# Sparse spectral surrogate models for deterministic and stochastic computer simulations

Nora Lüthen

▶ **To cite this version:**

Nora Lüthen. Sparse spectral surrogate models for deterministic and stochastic computer simulations. Statistics [math.ST]. ETH Zürich, 2022. English. NNT : . tel-04284700

**HAL Id: tel-04284700**
**https://hal.science/tel-04284700**

Submitted on 20 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

DISS. ETH NO. 28626

# Sparse spectral surrogate models for deterministic and stochastic computer simulations

*A thesis submitted to attain the degree of*

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

*presented by*

Nora Lüthen
M.Sc. Mathematics, University of Bonn
born on 26.03.1991

*accepted on the recommendation of*

Prof. Dr. Bruno Sudret, examiner
Dr. Sebastién Da Veiga, co-examiner
Dr. Stefano Marelli, co-examiner
Prof. Dr. Olivier Roustant, co-examiner

*15.09.2022*

# Abstract

Computer simulations are an invaluable tool for modeling and investigating real-world phenomena and processes. However, as any model, simulations are affected by uncertainty caused by imperfect knowledge or natural variability of their parameters, initial conditions, or input values. This leads to uncertainty in the model response, which needs to be quantified to make subsequent conclusions and decisions trustworthy. To alleviate the considerable cost of uncertainty analyses for expensive computational models, the latter are often replaced by surrogates, i.e., by approximations with an explicit functional form that can be created based on a rather small number of model evaluations, and can be evaluated at low cost.

Some computer models are affected by uncertainty only through their input parameters: for fixed values of the inputs, they always produce the same response. These models are called deterministic simulators. In contrast, models that feature inherent stochasticity are called stochastic simulators. The latter generate a different result each time they are run even if their input parameters are held at fixed values. In other words, they behave like random fields whose index set is the space of input parameters.

In this thesis, we investigate spectral surrogate models, which are a class of global non-intrusive methods that expand the computational model onto an orthonormal basis of a suitable function space. We focus on sparse expansions, i.e., representations that only include a small finite subset of the basis elements. Sparse representations are typically computed by regression with sparsity-encouraging constraints, often using ideas originating from the field of compressed sensing.

In particular, for deterministic simulators we explore the popular sparse polynomial chaos expansions (PCE) method, which utilizes a polynomial basis that is orthonormal with respect to the distribution of the input variables. We conduct an extensive literature survey as well as a benchmark of several promising methods on multiple models of varying dimensionality and complexity. The benchmark results are aggregated and visualized in a novel way to extract reliable recommendations about which methods should be used in practice.

We also investigate the recently proposed Poincaré chaos expansions, which rely on a generally non-polynomial basis consisting of eigenfunