time-consuming (Snelson and Ghahramani, 2005; Binois et al., 2018). As a result, replications can efficiently reduce the model complexity from using all the data to only focusing on the quantities extracted from the replications (Plumlee and Tuo, 2014). Third, some methods explore sequential designs of experiments that actively select new points to evaluate the simulator (Binois et al., 2019). In this case, a new point can be an existing point, and thus the associated model evaluation is a replication. Last but not least, replications can help validate the model, which offers a straightforward way to check the accuracy of the prediction.

A.2 Consistency of MLE for SPCE

In Chapter 5, we proved the consistency of MLE for generalized lambda model (GLaM). In this section, we study this property of MLE for SPCE. Since MLE is only applied to estimate the coefficients, we assume here a given set basis functions $\mathcal{A}$ and fixed variance of the noise variable σ.

In general, a SPCE is not "identifiable", meaning that two different sets of the coefficients can produce the same response distribution. For instance, for Z following a symmetric probability distribution centered at 0 (e.g., $Z\sim\mathcal{U}(-1,1)$ and $Z\sim\mathcal{N}(0,1)$ as used in Chapter 7), $-Z$ has the same distribution as Z, which leads to

$$\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x},Z\right)+\epsilon \overset{\mathrm{d}}{=} \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x},-Z\right)+\epsilon = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{c}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x},Z\right)+\epsilon. \tag{A.9}$$

As a result, the solution to the following optimization problem is not unique:

$$\boldsymbol{c}_0 = \arg\min_{\boldsymbol{c}\in\mathcal{C}} L(\boldsymbol{c}) \tag{A.10}$$

where $L(\boldsymbol{c}) = \mathbb{E}\left[\ell(\boldsymbol{c};\boldsymbol{X},Y)\right]$, and ℓ is the negative log-likelihood function. Consequently, the consistency of the estimation of the coefficients does not hold (the limiting coefficients are not unique). Nonetheless, we

investigate in the following the consistency in terms of the loss function, that is,

$$L(\hat{\boldsymbol{c}}_N) - L(\boldsymbol{c}_0) \xrightarrow{\text{a.s.}} 0, \tag{A.11}$$

where $\boldsymbol{c}_0$ is one of the minimizers solving Eq. (A.10). Eq. (A.11) indicates that the expected loss function is minimized with $N \to +\infty$, and thus the fitted model converges to the "projection" (in terms of the loss function) of the true model onto the set of response distributions defined by SPCE.

Lemma A.1. *If the parameter space $\mathcal{C}$ is a compact subset of $\mathbb{R}^{|\mathcal{A}|}$, and Y has a finite variance, the negative log-likelihood function parameterized by $\boldsymbol{c} \in \mathcal{C}$ forms a Glivenko–Cantelli class, that is,*

$$\sup_{\boldsymbol{c}\in\mathcal{C}} |L_N(\boldsymbol{c}) - L(\boldsymbol{c})| \xrightarrow{a.s.} 0, \tag{A.12}$$

where $L_N(\boldsymbol{c}) \overset{def}{=} \frac{1}{N}\sum_{i=1}^{N} \ell\left(\boldsymbol{c}; \boldsymbol{X}^{(i)}, Y^{(i)}\right)$ and $L(\boldsymbol{c}) \overset{def}{=} \mathbb{E}\left[\ell\left(\boldsymbol{c}; \boldsymbol{X}, Y\right)\right]$.

Proof. To prove this lemma, we rely on Newey and McFadden (1994, Lemma 2.4), where it is only necessary to show that the negative log-likelihood function $\ell(\boldsymbol{c}; \boldsymbol{x}, y)$ has the following properties:

1. $\ell(\boldsymbol{c}; \boldsymbol{x}, y)$ is continuous at each $\boldsymbol{c} \in \mathcal{C}$ for any $(\boldsymbol{x}, y) \in \mathcal{D}_{\boldsymbol{X}} \times \mathbb{R}$.
2. There is a function $G : \mathcal{D}_{\boldsymbol{X}} \times \mathbb{R} \to \mathbb{R}$ such that $|\ell(\boldsymbol{c}; \boldsymbol{x}, y)| \leqslant G(\boldsymbol{x}, y)$ for any $\boldsymbol{c} \in \mathcal{C}$ and $\mathbb{E}\left[G(\boldsymbol{X}, Y)\right] < +\infty$.

In the first step, we prove the continuity. Recall the negative log-likelihood function of SPCE

$$\begin{aligned} \ell(\boldsymbol{c}; \boldsymbol{x}, y) &= -\log\left(\mathbb{E}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2}{2\sigma^2}\right)\right]\right) \\ &= -\log\left(\mathbb{E}\left[\int_{\mathcal{D}_Z} h(Z, \boldsymbol{c}; \boldsymbol{x}, y)\right]\right), \end{aligned} \tag{A.13}$$

where h can be also expressed as $h(z, \boldsymbol{c}; \boldsymbol{x}, y) = \varphi\left(\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)\right)/\sigma\right)/\sigma$ with φ being the probability density function (PDF) of the standard normal distribution.

Because φ is continuous, and the basis functions $\psi_{\boldsymbol{\alpha}}$ are polynomials and thus continuous. Therefore, for any given $(\boldsymbol{x}, y) \in \mathcal{D}_{\boldsymbol{X}} \times \mathbb{R}$, $h(\cdot, \cdot; \boldsymbol{x}, y)$ is a continuous function in both $\boldsymbol{c}$ and z. Taking any sequence of coefficients $(\boldsymbol{c}_n)_{n\in\mathbb{N}}$ that converges to $\boldsymbol{c}$, i.e., $\lim_{n\to+\infty} \boldsymbol{c}_n = \boldsymbol{c}$, we define a sequence of functions $(h_n)_{n\in\mathbb{N}}$ by $h_n(z) = h(z, \boldsymbol{c}_n; \boldsymbol{x}, y)$. Because of the continuity of h in $\boldsymbol{c}$, h_n converges pointwise to $h(\cdot, \boldsymbol{c}; \boldsymbol{x}, y)$. Following its definition, h is positive and it is bounded by

$$h(z, \boldsymbol{c}; \boldsymbol{x}, y) \leqslant \frac{1}{\sigma\sqrt{2\pi}}. \tag{A.14}$$

The same properties hold for the sequence of functions $(h_n)_{n\in\mathbb{N}}$. It is clear that the constant upper bound is integrable with respect to the PDF f_Z. Applying the dominated convergence theorem gives

$$\lim_{n\to+\infty} \mathbb{E}\left[h_n(Z)\right] = \mathbb{E}\left[h(Z, \boldsymbol{c}; \boldsymbol{x}, y)\right]. \tag{A.15}$$

This means that the mapping $\boldsymbol{c} \mapsto \mathbb{E}\left[h(z, \boldsymbol{c}; \boldsymbol{x}, y)\right]$ is continuous. By composing this function with $-\log$, which is a continuous function, we prove that $\ell(\boldsymbol{c}; \boldsymbol{x}, y)$ is continuous at each $\boldsymbol{c} \in \mathcal{C}$ for any $(\boldsymbol{x}, y) \in \mathcal{D}_{\boldsymbol{X}} \times \mathbb{R}$.

In the second step, we will show the existence of the envelope function G with a finite expectation.

According to Eq. (A.14), we obtain

$$\ell(\boldsymbol{c}; \boldsymbol{x}, y) \geqslant -\log\left(\mathbb{E}\left[\frac{1}{\sqrt{2\pi}\sigma}\right]\right) = \log\left(\sqrt{2\pi}\sigma\right) \tag{A.16}$$

which provides a lower bound of $\ell(\boldsymbol{c}; \boldsymbol{x}, y)$.

For the upper bound, we apply Jensen's inequality to the negative log-likelihood function, which gives

$$-\log\left(\mathbb{E}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y-\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2}{2\sigma^2}\right)\right]\right) \leqslant \log\left(\sqrt{2\pi}\sigma\right) + \mathbb{E}\left[\frac{\left(y-\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2}{2\sigma^2}\right]. \tag{A.17}$$

For ease of notation, we denote the main part of the expectation term on the right-hand side as

$$\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) \overset{\text{def}}{=} \mathbb{E}\left[\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2\right] \geqslant 0 \tag{A.18}$$

Summarizing the results in Eq. (A.14) and Eq. (A.17) gives

$$\log\left(\sqrt{2\pi}\sigma\right) \leqslant \ell(\boldsymbol{c}; \boldsymbol{x}, y) \leqslant \frac{1}{2\sigma^2}\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) + \log(\sqrt{2\pi}\sigma). \tag{A.19}$$

By taking the absolute value, we obtain

$$\begin{aligned} |\ell(\boldsymbol{c}; \boldsymbol{x}, y)| &\leqslant \max\left(\left|\frac{1}{2\sigma^2}\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) + \log\left(\sqrt{2\pi}\sigma\right)\right|, \left|\log\left(\sqrt{2\pi}\sigma\right)\right|\right) \\ &\leqslant \left|\frac{1}{2\sigma^2}\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) + \log\left(\sqrt{2\pi}\sigma\right)\right| + \left|\log\left(\sqrt{2\pi}\sigma\right)\right| \\ &\leqslant \frac{1}{2\sigma^2}\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) + 2\left|\log\left(\sqrt{2\pi}\sigma\right)\right| \end{aligned} \tag{A.20}$$

To find an envelope function for $|\ell(\boldsymbol{c}; \boldsymbol{x}, y)|$, it is sufficient to find one for $\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y)$, which is investigated in the following derivations.

By separating the polynomials basis functions that are not related to $\boldsymbol{Z}$ in the expansion, $\tilde{G}$ becomes

$$\begin{aligned} \tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) =& \mathbb{E}\left[\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) - \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2\right] \\ =& \left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)^2 - 2\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)\mathbb{E}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right] \\ &+ \mathbb{E}\left[\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z)\right)^2\right], \end{aligned} \tag{A.21}$$

where $\mathcal{A}_m \overset{\text{def}}{=} \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_z = 0\}$ as defined in Eq. (7.27). Using the orthonormality of the polynomial chaos expansion (PCE) basis functions, Eq. (A.21) becomes

$$\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) = \left(y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \right)^2 + \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}), \tag{A.22}$$

where the basis function $\psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x})$ for $\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m$ is defined by only considering the components of $\boldsymbol{x}$ while ignoring those of $\boldsymbol{z}$.

Applying Jensen's inequality to the first part of Eq. (A.22) gives

$$\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) \leqslant (1 + |\mathcal{A}_m|) \left(y^2 + \left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}) \right) \right) + \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}). \tag{A.23}$$

Because $\mathcal{C}$ is a compact subset of $\mathbb{R}^{|\mathcal{A}|}$, it is bounded and closed (according to the Heine–Borel theorem). Define C the minimum radius of a ball centered at $\mathbf{0}$ that covers $\mathcal{C}$, and thus $|c_\alpha| \leqslant C$ for all $\alpha \in \mathcal{A}$. Therefore, $\tilde{G}$ is bounded by

$$\tilde{G}(\boldsymbol{c}; \boldsymbol{x}, y) \leqslant \bar{G}(\boldsymbol{x}, y) \overset{\text{def}}{=} (1 + |\mathcal{A}_m|) \left(y^2 + \left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} C^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}) \right) \right) + \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m} C^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}). \tag{A.24}$$

Taking the expectation of $\bar{G}$ with respect to the joint distribution of $(\boldsymbol{X}, Y)$ provides

$$\begin{aligned} \mathbb{E}\left[\bar{G}(\boldsymbol{X}, Y)\right] &= \mathbb{E}\left[(1 + |\mathcal{A}_m|) \left(Y^2 + \left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_m} C^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{X}) \right) \right) + \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m} C^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{X}) \right] \\ &= (1 + |\mathcal{A}_m|) \left(\mathbb{E}\left[Y^2\right] + |\mathcal{A}_m| C^2 \right) + (|\mathcal{A}| - |\mathcal{A}_m|) C^2 \\ &= (1 + |\mathcal{A}_m|) \mathbb{E}\left[Y^2\right] + \left(|\mathcal{A}| + |\mathcal{A}_m|^2 \right) C^2. \end{aligned} \tag{A.25}$$

By combining Eq. (A.20) with Eq. (A.24), we find a function independent of $\boldsymbol{c}$ that bounds ℓ:

$$|\ell(\boldsymbol{c}; \boldsymbol{x}, y)| \leqslant G(\boldsymbol{x}, y) \overset{\text{def}}{=} \frac{1}{2\sigma^2} \bar{G}(\boldsymbol{x}, y) + 2 \left| \log\left(\sqrt{2\pi}\sigma \right) \right|. \tag{A.26}$$

Based on Eq. (A.25), $\mathbb{E}\left[G(\boldsymbol{X}, Y)\right]$ is calculated by

$$\mathbb{E}\left[G(\boldsymbol{X}, Y)\right] = \frac{1}{2\sigma^2} \left((1 + |\mathcal{A}_m|) \mathbb{E}\left[Y^2\right] + \left(|\mathcal{A}| + |\mathcal{A}_m|^2 \right) C^2 \right) + 2 \left| \log\left(\sqrt{2\pi}\sigma \right) \right|. \tag{A.27}$$

Since Y has a finite variance, $\mathbb{E}\left[G(\boldsymbol{X}, Y)\right] < +\infty$.

Finally, as $\mathcal{C}$ is compact and ℓ is continuous and has an envelope function with a finite expectation, applying Newey and McFadden (1994, Lemma 2.4) proves this lemma. □

Theorem A.1. *If the parameter space $\mathcal{C}$ is compact and Y has a finite variance, MLE consistently minimizes the expected risk, that is,*

$$L(\hat{\boldsymbol{c}}_N) - L(\boldsymbol{c}_0) \xrightarrow{a.s.} 0. \tag{A.28}$$

Proof. Because $\boldsymbol{c}_0$ minimizes the expected loss, we have

$$0 \leqslant L(\hat{\boldsymbol{c}}_N) - L(\boldsymbol{c}_0) = L(\hat{\boldsymbol{c}}_N) - L_N(\hat{\boldsymbol{c}}_N) + L_N(\hat{\boldsymbol{c}}_N) - L_N(\boldsymbol{c}_0) + L_N(\boldsymbol{c}_0) - L(\boldsymbol{c}_0). \tag{A.29}$$

Since $\hat{\boldsymbol{c}}_N$ minimizes the empirical loss $L_N(\boldsymbol{c})$, $L_N(\hat{\boldsymbol{c}}_N) - L_N(\boldsymbol{c}_0) \leqslant 0$, and thus Eq. (A.29) becomes

$$0 \leqslant L(\hat{\boldsymbol{c}}_N) - L(\boldsymbol{c}_0) \leqslant L(\hat{\boldsymbol{c}}_N) - L_N(\hat{\boldsymbol{c}}_N) + L_N(\boldsymbol{c}_0) - L(\boldsymbol{c}_0). \tag{A.30}$$

Applying Lemma A.1 shows $L(\hat{\boldsymbol{c}}_N) - L_N(\hat{\boldsymbol{c}}_N) \xrightarrow{p} 0$ and $L_N(\boldsymbol{c}_0) - L(\boldsymbol{c}_0) \xrightarrow{p} 0$, which implies $L(\hat{\boldsymbol{c}}_N) - L(\boldsymbol{c}_0) \xrightarrow{p} 0$. Finally, applying Talagrand (1987, Theorem 22) extends the convergence in probability to almost sure convergence. □

The proof of the consistency of MLE for SPCE in this section requires more loose conditions than that for GLaM where the underlying model is assumed to be a GLaM. Even though the stochastic simulator is not a SPCE for a given set of basis functions, the estimator still converges to a model that minimizes the expected loss function. This is because the support of the response distribution of SPCE is $\mathbb{R}$ and independent of the unknown parameters, which allows for applying classical techniques to the proof. However, due to the non-identifiability of the response distribution of SPCE, the estimator is not consistent in terms of coefficients but minimizes consistently the expected loss — the Kullback–Leibler divergence between the stochastic simulator and the emulator.

B

A global sensitivity analysis framework for hybrid simulation with stochastic substructures

This chapter is a post-print of

> Tsokanas, N., Zhu, X., Abbiati, G., Marelli, S., Sudret, B., and Stojadinović, B. (2021). A global sensitivity analysis framework for hybrid simulation with stochastic substructures. *Frontiers in Built Environment*, 7:1–12. DOI:10.3389/fbuil.2021.778716.

differing from the published paper only in terms of layout and formatting.

Author contributions. **N. Tsokanas**: Conceptualization, Methodology, Software, Validation, Writing - Original Draft. **X. Zhu**: Conceptualization, Software, Writing - Original Draft. **G. Abbiati**: Conceptualization, Methodology, Software, Writing - Review & Editing. **S. Marelli**: Conceptualization, Writing - Review & Editing. **B. Sudret**: Supervision, Writing - Review & Editing. **B. Stojadinović**: Supervision, Funding acquisition, Writing - Review & Editing.

Abstract

Hybrid simulation is an experimental method used to investigate the dynamic response of a reference prototype structure by decomposing it to physically-tested and numerically-simulated substructures. The latter substructures interact with each other in a real-time feedback loop and their coupling forms the hybrid model. In this study, we extend our previous work on metamodel-based sensitivity analysis of deterministic hybrid models to the practically more relevant case of stochastic hybrid models. The aim is to cover a more realistic situation where the physical substructure response is not deterministic, as nominally identical specimens are, in practice, never actually identical. A generalized lambda surrogate model recently developed by some of the authors is proposed to surrogate the hybrid model response, and Sobol' sensitivity indices are computed for substructure

quantity of interest response quantiles. Normally, several repetitions of every single sample of the inputs parameters would be required to replicate the response of a stochastic hybrid model. In this regard, a great advantage of the proposed framework is that the generalized lambda surrogate model does not require repeated evaluations of the same sample. The effectiveness of the proposed hybrid simulation global sensitivity analysis framework is demonstrated using an experiment.

B.1 Introduction

Hybrid simulation (HS) is used to investigate the experimental dynamic response of a structural component or sub-assembly subjected to a realistic loading scenario, which includes the dynamic interaction with a credible yet virtual structural system. Coupled physical and numerical substructures (PS and NS) form the so-called hybrid model. Specifically, the PS is tested using servo-controlled actuators provided with force transducers, while the NS is instantiated in a structural analysis software. A time-stepping analysis algorithm computes the hybrid model response *on-the-fly*. This ensures displacement compatibility and force balance between NS and PS throughout the entire experiment. Additionally, HS is used to investigate the inner workings of a structural component beyond the linear regime without testing an entire structural assembly. As a result, the cost of experimentation is substantially reduced. A report by Schellenberg and co-workers provides a comprehensive review of HS methods and algorithms (Schellenberg et al., 2009).

In earthquake engineering, HS is the only viable solution for testing large structures (e.g., Abbiati et al., 2019; Moustafa and Mosalam, 2015; Bas et al., 2022). Similarly, HS has been recently proposed to test mooring lines of offshore structures for which hydrodynamic tests with sizeable scale are prohibitive (e.g., Sauder et al., 2018; Vilsen et al., 2019). HS is gaining popularity for component-level testing in fire engineering. The reason is that internal force redistribution occurring at the system level heavily influences the failure modes at the component level (e.g., Abbiati et al., 2020). More recently, HS has been combined with centrifuge testing for investigating soil-structure interaction problems (Idinyang et al., 2019).

In all the cases reviewed above, NS and related excitation are conceived as deterministic. However, in the majority of cases encountered in structural engineering, loading is stochastic, while the boundary conditions are highly uncertain. An exhaustive exploration of all possible load cases is clearly not an option given the experimental cost associated with a single hybrid model evaluation. Accordingly, Abbiati et al. (2021) proposed surrogate modeling to compute the variance-based global sensitivity analysis (GSA) of the response quantity of interest (QoI) of a given hybrid model with respect to a set of *input parameters* that characterize both substructures and loading excitations. In detail, polynomial chaos expansion (PCE) was used to construct a surrogate model (a.k.a. response surface) of the hybrid model response. Sobol' sensitivity indices of the QoIs were obtained as a by-product of polynomial coefficients as explained in Sudret (2008). The goal of such an approach was to reveal *what influences what* within the HS: This entails uncovering the inner workings of the PS, the unknown part of the hybrid model in both epistemic and aleatory sense. In that study, the PS was treated as *deterministic*, that is, aleatory uncertainty was neglected by assuming that nominally identical specimens have identical responses (plus some negligible measurement noise).

This assumption, however, is still far from a realistic scenario in structural testing. Nominally identical specimens are, in practice, never actually identical. Also, some sources of loading exerted through the PS are

inherently stochastic (e.g., fire or hydrodynamic loading). As a result, uncertainty, both aleatory or epistemic, always affects the PS structural behavior.

This paper extends the GSA framework for HS proposed in Abbiati et al. (2021) to the case of PSs with non-deterministic behavior. Similar to the original framework, the idea is to surrogate the hybrid model response as a function of the input parameters that can be controlled by the experimenter and originate from substructures and loading (physical and numerical). However, latent variables that do not appear in the input parameter vector make the hybrid model response stochastic. To account for the latter, the *generalized lambda model* recently developed by Zhu and Sudret (2020, 2021a) is used here to directly surrogate the probability density function (PDF) of the response QoI. This is achieved by means of the family of generalized lambda distributions, which are suitable to approximate a wide class of distributions commonly found in engineering contexts (Karian and Dudewicz, 2000). The parameters of the generalized lambda distributions are cast as functions of the input parameters of the hybrid model and approximated via PCE (Xiu and Karniadakis, 2002; Blatman and Sudret, 2011). Normally, several repetitions of every single sample of the inputs parameters would be required to replicate the response of a stochastic hybrid model. However, acquiring such repetitions would be impossible in a HS. Instead, by using the generalized lambda model presented in Zhu and Sudret (2021a) we can solve this problem, as the generalized lambda model can be computed in a non-intrusive manner (i.e., the model is considered as a black-box) and does not require repeated HSs for a single sample of input parameters. For these reasons, the generalized lambda model is well-suited to surrogate the response of a *stochastic hybrid model*. Variance-based GSA is uniquely defined for deterministic simulators (Saltelli, 2008). In the context of stochastic simulators, three alternative variants of Sobol' sensitivity indices are discussed in Zhu and Sudret (2021b), namely *classical*, *quantile-based* and *trajectory-based* Sobol' indices. In this work, quantile-based Sobol' indices are used for the GSA of the HS response QoI. The reasoning behind this selection is twofold: firstly the quantile functions of the QoI is more of interest for the presented case study as the QoI itself is stochastic; secondly computation of the classical and the trajectory-based Sobol' indices (but not of the quantile-based) would require control of the latent variables. However, controlling the latent variables in data obtained from physical experiments is generally not possible.

The effectiveness of the proposed framework is demonstrated for a 3-degree-of-freedom (DOF) hybrid model subjected to mechanical and thermal loading. Specifically, thermal loading is experimentally exerted on the PS so that temperature fluctuations (out of control of the experimenter) entail a stochastic response of the hybrid model. It should be noted that the proposed framework handles the hybrid model as a black-box, and hence it can be applied to any dynamic system investigated via HS.

This paper is organized as follows. Section B.2 introduces the generalized lambda model and describes the GSA framework for stochastic hybrid models. Section B.3 presents the 3-DOF hybrid model used to test the framework. Section B.4 discusses the results of the GSA of the stochastic hybrid model response. Finally, Section B.5 presents the overall conclusions of this study.

B.2 Global sensitivity analysis framework

Let a random variable vector $\boldsymbol{x} = (x_1, \ldots, x_M)^\intercal \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$, where $\mathcal{D}_{\boldsymbol{X}}$ denotes the range of definition of $\boldsymbol{x}$, represent the input parameters to a HS. Due to the random nature of the hybrid model response, for a given

set of parameters $\boldsymbol{x}$, the corresponding QoI $Y(\boldsymbol{x})$ is a random variable rather than a deterministic value. This is because some latent variables $\boldsymbol{z}$ cannot be identified or measured in the process, which makes it impossible to include all the relevant variables in $\boldsymbol{x}$. Therefore, a stochastic HS can be expressed as a mapping:

$$\mathcal{M}_s : \boldsymbol{x} \mapsto \mathcal{M}_s(\boldsymbol{x}, \boldsymbol{Z}). \tag{B.1}$$

The latent variables are grouped into a random vector $\boldsymbol{Z}$. Note that with a fixed $\boldsymbol{x}$ and $\boldsymbol{Z}$ varying according to some probability distribution, the HS output $Y(\boldsymbol{x})$ remains random.

The input parameters of the vector $\boldsymbol{x}$ are treated as random, modeled by known probability distributions and grouped into a random vector $\boldsymbol{X} = (X_1, \ldots, X_M)^{\intercal}$. $\boldsymbol{X}$ is characterized by its joint distribution with the PDF denoted by $f_{\boldsymbol{X}}$. Furthermore, we assume that X_i's are mutually independent, and thus the joint PDF is the product of marginal PDFs, i.e. $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$ with f_{X_i} being the marginal PDF of the i-th variable.

For a given set of input parameters $\boldsymbol{x}$, the QoI $Y(\boldsymbol{x})$ is a random variable characterized by an unknown conditional probability distribution. Therefore, representing the stochastic behavior of a HS consists in estimating the response distribution for any parameters $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. However, one simulation for $\boldsymbol{x}$ does not provide the whole probability distribution but rather a single realization of $Y(\boldsymbol{x})$. Hence, it is usually necessary to repeatedly conduct experiments for the same $\boldsymbol{x}$ (called replications) to have enough insight into the resulting hybrid model response probability distribution. This quickly becomes intractable when the number of $\boldsymbol{x}$'s to be investigated increases. To alleviate the burden, surrogate models can be constructed to emulate the stochastic behavior of a HS. Once a surrogate model is constructed, we can perform further analysis of the hybrid model response at a low cost, namely the GSA.

The simplest surrogate model of a stochastic HS involves additive Gaussian noise:

$$Y^{\mathrm{s}}(\boldsymbol{x}) = h(\boldsymbol{x}) + Z, \quad Z \sim \mathcal{N}(0, \sigma^2). \tag{B.2}$$

To build such a surrogate, one needs to estimate the mean function h and the noise variance σ^2. In this case, PCE (Xiu and Karniadakis, 2002; Berveiller et al., 2006) and Gaussian processes (Rasmussen and Williams, 2006) with a regression setup can be directly applied. However, Eq. (B.2) can be rather restrictive. To cover a wider group of problems, we choose to use the recently developed statistical model called the *generalized lambda model* (Zhu and Sudret, 2020, 2021a).

B.2.1 Generalized lambda models

A generalized lambda model (GLaM) assumes that the probability distribution of the hybrid model response QoI $Y(\boldsymbol{x})$ can be approximated by a generalized lambda distribution (GLD). The latter is a highly flexible four-parameter distribution family, which is able to approximate many common distributions such as normal, lognormal, uniform and extreme value distributions (Freimer et al., 1988). A GLD is defined by its *quantile function*:

$$Q(u; \boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4}\right), \tag{B.3}$$

where $u \in [0,1]$ and $\boldsymbol{\lambda} = \{\lambda_l : l = 1,\dots,4\}$ are the four distribution parameters. More precisely, λ_1 is the location parameter, λ_2 is the scaling parameter, and λ_3 and λ_4 are the shape parameters. To have valid quantile functions, λ_2 should be positive. From Eq. (B.3), we can derive the associated PDF:

$$f_Y(y;\boldsymbol{\lambda}) = \frac{\lambda_2}{u^{\lambda_3-1} + (1-u)^{\lambda_4-1}} \mathbb{1}_{[0,1]}(u), \text{ with } u = Q^{-1}(y;\boldsymbol{\lambda}), \tag{B.4}$$

where $\mathbb{1}_{[0,1]}$ is the indicator function. From the above equation, it is clear that evaluating the PDF for a particular y requires solving numerically the equation $u = Q^{-1}(y;\boldsymbol{\lambda})$.

Under this setup, varying $\boldsymbol{x}$ will lead to $Y(\boldsymbol{x})$ following a GLD with different distribution parameters $\boldsymbol{\lambda}$. In other words, λ_l's are functions of $\boldsymbol{x}$, which allows us to express the QoI as:

$$Y(\boldsymbol{x}) \sim \text{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{B.5}$$

Recall that the input parameters $\boldsymbol{x}$ are modelled as independent random variables $\boldsymbol{X}$ with joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$. Under appropriate assumptions (Ernst et al., 2012), each component of $\boldsymbol{\lambda}(\boldsymbol{x})$ admits a polynomial chaos (PC) representation:

$$\begin{aligned} \lambda_l(\boldsymbol{x}) &\approx \lambda_l^{\text{PC}}(\boldsymbol{x};\boldsymbol{c}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_l} c_{l,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad l = 1,3,4, \\ \lambda_2(\boldsymbol{x}) &\approx \lambda_2^{\text{PC}}(\boldsymbol{x};\boldsymbol{c}) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} c_{2,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right), \end{aligned} \tag{B.6}$$

where $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^M\}$ is a basis of multivariate polynomials that are mutually orthogonal with respect to the probability measure of $\boldsymbol{X}$, $\boldsymbol{\alpha}$ is a multi-index that identifies the polynomial degree in each of the input variables, $\mathcal{A}_l \subset \mathbb{N}^M$ is a truncated set defining a finite set of basis functions for $\lambda_l(\boldsymbol{x})$ and $\boldsymbol{c} = \{c_{l,\boldsymbol{\alpha}},\dots,c_{l,\boldsymbol{\alpha}}\}$ denotes the associated coefficients (see Zhu and Sudret, 2020 for details). Note that the polynomial chaos expansion for $\lambda_2(\boldsymbol{x})$ is built on the logarithmic transform so as to ensure that $\lambda_2^{\text{PC}}(\boldsymbol{x})$ is always positive. Combining Eq. (B.5) with Eq. (B.6), we define the generalized lambda surrogate model:

$$Y^{\text{GLaM}}(\boldsymbol{x}) \sim \text{GLD}\left(\lambda_1^{\text{PC}}(\boldsymbol{x};\boldsymbol{c}), \lambda_2^{\text{PC}}(\boldsymbol{x};\boldsymbol{c}), \lambda_3^{\text{PC}}(\boldsymbol{x};\boldsymbol{c}), \lambda_4^{\text{PC}}(\boldsymbol{x};\boldsymbol{c})\right). \tag{B.7}$$

To build a GLaM, we need to determine the associated coefficients $\boldsymbol{c}$. To avoid the need for replications, which may result in a large number of experiments, we use the method developed by Zhu and Sudret (2021a). We first generate a set of realizations of the input random vector $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}\right\}$, called the *experimental design* (ED). For each point of the ED, we conduct a HS and collect the corresponding QoI into $\mathcal{Y} = \left\{\boldsymbol{y}^{(1)},\dots,\boldsymbol{y}^{(N)}\right\}$. Note that each HS may correspond to a different realization of the latent variable $\boldsymbol{Z}$, which does not need to be explicitly known in the analysis. In a second step, we estimate $\boldsymbol{c}$ by maximizing the conditional likelihood, i.e. minimizing the negative log-likelihood, that is:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \mathsf{L}(\boldsymbol{c}) = \arg\min_{\boldsymbol{c}} \sum_{i=1}^{N} -\log\left(f^{\text{GLD}}\left(y^{(i)};\boldsymbol{\lambda}^{\text{PC}}\left(\boldsymbol{x}^{(i)};\boldsymbol{c}\right)\right)\right), \tag{B.8}$$

where f^{GLD} is the PDF of the GLD defined in Eq. (B.4). To solve the optimization problem, it is necessary to

determine the support of $\boldsymbol{c}$, which is equivalent to finding the truncation set $\mathcal{A}_l$ for each λ_l^{PC}. To this end, we plug the hybrid-LAR algorithm (Blatman and Sudret, 2011) into the modified feasible generalized least-squares framework (Zhu and Sudret, 2021a). The latter fits the mean and the variance function in an alternative way. The basis functions selected for these two functions are then used to represent $\lambda_1^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_2^{\mathrm{PC}}(\boldsymbol{x})$, respectively. As λ_3 and λ_4 mainly control the PDF shape of a GLD, which is expected not to change much when $\boldsymbol{x}$ is changed, we can pick polynomials with low degree, namely 0 or 1, for $\lambda_3^{\mathrm{PC}}(\boldsymbol{x})$ and $\lambda_4^{\mathrm{PC}}(\boldsymbol{x})$. After specifying the basis functions, we solve Eq. (B.4) to build the associated GLaM.

A great advantage of using the generalized lambda surrogate model presented in Zhu and Sudret (2021a) is that it does not require repeated replications of the same sample. The reason behind this feature is that GLaM works as a statistical model, imposing the shape of the response distribution and using a parametric form, namely the PCE, to represent the dependence of the distribution parameters on the input variables. The basis selection for λ_1 and λ_2 is performed in a data-driven manner (which allows us to detect potential heteroskedastic effects), whereas λ_3 and λ_4 are kept constant. If we use replications, information is rather concentrated on those replicated points. In contrast, when there are no replications, the training samples cover uniformly the design space and provide more "homogeneous" information.

B.2.2 Sobol' sensitivity indices

Variance-based sensitivity analysis has been extensively studied and successfully developed in the context of deterministic models (Abbiati et al., 2021; Saltelli, 2008). For a random vector $\boldsymbol{X}$ with independent components, any deterministic mapping $Y = \mathcal{M}_d(\boldsymbol{X})$ with $\mathrm{Var}\left[Y\right] < +\infty$ can be decomposed as (Sobol', 1993):

$$\begin{aligned}\mathcal{M}_d(\boldsymbol{x}) &= \mathcal{M}_0 + \sum_{i=1}^{M} \mathcal{M}_i(x_i) + \sum_{1\leqslant i<j\leqslant M} \mathcal{M}_{i,j}(x_i, x_j) + \ldots + \mathcal{M}_{1,\ldots,M}(x_1, \ldots, x_M) \\ &= \mathcal{M}_0 + \sum_{\mathbf{u}\neq\varnothing} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}),\end{aligned} \tag{B.9}$$

where $\mathcal{M}_0$ is constant and denotes the mean value of Y, $\mathbf{u} = \{i_1, \ldots, i_s\} \subset \{1, \ldots, M\}$ are index sets and $\boldsymbol{x}_{\mathbf{u}}$ is a subvector of $\boldsymbol{x}$ containing only the components indexed by $\mathbf{u}$. This decomposition is unique (Sobol', 1993), and the elementary functions $\mathcal{M}_{\mathbf{u}}$ are defined by conditional expectations:

$$\mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \overset{\text{def}}{=} \sum_{\mathbf{v}\subset\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}\right], \tag{B.10}$$

where $|\mathbf{u}|$ gives the cardinality of $\mathbf{u}$. Following this definition, $\mathcal{M}_0 = \mathbb{E}\left[Y\right]$ is the expected value of Y. Moreover, the various terms $\mathcal{M}_{\mathbf{u}}$ are orthogonal with each other w.r.t. the inner product induced by the input PDF. Thus we can decompose the variance of Y as:

$$\mathrm{Var}\left[Y\right] = \mathbb{E}\left[(Y - \mathcal{M}_0)^2\right] = \sum_{\substack{\mathbf{u}\subset\{1,\ldots,M\} \\ \mathbf{u}\neq\varnothing}} \mathrm{Var}\left[\mathcal{M}_{\mathbf{u}}\left(\boldsymbol{X}_{\mathbf{u}}\right)\right]. \tag{B.11}$$

Additionally, the definition in Eq. (B.10) allows for calculating the variance of $\mathcal{M}_{\mathbf{u}}\left(\boldsymbol{X}_{\mathbf{u}}\right)$ by:

$$V_{\mathbf{u}} \overset{\text{def}}{=} \operatorname{Var}\left[\mathcal{M}_{\mathbf{u}}\left(\boldsymbol{X}_{\mathbf{u}}\right)\right] = \sum_{\mathbf{v}\subset\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \operatorname{Var}\left[\mathbb{E}\left[Y \mid \boldsymbol{X}_{\mathbf{v}}\right]\right]. \tag{B.12}$$

The Sobol' index $S_{\mathbf{u}}$ is defined as the ratio of $V_{\mathbf{u}}$ to the total variance $\operatorname{Var}[Y]$ (Sobol', 1993):

$$S_{\mathbf{u}} \overset{\text{def}}{=} \frac{V_{\mathbf{u}}}{\operatorname{Var}[Y]}. \tag{B.13}$$

For $|\mathbf{u}| = 1$, we obtain the first-order Sobol' indices $\{S_i : i = 1, \ldots, p\}$, which represent the main effect of each input variable. Higher-order indices quantify the interactive effect within a given group of input variables. The total Sobol' index S_{T_i} account for all the effect related to X_i:

$$S_{T_i} \overset{\text{def}}{=} \sum_{\mathbf{u}\ni i} S_{\mathbf{u}}. \tag{B.14}$$

Due to the random nature of stochastic simulators, decomposition similar to Eq. (B.9) is generally impossible. Therefore, it is necessary to represent a stochastic model by a deterministic function to obtain the associated Sobol' indices. Based on the choice of the deterministic representation, we can have different extensions of the Sobol' indices (Zhu and Sudret, 2021b).

The most straightforward way is to include the latent variables within the input variables (Iooss and Ribatet, 2009). This leads to the underlying (yet unknown in practice) deterministic model $\mathcal{M}_s$ defined in Eq. (B.1). Decomposing this function, we have:

$$Y = \mathcal{M}_s(\boldsymbol{X}, \boldsymbol{Z}) = \mathcal{M}_0 + \sum_{\substack{\mathbf{u}\subset\{1,\ldots,p\} \\ \mathbf{u}\neq\varnothing}} \mathcal{M}_{\mathbf{u}}(\boldsymbol{X}_{\mathbf{u}}) + \mathcal{M}_{\boldsymbol{Z}}(\boldsymbol{Z}) + \mathcal{M}_{\boldsymbol{X},\boldsymbol{Z}}(\boldsymbol{X}, \boldsymbol{Z}). \tag{B.15}$$

As the definition of $\mathcal{M}_{\mathbf{u}}$ is the same as Eq. (B.10), the Eq. (B.12) for $V_{\mathbf{u}}$ still holds. This implies that $S_{\mathbf{u}}$ can be determined by the statistical dependence between Y and $\boldsymbol{X}_{\mathbf{u}}$. The definition of the total index S_{T_i} requires including the interactive effect between X_i and the latent variables $\boldsymbol{Z}$. However, interactions with $\boldsymbol{Z}$ cannot be determined by the response distribution but rather depend on the precise data generation process (i.e. how $\boldsymbol{Z}$ is present in the function $\mathcal{M}_s$). Since the latent variables $\boldsymbol{Z}$ are generally impossible to characterize and control in a real experiment, the total Sobol' indices cannot be assessed.

Alternatively to Eq. (B.15), certain summary quantities of the response random variable $Y(\boldsymbol{x})$ can be employed as a deterministic representation of $Y(\boldsymbol{x})$. This is particularly helpful when the selected summary quantity itself is of interest. Typical quantities are mean $m(\boldsymbol{x}) = \mathbb{E}[Y(\boldsymbol{x})]$, variance $v(\boldsymbol{x}) = \operatorname{Var}[Y(\boldsymbol{x})]$ (Iooss and Ribatet, 2009), and α-quantiles $q_\alpha(\boldsymbol{x})$ (Browne, 2017). As these functions are well-defined as deterministic functions of $\boldsymbol{x}$ (since the effect of the latent variables $\boldsymbol{Z}$ has been marginalized), the associated Sobol' indices follow directly from Eq. (B.9).

A generalized lambda surrogate model emulates the response distribution of a stochastic model, which fully captures the statistical dependence between the input variables $\boldsymbol{X}$ and the QoI Y. Therefore, such a surrogate allows for evaluating both types of Sobol' indices mentioned above. More precisely, we can apply either Monte Carlo simulations or polynomial chaos expansions to the easy-to-evaluate emulator (see Zhu and Sudret, 2021b for more details).

Recall that the presented GSA framework assumes that the input parameters in $\boldsymbol{x}$ are statistically indepen-

dent. Nevertheless, a generalized lambda surrogate model can emulate the response of a stochastic hybrid model even in the case of dependent input parameters (Zhu and Sudret, 2021a). This holds since the dependence of Y on the input parameters are not affected by the dependence within the input variables. Nevertheless, for the case of dependent input parameters, generalized Sobol' indices or alternative variance-based sensitivity analysis methods should be employed as described in Chastaing et al. (2015) and Marelli et al. (2019).

B.3 Experimental illustration of the proposed GSA framework

B.3.1 Stochastic hybrid model

The proposed GSA framework is illustrated using a stochastic 3-DOF hybrid model subjected to both thermal and mechanical loading, illustrated in Figure B.1. As can be appreciated from Figure B.1a, the hybrid model consists of a simply-supported beam provided with rotational elastic restraints. In this regard, u_1 and u_2 indicate the two rotational DOF, while u_3 is the axial DOF. Figure B.1b describes the substructuring of the hybrid model into PS and NS. The NS comprises two elastic rotational restraints, an axial spring and a linear dashpot whereas the PS coincides with the beam element. Specifically, the axial spring is characterized by a constant stiffness of $K_3 = 8{,}100 \times 10^3$ N/m. Two lumped rotational masses are defined by $J_1 = J_2 = 10$ kgm^2 while the lumped translational mass is defined by $M_3 = 5{,}000$ kg. The linear dash-pot is characterized by $C_3 = 1{,}129 \times 10^3$ Ns/m. The PS consists of an aluminum plate of 0.2×0.002 m rectangular cross-section and length $L = 0.47$ m. Accordingly, the cross-section of the plate is characterized by an area $A = 4 \times 10^{-4}$ m^2 and a moment of inertia $I = 66.67 \times 10^{-12}$ m^4. The Young's modulus, density and thermal expansion coefficient of aluminum are $E = 69.5$ GPa, $\rho = 2{,}700$ kg/m^3, and $\alpha = 23 \times 10^{-6}$ °C^{-1}, respectively. Since HS are conducted in *pseudodynamic* mode using a testing time scale equal to 50, the PS mass (rotational and translational) does not contribute to the hybrid model inertia. For the sake of clarity, all equations and plots in the following refer to simulation time, which is virtual and 50 times slower than the wall-clock time.

Mechanical loading is supplied as a bending moment history $F(t)$ applied to the right rotational DOF u_2, while thermal loading, applied by a heating lamp, is defined by a ramp & hold temperature history $\theta(t)$. The expressions of both read:

$$F(t) = \begin{cases} F_{max} \sin(\frac{\pi(t-t_0)}{T}) \,, t_0 \leqslant t \leqslant t_0 + T/2 \\ 0 \,, \text{elsewhere} \end{cases} \tag{B.16}$$

$$\theta(t) = \begin{cases} \dot{\theta} t \,, \theta(t) < \theta_{max} \\ \theta_{max} \,, \text{elsewhere} \end{cases} \tag{B.17}$$

where F_{max} and T are peak value and period of the half-side bending moment pulse applied to DOF u_2 with a time shift $t_0 = 0.1$ s; $\dot{\theta} = 21.5$ °C/s and θ_{max} are temperature rate and plateau characterizing the temperature

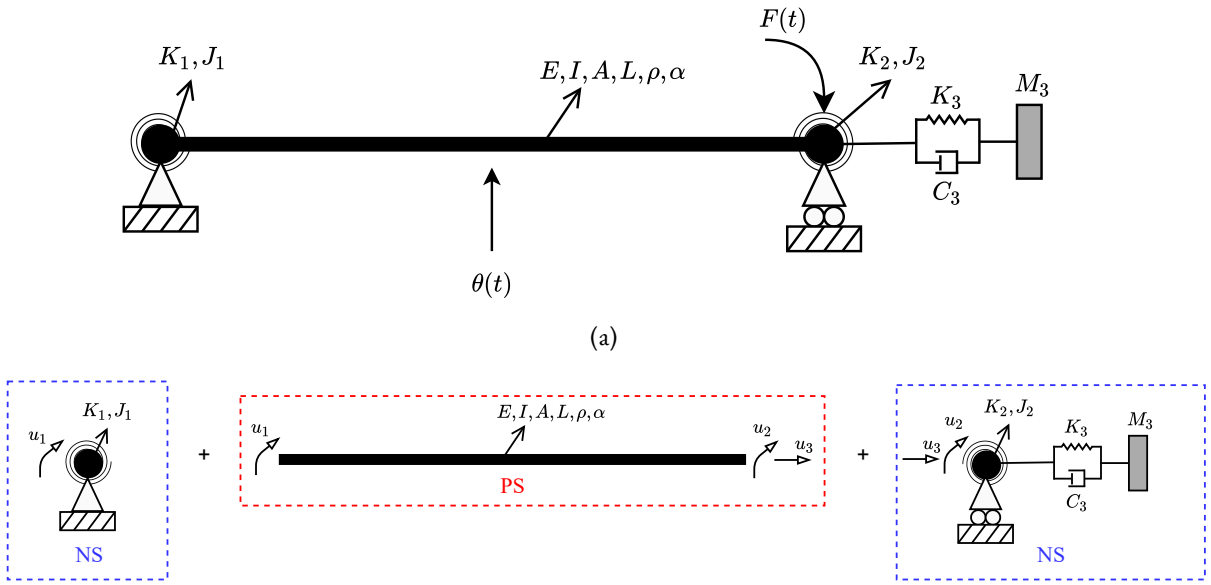
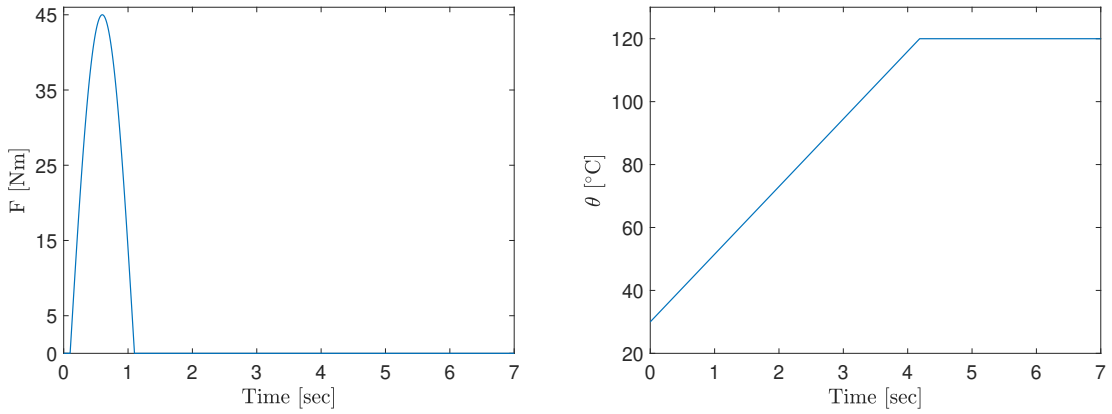


Figure B.1: Reference structural system: (a) prototype structure and (b) its hybrid model.

history imposed to the PS. For the sake of this example, Figure B.2a depicts the bending moment history computed for $F_{max} = 45$ Nm and $T = 1$ s. Similarly, Figure B.2b depicts the temperature history computed for $\theta_{max} = 120$ ℃.



(a) (b)

Figure B.2: Sample time history loading: (a) bending moment history for $F_{max} = 45$ Nm and $T = 1$ s; (b) temperature time history for $\theta_{max} = 120$ ℃.

Consistent with the motivations underlying the development of the GSA framework, the stiffness of the two elastic rotational springs, which play the role of boundary conditions to the PS, as well as the loading parameters, are selected as input parameters for the surrogate modeling phase. Recall that these parameters were chosen as inputs to the GSA framework, since in the majority of cases encountered in structural engineering, loading is stochastic, while the boundary conditions are highly uncertain. In line with the procedure described

Parameter	Description	Lower Bound	Upper Bound	Units
K_1	Rotational stiffness left spring	110.58	1,105.80	Nm/rad
K_2	Rotational stiffness right spring	110.58	1,105.80	Nm/rad
F_{max}	Bending moment pulse peak	40.00	50.00	Nm
T	Bending moment pulse period	0.50	2.00	sec
θ_{max}	Temperature plateau	100.00	130.00	℃

Table B.1: Input parameters of the hybrid model.

in Section B.2, the input parameters are described by independent uniform distributions, whose bounds are summarized in Table B.1.

It is important to remark that, in order to reduce the experimental effort required to validate the proposed framework, the hybrid model and loading excitation were designed such that the PS always remained in the linear response regime. As a result, HS were conducted using a single aluminum plate.

The response QoI selected for the GSA corresponds to the maximum absolute out-of-plane deflection of the tested aluminum plate, which is denoted as $u_{L,max}$.

B.3.2 Hybrid simulation setup

The 3-DOF HS test rig used to conduct HS is a stiff loading frame equipped with four electro-mechanical actuators and an infrared (IR) lamp module interfaced to an INDEL real-time computer (Abbiati et al., 2018). The 3-DOF HS test rig is designed to test plate specimens with an approximate footprint of 200 × 500 mm and thickness varying between 1 and 3 mm. Figure B.3 illustrates the architecture of the HS setup, including a close-up view of the plate specimen accommodation. Two axonometric views of the 3-DOF test rig, consisting of the main hardware components are shown in Figure B.4.

The moving parts of the test rig are colored in yellow, the plate specimen in brown and the fixed parts in gray. The latter are fixed to a reaction frame. In order to impose the u_1 and u_2 rotations, two rack-pinion systems (10) are installed along the vertical actuator y_1 and y_2 (1). The rack-pinion systems aim at transforming the commanded displacements from the actuators to rotational DOF, applied to the short edges of the plate specimen (6) through aluminum clamps (3). The two horizontal actuators x_1 and x_2 control the position of the moving frame mounted on profiled rail guides (4) and corresponding to the axial DOF u_3 of the plate specimen (6). A linear variable differential transformer (LVDT) measures the out-of-plane deflection at the mid-span of the plate specimen (labeled u_L in Figure B.3). A Type K thermocouple installed at the center of the plate provides the feedback signal for the control of the IR lamp, which imposes the temperature history $\theta(t)$.

The GINLink bus connects the actuator servo-drivers INDEL stand-alone controllers (SAC4), the IR lamp and all the data acquisition (DAQ) modules to the real-time computer INDEL stand-alone master (SAM4), which executes the HS software. The latter is developed in MATLAB/SIMULINK, compiled, and downloaded to the INDEL SAM4 from the Host-PC. At each simulation time step, the HS software generates the temperature command for the IR lamp. The latter command is generated using a predefined time history temperature response (see, for example, Figure B.2b). Also, at each time step of the simulation, the HS software imposes displacements u_1, u_2 and u_3 to the plate specimen, the PS. The latter displacements are computed as the response of the NS to the imposed bending moment time history response (see, for example, Figure B.2a). Using force transducers the HS software reads the corresponding restoring forces r_1, r_2 and r_3, due to the imposed dis-

placements and temperature commands, and uses them to solve the coupled equation of motion of the hybrid model. A comprehensive description of the time integration scheme used for HS is reported in Abbiati et al. (2019).

Forces were manually set to zero before starting the HS. An electric fan cools down the PS at the end of each test. Room temperature was quite stable and equal to 30 °C, namely $\theta(0) = 30$ °Cin Eq. (B.17), for the entire testing campaign.

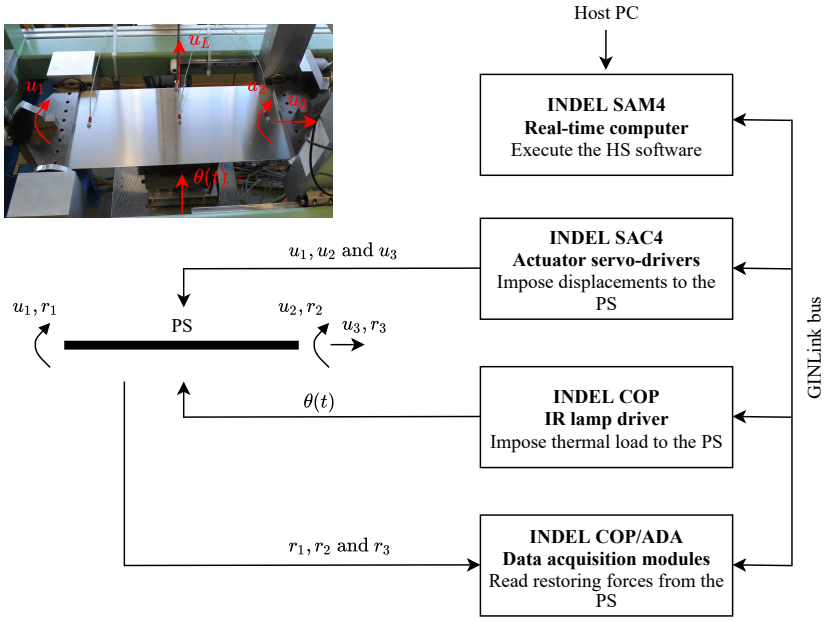
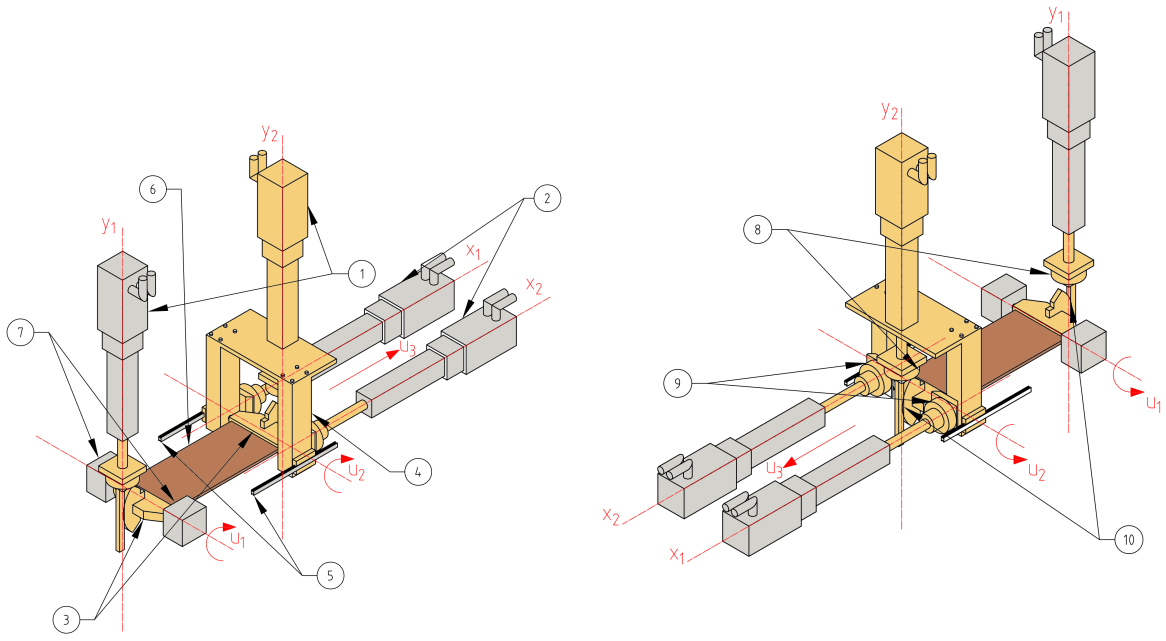


Figure B.3: Architecture of the 3-DOF HS test rig.

B.4 Results and discussion

The response of the hybrid model described in Section B.3.1 was evaluated using the HS setup described in Section B.3.2 on 200 samples of the input parameter vector generated using Latin hypercube sampling (Mckay et al., 2000). The resulting ED $\mathcal{X}, \mathcal{Y}$ was used for computing surrogate models. In a previous work of some of the authors (Abbiati et al., 2021), 200 samples were proven to be adequate to train PCE surrogate models with acceptably small validation error. In particular, in that work, PCE estimates converged in trustworthy values with ED size larger than 50 samples. In this regard, 200 samples of the input parameter space were used in this study as well, as an initial estimate. Results presented later on demonstrate that this number of samples was sufficient to train the generalized lambda surrogate model.

Figure B.5 reports out-of-plane and axial displacement histories obtained via HS for a single sample of input parameters, where $u_{L,max}$, $u_{3,max}$, $u_{L,0}$ and $u_{3,0}$ scalar quantities are highlighted. Specifically, $u_{L,max}$ corresponds to the QoI (see Eq. (B.18)) and $u_{3,max}$ to the absolute maximum axial displacement while $u_{L,0}$ and $u_{3,0}$ indicate the initial position of the out-of-plane displacement and axial axes respectively, relative to the value measured during the first HS. Figure B.6 describes the evolution of $u_{L,0}$ and $u_{3,0}$ scalar quantities over the entire experimental campaign.



(a) (b)

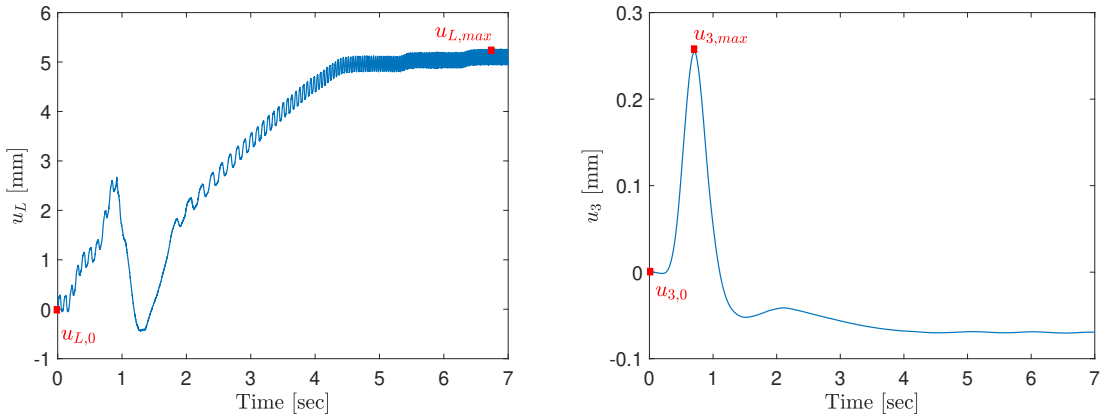
Figure B.4: Axonometric views of the 3-DOF HS test rig with its main components (the moving parts are colored in yellow, the plate specimen is brown, while those parts fixed to the reaction frame are gray): (a) front and (b) back view perspective.

B.4.1 Drift observed in measurement data

From Figure B.6 it is clear that both $u_{L,0}$ and $u_{3,0}$ quantities have a constant drift, which results in a total accumulated out-of-plane and axial displacements of 3.0 mm and 0.3 mm, respectively. Such a drift can be reasonably ascribed to the cumulative slippage of plate fixtures produced by heating/cooling cycles. This drift occurs regardless of the type of analysis that the HS setup was used for, and since the source of the drift is clear, it should be removed from the acquired raw data before any further post-processing. Accordingly, prior to the calculation of surrogate models, the effect of drift on $u_{L,max}$ was eliminated via linear detrending with respect to $u_{L,0}$. The detrended QoI is referred to as $\hat{u}_{L,max}$ and compared to original values in Figure B.7. Notably, $u_{L,0}$ is independent of the parameters of the hybrid model since the initial position of the PS was set by zeroing actuator forces.

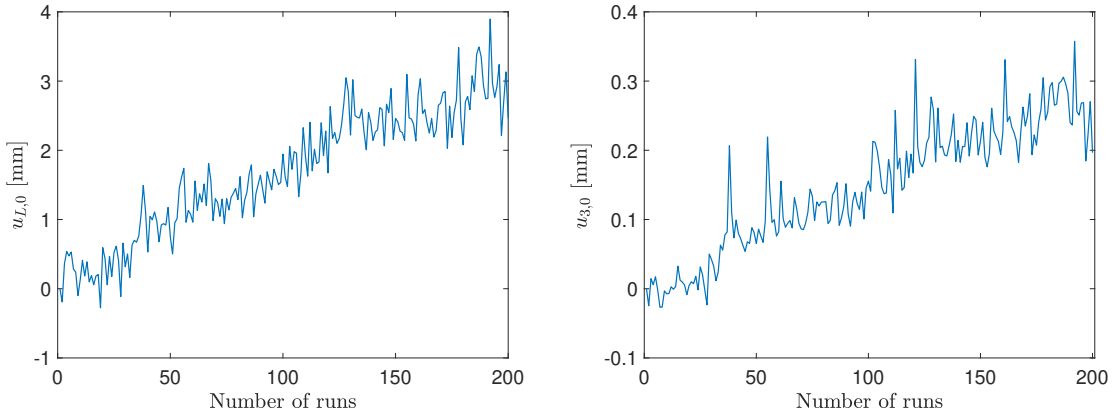
Consistent with the notation introduced in Section B.2, surrogate modeling was performed considering the following input parameter vector and response QoI:

$$\begin{aligned} \boldsymbol{X} &= (K_1, K_2, F_{max}, T, \theta_{max})^{\mathsf{T}}, \\ Y &= \hat{u}_{L,max}. \end{aligned} \tag{B.18}$$



(a) (b)

Figure B.5: Time history response of the hybrid model with $K_1 = 224.454$ Nm/rad, $K_2 = 118.235$ Nm/rad, $F_{max} = 48$ Nm, $T = 0.618$ s and $\theta_{max} = 116.73$ °C: (a) out-of-plane displacement and (b) axial displacement.



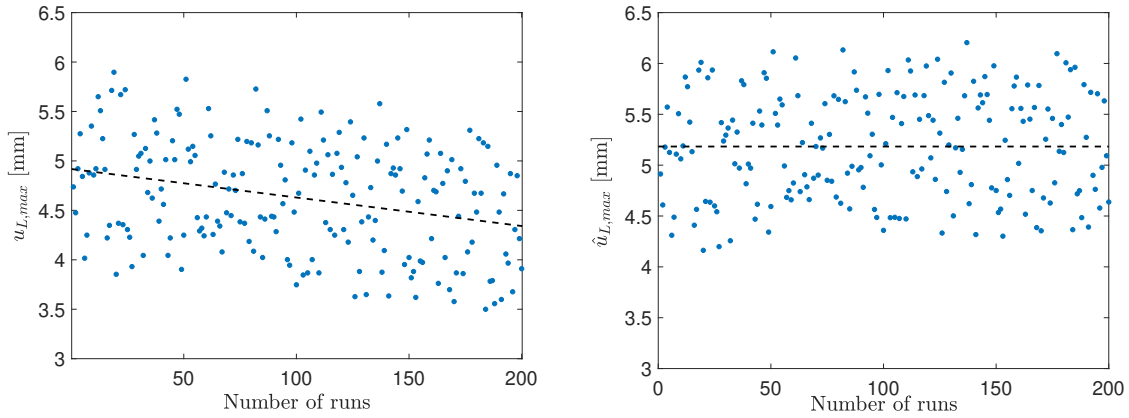
(a) (b)

Figure B.6: Drift in the hybrid model response: (a) u_L and (b) u_3

B.4.2 Global sensitivity analysis framework results

A GLaM of the hybrid model dynamic response QoI was computed as explained in Section B.2. In this study, we set the candidate degrees up to 5 for λ_1^{PC} and 3 for λ_2^{PC}. In order to validate the GLaM, 10 repeated HS were performed for two validation ED points, namely 58 and 157, associated with different regions of the input parameter space and characterized by appreciably different QoI values. For each validation ED point, Figure B.8 compares the GLaM prediction to the empirical distribution of the 10 related repetitions. It is observed that the GLaM correctly captures the empirical distribution of the QoI for both points. It is interesting to note that the computed GLaM model converged to zero-order polynomials for λ_2^{PC}, and that the two PDFs of Figure B.8 are similar in shape, thus suggesting an homoscedastic stochasticity of the hybrid model, which was further verified using a PCE surrogate model. Specifically, a residual analysis was performed on the difference between the measured QoI and its PCE.

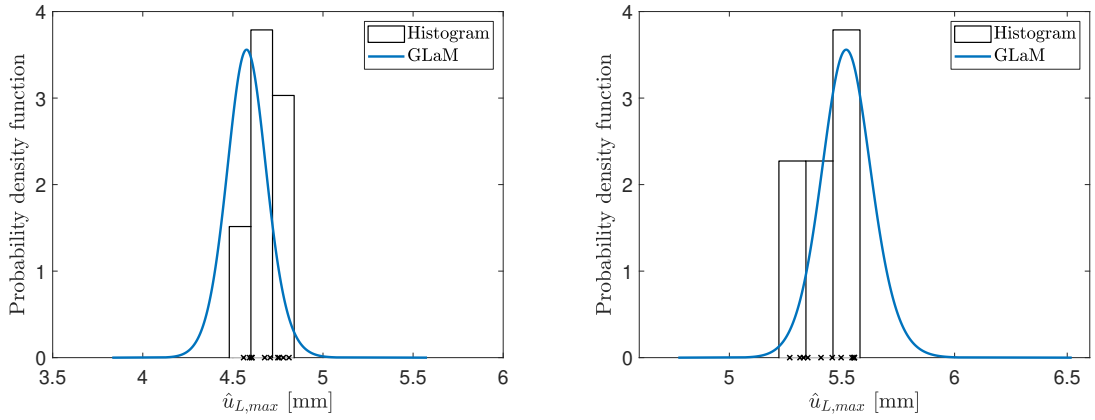
As highlighted by Torre et al. (2019), in presence of noisy data, PCE is a powerful *denoiser*, which natu-



(a) (b)

Figure B.7: Effect of detrending on the QoI: (a) original values ($u_{L,max}$) and (b) values after detrending ($\hat{u}_{L,max}$). Dashed lines indicate a linear trend of data.

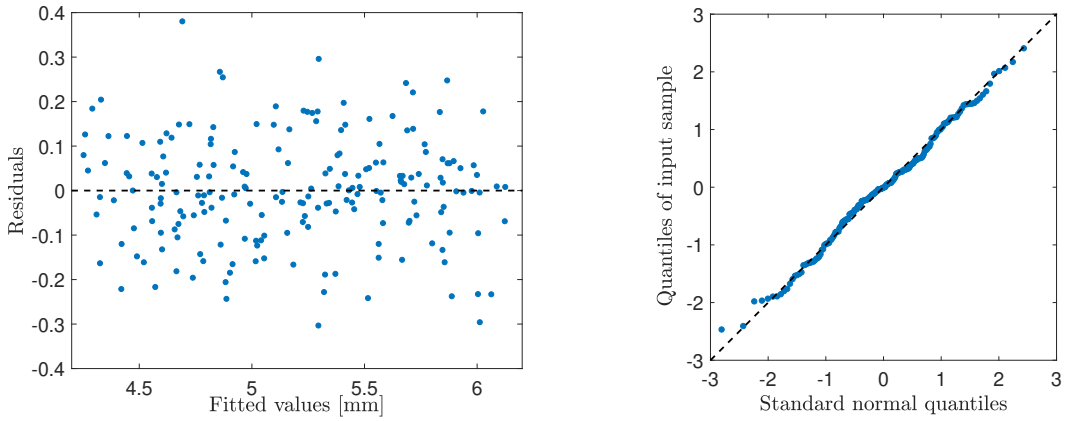
rally provides a surrogate of the average model response $\mathbb{E}_Z[Y|\boldsymbol{x}]$. In this regard, the Tukey-Anscombe plot (Anscombe and Tukey, 1963) of Figure B.9a compares the PCE output to the corresponding residual for each sample of the ED. The Q-Q plot of Figure B.9b compares the empirical quantiles of the residuals normalized to unit standard deviation to the theoretical values of a standard normal distribution $\mathcal{N}(0,1)$. The zero-average uniform scattering of residuals highlighted by the Tukey-Anscombe plot and the fairly good agreement between empirical and theoretical quantiles highlighted by the Q-Q plot confirms that the hybrid model response was affected by a Gaussian homoscedastic additive noise.



(a) (b)

Figure B.8: PDF of $\hat{u}_{L,max}$ predicted by the GLaM versus empirical distributions for ED points: (a) 58 and (b) 157. Cross markers denote the 10 related repetitions for each point.

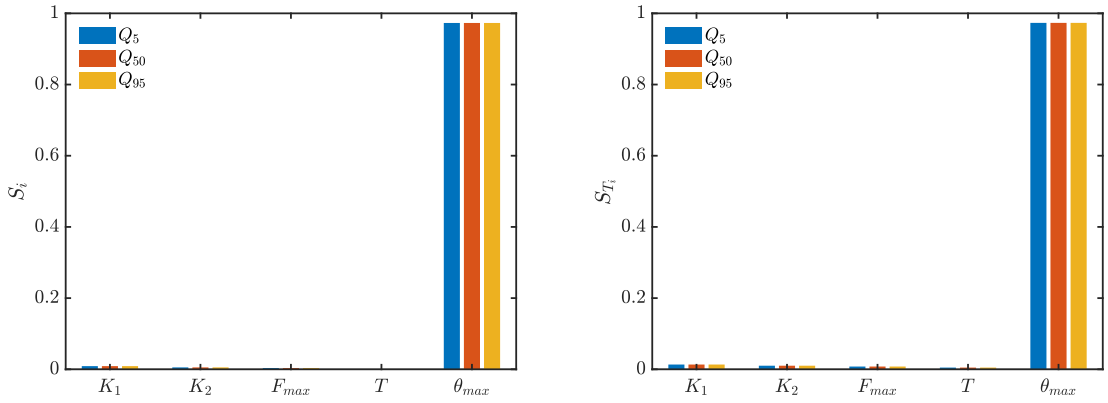
As reported in Section B.2.2, only first- and higher-order Sobol' indices but not total Sobol' indices can be obtained from the GLaM of the QoI. Instead, first and total Sobol' indices can be computed for the QoI quantiles. Accordingly, Figure B.10 provides first and total Sobol' indices of the 5, 50 and 95 % quantiles of $\hat{u}_{L,max}$. The results of the GSA indicate that the temperature plateau value θ_{max} is the most sensitive input parameter for the selected QoI. Additionally, the equal Sobol' indices values for each quantile unveil the homoscedastic



(a) (b)

Figure B.9: Analysis of QoI residuals with respect to PCE: (a) Tukey-Anscombe plot and (b) Q-Q plot.

response of the stochastic surrogate.



(a) (b)

Figure B.10: Sobol' indices of 5, 50 and 95 % quantiles of $\hat{u}_{L,max}$: (a) first order and (b) total.

The development and implementation of the surrogate modeling, as well as the GSA, was performed using the UQLab software framework developed by the Chair of Risk, Safety and Uncertainty Quantification in ETH Zurich (Marelli and Sudret, 2014).

B.5 Conclusions

This paper described a framework for global sensitivity analysis of stochastic hybrid models. A generalized lambda surrogate modeling technique is used to compute the Sobol' sensitivity indices for the quantiles of a response quantity of interest. The idea of using surrogate modeling to enable global sensitivity studies with few expensive-to-evaluate hybrid simulations was already presented in a previous work of the authors. However, in that work the physical substructure of the hybrid model was treated as deterministic, namely the associated aleatory uncertainties were neglected by assuming that nominally identical specimens have identical responses.

Nevertheless, this assumption is still far from a realistic scenario in structural testing, as nominally identical specimens are, in practice, never actually identical. Therefore, the novelty of this work lies in the extension of global sensitivity analysis to the case of stochastic hybrid models, covering the more realistic situation where the hybrid model response for two nominally identical physical substructures is not repeatable. A great advantage of the proposed framework is that the generalized lambda surrogate model does not require repeated evaluations of the same sample. On the other hand, an assumption of the framework is that the response distribution of the stochastic hybrid model can be approximated by the generalized lambda distribution. The main limitation of the generalized lambda surrogate model is that the generalized lambda distribution is flexible but cannot represent multimodal distributions. Nevertheless, one can use a mixture of generalized lambda models to bypass this limit. In addition, a generalized lambda surrogate model can emulate the response of a stochastic model even in the case of dependent input parameters. However, for the latter case, generalized Sobol' indices should be employed instead.

The effectiveness of the proposed framework is demonstrated in an experimental application consisting of a hybrid model with five parameters and subjected to mechanical and thermal loading. The results of the demonstration study highlight that the stochasticity of the particular hybrid model under consideration is homoscedastic with respect to the hybrid model parameters. Accordingly, both the first-order and total Sobol' sensitivity indices of 5, 50, and 95 % quantiles are almost identical. Moreover, the temperature plateau value of the thermal loading is the most sensitive parameter for the selected response quantity of interest. The outcome of the experiment demonstrates the effectiveness of the proposed global sensitivity analysis framework in revealing the inner workings of the hybrid model.

Global sensitivity analysis for stochastic hybrid models advances the current practices of hybrid simulation and establishes it as a tool capable to investigate the dynamic response of structural systems taking into account aleatory and epistemic uncertainties originating not only from numerical substructures and respective loading but also from physical specimens. The latter feature is of significant importance since the internal stochasticity of physical specimens is in general unknown and difficult to control.

Future research will address the issue of adaptive sampling of the parameter space of the stochastic hybrid model to minimize the experimental cost necessary to compute an accurate surrogate model for global sensitivity analysis.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764547. The sole responsibility of this publication lies with the author(s). The European Union is not responsible for any use that may be made of the information contained herein. The realization of the test rig work was funded by the Swiss Secretariat of Education Research and Innovation (SERI) - Swiss Space Office (SSO) [THERMICS MdP2016 Project (Thermo-Mechanical Virtualization of Hybrid Flax/Carbon Fiber Composite for Spacecraft Structures), grant number REF-1131-61001].

Data Availability Statement

All data and code that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abbiati, G., Covi, P., Tondini, N., Bursi, O. S., and Stojadinović, B. (2020). A real-time hybrid fire simulation method based on dynamic relaxation and partitioned time integration. *Journal of Engineering Mechanics*, 146(9):04020104.

Abbiati, G., Hey, V., Rion, J., and Stojadinović, B. (2018). Thermo-mechanical virtualization of hybrid flax/carbon fiber composite for spacecraft structures. thermics project, final report. Technical report, ETH Zurich.

Abbiati, G., Lanese, I., Cazzador, E., Bursi, O. S., and Pavese, A. (2019). A computational framework for fast-time hybrid simulation based on partitioned time integration and state-space modeling. *Structural Control and Health Monitoring*, 26(10):1–28.

Abbiati, G., Marelli, S., Tsokanas, N., Sudret, B., and Stojadinović, B. (2021). A global sensitivity analysis framework for hybrid simulation. *Mechanical Systems and Signal Processing*, 146.

Anscombe, F. J. and Tukey, J. W. (1963). The examination and analysis of residuals. *Technometrics*, 5:141–160.

Bas, E. E., Moustafa, M. A., and Pekcan, G. (2022). Compact Hybrid Simulation System: Validation and Applications for Braced Frames Seismic Testing. *Journal of Earthquake Engineering*, 26(3):1565–1594.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15(1–3):81–92.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Browne, T. (2017). *Regression Models and Sensitivity Analysis for Stochastic Simulators: Applications to Non-Destructive Examination*. PhD thesis, Université of Paris Descartes, Paris.

Chastaing, G., Gamboa, F., and Prieur, C. (2015). Generalized Sobol sensitivity indices for dependent variables: numerical methods. *Journal of Statistical Computation and Simulation*, 85(7):1306–1333.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Communications in Statistics - Theory and Methods*, 17:3547–3567.

Idinyang, S., Franza, A., Heron, C. M., and Marshall, A. M. (2019). Real-time data coupling for hybrid testing in a geotechnical centrifuge. *International Journal of Physical Modelling in Geotechnics*, 19(4):208–220.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94:1194–1204.

Karian, Z. A. and Dudewicz, E. J. (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press.

Marelli, S., Lamas, C., Konakli, K., Mylonas, C., Wiederkehr, P., and Sudret, B. (2019). UQLab user manual – Sensitivity analysis. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-106.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proceedings of the 2nd International Conference on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Mckay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input Variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

Moustafa, M. and Mosalam, K. (2015). Development of Hybrid Simulation System for Multi-Degree- of-Freedom Large-Scale Testing. In *6th International Conference on Advances in Experimental Structural Engineering*, University of Illinois, Urbana-Champaign, United States.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Saltelli, A., editor (2008). *Global sensitivity analysis: the primer*. John Wiley, Chichester, England ; Hoboken, NJ.

Sauder, T., Marelli, S., Larsen, K., and Sørensen, A. J. (2018). Active truncation of slender marine structures: Influence of the control system on fidelity. *Applied Ocean Research*, 74:154–169.

Schellenberg, A. H., Mahin, S. A., and Fenves, G. L. (2009). Advanced Implementation of Hybrid Simulation. Technical Report PEER 2009/104, Pacific Earthquake Engineering Research Center, University of California, Berkeley.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical and Computer Modelling*, 1:407–414.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93:964–979.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623.

Vilsen, S. A., Sauder, T., Sørensen, A. J., and Føre, M. (2019). Method for Real-Time Hybrid Model Testing of ocean structures: Case study on horizontal mooring systems. *Ocean Engineering*, 172:46–58.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021a). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

Zhu, X. and Sudret, B. (2021b). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815.

C

Impact of ploidy and pathogen life cycle on resistance durability

This chapter is a post-print of

> Saubin, M., de Mita, S., Zhu, X., Sudret, B., and Halkett, F. (2021). Impact of ploidy and pathogen life cycle on resistance durability. *Peer Community Journal*, 1. DOI:10.24072/pcjournal.10.

differing from the published paper only in terms of layout and formatting.

Author contributions. **M. Saubin**: Conceptualization, Methodology, Software, Validation, Writing - original draft. **S. de Mita**: Conceptualization, Methodology, Writing - Review & Editing. **X. Zhu**: Methodology, Writing - Review & Editing. **B. Sudret**: Methodology, Writing - Review & Editing. **F. Halkett**: Conceptualization, Supervision, Funding acquisition, Writing - Original Draft.

Abstract

The breeding of resistant hosts based on the gene-for-gene interaction is crucial to address epidemics of plant pathogens in agroecosystems. Resistant host deployment strategies are developed and studied worldwide to decrease the probability of resistance breakdown and increase the resistance durability in various pathosystems. A major component of deployment strategies is the proportion of resistant hosts in the landscape. However, the impact of this proportion on resistance durability remains unclear for diploid pathogens with complex life cycles. In this study, we modelled pathogen population dynamics and genetic evolution at the virulence locus to assess the impact of the ploidy (haploid or diploid) and the pathogen's life cycle (with or without host alternation) on resistance durability. Ploidy has a strong impact on evolutionary trajectories, with much greater stochasticity and delayed times of resistance breakdown for diploids. This result emphasises the importance of genetic drift in this system: as the virulent allele is recessive, positive selection on resistant hosts only applies to

homozygous (virulent) individuals, which may lead to population collapse at low frequencies of the virulent allele. We also observed differences in the effect of host deployment depending on the pathogen's life cycle. With host alternation, the probability that the pathogen population collapses strongly increases with the proportion of resistant hosts in the landscape. Therefore, resistance breakdown events occurring at high proportions of resistant hosts frequently amount to evolutionary rescue. Last, life cycles correspond to two selection regimes: without host alternation (soft selection) the resistance breakdown is mainly driven by the migration rate. Conversely, host alternation (hard selection) resembles an all-or-nothing game, with stochastic trajectories caused by the recurrent allele redistributions on the alternate host.

C.1 Introduction

Plant pathogens can quickly evolve (Perkins et al., 2013), and the loss of host genetic diversity in agroecosystems compared to natural ecosystems can enhance the spread of epidemics (Mundt, 2002; Burdon and Thrall, 2008; Garrett et al., 2009; Haas et al., 2011; Ostfeld and Keesing, 2012; Zhan et al., 2015). In this context, many plant protection strategies are developed and studied worldwide (Bousset and Chèvre, 2013), particularly spatio-temporal host resistance deployment strategies (Mundt, 2002; Gilligan and van den Bosch, 2008; Sapoukhina et al., 2009; Burdon et al., 2014; Djian-Caporalino et al., 2014; Fabre et al., 2015; Bousset et al., 2018; Rimbaud et al., 2018). However these modelling studies seldom account for pathogen differences in life cycle and ploidy levels.

While quantitative resistance has gained interest (Pilet-Nayel et al., 2017), the breeding of disease resistant plants is still often based on the gene-for-gene interaction (Person et al., 1962; Zhan et al., 2015). In the simplest case of specific response, the result of the infection is determined by the interaction between a locus in the plant (the resistance gene) and in the pathogen (the avirulence gene; Flor, 1971). This interaction leads to an all-or-nothing response and therefore such resistances are called qualitative. Qualitative resistances often rely on the recognition of a specific pathogen molecule (an effector protein for instance) by a plant immune receptor (Lo Presti et al., 2015). If the pathogen is recognised by the plant, the infection is stopped and the plant is called *resistant*. But the pathogen species evolves in multiple ways to escape host recognition (Rouxel and Balesdent, 2017). When a pathogen can infect a resistant host it is called *virulent*, as opposed to *avirulent* individuals. For avirulent individuals, if the product of the avirulence gene is not recognised by the plant, the infection occurs and the plant is called *susceptible*. Hence, virulent individuals can infect both susceptible and resistant hosts, while avirulent individuals can only infect susceptible hosts. In its simplest cases, the avirulence gene exists in two versions: the avirulent Avr allele and the virulent avr allele. The plant resistance can thus be overcome by a mutation of the Avr allele which modifies the pathogen recognition by the plant. The Avr allele is then replaced by a virulent avr allele which leads to a virulent pathogen (Stukenbrock and McDonald, 2009).

In natural systems, the constant turnover of resistance and avirulence genes results from a strong coevolutionary interaction between both species (Zhan et al., 2014), represented by the concept of arms-race (Brown and Tellier, 2011). On both sides, the most adapted allele can spread in the population, sometimes replacing alleles conferring lower fitness to individuals (Brown and Tellier, 2011; Persoons et al., 2017). These genes are under strong selective pressure and at each selective event a selective sweep can occur and drastically reduce the genetic diversity of both species (Oleksyk et al., 2010; Terauchi and Yoshida, 2010). In natural populations, rare

host genotypes can be maintained by negative frequency-dependent selection, resulting in the preservation of host polymorphism (Lewontin, 1958). In agroecosystems, however, pure crops of resistant hosts hinder this maintenance of polymorphism (Zhan et al., 2015). Therefore, the issue of such resistance deployments is often a resistance breakdown, *i.e.* the failing of the host to remain resistant to the pathogen, which can result in severe epidemics (Johnson, 1984; Pink and Puddephat, 1999; Brown and Tellier, 2011; Burdon et al., 2016). Such a resistance breakdown can occur more or less quickly, depending on the pathosystem considered and the environmental conditions (Van den Bosch and Gilligan, 2003). This observation raises the question of resistance durability, which can be defined as the time until the virulent population reaches a given threshold in the pathogen population. Definitions of resistance durability can have diverse acceptations depending on the threshold considered (Van den Bosch and Gilligan, 2003; Pietravalle et al., 2006; Brown, 2015; Carolan et al., 2017; Lof et al., 2017; Pacilly et al., 2018; Rimbaud et al., 2021). Considering several thresholds can help in capturing different steps of the pathogen dynamics.

Resistance durability becomes a major economical issue when epidemics impact crop yields. Therefore, it has often been studied through the modelling of epidemics spread in agricultural landscapes (Rimbaud et al., 2021). Such models can couple epidemiological and evolutionary processes, and often aim to study the influence of different biological parameters on the emergence of pathogens, their specialisation to the host plant, the evolutionary dynamics of virulence, or on the resistance durability (Van den Bosch and Gilligan, 2003). *Virulence* is defined here as the ability for a pathogen individual to infect a resistant host, in accordance to the phytopathology literature (Flor, 1971; McDonald and Linde, 2002). These parameters can be specific to the host plant (proportion of resistant host in the landscape, their spatial and temporal distribution) or to the pathogen (life cycle, mutation rate, dispersal; Fabre et al., 2012, 2015; Papaïx et al., 2015, 2018; Soularue et al., 2017). These models often represent haploid pathogens with a virulent and an avirulent genotype, evolving purely asexually on a landscape composed of compartments, gathering resistant or susceptible hosts (Pietravalle et al., 2006; Lof and van der Werf, 2017; Lof et al., 2017; Pacilly et al., 2018). Regarding the pathogen, high risks of resistance breakdown are observed for pathogen populations with high gene flow and mutation rates, large effective population sizes, and partially asexual reproductions (McDonald and Linde, 2002). Regarding the host, the increase in the proportion of resistant hosts should increase the selection pressure, hence weakening the resistance durability (Van den Bosch and Gilligan, 2003; Pietravalle et al., 2006). However, a large proportion of resistant hosts also reduces the initial size of the pathogen population and thus the risk of resistance breakdown (Pacilly et al., 2018), partly because a small population size reduces the likelihood that a virulent individual will emerge through mutation.

However, the impact of host resistance deployment on resistance durability remains unclear when the pathogen is diploid (like rust fungi, oomycetes, or nematodes). When the product of the avirulence gene is a specific molecule like an effector protein, the pathogen is virulent only if this product is not detected by the product of the corresponding resistance gene in the host (Stukenbrock and McDonald, 2009). Therefore, for a diploid individual the pathogen is virulent only if the products of both alleles avoid detection by the host. In other words, in the classical gene-for-gene interaction the virulent allele is recessive (Thrall et al., 2016). Consequently, a heterozygous Avr/avr individual is phenotypically avirulent, and the selective advantage of the virulence is effective among homozygous avr/avr individuals only. At low frequency, avr alleles are then carried by heterozygous individuals and mostly subjected to drift.

Diploid pathogens exhibit a large variability of life cycles (Agrios, 2005). We can especially distinguish

autoecious pathogens, which complete their life cycle on a unique host species, from heteroecious pathogens which need two different and successive host species to complete their life cycle (Moran, 1992; Lorrain et al., 2019). This presence or absence of an alternate host species could also affect the influence of host deployment strategy on resistance durability. Moreover, most studies focus on purely asexual pathogens, but the highest risks of resistance breakdown were observed for mixed reproduction systems (McDonald and Linde, 2002), with the best invaders combining high rates of asexual reproduction and rare events of sex (Bazin et al., 2014). Yet, the allelic redistribution resulting from a sexual reproduction event could have an even stronger impact on resistance durability when the pathogen is diploid.

To study resistance durability and evolutionary forces shaping the system, the understanding of the evolution of gene and virulence allele frequencies is needed. Coupling epidemiology and population genetics provides insights on both short and long time scales. It allows in particular detailed analyses of transition periods (Day and Proulx, 2004; Day and Gandon, 2007; Bolker et al., 2010), through variables like the pathogen population size, affecting both the disease incidence in epidemiology and the impact of genetic drift in population genetics (McDonald, 2004). This approach is also crucial for highlighting transient effects like evolutionary rescue, *i.e.* the genetic adaptation of a population to a new environment, thus preventing its extinction (Martin et al., 2013; Alexander et al., 2014). Evolutionary rescue as a process leading to resistance breakdown has not received consideration so far.

The virulence of pathogens can be associated with a fitness cost on susceptible hosts (Leach et al., 2001; Thrall and Burdon, 2003; Montarry et al., 2010; Laine and Barrès, 2013; Bousset et al., 2018; Nilusmas et al., 2020), sometimes referred to as the cost of pathogenicity (Sacristán and García-Arenal, 2008). This fitness cost has been shown to have a strong impact on resistance durability (Fabre et al., 2012). However, depending on the avirulence gene considered, such a fitness cost is not systematic (see Leach et al., 2001 for a review). Therefore, in the absence of data, it could be more conservative of the risk of breakdown not to consider fitness cost while modelling resistance durability.

In this paper, we aim to broaden our understanding about the impact of the ploidy and the life cycle of pathogens on resistance durability. We used a non-spatialised model coupling population dynamics and population genetics to simulate the evolution of pathogens on susceptible and resistant hosts. We investigated the effects of resistant host deployment and pathogen demography on resistance durability, for a population of diploid and partially clonal pathogens, and compared the results to those obtained for haploid pathogens. Two different life cycles were implemented specifically: with or without host alternation for the sexual reproduction of the pathogen population. We assessed the resistance durability in two steps. First we examined the dynamics of fixation of the virulent allele in the population, and considered in parallel the cases when the pathogen population could go extinct, to highlight evolutionary rescue events. Then we focused on the invasion and resistance breakdown events, and disentangle the relationship between the durability of resistance and the dynamics of virulent populations after the invasion of the resistant plants.

C.2 Model description

C.2.1 Model overview

The model is individual-based, forward-time and non-spatialised, and couples population dynamics and population genetics to study the evolution of a population of pathogens for a succession of generations. It allows us to follow the evolutionary trajectory of different genotypes at the avirulence locus of pathogens through time. We consider a life cycle usually found in temperate pathogen species, which alternate rounds of clonal reproduction with an annual event of sexual reproduction (Agrios, 2005). This model is designed in four variants to represent haploid or diploid pathogens with two distinct life cycles: with or without host alternation for the sexual reproduction (Figure C.1). Without host alternation, the model represents the evolution in time of a population of pathogens on two static host compartments: a resistant compartment (R) and a susceptible compartment (S). Fixed carrying capacities of pathogens, K_R and K_S, are respectively assigned to R and S compartments and represent the maximum amount of pathogens that each compartment can contain. With host alternation, the alternate host compartment (A) is added, where the sexual reproduction takes place. This static compartment is assumed to be sufficiently large and thus with unbounded population size. Note that the life cycle with host alternation for haploid pathogens was added for the sake of comparison but has no real biological meaning, because no haploid pathogen display this life cycle.

C.2.2 Demographic evolution of the pathogen population

C.2.2.1 Reproduction events

Each discrete generation corresponds to a reproduction event, either sexual or asexual. Each year is composed of g non-overlapping generations, with one annual sexual reproduction event followed by a succession of $g-1$ asexual reproduction events. In our simulations, we considered a year composed of $g = 11$ generations. At each reproduction event, parents give way to offspring and the new population is composed exclusively of new individuals. The within-compartment dynamics of the pathogen population are provided by the following equations:

Before each sexual reproduction event, a proportion $Reduct$ of pathogen individuals is picked randomly to form the parental population. We fixed $Reduct = 0.2$ to cope to pathogen life cycles displaying drastic reduction in population size during sexual reproduction which usually takes place in winter.

For the sexual reproduction event itself, the population size is considered constant before and after the reproduction event:

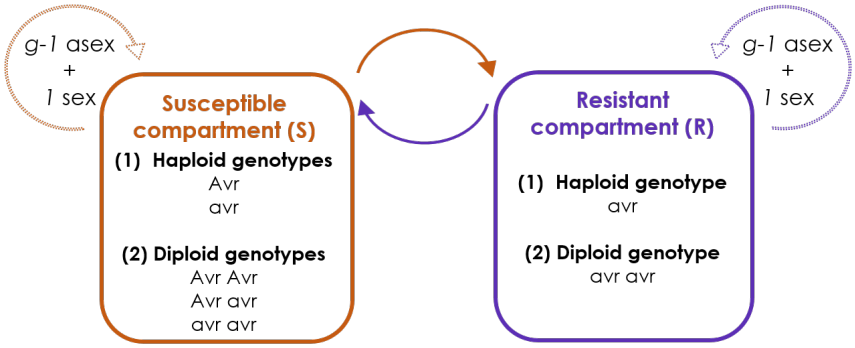
$$N_{n+1} = N_n, \tag{C.1}$$

with N_{n+1} the population size at generation $n+1$ and N_n the population size at generation n. Sexual offspring genotypes are obtained through random mating within the parental population.

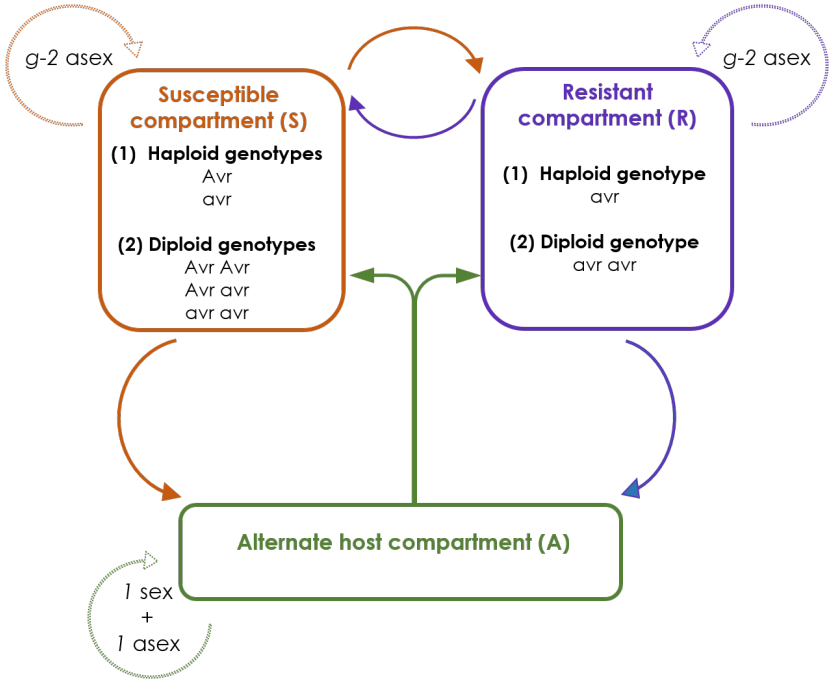
For the asexual reproduction following the sexual reproduction in the A compartment in the life cycle with host alternation, the population growth is exponential, with the following relation:

$$N_{n+1} = r \times N_n, \tag{C.2}$$

with r the growth rate of the pathogen population.



(a) **Without** host alternation



(b) **With** host alternation

Figure C.1: Model representation for two different life cycles: (a) without or (b) with host alternation. g corresponds to the total number of generations (asexual plus sexual) in a year. Dashed arrows represent reproduction events, and solid arrows represent migration events occurring at each generation. $asex$ stands for asexual reproduction events, and sex for sexual reproduction events. avr denotes the virulent recessive allele, and Avr the avirulent dominant allele.

For each asexual reproduction in the R or S compartments, the population growth is logistic, with the following relation:

$$N_{n+1} = N_n + (r-1) \times N_n \times (1 - \frac{N_n}{K}), \tag{C.3}$$

with K the carrying capacity of the compartment (K_R or K_S for R or S compartment respectively). For all clonal reproduction events, offspring genotypes are drawn randomly from the parental population, with replacement, considering the same reproduction rate for all pathogen genotypes.

C.2.2.2 Migration events

A regular two-way migration event takes place each generation before the reproduction event, between individuals evolving in the R and S compartments. The number of migrant individuals is determined by a migration rate (mig) multiplied by the number of individuals on the compartment of origin. Migrant individuals succeed to invade the compartment of arrival, even if the number of individuals on this compartment reached the maximum carrying capacity. The population size on each compartment is restricted to the carrying capacity during reproduction events only, and not during migration events. Thereby, this choice enables the immigration of new pathogens regardless of the size of the population, as it is observed in natural populations for plant pathogens.

For the life cycle with host alternation, the annual sexual reproduction event coincides with the obligate migration of the entire pathogen population to and from the alternate host. The first migration event takes place once every year after $g-2$ asexual reproduction events in the R and S compartments (Figure C.1). For this migration event, an established proportion of individuals $Reduct$ is picked randomly from R and S compartments to migrate to the A compartment. All remaining individuals die in the R and S compartments, because the sexual reproduction is mandatory to complete the life cycle. After the two reproduction events (sexual and asexual) in the A compartment, the second migration event redistributes randomly all individuals from the compartment A to R and S compartments, in proportion to the relative size of R and S compartments (Figure C.1).

C.2.3 Genetic evolution of the pathogen population

To better highlight the effect of drift among other evolutionary forces, we did not consider mutation, that is, there is no change by chance of allelic state. This amounts to study evolution of the pathogen population from standing genetic variation (Barrett et al., 2008). The avirulence gene exists at two possible states: the Avr allele and the avr allele. For haploid pathogens, the Avr allele leads to avirulent individuals surviving only in the S compartment (and in the A compartment in the case of host alternation), while the avr allele leads to virulent ones capable to survive on all compartments without any fitness cost (Leach et al., 2001; Brown, 2015). For diploid pathogens, Avr is dominant and avr is recessive. Thus, individuals with genotypes Avr/Avr, Avr/avr and avr/avr survive with equal fitness in the S and A compartments, while only individuals with the virulent genotype avr/avr survive in the R compartment. Every avirulent individual (Avr for haploids, and Avr/Avr or Avr/avr for diploids) migrating to the R compartment dies before any subsequent migration or reproduction event.

Besides the demographic evolution of pathogen populations, the model describes the evolution of allelic and genotypic frequencies through generations in each compartment. Reproduction events can change allelic and genotypic frequencies. In particular, the annual sexual reproduction is the only event responsible for allele reshuffling in diploid individuals. For haploid pathogens, as only one locus is studied, the sexual reproduction event amounts to asexual reproduction, with differences in the size of the offspring population only.

Resistance durability is evaluated at four steps representing different proportions of virulent individuals in the population: (1) the time of apparition of the first virulent individual on the R compartment; (2) the time of invasion of the R compartment (1‰ of the R compartment occupied); (3) the time of resistance breakdown (1% of the R compartment occupied); and (4) the time of fixation of the virulence (all individuals are virulent, i.e only avr alleles remain). The thresholds of 1‰ and 1% were arbitrarily fixed to correspond to (i) the estab-

lishment of a pathogen population on the R compartment for the invasion and (ii) the detection of the virulent population on the R compartment for the resistance breakdown, respectively.

C.2.4 Implementation of model analyses

The model was implemented in Python (version 3.7; van Rossum, 1995), with the package "simuPOP" (Peng and Kimmel, 2005). The starting point of each replicate simulated was a population of 2,000 individuals in the susceptible compartment. A proportion f_{avr} of virulent alleles was introduced initially in the pathogen population at Hardy-Weinberg equilibrium, as standing genetic variation. For diploid individuals, homozygous avr/avr individuals could therefore be initially present, depending on f_{avr}. All simulations were run with a fixed total carrying capacity for the host population size, $K = K_R + K_S = 100\,000$, but a variable proportion of the size of the R compartment $propR = \frac{K_R}{K}$.

Preliminary analyses were carried out to study demographic and genetic outcomes with varying parameters. These analyses enabled six variables of interest to be identified: the initial frequency of avr allele (f_{avr}), the migration rate (mig), the growth rate (r), the proportion of resistant hosts in the landscape ($propR$), the ploidy ($Ploidy$) and the life cycle ($Cycle$). Statistical analyses were performed on simulations with quantitative input parameters picked randomly from known distributions, resulting into a random simulation design (Table C.1). The same simulation design was run four times, once for each combination of categorical input parameters ($Ploidy$ and $Cycle$). To investigate further the impact of the input parameters on the simulation outcome in specific cases and to present the model results in a more didactic form, a regular simulation design was developed to complement the random design (Table C.1).

This regular simulation design allowed us to present the results in a more conventional form. For both the random and the regular simulation design, simulations were run for each combination of parameters for 100 years (1,100 generations) with 100 replicates. During this period, nearly all replicates reached equilibrium (fixation of one allele or extinction of the population).

A principal component analysis was performed on the data obtained with the random simulation design using R (R Core Team, 2018), on the following output variables: the frequency of extinction of population ($freq\_ext$), the frequency of fixation of the Avr allele in the population ($freq\_fix\_Avr$), the frequency of fixation of the avr allele ($freq\_fix\_avr$) and the generation of fixation of avr (gen_fix_avr). To study the influence of the six input parameters ($Ploidy$, $Cycle$, f_{avr}, mig, r, and $propR$) on the three main output variables selected ($freq\_ext$, $freq\_fix\_avr$, and gen_fix_avr), generalized linear models (GLM) were performed on R. GLM on $freq\_ext$ and $freq\_fix\_avr$ were performed with a Logistic link function, and the GLM on gen_fix_avr was performed with a Gamma link function.

To analyse further the temporal dynamics of avr allele frequency and population size, simulations were run recording population size and allelic states over time. Because these simulations were time- and memory-consuming, they were run on a restricted simulation design with only 24 combinations of parameters (Table C.1). The generation of fixation of the avr allele was thus decomposed into two distinct output variables: the year of invasion of the R compartment and the time elapsed between the invasion and the avr allele fixation in the population. The influence of three parameters ($propR$, $Ploidy$ and $Cycle$) on these two output variables was studied with 1,000 replicates for each combination through 1,100 generations. For these two output variables, GLM were performed with a Gamma link function.

For each general linear model developed, a dominance analysis was performed with the R package "dominanceanalysis" (Bustos Navarrete and Coutinho Soares, 2020) to compare the relative importance of the input parameters on the five output variables described. Estimated general dominance were calculated using bootstrap average values with the corresponding standard errors for each predictor with 100 bootstrap resamples, with McFadden's indices (McFadden, 1974).

Calculations of a growth rate threshold r_0 were carried out on Python for several parameter combinations. This value determines the growth rate below which the population goes extinct before the end of the simulation if there are no virulent individuals, therefore if the R compartment remains empty.

Table C.1: Input parameters and their range of variations for the three simulation designs. For both the random and regular simulation designs, 100 replicates were run for each combination to analyse the equilibrium reached by the population of pathogens after 1,100 generations. For the restricted simulation design, 1,000 replicates were run for each combination and the allele frequencies and population sizes were monitored through all 1,100 generations simulated.

Variable	Description	Random design		Regular design	Restricted design
		Distribution	Interval		
f_{avr}	Initial frequency of the avr allele in the pathogen's population	Log-uniform	[log(0.0005), log(0.3)]	**6 levels**: 0.0005; 0.002; 0.01; 0.05; 0.15; 0.3	0.02
mig	Migration rate between R and S compartments	Log-uniform	[log(0.001), log(0.2)]	**3 levels**: 0.05; 0.1; 0.2	**2 levels**: 0.001; 0.05
r	Growth rate of the pathogen	Uniform	[1.1, 2]	**10 levels** : 1.1; 1.2; 1.3; 1.4; 1.5; 1.6; 1.7; 1.8; 1.9; 2.0	1.5
$propR$	Proportion of resistant hosts in the landscape	Uniform	]0, 1[	**9 levels**: 0.05; 0.1; 0.2; 0.35; 0.5; 0.65; 0.8; 0.9; 0.95	**3 levels**: 0.1; 0.5; 0.9
$Ploidy$	Ploidy of the pathogen	-	Haploid or Diploid	**2 levels**: Haploid; Diploid	**2 levels**: Haploid; Diploid
$Cycle$	Life cycle of the pathogen	-	Without or with host alternation	**2 levels**: Without host alternation; With host alternation	**2 levels**: Without host alternation; With host alternation

C.3 Results

C.3.1 Model behaviour

Since the model leads to stochastic outputs, we first analysed model behaviour in order to identify sound output variables. We visualised both population size and allele frequency dynamics through generations. The trajectory of each simulation either lead to the extinction of the entire pathogen population or to the fixation of one allele, provided that simulations last long enough. An example of such model behaviour is illustrated in Figure C.9 with four replicates, assuming diploid pathogens with host alternation. In this example, population sizes display cyclical dynamics due to the annual migration event on the A compartment. Three out of five replicates lead to population extinctions, while in the two other replicates, some pathogen individuals succeed to invade the R compartment after the initial phase of population dynamics collapse, leading to the fixation of the avr allele. These two dynamics with the survival of the population following a genetic adaptation to harsh environment illustrates evolutionary rescue. Interestingly, all replicates succeed to invade the R compartment at some time, but - because of host alternation - the annual redistribution of individuals breaks the invasion dynamics of the R compartment. Therefore, invasion does not necessarily lead to avr fixation.

In the following, we will summarise simulation results with four output variables, computed over replicates: the frequency of extinction, $freq\_ext$; the frequency of fixation of Avr or avr allele, $freq\_fix\_Avr$ or $freq\_fix\_avr$, respectively; and the generation of fixation of avr allele, gen_fix_avr. The later output variable provides insights on the durability of resistance.

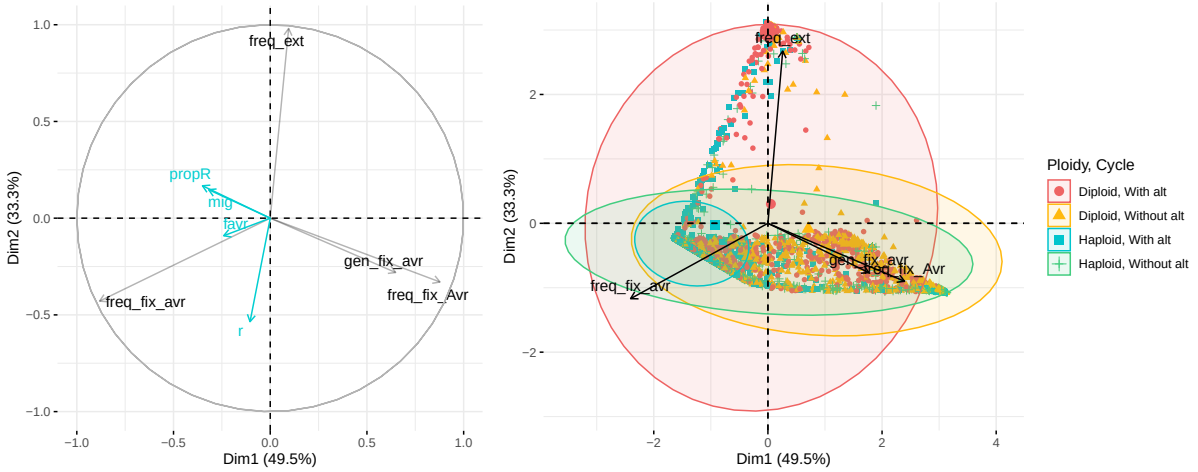
C.3.2 Sensitivity analyses

To identify the most significant parameters on the different output variables, we conducted a PCA analysis, general linear models and dominance analyses.

The PCA analysis was performed on the four output variables, with the first and second axes accounting for 49.5% and 33.3% of the total variability respectively (Figure C.2). The most influential parameters of interest on the output variables were the growth rate r, negatively correlated with the frequency of extinction of population $freq\_ext$. The initial frequency of avr alleles in the population f_{avr} was positively correlated with the frequency of fixation of the avr allele $freq\_fix\_avr$. The migration rate mig and the proportion of resistant hosts in the landscape $propR$ were negatively correlated with both the frequency of fixation of the Avr allele $freq\_fix\_Avr$, and the generation of fixation of the avr allele gen_fix_avr. The qualitative input parameters ($Ploidy$ and $Cycle$) were studied by representing each of the combinations of parameters of the random simulation design colored according to its ploidy and life cycle (Figure C.2.b). This PCA highlights a higher frequency of extinctions of population for diploids with host alternation. Moreover, simulations without host alternation lead to higher frequencies of fixation of Avr, and longer times to avr fixation. The ellipses also illustrate that haploid individuals with host alternation lead to less variable outcomes and to higher frequencies of fixation of avr.

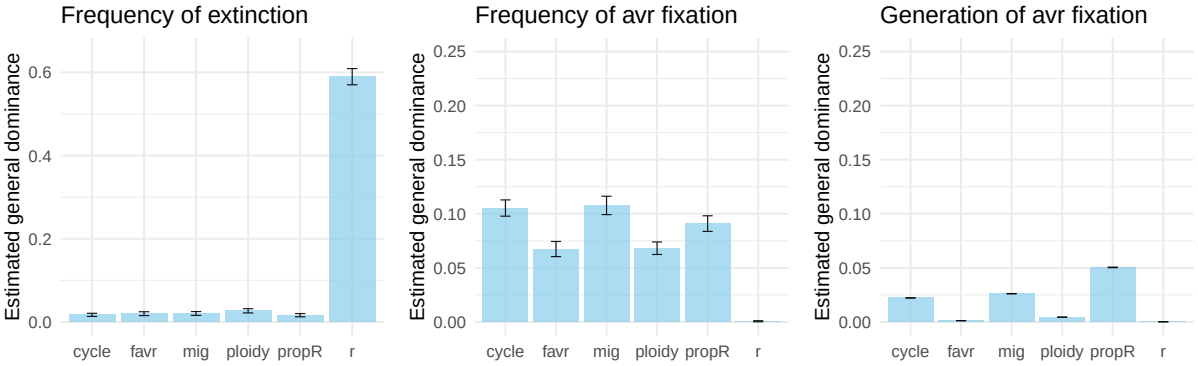
Dominance analyses highlight the high impact of r on $freq\_ext$ (Figure C.3.a), which is confirmed by the analysis of Sobol' indices (Figure C.8). For $freq\_fix\_avr$ and gen_fix_avr, the influence of model parameters is more balanced with a lower contribution of the input parameters on gen_fix_avr (Figure C.3.b, c).

Overall, this analysis points out that $freq\_ext$ and $freq\_fix\_avr$ are relatively well explained while gen_fix_avr is more stochastic.



(a) Correlation circle (b) Individual representation

Figure C.2: Principal component analysis on four output variables: $freq\_ext$, $freq\_fix\_Avr$, $freq\_fix\_avr$, and gen_fix_avr. Two results are displayed: (a) the correlation circle on four output variables. The quantitative parameters f_{avr}, mig, r, and $propR$ are represented informatively in blue on the correlation circle and do not contribute to the variability; (b) the individual representation of simulations which represents the influence of the pathogen ploidy and life cycle. Each point represents a different combination of parameters with 100 replicates. Ellipses correspond to the 95% multivariate distribution.



(a) (b) (c)

Figure C.3: Estimated general dominance of each predictor calculated from general linear models applied to three output variables of the random simulation design: the frequency of extinction of population, the frequency of fixation of the avr allele among replicates with surviving populations, and the generation of fixation of the avr allele. For each predictor the general dominance was estimated from bootstrap average values with the corresponding standard error for 100 bootstrap resamples.

C.3.3 Patterns of virulence fixation

Three different and exclusive equilibria are observed: the extinction of the pathogen population, the fixation of the avr allele and the fixation of the Avr allele. The frequencies of these outputs among replicates are represented depending on f_{avr}, r, $propR$ and $Cycle$, for haploids and diploids (Figures C.4 and C.12). For both ploidy levels, there is an increase in the frequency of fixation of the avr allele with the increase in f_{avr} and r. This representation also highlights the existence of a growth rate threshold r_0 above which there is fixation of either the avr allele or the Avr allele, and below which there is instead either fixation of the avr allele or extinction of the population. In other words, for a growth rate below r_0 the pathogen population only survives when virulent individuals invade the R compartment, which corresponds to evolutionary rescue. Evolutionary rescue is particularly noticeable for the life cycle with host alternation because in this case, r_0 increases with $propR$.

Above r_0, we observe a gradient between the predominant fixation of the two alleles depending of f_{avr}, with slightly different patterns influenced by $propR$, $Cycle$ and r for diploids (Figure C.4). The influence of the life cycle on the fixation pattern is the most noticeable for low values of $propR$ and r. The frequency of fixation of the avr allele appears maximal for intermediate values of $propR$.

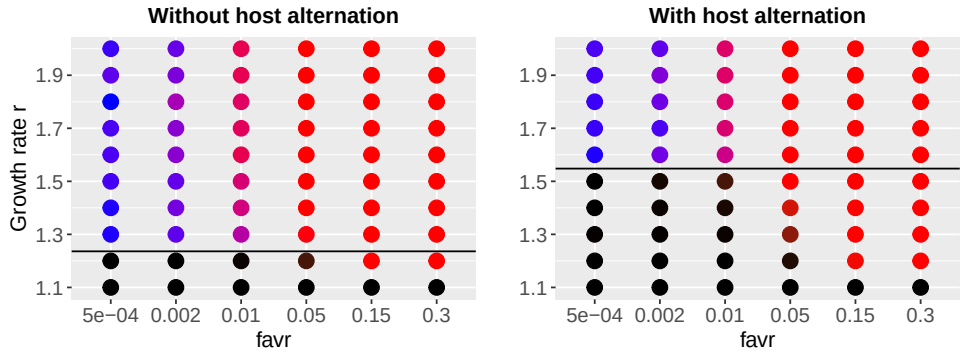
To examine further the influence of $propR$ on the probability of fixation of the avr allele, we plotted the evolution of the frequency of fixation of the avr allele among all replicates of the regular simulation design depending on $propR$ and r for a fixed value of f_{avr} (Figure C.5). For haploid individuals with host alternation, the frequency of fixation of the avr allele increases very slightly with $propR$. For haploids without host alternation, a plateau is observed for intermediate values of $propR$. For diploids, the distribution is shifted with a peak of maximal proportion of avr fixation for a slightly lower value of $propR$.

C.3.4 Variations in the speed of virulence spread

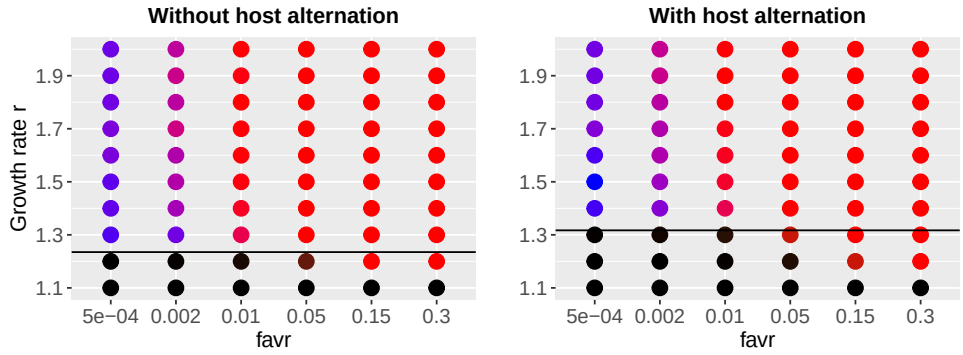
To analyse in more details the dynamics of virulence spread, we examine two time points, reflecting two measures of resistance durability: the invasion of the R compartment and the resistance breakdown event. Invasion and resistance breakdown were defined as the first year when at least 1‰ and 1% of the resistant compartment were occupied by pathogens, respectively. Distributions of these two measures of resistance durability were plotted for three values of $propR$, with and without host alternation, only for the replicates for which we observed eventually a fixation of the avr allele. To broaden the picture, we monitored also the evolutionary trajectory of the avr allele from the invasion of the R compartment. The dynamics of invasion is mainly driven by the ploidy level and the dynamics of virulence spread is mainly driven by $propR$ (Figure C.10).

For haploid individuals resistance breakdown occur very rapidly: during the first or second year of simulation, regardless of the life cycle (figure not shown). There is a small delay in the time of the resistance breakdown with the increase in $propR$, especially without host alternation.

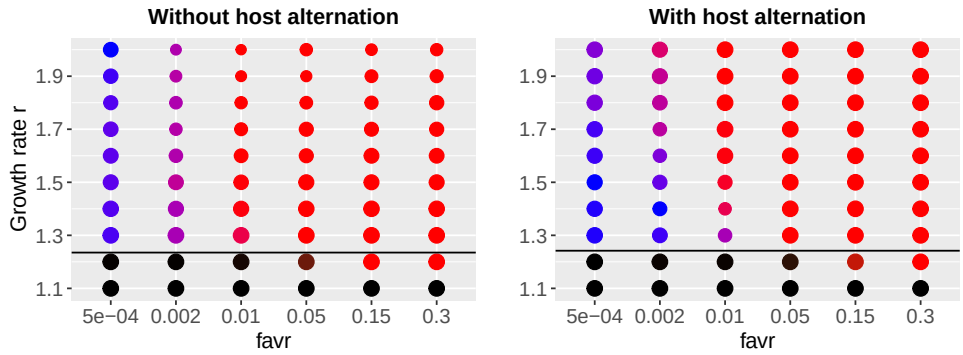
For diploid individuals, we observed longer periods before invasion and resistance breakdown and higher kurtosis. Assuming a strong migration rate, there are few differences between life cycles on the time of invasion. Both life cycles display an increase in kurtosis that goes hand in hand with the increase in $propR$. Without host alternation, distributions of the year of resistance breakdown and invasion time are similar, but with a one-year lag. Conversely, with host alternation, there are strong changes in the distributions that flatten out when considering the year of resistance breakdown (Figure C.6). Assuming a low migration rate (i.e. for telluric pathogens),



(a) $propR = 0.9$



(b) $propR = 0.5$



(c) $propR = 0.1$

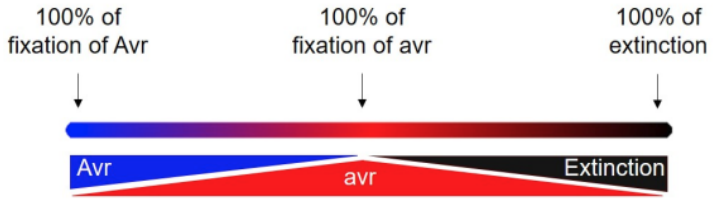


Figure C.4: For diploid pathogens, representation of the frequencies of population extinction or fixation of alleles Avr or avr for each combination of four parameters: f_{avr}, r, $propR$ and $cycle$, with $mig = 0.05$. On each graph the black line corresponds to the calculated value of the growth rate threshold r_0 below which the population dies if it does not expand to the R compartment. The surface of each plotted result is proportional to the number of simulations, among the 100 replicates, for which an equilibrium was reached at the end of the 1,100 generations simulated.

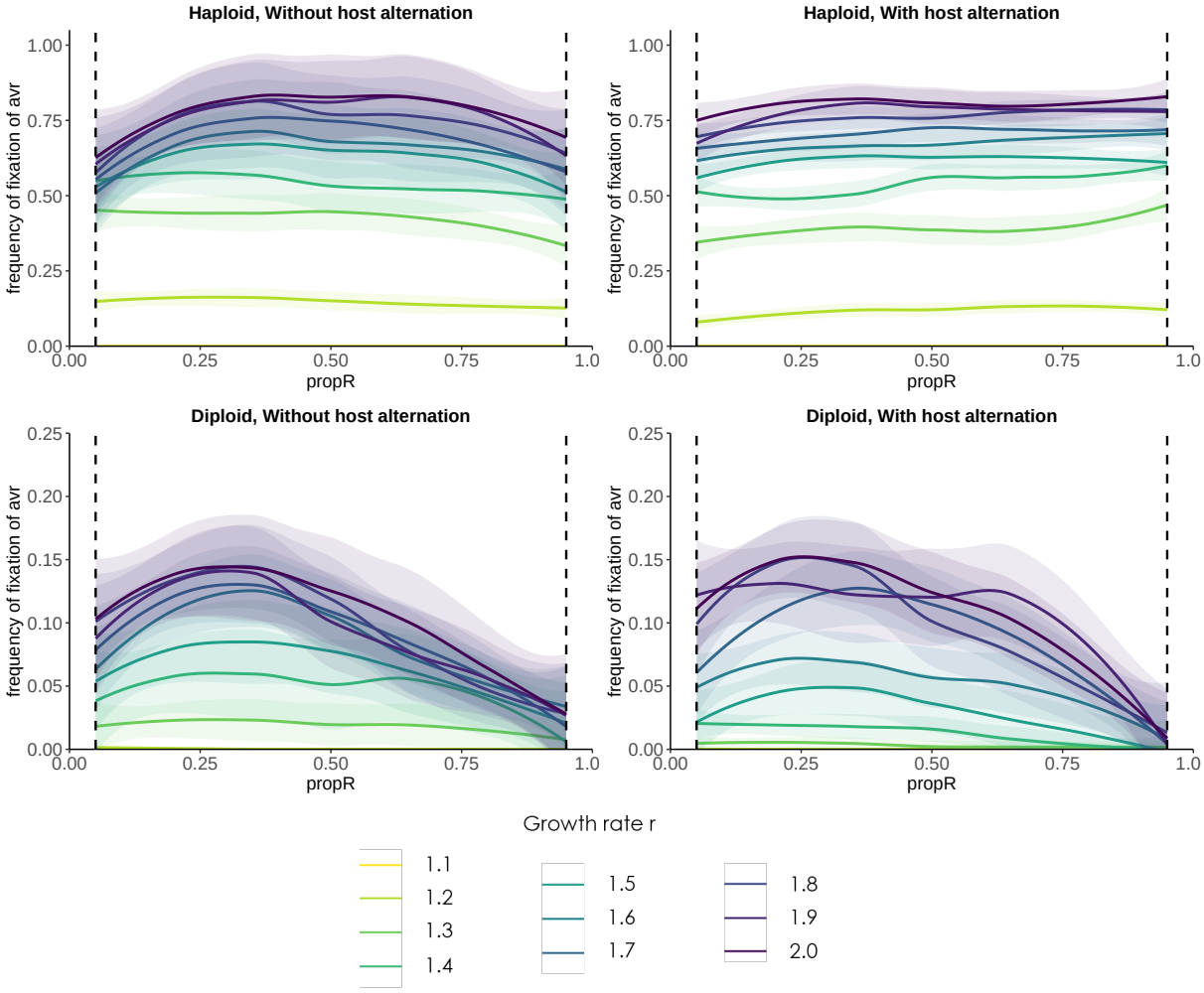
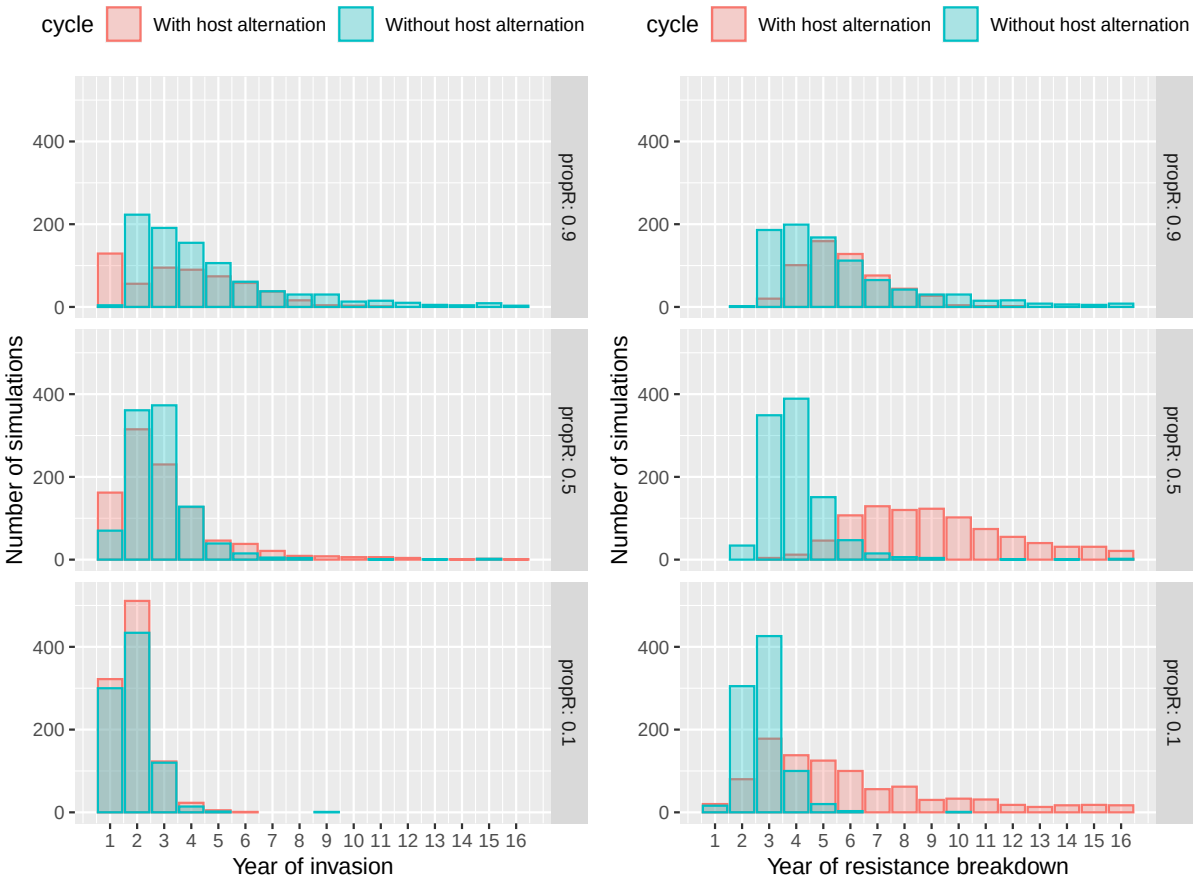


Figure C.5: Evolution of the frequency of fixation of the avr allele depending on $propR$ for r varying between 1.1 and 2.0, with $f_{avr} = 0.0005$. Simulations were performed without and with host alternation, for haploid and diploid individuals, with 100 replicates for each combination of parameters. The plotted results correspond to the local regression on the frequency of fixation of the avr allele with the 95% confidence intervals associated with each regression. The vertical dotted lines correspond to the bounds of simulated values of $propR$ for this regular experimental design.

distributions of the year of invasion and resistance breakdown remain unchanged with host alternation, while these distributions flatten out considerably without host alternation (Figure C.11). Note that in both migration regimes, with host alternation we observe a bimodal distribution of invasion year for $propR = 0.9$, with many invasion events in the first year of simulations. Early invasion events result from the initial redistribution of pathogen individuals following sexual reproduction on the alternate host: it is all the more likely that a virulent individual arrives on the R compartment the more predominant it is in the landscape.

In a last step, the evolution of the frequency of the avr allele in the population was studied along with the evolution of the population sizes through generations, from the invasion (Figure C.7, see Figure C.13 for haploids). For both life cycles, we observe an increase in the speed of fixation of the avr allele with the increase in $propR$. The median speed of fixation is higher with host alternation for haploids, and highly dependant of $propR$ without host alternation for diploids (Figure C.10). We also observe differences in stochasticity levels



(a) Year of invasion

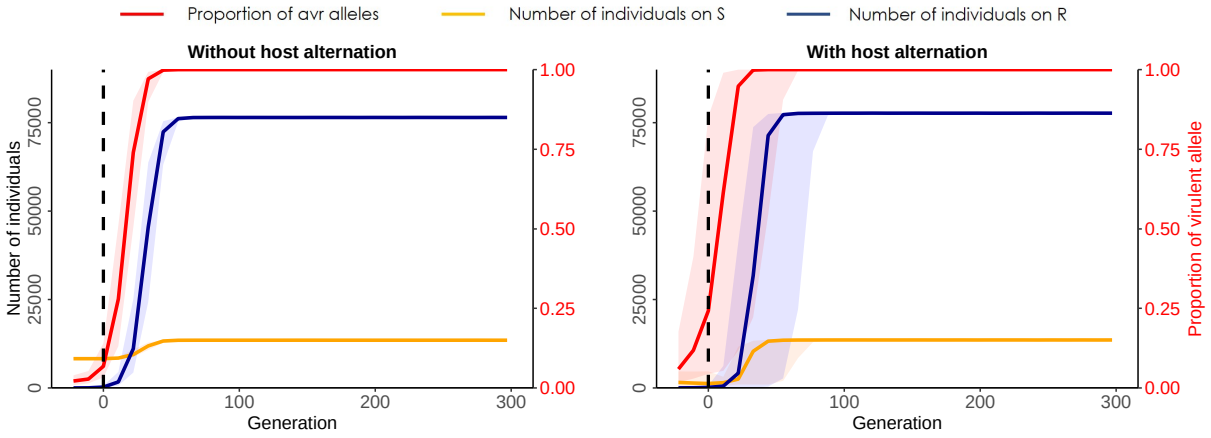
(b) Year of resistance breakdown

Figure C.6: Histograms of (a) the year of invasion and (b) the year of resistance breakdown depending on the life cycle of the diploid pathogen, for three values of $propR$. Simulations were performed with $f_{avr} = 0.02$, $mig = 0.05$, and $r = 1.5$. The plotted results were obtained from the restricted simulation design, and correspond to all simulations among the 1,000 replicates per combinations for which we observed a fixation of the avr allele in the population.

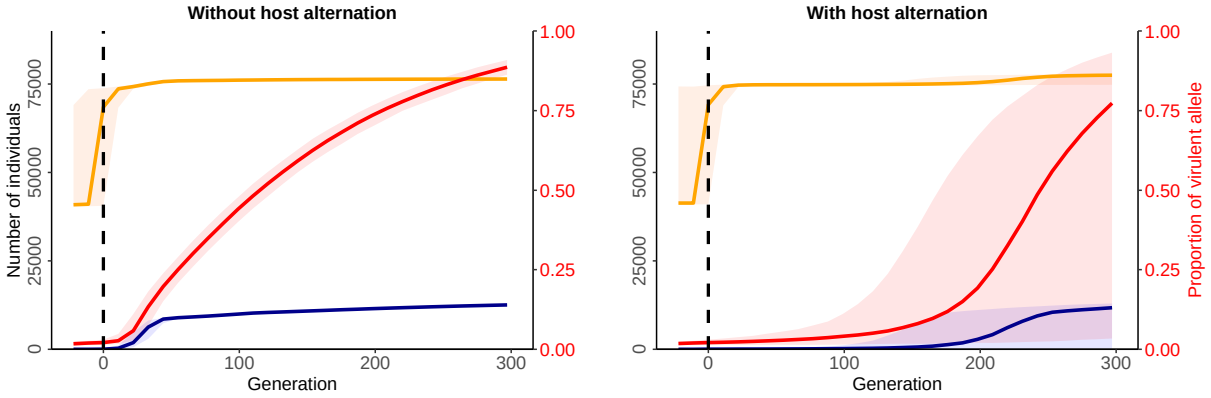
depending on the ploidy and the life cycle. For haploids, the dynamics of virulence fixation is almost deterministic. For diploids, the dynamics are more variable, with a highly stochastic behaviour for the life cycle with host alternation. Moreover, the results of GLM, the dominance analysis and the comparison of both figures show that independently of $propR$ and of the life cycle, the increase in the proportion of avr allele is faster for haploid than for diploid individuals.

Focusing on diploid simulations, Figure C.7.b highlights the existence of an evolutionary rescue effect, for the life cycle with host alternation and a high value of $propR$. The median population size decreases through time and almost reached 0 before the 20th generation following the invasion, when an increase in the proportion of avr alleles lead to an increase in the population size in the R compartment, followed by an increase in the population size of the S compartment, preventing the extinction of the population.

Interestingly, the speed of invasion is mainly driven by the ploidy, while the speed of fixation of the avr allele from the invasion is mainly driven by the landscape composition $propR$ (Figure C.10).



(a) $propR = 0.9$



(b) $propR = 0.1$

Figure C.7: For diploid pathogens, evolution of the population size in the R and S compartments (on the left scale) and of the frequency or avr alleles in the S compartment (on the right scale) through generations. Simulations were performed without and with host alternation, for two values of $propR$: (a) 0.1 and (b) 0.9, with $f_{avr} = 0.02$, $mig = 0.05$, and $r = 1.5$. For each simulation, generation 0 corresponds to the generation at which the invasion occurred. For each combination of parameters, simulations were performed on 1,000 replicates. The plotted results correspond to the median results (frequency of avr alleles or population size) for all simulations among the 1,000 replicates for which we observed a fixation of the avr allele in the population. Coloured intervals correspond to the 95% confidence intervals.

C.4 Discussion

C.4.1 Deep impact of the ploidy on resistance durability

Lof et al. (2017) demonstrated that the pre-existence of virulent alleles in the pathogen population could greatly diminish resistance durability. In the present study, we varied the initial frequency of avr alleles in the population but focused only on cases where this allele was already present at the beginning of the simulation, which corresponds to standing genetic variation (Barrett et al., 2008; Alexander et al., 2014). Our results illustrated a strong positive relationship between the initial frequency of virulent alleles and the probability of invasion and resistance breakdown. For haploid individuals, we found no stochasticity in the time of occurrence of the invasion, which occurred in the first year of the simulation for all replicates. Thus, for simulations with hap-

loid pathogens, almost as soon as one virulent individual invaded the resistant compartment, it was selected and the resistance breakdown occurred. This result explains why a lot of models on haploid individuals focus on the probability of apparition of the first virulent individual, in particular by mutation (Fabre et al., 2015; Papaïx et al., 2015). Our results for simulations with haploid pathogens also highlighted low stochasticity in the increase in the proportion of virulent alleles in the population after the invasion. The results obtained with haploids were consistent with previous studies on resistance durability (Pacilly et al., 2018), which permitted us to consider this model as the reference model, in order to study the influence of the diploid state on the system dynamics. Diploid individuals, however, display strongly different evolutionary trajectories. In particular, we observed higher stochasticities in the evolution of the virulent allele frequency, both before and after the resistance breakdown. This is mainly caused by the recessivity of the avr allele, according to the gene-for-gene model. Before the invasion, the avr allele is rare and mostly at the heterozygous state, hence leading to phenotypically avirulent individuals. Therefore, the avr allele is poorly selected, and variations in its frequency are mostly driven by genetic drift, which induces high stochasticity among replicates. This effect should be strengthened in species with small effective population sizes, such as in cyst nematodes (Gracianne et al., 2016; Montarry et al., 2019).

As a consequence, we observed lower frequencies of avr fixation and higher extinction rates for diploid individuals, independently of the life cycle and the host deployment strategy. Moreover, simulations with diploid pathogens resulted in lower speeds of fixation of the virulent allele, that is, higher resistance durability. Because of the heterozygous Avr/avr state, avr alleles are not necessarily selected and their presence in the population does not inevitably lead to an immediate resistance breakdown, as observed for haploid individuals. Thus, besides its impact on the stochasticity of the results, the vulnerability of the virulent allele at the heterozygous state is also responsible for an increase in resistance durability. The impact of the landscape composition on resistance durability also differs with the ploidy. Consistently with the work of Van den Bosch and Gilligan (2003) and Pietravalle et al. (2006), for haploid individuals the increase in $propR$ leads to a strong increase in the speed of fixation of the avr allele, thus decreasing the resistance durability. In all cases except for haploids with host alternation, this result was accompanied in the present study by a maximum frequency of avr fixation for intermediate values of $propR$. This non-linear relationship is similar to the one highlighted for haploids by Pacilly et al. (2018), and is caused by two distinct mechanisms. At low proportions of resistant hosts in the landscape, the selective pressure on the avr allele is sufficiently low to limit the increase in the virulence in the pathogen population. At high values of $propR$, the opposition between selection and drift is magnified: on one hand the selective pressure is high and imposes a rapid pace of adaptation; on the other hand the small size of the S compartment increases genetic drift and the risk of extinction of the avr allele, provided that the R compartment is not invaded. Hence, in most cases the virulent allele is lost if it does not spread quickly enough in the population: either $r > r_0$ which lead to a fixation of the avirulent Avr allele, either $r < r_0$ and the population goes extinct. Therefore, it would be possible to reduce the probability of invasion for diploid pathogens with either very low or very high proportions of resistant hosts in the landscape. However the increase in the proportion of resistant hosts is at the risk of weaker resistance durability: if the resistance breakdown occurs, it occurs more rapidly. For haploid pathogens with host alternation, we observed an almost constant and slightly increasing frequency of fixation of avr with the increase in $propR$. In this case and contrary to diploids with host alternation, the increase in the proportion of avr alleles on the resistant host is not counteracted by the allele reshuffling during the sexual reproduction event on the alternate host. For diploids, this allelic reshuffling

causes a rise in the number of phenotypically avirulent Avr/avr heterozygous individuals, which will die if the redistribution following the sexual reproduction lead them on the resistant host. This can result in a drastic drop in the avr allele proportion while for haploids, the proportion of resistant hosts in the landscape does not increase the mortality rate of individuals carrying the virulent allele, because these haploid individuals are necessarily surviving on the resistant host.

C.4.2 Life cycles impose different selection regimes and lead to contrasted evolutionary trajectories

The two different life cycles considered in this study - with or without host alternation - can be assimilated to hard and soft selection respectively (Wallace, 1975; Christiansen, 1975). Soft selection is expected to protect polymorphims, and hence promote local adaptation, while hard selection resembles an all or nothing game, that is to adapt to the encountered environment or to perish. Here host alternation can lead to faster evolution of allelic frequencies, with higher speeds of virulence fixation, especially for high values of $propR$. Without host alternation on the contrary, the evolution of virulence alleles are buffered, which result in more constrained dynamics. The increase in gene flow resulting from host alternation limits natural selection and local adaptation (Lenormand, 2002), especially because of the dispersal of non-adapted individuals on resistant hosts. The life cycle with host alternation is thus characterized by higher probabilities of population extinctions, and strong dependency of the growth rate threshold r_0 and the landscape composition $propR$. For diploids with host alternation, contrary to the local adaptation on each compartment without host alternation, the forced allelic reshuffling on the alternate host is responsible for the increase in the number of Avr/avr heterozygous individuals. Because the avr allele is recessive, a large proportion of these newly-produced individuals die from the redistribution on the resistant compartment following the sexual reproduction. Noticeably, the reduction in virulence fixation at high proportions of resistant host discussed above hence results from two mechanisms in diploids: impediment of local adaptation without host alternation or increase in selective pressure with host alternation.

For diploid individuals, we also observed contrasting patterns of variation in the evolution of the avr allele frequency before and after the invasion, depending on the life cycle. The time of invasion is more stochastic without host alternation, while the speed of increase in the avr allele frequency from the invasion is far more stochastic with host alternation. The first step relies essentially on the probability of encounter between a virulent individual and a resistant host. Without host alternation the encounter is restrained to the probability that a virulent individual migrates from the susceptible to the resistant host during the vegetative season. The host alternation reinforces gene-flow, with the annual migration event that redistributes pathogen individuals among all host plants, thereby favoring the encounter. Interestingly, in the case of host alternation, early infections of resistant host (invasion) does not readily translate into population establishment on that host (disease outbreak leading to resistance breakdown). At the end of the vegetative season and initial invasion, the sexual reproduction on alternate host reshuffles allele frequencies, and thus breaks virulent (homozygous) individuals into mostly avirulent (heterozygous) individuals. These up and down selection phases amplify the effect of genetic drift and lead to nearly chaotic evolutionary trajectories, resulting in a resistance durability all the more difficult to predict. Without host alternation, virulent individuals mate with each others and the homozygous state is sheltered, which results in a strict time lag between initial invasion and population outbreak. Overall

our model is in accordance with the framework proposed by McDonald and Linde (2002) which highlights the importance of gene flow as an impediment to resistance durability. Our analysis completes this framework, taking into account the variation in life cycles.

The life cycle also plays a role in the frequency of observation of evolutionary rescue effects. Carolan et al. (2017) highlighted the impact of the growth rate of the pathogen on resistance durability, by presenting the limitation of the growth rate as a mean to increase resistance durability. In accordance with this study, we displayed a growth rate threshold r_0 below which the pathogen population goes extinct if it does not invade the resistant compartment. Hence, for a growth rate below r_0, the genetic adaptation of the pathogen population is the only way for the population to survive, which is a classical example of evolutionary rescue. In the current study, r_0, and thus the observation range of evolutionary rescue, is highly dependent on the life cycle. Without host alternation, the redistribution of individuals between compartments and the mortality rate is limited, which leads to a quite low r_0, independently of the proportion of resistant hosts in the landscape. With host alternation, however, the redistribution event occurring each year from the alternate host to the S and R compartments leads to a strong dependence of r_0 on the landscape composition, with an increase in the observation range of evolutionary rescue with the proportion of resistant hosts.

C.5 Conclusion

Short-term epidemiological control is predicted to be optimal for a landscape composed of a high proportion of resistant hosts in a low degree of spatial aggregation (Holt and Chancellor, 1999; Skelsey et al., 2010; Papaïx et al., 2014, 2018). However other authors also highlighted that optimal resistance durability could be obtained by reducing the proportion of resistant hosts (Pink and Puddephat, 1999; Pietravalle et al., 2006; Fabre et al., 2012; Papaïx et al., 2018), thus minimising the selection pressure on the pathogen population. In the current study, the minimisation of the probability of fixation of the virulent allele for a diploid pathogen population was obtained either at very low or very high proportions of resistant hosts in the landscape. In cases where the population does not go extinct and the virulent allele increases in proportion, however, the proportion of resistant hosts in the landscape strongly impacts the speed of increase, and thus the resistance durability. Consistently with Van den Bosch and Gilligan (2003), we displayed that for a diploid pathogen population with standing genetic variation, the increase in the proportion of resistant hosts decreases resistance durability. In particular, with host alternation both the invasion and the fixation of the virulent allele in the population can occur very quickly. However, in such a case, the evolutionary trajectory of the virulent allele is all the more stochastic and durability is thus difficult to predict. Without host alternation (i.e. for the majority of pathogen species) early detection and population control measures would increase resistance durability. However such prophylactic measures are made all the more difficult by the strong unpredictability of the invasion date. For the few species with host alternation, a massive intervention could durably control a population of pathogens, such as the eradication of the alternate host species *Berberis vulgaris* for wheat stem rust control (Peterson, 2018). Overall, the high stochasticity of evolutionary trajectory impedes accurate forecasts of resistance durability for diploid organisms.

C.6 Perspectives

In the current study, we considered a single qualitative resistance gene. The combination of several resistance genes is often studied and deployed to increase resistance durability (Djian-Caporalino et al., 2014; Mundt, 2014; Djidjou-Demasse et al., 2017; Rimbaud et al., 2018). These combinations of resistances can occur at the plant scale with multiple resistance genes (pyramiding) inside one host genotype, or at the landscape scale with multiple resistance deployments in time or space (Mundt, 2002; Van den Bosch and Gilligan, 2003). Some resistant cultivars progressively introduced in the landscape are composed of different combinations of qualitative resistance genes, resulting in an evolving selective pressure through time (Goyeau and Lannou, 2011). Building on our results, we can extrapolate on the impact of the ploidy and the life cycle on resistance durability for these different strategies of deployment. Hence, rotating cultures with different resistances would amount to force gene flow, favoring the encounter of pathogen individuals with new hosts without an actual migration. This would be comparable to the hard selection regime observed with host alternation, and we expect similar results. With the pyramiding of several resistance genes in same host, meanwhile, we would expect a higher short term efficiency than with a single resistance, but with a higher risk of rapid resistance breakdown by multi-virulent individuals due to the inducing of a strong selective pressure. This may especially be true for pathogens with host alternation because of the increased probability of mating between pathogens with different virulent profiles when they meet on the alternate host.

Data accessibility

Python simulation code and results, as well as R script for result analyses are available on Zenodo repository: (DOI:10.5281/zenodo.4892587).

Acknowledgements

We warmly thank Josselin Montarry, Lydia Bousset, Jean-Paul Soularue, Frédéric Grognard, Bénédicte Fabre, Clémentine Louet, Pascal Frey, and Cyril Dutech for constructive comments on previous versions of the manuscript. We thank Frédéric Fabre for insightful discussions on sensibility analyses. We thank AgroParisTech for support. We also thank Hirohisa Kishino, Loup Rimbaud and one anonymous reviewer for detailed comments and suggestions which helped improve the manuscript. A previous version of this article has been peer-reviewed and recommended by Peer Community In Evolutionary Biology (DOI:10.24072/pci.evolbiol.100131).

Funding

This work was supported by grants from the French National Research Agency (ANR-18-CE32-0001, Clonix2D project; ANR-11-LABX-0002-01, Cluster of Excellence ARBRE). Méline Saubin was supported by a PhD fellowship from INRAE and the French National Research Agency (ANR-18-CE32-0001, Clonix2D project).

Conflict of interest disclosure

The authors of this article declare that they have no financial conflict of interest with the content of this article.

C.A Appendix

C.A.1 Sobol' indices

Sensitivity analyses were performed with the calculation of Sobol' indices (first-order, second-order and total-order) with the R package "sensitivity" (Iooss et al., 2021). Sobol' indices were calculated to study the importance of each of the six parameters of interest on the output variable $freq\_ext$ only (Figure C.8). These calculations were based on the results issued from the random simulation design.

For the four remaining outputs ($freq\_fix$, gen_fix, the year of occurrence of the invasion, and the time elapsed between the invasion and the fixation of the avr allele), only the simulations not leading to extinction were retained for the sensitivity analyses. Thus, the input combinations of parameters retained depended strongly on the results of the output variable $freq\_ext$, the independence hypothesis of the input parameters were then not verified and Sobol' indices could not be calculated for these four remaining output variables. Further analyses would be necessary to disentangle the effect of each parameter of interest on these remaining output variables.

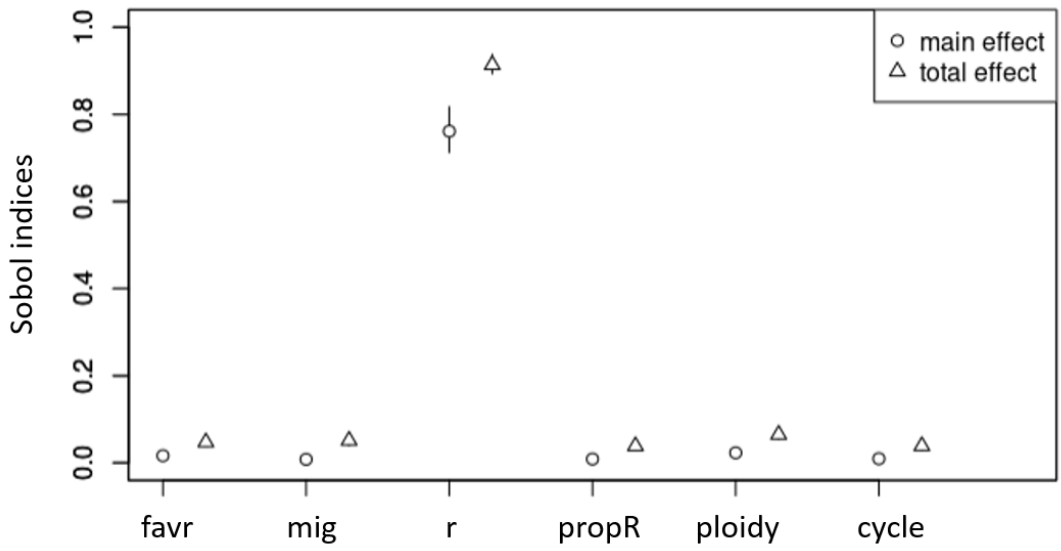


Figure C.8: Sobol' indices to evaluate the influence of six variables of interest on the frequency of extinctions among simulations. Main effect correspond to first-order Sobol' indices, and total effect correspond to total-order Sobol' indices.

C.a.2 Supplementary figures

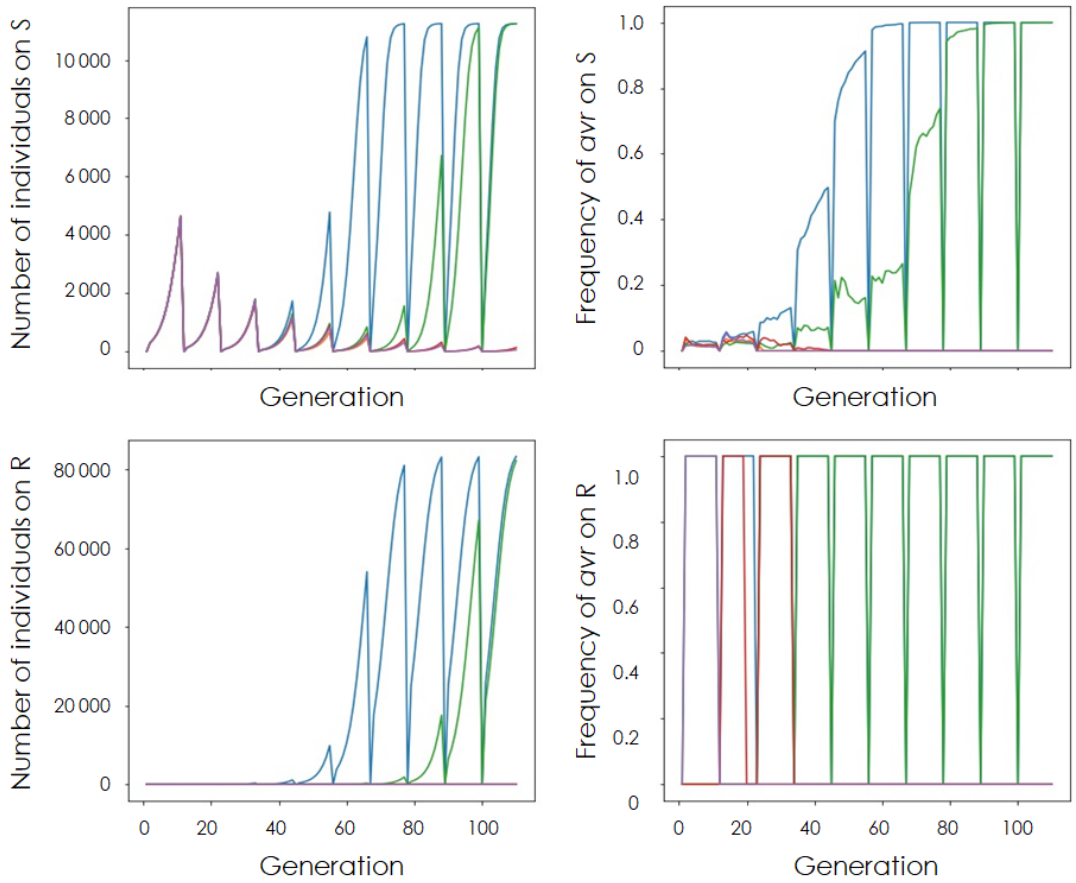


Figure C.9: Example of the simulated populations demographic (on the left) and virulent allele frequency (on the right) dynamics through time on S and R compartments. The model was run for four replicates with diploid individuals and host alternation, $propR = 0.9$, $r = 1.5$, $f_{avr} = 0.025$, and $mig = 0.05$. Each color represents a distinct replicate.

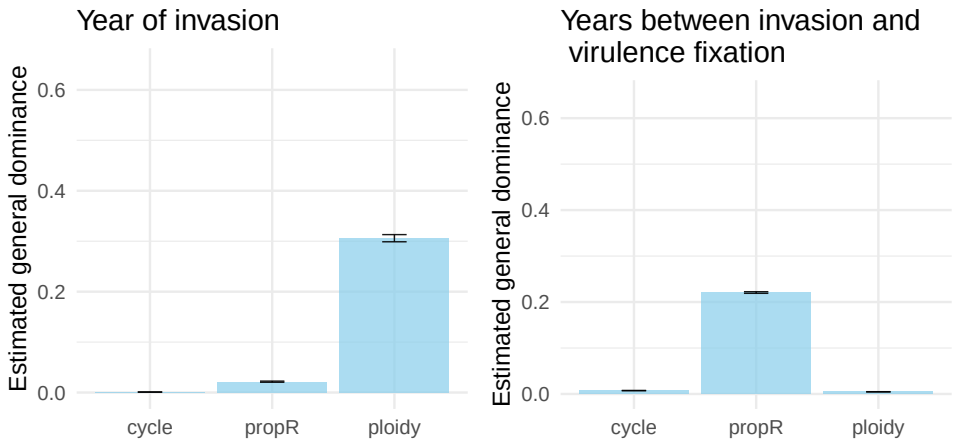
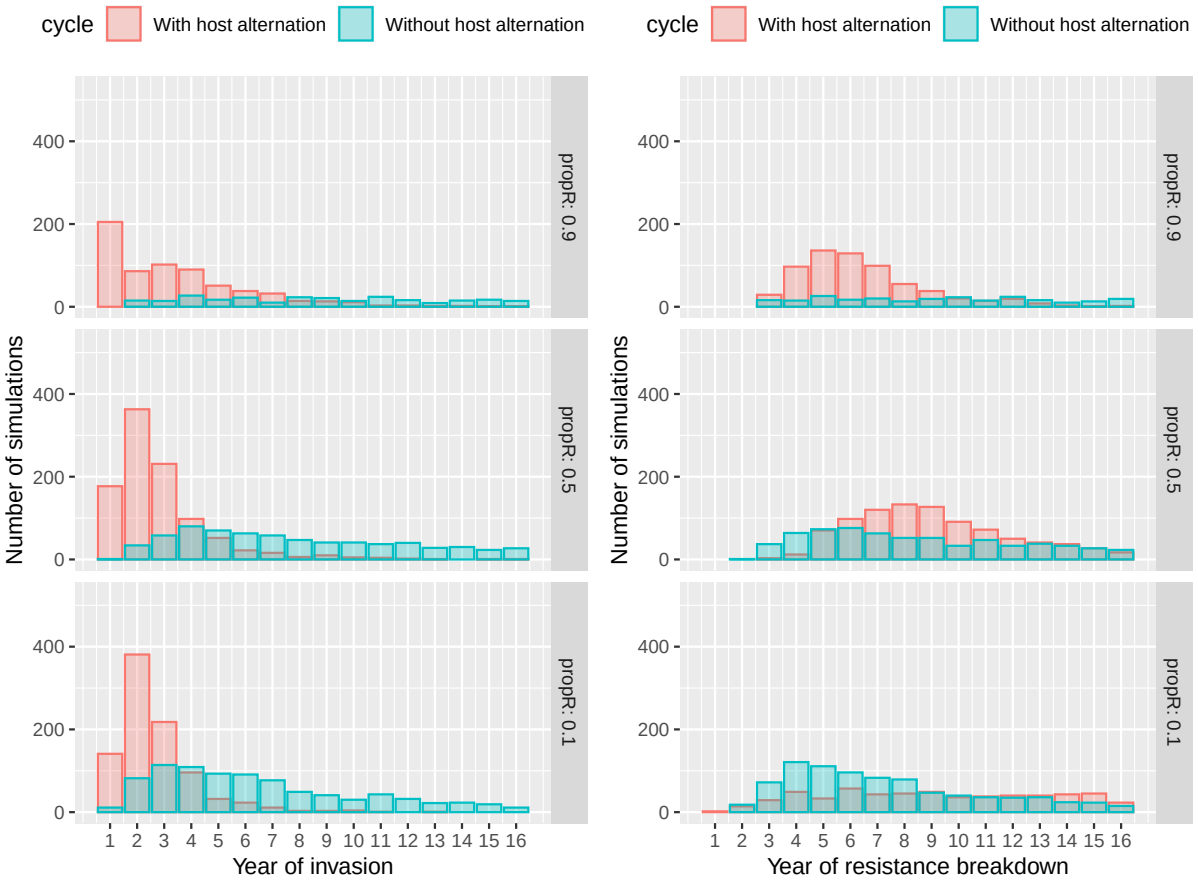


Figure C.10: Estimated general dominance of each predictor calculated from general linear models applied to two output variables of the restricted simulation design: the year of invasion and the time elapsed between the invasion and the fixation of the *avr* allele. For each predictor the general dominance was estimated from bootstrap average values with the corresponding standard error for 100 bootstrap resamples.



(a) Year of invasion

(b) Year of resistance breakdown

Figure C.11: Histograms of (a) the year of invasion and (b) the year of resistance breakdown depending on the life cycle of the diploid pathogen, for three values of $propR$. Simulations were performed with $f_{avr} = 0.02$, $mig = 0.001$, and $r = 1.5$. The plotted results were obtained from the restricted simulation design, and correspond to all simulations among the 1 000 replicates per combinations for which at least 80% of the R compartment is occupied at the end of the simulation.

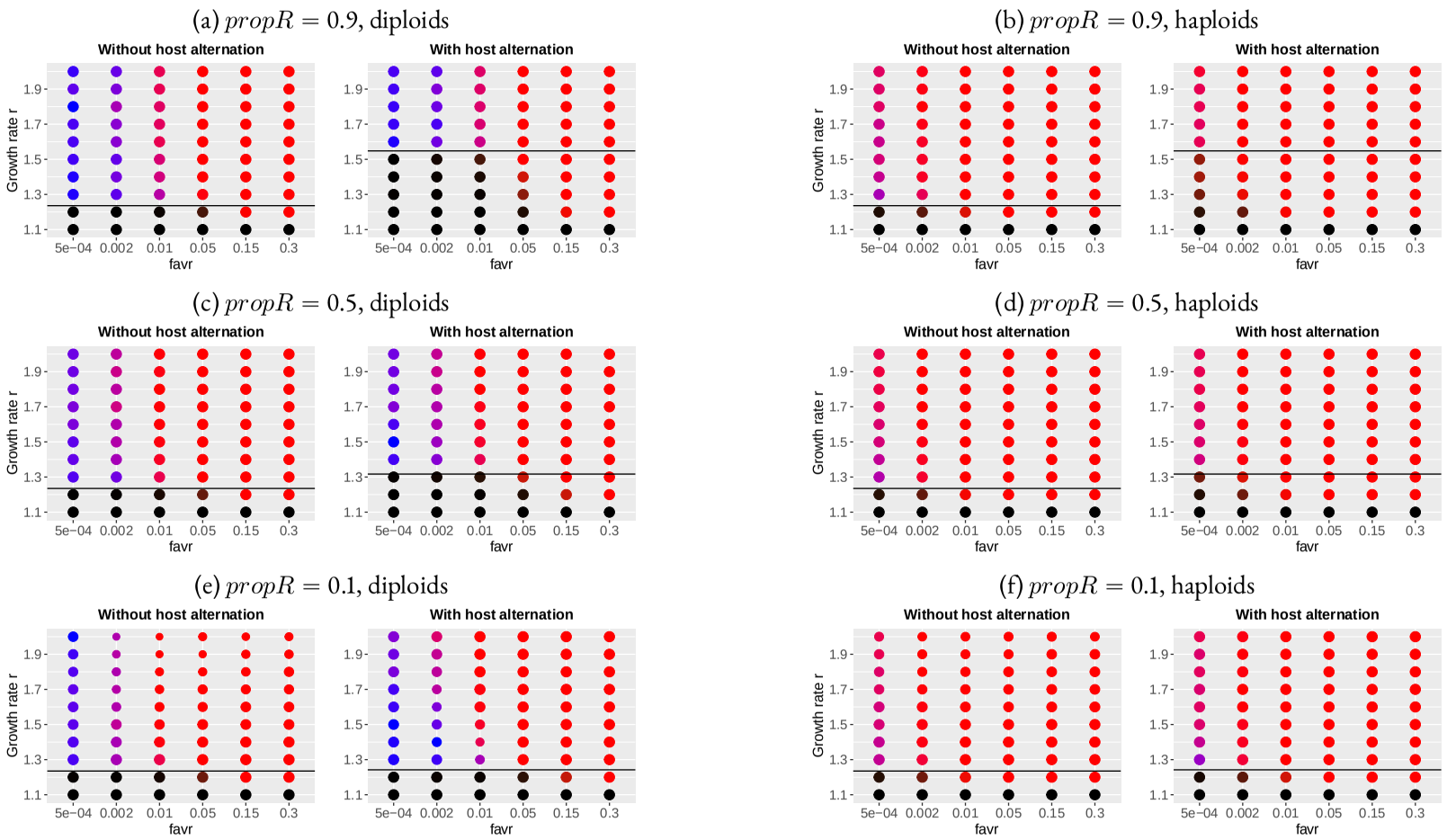


Figure C.12: Representation of the frequencies of population extinction or fixation of alleles Avr or avr for each combination of five parameters: f_{avr}, r, $propR$, $ploidy$ and $cycle$, with $mig = 0.05$. On each graph the black line corresponds to the calculated value of the growth rate threshold r_0 below which the population dies if it does not expand to the R compartment. The surface of each plotted result is proportional to the number of simulations, among the 100 replicates, for which an equilibrium was reached at the end of the 1100 generations simulated. In colored dots, red corresponds to the fixation of the avr allele, blue to the fixation of the Avr allele, and black to the extinction of the population. Dot color shades indicate simulation results among replicates. The left part ((a), (c), and (e)) corresponds to Figure C.4, presented in the main document.

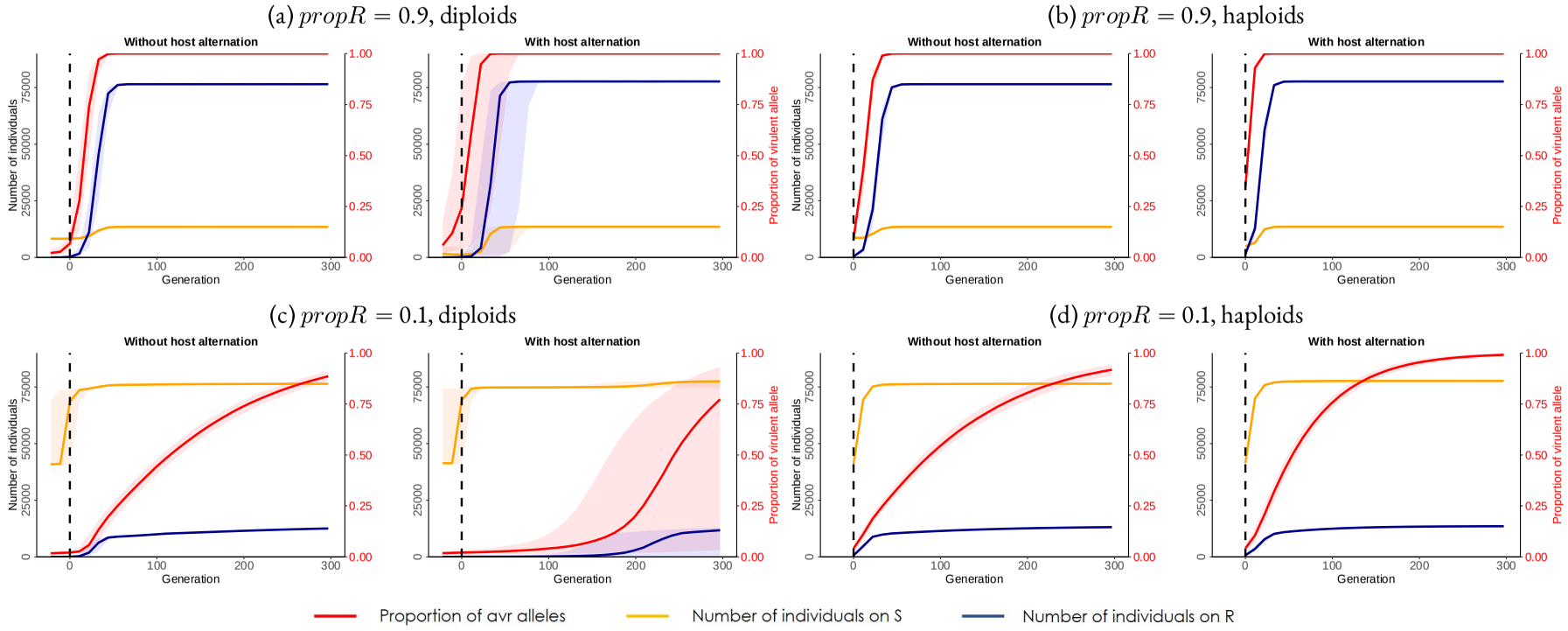


Figure C.13: Evolution of the population size in the R and S compartments (on the left scale) and of the frequency or avr alleles in the S compartment (on the right scale) through generations. Simulations were performed without and with host alternation, for (a, c) diploid and (b, d) haploid pathogens, for two values of $propR$: (a, b) 0.9 and (c, d) 0.1, with $f_{avr} = 0.02$, $mig = 0.05$, and $r = 1.5$. For each simulation, generation 0 corresponds to the generation at which the invasion occurred. For each combination of parameters, simulations were performed on 1 000 replicates. The plotted results correspond to the median results (frequency of avr alleles or population size) for all simulations among the 1 000 replicates for which we observed a fixation of the avr allele in the population. Coloured intervals correspond to the 95% confidence intervals. The left part ((a) and (c)) corresponds to Figure C.7, presented in the main document.

References

Agrios, G. N. (2005). *Plant pathology*. Elsevier.

Alexander, H. K., Martin, G., Martin, O. Y., and Bonhoeffer, S. (2014). Evolutionary rescue: Linking theory for conservation and medicine. *Evolutionary Applications*, 7(10):1161–1179.

Barrett, L. G., Thrall, P. H., Burdon, J. J., and Linde, C. C. (2008). Life history determines genetic structure and evolutionary potential of host-parasite interactions. *Trends in Ecology and Evolution*, 23(12):678–685.

Bazin, E., Mathé-Hubert, H., Facon, B., Carlier, J., and Ravigné, V. (2014). The effect of mating system on invasiveness: Some genetic load may be advantageous when invading new environments. *Biological Invasions*, 16(4):875–886.

Bolker, B. M., Nanda, A., and Shah, D. (2010). Transient virulence of emerging pathogens. *Journal of the Royal Society Interface*, 7.

Bousset, L. and Chèvre, A. M. (2013). Stable epidemic control in crops based on evolutionary principles: Adjusting the metapopulation concept to agro-ecosystems. *Agriculture, Ecosystems and Environment*, 165:118–129.

Bousset, L., Sprague, S. J., Thrall, P. H., and Barrett, L. G. (2018). Spatio-temporal connectivity and host resistance influence evolutionary and epidemiological dynamics of the canola pathogen *Leptosphaeria maculans*. *Evolutionary Applications*, 11:1354–1370.

Brown, J. K. M. (2015). Durable resistance of crops to disease: a darwinian perspective. *Annual Review of Phytopathology*, 53:513–539.

Brown, J. K. M. and Tellier, A. (2011). Plant-parasite coevolution: bridging the gap between genetics and ecology. *Annual Review of Phytopathology*, 49(1):345–367.

Burdon, J. J., Barrett, L. G., Rebetzke, G., and Thrall, P. H. (2014). Guiding deployment of resistance in cereals using evolutionary principles. *Evolutionary Applications*, 7(6):609–624.

Burdon, J. J. and Thrall, P. H. (2008). Pathogen evolution across the agro-ecological interface: implications for disease management. *Evolutionary Applications*, 1(1):57–65.

Burdon, J. J., Zhan, J., Barrett, L. G., Papaïx, J., and Thrall, P. H. (2016). Addressing the challenges of pathogen evolution on the world's arable crops. *Phytopathology*, 106(10):1117–1127.

Bustos Navarrete, C. and Coutinho Soares, F. (2020). *dominanceanalysis: Dominance Analysis*. R package version 2.0.0.

Carolan, K., Helps, J., Van Den Berg, F., Bain, R., Paveley, N., and Van Den Bosch, F. (2017). Extending the durability of cultivar resistance by limiting epidemic growth rates. *Proceedings of the Royal Society B: Biological Sciences*, 284.

Christiansen, F. B. (1975). Hard and soft selection in a subdivided sopulation. *The American Naturalist*, 109(965):11–16.

Day, T. and Gandon, S. (2007). Applying population-genetic models in theoretical evolutionary epidemiology. *Ecology Letters*, 10:876–888.

Day, T. and Proulx, S. R. (2004). A general theory for the evolutionary dynamics of virulence. *The American naturalist*, 163(4):E40–E63.

Djian-Caporalino, C., Palloix, A., Fazari, A., Marteu, N., Barbary, A., Abad, P., Sage-Palloix, A. M., Mateille, T., Risso, S., Lanza, R., Taussig, C., and Castagnone-Sereno, P. (2014). Pyramiding, alternating or mixing: Comparative performances of deployment strategies of nematode resistance genes to promote plant resistance efficiency and durability. *BMC Plant Biology*, 14:53.

Djidjou-Demasse, R., Moury, B., and Fabre, F. (2017). Mosaics often outperform pyramids: Insights from a model comparing strategies for the deployment of plant resistance genes against viruses in agricultural landscapes. *New Phytologist*, pages 239–253.

Fabre, F., Rousseau, E., Mailleret, L., and Moury, B. (2012). Durable strategies to deploy plant resistance in agricultural landscapes. *New Phytologist*, 193(4):1064–1075.

Fabre, F., Rousseau, E., Mailleret, L., and Moury, B. (2015). Epidemiological and evolutionary management of plant resistance: Optimizing the deployment of cultivar mixtures in time and space in agricultural landscapes. *Evolutionary Applications*, 8:919–932.

Flor, H. H. (1971). Current status of the gene-for-gene concept. *Annual Review of Phytopathology*, 9(1):275–296.

Garrett, K. A., Zúñiga, L. N., Roncal, E., Forbes, G. A., Mundt, C. C., Su, Z., and Nelson, R. J. (2009). Intraspecific functional diversity in hosts and its effect on disease risk across a climatic gradient. *Ecological Applications*, 19(7):1868–1883.

Gilligan, C. A. and van den Bosch, F. (2008). Epidemiological models for invasion and persistence of pathogens. *Annual Review of Phytopathology*, 46:385–418.

Goyeau, H. and Lannou, C. (2011). Specific resistance to leaf rust expressed at the seedling stage in cultivars grown in France from 1983 to 2007. *Euphytica*, 178:45–62.

Gracianne, C., Jan, P. L., Fournet, S., Olivier, E., Arnaud, J. F., Porte, C., Bardou-Valette, S., Denis, M. C., and Petit, E. J. (2016). Temporal sampling helps unravel the genetic structure of naturally occurring populations of a phytoparasitic nematode. 2. Separating the relative effects of gene flow and genetic drift. *Evolutionary Applications*, 9(8):1005–1016.

Haas, S. E., Hooten, M. B., Rizzo, D. M., and Meentemeyer, R. K. (2011). Forest species diversity reduces disease risk in a generalist plant pathogen invasion. *Ecology Letters*, 14:1108–1116.

Holt, J. and Chancellor, T. C. B. (1999). Modelling the spatio-temporal deployment of resistant varieties to reduce the incidence of rice tungro disease in a dynamic cropping system. *Plant Pathology*, 48:453–461.

Iooss, B., Da Veiga, S., Janon, A., Pujol, G., with contributions from Broto, B., Boumhaout, K., Delage, T., El Amri, R., Fruth, J., Gilquin, L., Guillaume, J., Il Idrissi, M., Le Gratiet, L., Lemaitre, P., Marrel, A., Meynaoui, A., Nelson, B. L., Monari, F., Oomen, R., Rakovec, O., Ramos, B., Roustant, O., Song, E., Staum, J., Sueur, R., Touati, T., and Weber, F. (2021). *sensitivity: Global Sensitivity Analysis of Model Outputs*. R package version 1.24.0.

Johnson, R. (1984). A critical analysis of durable resistance. *Annual Review of Phytopathology*, 22:309–330.

Laine, A. L. and Barrès, B. (2013). Epidemiological and evolutionary consequences of life-history trade-offs in pathogens. *Plant Pathology*, 62:96–105.

Leach, J. E., Vera Cruz, C. M., Bai, J., and Leung, H. (2001). Pathogen fitness penalty as a predictor of durability of disease resistance genes. *Annual Review of Ecology, Evolution, and Systematics*, 39:187–224.

Lenormand, T. (2002). Gene flow and the limits to natural selection. *Trends in Ecology and Evolution*, 17(4):183–189.

Lewontin, R. C. (1958). A general method for investigating the equilibrium of gene frequency in a population. *Genetics*, 43:419–434.

Lo Presti, L., Lanver, D., Schweizer, G., Tanaka, S., Liang, L., Tollot, M., Zuccaro, A., Reissmann, S., and Kahmann, R. (2015). Fungal effectors and plant susceptibility. *Annual Review of Plant Biology*, 66:513–545.

Lof, M. E., De Vallavieille-Pope, C., and Van Der Werf, W. (2017). Achieving durable resistance against plant diseases: Scenario analyses with a national-scale spatially explicit model for a wind-dispersed plant pathogen. *Phytopathology*, 107:580–589.

Lof, M. E. and van der Werf, W. (2017). Modelling the effect of gene deployment strategies on durability of plant resistance under selection. *Crop Protection*, 97:10–17.

Lorrain, C., Gonçalves dos Santos, K. C., Germain, H., Hecker, A., and Duplessis, S. (2019). Advances in understanding obligate biotrophy in rust fungi. *New Phytologist*, 222(3):1190–1206.

Martin, G., Aguilée, R., Ramsayer, J., Kaltz, O., and Ronce, O. (2013). The probability of evolutionary rescue: towards a quantitative comparison between theory and evolution experiments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368.

McDonald, B. A. (2004). Population genetics of plant pathogens. *The Plant Health Instructor*.

McDonald, B. A. and Linde, C. (2002). Pathogen population genetics , evolutionary potential, and durable resistance. *Annual Review of Phytopathology*, 40(1):349–379.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In *Frontiers in econometrics*, pages 104–142. Institute of Urban and Regional Development, University of California Oakland.

Montarry, J., Bardou-Valette, S., Mabon, R., Jan, P. L., Fournet, S., Grenier, E., and Petit, E. J. (2019). Exploring the causes of small effective population sizes in cyst nematodes using artificial *Globodera pallida* populations. *Proceedings of the Royal Society B: Biological Sciences*, 286(1894):20182359.

Montarry, J., Hamelin, F. M., Glais, I., Corbière, R., and Andrivon, D. (2010). Fitness costs associated with unnecessary virulence factors and life history traits: evolutionary insights from the potato late blight pathogen *Phytophthora infestans*. *BMC Evolutionary Biology*, 10:283.

Moran, N. A. (1992). The evolution of aphid life cycles. *Annual review of entomology*, 37(129):321–348.

Mundt, C. C. (2002). Use of multiline cultivars and cultivar mixtures for disease management. *Annual Review of Phytopathology*, 40(1):381–410.

Mundt, C. C. (2014). Durable resistance: A key to sustainable management of pathogens and pests. *Infection, Genetics and Evolution*, 27:1567–1348.

Nilusmas, S., Mercat, M., Perrot, T., Djian-Caporalino, C., Castagnone-Sereno, P., Touzeau, S., Calcagno, V., and Mailleret, L. (2020). Multiseasonal modelling of plant-nematode interactions reveals efficient plant resistance deployment strategies. *Evolutionary Applications*, 13(9):2206–2221.

Oleksyk, T. K., Smith, M. W., and O'Brien, S. J. (2010). Genome-wide scans for footprints of natural selection. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365:185–205.

Ostfeld, R. S. and Keesing, F. (2012). Effects of host diversity on infectious disease. *Annual Review of Ecology, Evolution, and Systematics*, 43(1):157–182.

Pacilly, F. C. A., Hofstede, G. J., Lammerts van Bueren, E. T., Kessel, G. J. T., and Groot, J. C. J. (2018). Simulating crop-disease interactions in agricultural landscapes to analyse the effectiveness of host resistance in disease control: The case of potato late blight. *Ecological Modelling*, 378:1–12.

Papaïx, J., Adamczyk-Chauvat, K., Bouvier, A., Kiêu, K., Touzeau, S., Lannou, C., and Monod, H. (2014). Pathogen population dynamics in agricultural landscapes: The Ddal modelling framework. *Infection, Genetics and Evolution*, 27:509–520.

Papaïx, J., Burdon, J. J., Zhan, J., and Thrall, P. H. (2015). Crop pathogen emergence and evolution in agro-ecological landscapes. *Evolutionary Applications*, 8(4):385–402.

Papaïx, J., Rimbaud, L., Burdon, J. J., Zhan, J., and Thrall, P. H. (2018). Differential impact of landscape-scale strategies for crop cultivar deployment on disease dynamics, resistance durability and long-term evolutionary control. *Evolutionary Applications*, 11(5):705–717.

Peng, B. and Kimmel, M. (2005). simuPOP: A forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687.

Perkins, T. A., Phillips, B. L., Baskett, M. L., and Hastings, A. (2013). Evolution of dispersal and life history interact to drive accelerating spread of an invasive species. *Ecology Letters*, 16(8):1079–1087.

Person, C., Samborski, D. J., and Rohringer, R. (1962). The gene-for-gene concept. *Nature*, 194:561–562.

Persoons, A., Hayden, K. J., Fabre, B., Frey, P., De Mita, S., Tellier, A., and Halkett, F. (2017). The escalatory Red Queen: Population extinction and replacement following arms race dynamics in poplar rust. *Molecular Ecology*.

Peterson, P. D. (2018). The barberry eradication program in Minnesota for stem rust control: a case study. *Annual Review of Phytopathology*, 56:203–223.

Pietravalle, S., Lemarié, S., and Van Den Bosch, F. (2006). Durability of resistance and cost of virulence. *European Journal of Plant Pathology*, 114:107–116.

Pilet-Nayel, M. L., Moury, B., Caffier, V., Montarry, J., Kerlan, M. C., Fournet, S., Durel, C. E., and Delourme, R. (2017). Quantitative resistance to plant pathogens in pyramiding strategies for durable crop protection. *Frontiers in Plant Science*, 8:1838.

Pink, D. and Puddephat, I. (1999). Deployment of disease resistance genes by plant transformation - a 'mix and match' approach. *Trends in Plant Science*, 4(2):71–75.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rimbaud, L., Fabre, F., Papaïx, J., Moury, B., Lannou, C., Barrett, L. G., and Thrall, P. H. (2021). Models of plant resistance deployment. *Annual Review of Phytopathology*, 59.

Rimbaud, L., Papaïx, J., Barrett, L. G., Burdon, J. J., and Thrall, P. H. (2018). Mosaics, mixtures, rotations or pyramiding: What is the optimal strategy to deploy major gene resistance? *Evolutionary Applications*, 11:1791–1810.

Rouxel, T. and Balesdent, M. H. (2017). Life, death and rebirth of avirulence effectors in a fungal pathogen of Brassica crops, *Leptosphaeria maculans*. *New Phytologist*, 214(2):526–532.

Sacristán, S. and García-Arenal, F. (2008). The evolution of virulence and pathogenicity in plant pathogen populations. *Molecular Plant Pathology*, 9(3):369–384.

Sapoukhina, N., Durel, C. E., and Le Cam, B. (2009). Spatial deployment of gene-for-gene resistance governs evolution and spread of pathogen populations. *Theoretical Ecology*, 2(4):229–238.

Skelsey, P., Rossing, W. A. H., Kessel, G. J. T., and Van Der Werf, W. (2010). Invasion of *Phytophthora infestans* at the landscape level: How do spatial scale and weather modulate the consequences of spatial heterogeneity in host resistance? *Phytopathology*, 100(11):1146–1161.

Soularue, J. P., Robin, C., Desprez-Loustau, M. L., and Dutech, C. (2017). Short rotations in forest plantations accelerate virulence evolution in root-rot pathogenic fungi. *Forests*, 8(6):205.

Stukenbrock, E. H. and McDonald, B. A. (2009). Population genetics of fungal and oomycete effectors involved in gene-for-gene interactions. *Molecular Plant-Microbe Interactions*, 22(4):371–380.

Terauchi, R. and Yoshida, K. (2010). Towards population genomics of effector-effector target interactions. *New Phytologist*, 187(4):929–939.

Thrall, P. H., Barrett, L. G., Dodds, P. N., and Burdon, J. J. (2016). Epidemiological and evolutionary outcomes in gene-for-gene and matching allele models. *Frontiers in Plant Science*, 6:1084.

Thrall, P. H. and Burdon, J. J. (2003). Evolution of virulence in a plant host-pathogen metapopulation. *Science*, 299:1735–1737.

Van den Bosch, F. and Gilligan, C. A. (2003). Measures of durability of resistance. *Phytopathology*, 93(5):616–625.

van Rossum, G. (1995). Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI).

Wallace, B. (1975). Hard and soft selection revisited. *Evolution*, 29:465–473.

Zhan, J., Thrall, P. H., and Burdon, J. J. (2014). Achieving sustainable plant disease management through evolutionary principles. *Trends in Plant Science*, 19(9):570–575.

Zhan, J., Thrall, P. H., Papaïx, J., Xie, L., and Burdon, J. J. (2015). Playing on a aathogen's weakness: using evolution to guide sustainable plant disease control strategies. *Annual Review of Phytopathology*, 53(1):19–43.

Acronyms

a.s.	almost surely
AOLS	adaptive ordinary least-squares
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
CV	cross-validation
DOF	degree of freedom
ED	experimental design
EDP	engineering demand parameter
EPSD	evolutionary power spectral density
FGLS	feasible generalized least-squares
FKML	Freimer–Kollia–Mudholkar–Lin
GLaM	generalized lambda model
GLD	generalized lambda distribution
GLM	generalized linear model
GMM	generalized method of moments
GMPE	ground motion predictive equation
GSA	global sensitivity analysis
HS	hybrid simulation
i.i.d.	independent and identically distributed
IM	intensity measure
KCDE	kernel conditional density estimator
KDE	kernel density estimator
KT-PSD	Kanai-Tajimi power spectral density
LAR	least-angle regression
LASSO	least absolute shrinkage and selection operator
LHS	Latin hypercube sampling
LM	linear model
LOO	leave-one-out
MCMC	Markov chain Monte Carlo
MLE	maximum likelihood estimation
NS	numerical substructure
OLS	ordinary least-squares
PBEE	performance-based earthquake engineering
PCA	principal component analysis
PCE	polynomial chaos expansion
PDF	probability density function
PGA	peak ground acceleration
PS	physical substructure
PSHA	probabilistic seismic hazard analysis
QoI	quantity of interest
SA	spectral acceleration
SDE	stochastic differential equation
SGMM	stochastic ground motion model
SIR	susceptible-infected-recovered
SPCE	stochastic polynomial chaos expansion
std	standard deviation
WLS	weighted least-squares

Bibliography

Abbiati, G., Broccardo, M., Marelli, S., and Paolacci, F. (2021a). Seismic fragility analysis based on artificial ground motions and surrogate modeling of validated structural simulators. *Earthquake Engineering & Structural Dynamics*, 9:2314–2333.

Abbiati, G., Covi, P., Tondini, N., Bursi, O. S., and Stojadinović, B. (2020). A real-time hybrid fire simulation method based on dynamic relaxation and partitioned time integration. *Journal of Engineering Mechanics*, 146(9):04020104.

Abbiati, G., Hey, V., Rion, J., and Stojadinović, B. (2018). Thermo-mechanical virtualization of hybrid flax/carbon fiber composite for spacecraft structures. thermics project, final report. Technical report, ETH Zurich.

Abbiati, G., Lanese, I., Cazzador, E., Bursi, O. S., and Pavese, A. (2019). A computational framework for fast-time hybrid simulation based on partitioned time integration and state-space modeling. *Structural Control and Health Monitoring*, 26(10):1–28.

Abbiati, G., Marelli, S., Tsokanas, N., Sudret, B., and Stojadinović, B. (2021b). A global sensitivity analysis framework for hybrid simulation. *Mechanical Systems and Signal Processing*, 146.

Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators—application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67–77.

Abraham, S., Raisee, M., Ghorbaniasl, G., Contino, F., and Lacor, F. (2017). A robust and efficient stepwise regression method for building sparse polynomial chaos expansions. *Journal of Computational Physics*, 332:461–474.

Abramowitz, M. and Stegun, I. A. (1970). *Handbook of mathematical functions*. Dover Publications, Inc.

Acerbi, C. (2002). Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking and Finance*, 26:1505–1518.

Agrios, G. N. (2005). *Plant pathology*. Elsevier.

Alexander, H. K., Martin, G., Martin, O. Y., and Bonhoeffer, S. (2014). Evolutionary rescue: Linking theory for conservation and medicine. *Evolutionary Applications*, 7(10):1161–1179.

Alexanderian, A., Winokur, J., Sraj, I., Srinivasan, A., Iskandarani, M., Thacker, W. C., and Knio, O. M. (2012). Global sensitivity analysis in an ocean general circulation model: a sparse spectral projection approach. *Computational Geosciences*, 16(3):757–778.

Allaire, G. (2005). *Analyse numérique et optimisation*. Ecole Polytechnique.

Amemiya, T. (1977). A note on a heteroscedastic model. *Journal of Econometrics*, 6(3):365–370.

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58:371–382.

Anscombe, F. J. and Tukey, J. W. (1963). The examination and analysis of residuals. *Technometrics*, 5:141–160.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning (ICML' 17), Sydney, Australia*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223.

Arnold, D. V. and Hansen, N. (2012). A (1+1)-CMA-ES for constrained optimisation. In Soule, T. and Moore, J. H., editors, *Proceedings of the Genetic and Evolutionary Computation Conference 2012 (GECCO 2012) (Philadelphia, PA)*, pages 297–304.

Asher, M. J., Croke, B. F. W., Jakeman, A. J., and Peeters, L. J. M. (2015). A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51(8):5957–5973.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions—application to computational electromagnetic dosimetry. *International Journal for Uncertainty Quantification*, 9:351–363.

Azzi, S., Sudret, B., and Wiart, J. (2020). Sensitivity analysis for stochastic simulators using differential entropy. *International Journal for Uncertainty Quantification*, 10:25–33.

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2010). Bayesian compressive sensing using Laplace priors. *IEEE Transactions on Image Processing*, 19(1):53–63.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecifications. *Computational Statistics & Data Analysis*, 66:55–69.

Bachoc, F., Bois, G., Garnier, J., and Martinez, J.-M. (2014). Calibration and improved prediction of computer models by universal kriging. *Nuclear Science and Engineering*, 176(1):81–97.

Baker, J. W. (2015). Efficient analytical fragility function fitting using dynamic structural analysis. *Earthquake Spectra*, 31(1):579–599.

Baker, J. W. and Cornell, C. A. (2005). A vector-valued ground motion intensity measure consisting of spectral acceleration and epsilon. *Earthquake Engineering & Structural Dynamics*, 34:1193–1217.

Baker, J. W. and Cornell, C. A. (2006). Spectral shape, epsilon and record selection. *Earthquake Engineering & Structural Dynamics*, 35:1077–1095.

Barrett, L. G., Thrall, P. H., Burdon, J. J., and Linde, C. C. (2008). Life history determines genetic structure and evolutionary potential of host-parasite interactions. *Trends in Ecology and Evolution*, 23(12):678–685.

Bas, E. E., Moustafa, M. A., and Pekcan, G. (2022). Compact Hybrid Simulation System: Validation and Applications for Braced Frames Seismic Testing. *Journal of Earthquake Engineering*, 26(3):1565–1594.

Batchu, S. (2022). Aerospace engineering online courses. https://www.stressebook.com/.

Bau, D., Zhu, J., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., and Torralba, A. (2019). Seeing what a GAN cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, pages 4502–4511.

Bazin, E., Mathé-Hubert, H., Facon, B., Carlier, J., and Ravigné, V. (2014). The effect of mating system on invasiveness: Some genetic load may be advantageous when invading new environments. *Biological Invasions*, 16(4):875–886.

Bertin, K., Lacour, C., and Rivoirard, V. (2016). Adaptive pointwise estimation of conditional density function. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 52(2):939–980.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15(1–3):81–92.

Bhattacharya, P. K. and Gangopadhyay, A. K. (1990). Kernel and nearest-neighbor estimation of a conditional quantile. *The Annals of Statistics*, 18(3):1400–1415.

Billingsley, P. (1995). *Probability and Measure*. John Wiley & Sons, 3rd edition.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27:808–821.

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2019). Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61:7–23.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Advanced Texts in Econometrics. Oxford University Press, Oxford, UK.

Blatman, G. (2009). *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand.

Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367.

Bogachev, V. I. (2007). *Measure Theory*. Springer Berlin, Heidelberg.

Bolker, B. M., Nanda, A., and Shah, D. (2010). Transient virulence of emerging pathogens. *Journal of the Royal Society Interface*, 7.

Borgonovo, E. (2007). A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92:771–784.

Borgonovo, E. (2017). *Sensitivity analysis – An Introduction for the Management Scientist*. Springer.

Borgonovo, E., Figalli, A., Plischke, E., and Savarè, G. (2022). Probabilistic sensitivity with optimal transport. *In preparation*.

Borgonovo, E., Tarantola, S., Plischke, E., and Morris, M. D. (2014). Transformations and invariance in the sensitivity analysis of computer experiments. *Journal of the Royal Statistical Society: Series B*, 76(5):925–947.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.

Bousset, L. and Chèvre, A. M. (2013). Stable epidemic control in crops based on evolutionary principles: Adjusting the metapopulation concept to agro-ecosystems. *Agriculture, Ecosystems and Environment*, 165:118–129.

Bousset, L., Sprague, S. J., Thrall, P. H., and Barrett, L. G. (2018). Spatio-temporal connectivity and host resistance influence evolutionary and epidemiological dynamics of the canola pathogen *Leptosphaeria maculans*. *Evolutionary Applications*, 11:1354–1370.

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45:5–32.

Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer New York, 1st edition.

Britton, T. (2010). Stochastic epidemic models: A survey. *Mathematical Biosciences*, 225:24–35.

Broccardo, M. and Dabaghi, M. (2017). A spectral-based stochastic ground motion model with a non-parametric time-modulating function. In *12th International Conference on Structural Safety and Reliability, Vienna, Austria*, volume 2017, pages 1–10.

Broccardo, M. and Dabaghi, M. (2019). Preliminary validation of a spectral-based stochastic ground motion model with a non-parametric time-modulating function. In *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13), Seoul, South Korea*.

Brown, J. K. M. (2015). Durable resistance of crops to disease: a darwinian perspective. *Annual Review of Phytopathology*, 53:513–539.

Brown, J. K. M. and Tellier, A. (2011). Plant-parasite coevolution: bridging the gap between genetics and ecology. *Annual Review of Phytopathology*, 49(1):345–367.

Brown, S., Beck, J., Mahgerefteh, H., and Fraga, E. S. (2013). Global sensitivity analysis of the impact of impurities on CO2 pipeline failure. *Reliability Engineering & System Safety*, 115:43–54.

Browne, T. (2017). *Regression Models and Sensitivity Analysis for Stochastic Simulators: Applications to Non-Destructive Examination*. PhD thesis, Université of Paris Descartes, Paris.

Browne, T., Iooss, B., Le Gratiet, L., Lonchampt, J., and Rémy, E. (2016). Stochastic simulators based optimization by Gaussian process metamodels—application to maintenance investments planning issues. *Quality and Reliability Engineering International*, 32(6):2067–2080.

Burden, R. L., Faires, J. D., and Burden, A. M. (2015). *Numerical analysis*. Cengage Learning.

Burdon, J. J., Barrett, L. G., Rebetzke, G., and Thrall, P. H. (2014). Guiding deployment of resistance in cereals using evolutionary principles. *Evolutionary Applications*, 7(6):609–624.

Burdon, J. J. and Thrall, P. H. (2008). Pathogen evolution across the agro-ecological interface: implications for disease management. *Evolutionary Applications*, 1(1):57–65.

Burdon, J. J., Zhan, J., Barrett, L. G., Papaïx, J., and Thrall, P. H. (2016). Addressing the challenges of pathogen evolution on the world's arable crops. *Phytopathology*, 106(10):1117–1127.

Bustos Navarrete, C. and Coutinho Soares, F. (2020). *dominanceanalysis: Dominance Analysis*. R package version 2.0.0.

Carolan, K., Helps, J., Van Den Berg, F., Bain, R., Paveley, N., and Van Den Bosch, F. (2017). Extending the durability of cultivar resistance by limiting epidemic growth rates. *Proceedings of the Royal Society B: Biological Sciences*, 284.

Cartwright, H. (2015). *Artificial neural networks*. Humana Press, 2nd edition.

Chalabi, Y., Scott, D. J., and Würtz, D. (2011). The generalized lambda distribution as an alternative to model financial returns. Technical report, Eidgenössische Technische Hochschule and University of Auckland.

Chan, A. B. and Dong, D. (2011). Generalized gaussian process models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado, USA*, pages 2681–2688.

Chapelle, O., Vapnik, V., and Bengio, Y. (2002). Model selection for small sample regression. *Machine Learning*, 48(1):9–23.

Chastaing, G., Gamboa, F., and Prieur, C. (2015). Generalized Sobol sensitivity indices for dependent variables: numerical methods. *Journal of Statistical Computation and Simulation*, 85(7):1306–1333.

Chen, X., Linton, O., and Robinson, P. M. (2001). The estimation of conditional densities. STICERD - Econometrics Paper Series 415, Suntory and Toyota International Centres for Economics and Related Disciplines, LSE.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. (2021). Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118.

Chevreuil, M., Lebrun, R., Nouy, A., and Rai, P. (2015). A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):897–921.

Christiansen, F. B. (1975). Hard and soft selection in a subdivided sopulation. *The American Naturalist*, 109(965):11–16.

Corlu, C. G. and Meterelliyoz, M. (2016). Estimating the parameters of the generalized lambda distribution: Which method performs best? *Communications in Statistics - Simulation and Computation*, 45(7):2276–2296.

Cornell, C. A. (1968). Engineering seismic risk analysis. *Bulletin of the Seismological Society of America*, 58(5):1583–1606.

Cornell, C. A., Jalayer, F., Hamburger, R. O., and Foutch, D. A. (2002). Probabilistic basis for 2000 SAC federal emergency management agency steel moment frame guidelines. *Earthquake Engineering & Structural Dynamics*, 128:526–533.

Cornell, C. A. and Krawinkler, H. (2000). Progress and challenges in seismic performance assessment. *PEER center news*, 3(2):1–3.

Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31:377–403.

Cuevas, E. (2020). An agent-based model to evaluate the covid-19 transmission risks in facilities. *Computers in Biology and Medicine*, 121:103827.

Czado, C. (2019). *Analyzing Dependent Data with Vine Copulas*. Lecture Notes in Statistics. Springer.

da Veiga, S. (2021). Kernel-based anova decomposition and shapley effects — application to global sensitivity analysis.

Dacidian, M. and Carroll, R. J. (1987). Variance function estimation. *Journal of the American Statistical Association*, 400:1079–1091.

Davidian, M. and Carroll, R. J. (1987). Variance function estimation. *Journal of the American Statistical Association*, 82:1079–1091.

Day, T. and Gandon, S. (2007). Applying population-genetic models in theoretical evolutionary epidemiology. *Ecology Letters*, 10:876–888.

Day, T. and Proulx, S. R. (2004). A general theory for the evolutionary dynamics of virulence. *The American naturalist*, 163(4):E40–E63.

de Rocquigny, E. (2006). La maîtrise des incertitudes dans un contexte industriel : 1^e partie – Une approche méthodologique globale basée sur des exemples. *Journal de la Société Française de Statistique*, 147(3):33–71.

de Rocquigny, E., Devictor, N., and Tatantola, S. (2008). *Uncertainty in Industrial Practice: A guide to Quantitative Uncertainty Management*. John Wiley & Sons, New York.

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112.

Desceliers, C., Ghanem, R. G., and Soize, C. (2006). Maximum likelihood estimation of stochastic chaos representations from experimental data. *International Journal for Numerical Methods in Engineering*, 66:978–1001.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France*.

Djian-Caporalino, C., Palloix, A., Fazari, A., Marteu, N., Barbary, A., Abad, P., Sage-Palloix, A. M., Mateille, T., Risso, S., Lanza, R., Taussig, C., and Castagnone-Sereno, P. (2014). Pyramiding, alternating or mixing: Comparative performances of deployment strategies of nematode resistance genes to promote plant resistance efficiency and durability. *BMC Plant Biology*, 14:53.

Djidjou-Demasse, R., Moury, B., and Fabre, F. (2017). Mosaics often outperform pyramids: Insights from a model comparing strategies for the deployment of plant resistance genes against viruses in agricultural landscapes. *New Phytologist*, pages 239–253.

Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034.

Dubourg, V. (2011). *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France.

Duong, T. and Hazelton, M. L. (2005). Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32:485–506.

Durrett, R. (2019). *Probability: Theory and Examples*. Cambridge University Press, 5th edition.

Echard, B., Gayton, N., and Lemaire, M. (2011). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*, 33(2):145–154.

Efromovich, S. (2010). Dimension reduction and adaptation in conditional density estimation. *Journal of the American Statistical Association*, 105:761–774.

Efron, B. (1979). Bootstrap methods: another look at the Jackknife. *The Annals of Statistics*, 7(1):1–26.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32:407–499.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017.

Embrechts, P. and Hofert, M. (2013). A note on generalized inverses. *Mathematical Methods of Operations Research*, 277:423–432.

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339.

Eurocode (2004). *Eurocode — Basis of Structural Design*.

Everitt, B. S. (1984). *An Introduction to Latent Variables Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fabre, F., Rousseau, E., Mailleret, L., and Moury, B. (2012). Durable strategies to deploy plant resistance in agricultural landscapes. *New Phytologist*, 193(4):1064–1075.

Fabre, F., Rousseau, E., Mailleret, L., and Moury, B. (2015). Epidemiological and evolutionary management of plant resistance: Optimizing the deployment of cultivar mixtures in time and space in agricultural landscapes. *Evolutionary Applications*, 8:919–932.

Fadikar, A., Higdon, D., Chen, J., Lewis, B., Venkatramanan, S., and Marathe, M. (2018). Calibrating a stochastic, agent-based model using quantile-based emulation. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1685–1706.

Fajraoui, N., Marelli, S., and Sudret, B. (2017). Sequential design of experiment for sparse polynomial chaos expansions. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1061–1085.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J., Liu, H., Sun, Q., and Zhang, T. (2018). I-LAMM for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *The Annals of Statistics*, 46(2):814–841.

Fan, J., Yao, Q., and Tong, H. (1996). Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206.

Fan, J. and Yao, Q. W. (1998). Efficient estimation of conditional variance functions in stochastic regression. *Biometrika*, 85:645–660.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Federal Emergency Management Agency (2000). Commentary for the seismic rehabilitation of buildings. Technical report.

Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition.

Flor, H. H. (1971). Current status of the gene-for-gene concept. *Annual Review of Phytopathology*, 9(1):275–296.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. Wiley.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Communications in Statistics - Theory and Methods*, 17:3547–3567.

Freud, G. (1971). *Orthogonal Polynomials*. Pergamon, 1st edition.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67.

Garrett, K. A., Zúñiga, L. N., Roncal, E., Forbes, G. A., Mundt, C. C., Su, Z., and Nelson, R. J. (2009). Intraspecific functional diversity in hosts and its effect on disease risk across a climatic gradient. *Ecological Applications*, 19(7):1868–1883.

Gautier, A., Ginsbourger, D., and Pirot, G. (2021). Goal-oriented adaptive sampling under random field modelling of response probability distributions. *ESAIM: Proceedings and Surveys*, 71:89–100.

Gautschi, W. (2004). *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press.

Genevay, A., Peyre, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*, volume 84, pages 1608–1617.

Gentle, J. E. (2003). *Random Number Generation and Monte Carlo Methods*. Statistics and Computing. Springer, 2nd edition.

Ghanem, R. G. (2017). *Handbook of uncertainty quantification*. Springer, Cham, Switzerland.

Ghanem, R. G. and Spanos, P. (2003). *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, Mineola, 2nd edition.

Ghiocel, D. M. and Ghanem, R. G. (2002). Stochastic finite element analysis of seismic soil-structure interaction. *Journal of Engineering Mechanics*, 128:66–77.

Ghosh, S. and Chakraborty, S. (2020). Seismic fragility analysis of structures based on Bayesian linear regression demand models. *Probabilistic Engineering Mechanics*, 61:103081.

Ghosh, S., Roy, A., and Chakraborty, S. (2021). Kriging metamodeling-based Monte Carlo simulation for improved seismic fragility analysis of structures. *Journal of Earthquake Engineering*, 25:1316–1336.

Gidaris, I., Taflanidis, A. A., and Mavroeidis, G. P. (2015). Kriging metamodeling in seismic risk assessment based on stochastic ground motion models. *Earthquake Engineering & Structural Dynamics*, 44:2377–2399.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81:2340–2361.

Gilligan, C. A. and van den Bosch, F. (2008). Epidemiological models for invasion and persistence of pathogens. *Annual Review of Phytopathology*, 46:385–418.

Goldberg, P. W., Williams, C. K. I., and Bishop, C. M. (1997). Regression with input-dependent noise: A Gaussian process treatment. In *Proceedings of the 10th International Conference on Advances in Neural Information Processing Systems (NIPS 1997), Colorado, USA*, pages 493–499.

Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Proceedings of the 28th Conference on Advances in Neural Information Processing Systems (NIPS 2014), Montreal, Canada*, volume 27.

Goyeau, H. and Lannou, C. (2011). Specific resistance to leaf rust expressed at the seedling stage in cultivars grown in France from 1983 to 2007. *Euphytica*, 178:45–62.

Gracianne, C., Jan, P. L., Fournet, S., Olivier, E., Arnaud, J. F., Porte, C., Bardou-Valette, S., Denis, M. C., and Petit, E. J. (2016). Temporal sampling helps unravel the genetic structure of naturally occurring populations of a phytoparasitic nematode. 2. Separating the relative effects of gene flow and genetic drift. *Evolutionary Applications*, 9(8):1005–1016.

Grant, E. (2007). *A History of Natural Philosophy: From the Ancient World to the Nineteenth Century*. Cambridge University Press.

Gray, A., Greenhalgh, D., Hu, L., Mao, X., and Pan, J. (2011). A stochastic differential equation SIS epidemic model. *SIAM Journal on Applied Mathematics*, 71(3):876–902.

Grigoriu, M. (2011). To scale or not to scale seismic ground-acceleration records. *Journal of Engineering Mechanics*, 137(4):284–293.

Haas, S. E., Hooten, M. B., Rizzo, D. M., and Meentemeyer, R. K. (2011). Forest species diversity reduces disease risk in a generalist plant pathogen invasion. *Ecology Letters*, 14:1108–1116.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99:1015–1026.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90.

Handa, K. and Andersson, K. (1981). Application of finite element methods in the statistical analysis of structures. In Moan, T. and Shinozuka, M., editors, *Proceedings of the 3rd International Conference on Structural Safety and Reliability (ICOSSAR'81), Trondheim, Norway*, pages 409–420.

Hanin, B. (2018). Which neural net architectures give rise to exploding and vanishing gradients? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Proceedings of the 32nd International Conference on Advances in Neural Information Processing Systems (NIPS 2018)*, volume 31.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica*, 50:1029–1054.

Härdle, W. and Tsybakov, A. (1997). Local polynomial estimators of the volatility function in nonparametric autoregression. *Journal of Econometrics*, 81(1):223–242.

Harenberg, D., Marelli, S., Sudret, B., and Winschel, V. (2019). Uncertainty quantification and global sensitivity analysis for economic models. *Quantitative Economics*, 10(1):1–41.

Hart, J. L., Alexanderian, A., and Gremaud, P. A. (2016). Efficient computation of Sobol' indices for stochastic models. *SIAM Journal on Scientific Computing*, 39:1514–1530.

Harvey, A. C. (1976). Estimating regression models with multiplicative heteroscedasticity. *Econometrica*, 44:461–465.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer.

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27:1–32.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 770–778.

He, X. (1997). Quantile curves without crossing. *The American Statistician*, 51(2):186–192.

Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B. (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10):1175–1209.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6:327–343.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *The Annals of Mathematical Statistics*, 19:293–325.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Holmes, M. P., Gray, A. G., and Isbell, C. L. (2007). Fast nonparametric conditional density estimation. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI'07), Vancouver, Canada*, pages 175–182.

Holt, J. and Chancellor, T. C. B. (1999). Modelling the spatio-temporal deployment of resistant varieties to reduce the incidence of rice tungro disease in a dynamic cropping system. *Plant Pathology*, 48:453–461.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of non linear models. *Reliability Engineering & System Safety*, 52:1–17.

Huoh, Y. (2013). *Sensitivity Analysis of Stochastic Simulators with Information Theory*. PhD thesis, University of California, Berkeley.

Idinyang, S., Franza, A., Heron, C. M., and Marshall, A. M. (2019). Real-time data coupling for hybrid testing in a geotechnical centrifuge. *International Journal of Physical Modelling in Geotechnics*, 19(4):208–220.

Iooss, B., Da Veiga, S., Janon, A., Pujol, G., with contributions from Broto, B., Boumhaout, K., Delage, T., El Amri, R., Fruth, J., Gilquin, L., Guillaume, J., Il Idrissi, M., Le Gratiet, L., Lemaitre, P., Marrel, A., Meynaoui, A., Nelson, B. L., Monari, F., Oomen, R., Rakovec, O., Ramos, B., Roustant, O., Song, E., Staum, J., Sueur, R., Touati, T., and Weber, F. (2021). *sensitivity: Global Sensitivity Analysis of Model Outputs*. R package version 1.24.0.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94:1194–1204.

Iversen, E. B., Morales, J. M., Moller, J. K., Mao, X., and Madsen, H. (2015). Short-term probabilistic forecasting of wind speed using stochastic differential equations. *International Journal of Forecasting*, 32:981–990.

Izbicki, R. and Lee, A. B. (2017). Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics*, 11:2800–2831.

Jacod, J. and Protter, P. (2004). *Probability Essentials*. Springer, 2nd edition.

Jakeman, J. D., Franzelin, F., Natayan, A., Eldred, M., and Plfüger, D. (2019). Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351:643–666.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Janon, A., Klein, T., Lagnoux, A., Nodet, M., and Prieur, C. (2014). Asymptotic normality and efficiency of two Sobol' index estimators. *ESAIM: Probability and Statistics*, 18:342–364.

Jaynes, E. T. (2003). *Probability theory: the logic of science*. Cambridge University Press.

Jennrich, R. I. and Sampson, P. F. (1968). Application of stepwise regression to non-linear estimation. *Technometrics*, 10(1):63–72.

Jimenez, M. N., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 5:378–402.

Johnson, R. (1984). A critical analysis of durable resistance. *Annual Review of Phytopathology*, 22:309–330.

Joint Committee on Structural Safety (2002). *JCSS probabilistic model code*.

Jonkman, B. J. (2009). TurbSim user's guide: version 1.50. Technical report, National Renewable Energy Laboratory, U.S. Department of Energy.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-MW reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory, U.S. Department of Energy.

Kantorovich, L. V. and Akilov, G. P. (1982). *Functional Analysis*. Pergamon, 2nd edition.

Karhunen, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. PhD thesis, University of Helsinki.

Karian, Z. A. and Dudewicz, E. J. (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press.

Karian, Z. A. and Dudewicz, E. J. (2010). *Handbook of Fitting Statistical Distributions with R*. Taylor & Francis.

Kemna, A. G. Z. and Vorst, A. C. F. (1990). A pricing method for options based on average asset values. *Journal of Banking & Finance*, 14:113–129.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2007). Most likely heteroscedastic gaussian process regression. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07), Corvalis, Oregon, USA*, pages 393–400.

Kiani, J. and Khanmohammadi, M. (2015). New approach for selection of real input ground motion records for incremental dynamic analysis (IDA). *J. Earthq. Eng.*, 19:592–623.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada*.

Kloeden, P. E. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer.

Koenker, R. (2017). Quantile regression: 40 years on. *Annual Review of Economics*, 9:155–176.

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46:33–50.

Kolmogorov, A. (1933). Die elementare Wahrscheinlichkeitsrechnung. In *Grundbegriffe der Wahrscheinlichkeitsrechnung*, pages 1–13. Springer Berlin Heidelberg.

König, H. (1986). *Eigenvalue Distribution of Compact Operators*. Operator Theory: Advances and Applications. Birkhäuser Basel.

Konishi, S. and Kitagawa, G. (2008). *Information Criteria and Statistical Modeling*. Springer Series in Statistics. Springer.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139.

Lacour, C. (2015). Contributions à l'estimation non-paramétrique adaptative : estimation de loi conditionnelle et déconvolution. Habilitation à diriger des recherches, Université Paris-Sud, France.

Laine, A. L. and Barrès, B. (2013). Epidemiological and evolutionary consequences of life-history trade-offs in pathogens. *Plant Pathology*, 62:96–105.

Lakhany, A. and Massuer, H. (2000). Estimating the parameters of the generalised lambda distribution. *Algo Research Quarterly*, 3(3):47–58.

Landau, L. D. and Lifshitz, E. M. (1987). *Fluid Mechanics*. Pergamon, 2nd edition.

Laplace, P. S. (1814). *Essai philosophique sur les probabilit/'es*.

Lataniotis, C. (2019). *Data-driven uncertainty quantification for high-dimensional engineering problems*. PhD thesis, ETH Zurich.

Lataniotis, C., Marelli, S., and Sudret, B. (2018). The Gaussian process modelling module in UQLab. *Journal of Soft Computing in Civil Engineering*, 2(3):91–116.

Lauvernet, C. and Helbert, C. (2020). Metamodeling methods that incorporate qualitative variables for improved design of vegetative filter strips. *Reliability Engineering & System Safety*, 204:107083.

Lázaro-Gredilla, M. and Titsias, M. K. (2011). Variational heteroscedastic gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11), Bellevue, Washington, USA*, pages 841–848.

Le Matre, O. P., Knio, O. M., Najm, N. H., and Ghanem, R. G. (2004). Uncertainty propagation using Wiener–Haar expansions. *Journal of Computational Physics*, 224:560–586.

Leach, J. E., Vera Cruz, C. M., Bai, J., and Leung, H. (2001). Pathogen fitness penalty as a predictor of durability of disease resistance genes. *Annual Review of Ecology, Evolution, and Systematics*, 39:187–224.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 512:436–444.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Lenk, P. J. (1991). Towards a practicable Bayesian nonparametric density estimator. *Biometrika*, 78(3):531–543.

Lenormand, T. (2002). Gene flow and the limits to natural selection. *Trends in Ecology and Evolution*, 17(4):183–189.

LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press.

Lewontin, R. C. (1958). A general method for investigating the equilibrium of gene frequency in a population. *Genetics*, 43:419–434.

Li, Q., Lin, J., and Racine, J. S. (2013). Optimal bandwidth selection for nonparametric conditional distribution and quantile functions. *Journal of Business & Economic Statistics*, 31:57–65.

Li, X., Sudarsanam, N., and Frey, D. D. (2006). Regularities in data from factorial experiments. *Complexity*, 11(5):32–45.

Lo Presti, L., Lanver, D., Schweizer, G., Tanaka, S., Liang, L., Tollot, M., Zuccaro, A., Reissmann, S., and Kahmann, R. (2015). Fungal effectors and plant susceptibility. *Annual Review of Plant Biology*, 66:513–545.

Loève, M. (1977). *Probability theory I*. Springer New York.

Loève, M. (1977–1978). *Probability Theory*. Number 45–46 in Graduate Texts in Mathematics. Springer-Verlag, New York, 4 edition.

Lof, M. E., De Vallavieille-Pope, C., and Van Der Werf, W. (2017). Achieving durable resistance against plant diseases: Scenario analyses with a national-scale spatially explicit model for a wind-dispersed plant pathogen. *Phytopathology*, 107:580–589.

Lof, M. E. and van der Werf, W. (2017). Modelling the effect of gene deployment strategies on durability of plant resistance under selection. *Crop Protection*, 97:10–17.

Loh, W.-Y. (2014). Fifty years of classification and regression trees. *International Statistical Review*, 82:329–348.

Lorrain, C., Gonçalves dos Santos, K. C., Germain, H., Hecker, A., and Duplessis, S. (2019). Advances in understanding obligate biotrophy in rust fungi. *New Phytologist*, 222(3):1190–1206.

Luco, N. and Bazzurro, P. (2007). Does amplitude scaling of ground motion records result in biased nonlinear structural drift responses. *Earthquake Engineering & Structural Dynamics*, 36:1813–1835.

Lum, K. and Gelfand, A. E. (2012). Spatial quantile multiple regression using the asymmetric Laplace process. *Bayesian Analysis*, 7(2):235–258.

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022a). A benchmark of basis-adaptive sparse polynomial chaos expansions for engineering regression problems. *International Journal for Uncertainty Quantification*, 12:49–74.

Lüthen, N., Marelli, S., and Sudret, B. (2022b). A spectral surrogate model for stochastic simulators computed from trajectory samples. *Submitted*.

Lüthen, N., Roustant, O., Gamboa, F., Iooss, B., Marelli, S., and Sudret, B. (2022c). Global sensitivity analysis using derivative-based sparse Poincaré chaos expansions. *Submitted*.

MacKay, D. J. C. (1999). Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068.

MacKay, D. J. C. (2013). *Information theory, inference and learning algorithms*. Cambridge University Pres.

Mackie, K. and Stojadinović, B. (2003). *Seismic demands for performance-based design of bridges*. Pacific Earthquake Engineering Research Center Berkeley.

Mai, C. V., Konakli, K., and Sudret, B. (2017). Seismic fragility curves for structures using non-parametric representations. *Frontiers of Structural and Civil Engineering*, 11(2):169–186.

Mallat, S. (2009). *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition.

Mammen, E. (1992). *When Does Bootstrap Work?* Lecture Notes in Statistics. Springer New York.

Marelli, S., Lamas, C., Konakli, K., Mylonas, C., Wiederkehr, P., and Sudret, B. (2019). UQLab user manual – Sensitivity analysis. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-106.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proceedings of the 2nd International Conference on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Marelli, S. and Sudret, B. (2018). An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*, 75:67–74.

Marelli, S. and Sudret, B. (2019). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-104.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22:833–847.

Martin, G., Aguilée, R., Ramsayer, J., Kaltz, O., and Ronce, O. (2013). The probability of evolutionary rescue: towards a quantitative comparison between theory and evolution experiments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition.

McDonald, B. A. (2004). Population genetics of plant pathogens. *The Plant Health Instructor*.

McDonald, B. A. and Linde, C. (2002). Pathogen population genetics , evolutionary potential, and durable resistance. *Annual Review of Phytopathology*, 40(1):349–379.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In *Frontiers in econometrics*, pages 104–142. Institute of Urban and Regional Development, University of California Oakland.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Mckay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input Variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ.

Melchers, R. and Beck, A. (2018). *Structural reliability analysis and prediction*. John Wiley & Sons.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446.

Merl, D., Johnson, L. R., Gramacy, R. B., and Mangel, M. (2009). A statistical framework for the adaptive management of epidemiological interventions. *PLoS ONE*, 4:e5089.

Meyer, Y. (1993). *Wavelets and Operators*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press.

Millman, D., King, P., and Beran, P. (2005). Airfoil pitch-and-plunge bifurcation behaviour with Fourier chaos expansion. *Journal of Aircraft*, 42(2):376–384.

Minh, H. Q., Niyogi, P., and Yao, Y. (2006). Mercer's theorem, feature maps, and smoothing. In Lugosi, G. and Simon, H. U., editors, *Learning Theory*, pages 154–168. Springer Berlin Heidelberg.

Modica, A. and Stafford, P. J. (2014). Vector fragility surfaces for reinforced concrete frames in Europe. *Bulletin of Earthquake Engineering*, 12:1725–1753.

Montarry, J., Bardou-Valette, S., Mabon, R., Jan, P. L., Fournet, S., Grenier, E., and Petit, E. J. (2019). Exploring the causes of small effective population sizes in cyst nematodes using artificial *Globodera pallida* populations. *Proceedings of the Royal Society B: Biological Sciences*, 286(1894):20182359.

Montarry, J., Hamelin, F. M., Glais, I., Corbière, R., and Andrivon, D. (2010). Fitness costs associated with unnecessary virulence factors and life history traits: evolutionary insights from the potato late blight pathogen *Phytophthora infestans*. *BMC Evolutionary Biology*, 10:283.

Montgomery, D. C. (2004). *Design and analysis of experiments*. John Wiley & Sons, New York.

Moran, N. A. (1992). The evolution of aphid life cycles. *Annual review of entomology*, 37(129):321–348.

Moustafa, M. and Mosalam, K. (2015). Development of Hybrid Simulation System for Multi-Degree- of-Freedom Large-Scale Testing. In *6th International Conference on Advances in Experimental Structural Engineering*, University of Illinois, Urbana-Champaign, United States.

Moustapha, M. (2016). *Adaptive surrogate models for the reliable lightweight design of automotive body structures*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France.

Moustapha, M., Lataniotis, C., Wiederkehr, P., Wagner, P.-R., Wicaksono, D., Marelli, S., and Sudret, B. (2019). UQLib User Manual. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.3-201.

Moustapha, M., Marelli, S., and Sudret, B. (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety*, 96:102714.

Moustapha, M., Sudret, B., Bourinet, J.-M., and Guillaume, B. (2018). Comparative study of Kriging and support vector regression for structural engineering applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 4(2):04018005.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:116–155.

Mundt, C. C. (2002). Use of multiline cultivars and cultivar mixtures for disease management. *Annual Review of Phytopathology*, 40(1):381–410.

Mundt, C. C. (2014). Durable resistance: A key to sustainable management of pathogens and pests. *Infection, Genetics and Evolution*, 27:1567–1348.

Murcia, J. P., Réthoré, P. E., Dimitrov, N., Natarajan, A., Sørensen, J. D., Graf, P., and Kim, T. (2018). Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. *Renewable Energy*, 119:910–922.

Nataf, A. (1962). Détermination des distributions dont les marges sont données. *Comptes Rendus de l'Academie des Sciences*, 225:42–43.

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function. *Biometrika*, 74(2):221–232.

Newey, W. K. and McFadden, D. (1994). Chapter 36 large sample estimation and hypothesis testing. In *Handbook of Econometrics*, volume 4, pages 2111–2245. Elsevier.

Nielson, B. G. and DesRoches, R. (2007). Seismic fragility methodology for highway bridges using a component level approach. *Earthquake Engineering & Structural Dynamics*, 36(6):823–839.

Nilusmas, S., Mercat, M., Perrot, T., Djian-Caporalino, C., Castagnone-Sereno, P., Touzeau, S., Calcagno, V., and Mailleret, L. (2020). Multiseasonal modelling of plant-nematode interactions reveals efficient plant resistance deployment strategies. *Evolutionary Applications*, 13(9):2206–2221.

Noh, H. Y., Lallemant, D., and Kiremidjian, A. S. (2015). Development of empirical and analytical fragility functions using kernel smoothing methods. *Earthquake Engineering & Structural Dynamics*, 44:1163–1180.

O'Hagan, A. (2019). Expert knowledge elicitation: Subjective but scientific. *The American Statistician*, 73:69–81.

Oleksyk, T. K., Smith, M. W., and O'Brien, S. J. (2010). Genome-wide scans for footprints of natural selection. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365:185–205.

Ostfeld, R. S. and Keesing, F. (2012). Effects of host diversity on infectious disease. *Annual Review of Ecology, Evolution, and Systematics*, 43(1):157–182.

Pacific Earthquake Engineering and Research Center (2004). *OpenSees: The Open System for Earthquake Engineering Simulation*.

Pacilly, F. C. A., Hofstede, G. J., Lammerts van Bueren, E. T., Kessel, G. J. T., and Groot, J. C. J. (2018). Simulating crop-disease interactions in agricultural landscapes to analyse the effectiveness of host resistance in disease control: The case of potato late blight. *Ecological Modelling*, 378:1–12.

Papaïx, J., Adamczyk-Chauvat, K., Bouvier, A., Kiêu, K., Touzeau, S., Lannou, C., and Monod, H. (2014). Pathogen population dynamics in agricultural landscapes: The Ddal modelling framework. *Infection, Genetics and Evolution*, 27:509–520.

Papaïx, J., Burdon, J. J., Zhan, J., and Thrall, P. H. (2015). Crop pathogen emergence and evolution in agro-ecological landscapes. *Evolutionary Applications*, 8(4):385–402.

Papaïx, J., Rimbaud, L., Burdon, J. J., Zhan, J., and Thrall, P. H. (2018). Differential impact of landscape-scale strategies for crop cultivar deployment on disease dynamics, resistance durability and long-term evolutionary control. *Evolutionary Applications*, 11(5):705–717.

Peng, B. and Kimmel, M. (2005). simuPOP: A forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687.

Perkins, T. A., Phillips, B. L., Baskett, M. L., and Hastings, A. (2013). Evolution of dispersal and life history interact to drive accelerating spread of an invasive species. *Ecology Letters*, 16(8):1079–1087.

Person, C., Samborski, D. J., and Rohringer, R. (1962). The gene-for-gene concept. *Nature*, 194:561–562.

Persoons, A., Hayden, K. J., Fabre, B., Frey, P., De Mita, S., Tellier, A., and Halkett, F. (2017). The escalatory Red Queen: Population extinction and replacement following arms race dynamics in poplar rust. *Molecular Ecology*.

Peterson, P. D. (2018). The barberry eradication program in Minnesota for stem rust control: a case study. *Annual Review of Phytopathology*, 56:203–223.

Pietravalle, S., Lemarié, S., and Van Den Bosch, F. (2006). Durability of resistance and cost of virulence. *European Journal of Plant Pathology*, 114:107–116.

Pilet-Nayel, M. L., Moury, B., Caffier, V., Montarry, J., Kerlan, M. C., Fournet, S., Durel, C. E., and Delourme, R. (2017). Quantitative resistance to plant pathogens in pyramiding strategies for durable crop protection. *Frontiers in Plant Science*, 8:1838.

Pink, D. and Puddephat, I. (1999). Deployment of disease resistance genes by plant transformation - a 'mix and match' approach. *Trends in Plant Science*, 4(2):71–75.

Plumlee, M. and Tuo, R. (2014). Building accurate emulators for stochastic simulations via quantile Kriging. *Technometrics*, 56:466–473.

Priestley, M. B. (1965). Evolutionary spectra and non-stationary processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):204–229.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Reddy, K. and Clinton, V. (2016). Simulating stock prices using geometric Brownian motion: Evidence from Australian companies. *Australasian Accounting, Business and Finance Journal*, 10(3):23–47.

Reed, M. and Simon, B. (1972). *Methods of modern mathematical physics*, volume 1: Functional analysis. Academic press.

Ren, Y., Li, J., Luo, Y., and Zhu, J. (2016). Conditional generative moment-matching networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*, pages 2936–2944.

Rezaeian, A. and Der Kiureghian, A. (2008). A stochastic ground motion model with separable temporal and spectral nonstationarities. *Earthquake Engineering & Structural Dynamics*, 37:1565–1584.

Rezaeian, A. and Der Kiureghian, A. (2010). Simulation of synthetic ground motions for specified earthquake and site characteristics. *Earthquake Engineering & Structural Dynamics*, 39:1155–1180.

Riesz, M. (1923). Sur le problème des moments et le théorème de parseval correspondant. *Acta Scientiarum Mathematicarum*, 1:209–225.

Riihimäki, J. and Vehtari, A. (2014). Laplace approximation for logistic Gaussian process density estimation and regression. *Bayesian Analysis*, 9(2):425–448.

Rimbaud, L., Fabre, F., Papaïx, J., Moury, B., Lannou, C., Barrett, L. G., and Thrall, P. H. (2021). Models of plant resistance deployment. *Annual Review of Phytopathology*, 59.

Rimbaud, L., Papaïx, J., Barrett, L. G., Burdon, J. J., and Thrall, P. H. (2018). Mosaics, mixtures, rotations or pyramiding: What is the optimal strategy to deploy major gene resistance? *Evolutionary Applications*, 11:1791–1810.

Rokach, L. and Maimon, O. (2005). Decision trees. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 165–192. Springer.

Romano, Y., Patterson, E., and Candès, E. (2019). Conformalized quantile regression. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NIPS 2019)*, volume 32.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23:470–472.

Rouah, F. D. (2013). *The Heston Model in Matlab and C#*. John Wiley & Sons, New Jersey.

Roustant, O., Barthe, F., and Iooss, B. (2017). Poincaré inequalities on intervals – application to sensitivity analysis. *Electronic Journal of Statistics*, 11(2):3081–3119.

Rouxel, T. and Balesdent, M. H. (2017). Life, death and rebirth of avirulence effectors in a fungal pathogen of Brassica crops, *Leptosphaeria maculans*. *New Phytologist*, 214(2):526–532.

Sacristán, S. and García-Arenal, F. (2008). The evolution of virulence and pathogenicity in plant pathogen populations. *Molecular Plant Pathology*, 9(3):369–384.

Sadler, W. A. and Smith, M. H. (1985). Estimation of the response error relationship in immunoassay. *Clinical Chemistry*, 31:1802–1805.

Saltelli, A., editor (2008). *Global sensitivity analysis: the primer*. John Wiley, Chichester, England ; Hoboken, NJ.

Saltelli, A., Chan, K., and Scott, E. M., editors (2000). *Sensitivity analysis*. John Wiley & Sons.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Sapoukhina, N., Durel, C. E., and Le Cam, B. (2009). Spatial deployment of gene-for-gene resistance governs evolution and spread of pathogen populations. *Theoretical Ecology*, 2(4):229–238.

Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., and Thornton, P. (2014). Dimensionality reduction for complex models via Bayesian compressive sensing. *International Journal for Uncertainty Quantification*, 4(1):63–93.

Saubin, M., de Mita, S., Zhu, X., Sudret, B., and Halkett, F. (2021). Impact of ploidy and pathogen life cycle on resistance durability. *Peer Community Journal – Evolutionary Biology*, 1.

Sauder, T., Marelli, S., Larsen, K., and Sørensen, A. J. (2018). Active truncation of slender marine structures: Influence of the control system on fidelity. *Applied Ocean Research*, 74:154–169.

Saul, A. D., Hensman, J., Vehtari, A., and Lawrence, N. D. (2016). Chained Gaussian processes. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 1431–1440.

Schellenberg, A. H., Mahin, S. A., and Fenves, G. L. (2009). Advanced Implementation of Hybrid Simulation. Technical Report PEER 2009/104, Pacific Earthquake Engineering Research Center, University of California, Berkeley.

Schöbi, R. (2017). *Surrogate models for uncertainty quantification in the context of imprecise probability modelling*. PhD thesis, ETH Zurich (Swizerland).

Schöbi, R., Sudret, B., and Wiart, J. (2015). Polynomial-chaos-based Kriging. *International Journal for Uncertainty Quantification*, 5(2):171–193.

Schölkopf, B. and Smola, A. S. (2002). *Learning with kernels*. MIT Press.

Schwab, C. and Todor, R. A. (2006). Karhunen–loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122.

Sesia, M. and Romano, Y. (2021). Conformal prediction using conditional histograms. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., and Vaughan, J. W., editors, *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, pages 6304–6315.

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Seyedi, D. M., Gehl, P., Douglas, J., Davenne, L., Mezher, N., and Ghavamian, S. (2010). Development of seismic fragility surfaces for reinforced concrete buildings by means of nonlinear time-history analysis. *Earthquake Engineering & Structural Dynamics*, 39:91–108.

Shattock, A. J., Le Rutte, E. A., Dünner, R. P., Sen, S., Kelly, S. L., Chitnis, N., and Penny, M. A. (2022). Impact of vaccination and non-pharmaceutical interventions on sars-cov-2 dynamics in switzerland. *Epidemics*, 38:100535.

Shinozuka, M. and Deodatis, G. (1991). Simulation of stochastic processes by spectral representation. *Applied Mechanics Reviews*, 44(4):191–204.

Shinozuka, M., Feng, M., Lee, J., and Naganuma, T. (2000). Statistical analysis of fragility curves. *Journal of Engineering Mechanics*, 126:1224–1231.

Shreve, S. (2004). *Stochastic Calculus for Finance II*. Springer, New York.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Number 26 in Monographs on statistics and applied probability. Chapman and Hall, London.

Skelsey, P., Rossing, W. A. H., Kessel, G. J. T., and Van Der Werf, W. (2010). Invasion of *Phytophthora infestans* at the landscape level: How do spatial scale and weather modulate the consequences of spatial heterogeneity in host resistance? *Phytopathology*, 100(11):1146–1161.

Slot, R. M. M., Sørensen, J. D., Sudret, B., Svenningsen, L., and Thogersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162.

Smerzini, C. and Pitilakis, K. (2018). Seismic risk assessment at urban scale from 3D physics-based numerical modeling: the case of Thessaloniki. *Bulletin of Earthquake Engineering*, 16:2609–2631.

Smith, G. D. (1985). *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press, 3rd edition.

Smolyak, S. A. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics Doklady*, 4:240–243.

Snelson, E. and Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Proceedings of the 18th International Conference on Advances in Neural Information Processing Systems (NIPS 2005)*, volume 18.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NIPS 2012), Colorado, USA*, pages 2951–2959. Curran Associates, Inc.

Sobol', I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Computational Mathematics and Mathematical Physics*, 7:86–112.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical and Computer Modelling*, 1:407–414.

Soize, C. (2017). *Uncertainty quantification*. Springer-Verlag GmbH.

Soize, C. and Ghanem, R. G. (2004). Physical systems with random uncertainties: Chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26(2):395–410.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09), Montreal, Quebec, Canada*, pages 961–968.

Soularue, J. P., Robin, C., Desprez-Loustau, M. L., and Dutech, C. (2017). Short rotations in forest plantations accelerate virulence evolution in root-rot pathogenic fungi. *Forests*, 8(6):205.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637.

Steinwart, I. and Scovel, C. (2012). Mercer's theorem on general domains: On the interaction between measures, kernels, and rkhss. *Constructive Approximation*, 35:363–417.

Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118–171.

Stukenbrock, E. H. and McDonald, B. A. (2009). Population genetics of fungal and oomycete effectors involved in gene-for-gene interactions. *Molecular Plant-Microbe Interactions*, 22(4):371–380.

Su, S. (2007). Numerical maximum log likelihood estimation for generalized lambda distributions. *Computational Statistics & Data Analysis*, 51(8):3983–3998.

Sudret, B. (2007). Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods. Habilitation à diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France (229 pages).

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93:964–979.

Sudret, B. (2015). Polynomial chaos expansions and stochastic finite element methods. In Phoon, K.-K. and Ching, J., editors, *Risk and Realiability in Geotechnical Engineering*, Risk and Reliability in Geotechnical Engineering, chapter 6, pages 265–300. Taylor and Francis.

Sudret, B., Berveiller, M., and Lemaire, M. (2006). A stochastic finite element procedure for moment and reliability analysis. *European Journal of Computational Mechanics*, 15(7–8):825–866.

Sullivan, T. J. (2015). *Introduction to uncertainty quantification*. Springer-Verlag GmbH.

Taflanidis, A. A. and Beck, J. L. (2009). Life-cycle cost optimal design of passive dissipative devices. *Structural Safety*, 31(6):508–522.

Takeuchi, I., Le, Q. V., Sears, T. D., and Smola, A. J. (2006). Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7(45):1231–1264.

Talagrand, M. (1987). The Glivenko-Cantelli problem. *Annals of Probability*, 15:837–870.

Terauchi, R. and Yoshida, K. (2010). Towards population genomics of effector-effector target interactions. *New Phytologist*, 187(4):929–939.

Thomson, W. (1901). I. nineteenth century clouds over the dynamical theory of heat and light. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(7):1–40.

Thrall, P. H., Barrett, L. G., Dodds, P. N., and Burdon, J. J. (2016). Epidemiological and evolutionary outcomes in gene-for-gene and matching allele models. *Frontiers in Plant Science*, 6:1084.

Thrall, P. H. and Burdon, J. J. (2003). Evolution of virulence in a plant host-pathogen metapopulation. *Science*, 299:1735–1737.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tokdar, S. T. and Ghosh, J. K. (2007). Posterior consistency of logistic gaussian process priors in density estimation. *Journal of Statistical Planning and Inference*, 137(1):34–42.

Torossian, L., Picheny, V., Faivre, R., and Garivier, A. (2020). A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety*, 201.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019a). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019b). A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas. *Probabilistic Engineering Mechanics*, 55:1–16.

Trigub, R. M. and Belinsky, E. S. (2004). *Fourier Analysis and Approximation of Functions*. Springer, Dordrecht, Netherlands.

Tröndle, T., Lilliestam, J., Marelli, S., and Pfenninger, S. (2020). Appropriate technology: The relationship between geographic scale, cost, and technology mix of fully renewable electricity systems in europe. *Joule*, 4:1929–1948.

Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666.

Tsokanas, N., Zhu, X., Abbiati, G., Marelli, S., Sudret, B., and Stojadinović, B. (2021). A global sensitivity analysis framework for hybrid simulation with stochastic substructures. *Frontiers in Built Environment*, 7:778716.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, Cambridge, New York.

Tucker, H. G. (1959). A generalization of the Glivenko–Cantelli theorem. *The Annals of Mathematical Statistics*, 30(3):828–830.

Vamvatsikos, D. and Cornell, C. (2002). Incremental dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 31:491–514.

van de Geer, S. (2000). *Empirical Processes in M-Estimation*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

Van den Bosch, F. and Gilligan, C. A. (2003). Measures of durability of resistance. *Phytopathology*, 93(5):616–625.

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

van Rossum, G. (1995). Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI).

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Villani, C. (2009). *Optimal Transport, Old and New*. Springer, Berlin.

Vilsen, S. A., Sauder, T., Sørensen, A. J., and Føre, M. (2019). Method for Real-Time Hybrid Model Testing of ocean structures: Case study on horizontal mooring systems. *Ocean Engineering*, 172:46–58.

Vlachos, C., Papakonstantinou, K. G., and Deodatis, G. (2016). A multi-modal analytical non-stationary spectral model for characterization and stochastic simulation of earthquake ground motions. *Soil Dynamics and Earthquake Engineering*, 80:177–191.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer-Verlag.

Wagner, P.-R., Fahrni, R., Klippel, M., Frangi, A., and Sudret, B. (2020). Bayesian calibration and sensitivity analysis of heat transfer models for fire insulation panels. *Engineering Structures*, 205:110063.

Wagner, P.-R., Marelli, S., Papaioannou, I., Straub, D., and Sudret, B. (2022). Rare event estimation using stochastic spectral embedding. *Structural Safety*, 96:102179.

Wallace, B. (1975). Hard and soft selection revisited. *Evolution*, 29:465–473.

Wand, M. and Jones, M. C. (1995). *Kernel smoothing*. Chapman & Hall/CRC.

Wen, Y.-K. (1976). Method for random vibration of hysteretic systems. *J. Eng. Mech.*, 102(2):249–263.

Windrum, P., Fagiolo, G., and Moneta, A. (2007). Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation*, 10(2):1–8.

Winsberg, E. (2019). Computer simulations in science. https://plato.stanford.edu/entries/simulations-science/.

Wooldridge, J. M. (2013). *Introductory Econometrics: A Modern Approach*. Cengage Learning, 5th edition.

Xiu, D. (2010). *Numerical methods for stochastic computations – A spectral method approach*. Princeton University press.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644.

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100:9–34.

Zhan, J., Thrall, P. H., and Burdon, J. J. (2014). Achieving sustainable plant disease management through evolutionary principles. *Trends in Plant Science*, 19(9):570–575.

Zhan, J., Thrall, P. H., Papaïx, J., Xie, L., and Burdon, J. J. (2015). Playing on a aathogen's weakness: using evolution to guide sustainable plant disease control strategies. *Annual Review of Phytopathology*, 53(1):19–43.

Zhu, X., Broccardo, M., and Sudret, B. (2022). Use of generalized lambda models for seismic fragility analysis. In *Proceedings of the 8th International Symposium on Reliability Engineering and Risk Management (ISRERM), Hannover, Germany*.

Zhu, X., Broccardo, M., and Sudret, B. (2023). Seismic fragility analysis using stochastic polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 72:103413.

Zhu, X. and Sudret, B. (2018). Surrogating the response PDF of stochastic simulators using parametric & semi-parametric representations. In *MascotNum Workshop, Ecole Centrale Nantes (France), March 21-23 (poster)*.

Zhu, X. and Sudret, B. (2019a). Emulating the response PDF of stochastic simulators using sparse generalized lambda models. In *MascotNum Workshop, IFPEN (Rueil-Malmaison, France), March 18-20 (poster)*.

Zhu, X. and Sudret, B. (2019b). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. In *9th International Conference on Sensitivity Analysis of Model Output (SAMO 2019), Universitat Oberta de Catalunya, Barcelona, Spain*. (Talk given by B. Sudret).

Zhu, X. and Sudret, B. (2019c). Surrogating the response PDF of stochastic simulators using generalized lambda distributions. In Song, J., editor, *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13), Seoul, South Korea*.

Zhu, X. and Sudret, B. (2019d). Use of generalized lambda distributions to emulate stochastic simulators. In *3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019), Crete Island, Greece*.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10:249–275.

Zhu, X. and Sudret, B. (2021a). Construction of sparse polynomial chaos surrogate models for simulators with mixed continuous and categorical variables. In *4th International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2021), Athens, Greece, June 24-26*.

Zhu, X. and Sudret, B. (2021b). Emulating the response distribution of stochastic simulators. In *MascotNum Workshop, Centre Paul Langevin, Aussois (France), April 28-30*. (Talk given by X. Zhu).

Zhu, X. and Sudret, B. (2021c). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification*, 9:1345–1380.

Zhu, X. and Sudret, B. (2021d). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815.

Zhu, X. and Sudret, B. (2021e). Metamodels of stochastic simulators using polynomial chaos expansions with latent variables. In *Engineering Mechanics Institute Conference and Probabilistic Mechanics and Reliability Conference (EMI/PMC 2021), Columbia University, New-York, USA*. (Talk given by B. Sudret).

Zhu, X. and Sudret, B. (2021f). Stochastic polynomial chaos expansions for emulating stochastic simulators. In *MascotNum Workshop on Stochastic Simulators, March 11th*.

Zhu, X. and Sudret, B. (2022a). Extension of polynomial chaos expansions to the metamodeling of stochastic simulators. In *SIAM Conference on Uncertainty Quantification (UQ22), Atlanta, USA*.

Zhu, X. and Sudret, B. (2022b). Introducing latent variables in polynomial chaos expansions to surrogate stochastic simulators. In Li, J., Spanos, P. D., Chen, J. B., and Peng, Y. B., editors, *Proceedings of the 13th International Conference on Structural Safety and Reliability (ICOSSAR 2021-2022), Shanghai, China*.

Zhu, X. and Sudret, B. (2023). Stochastic polynomial chaos expansions to emulate stochastic simulators. *International Journal for Uncertainty Quantification*, 13:31–52.

Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. (2013). *The Finite Element Method: its Basis and Fundamentals*. Butterworth-Heinemann, 7th edition.

Zou, H. (2006). The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533.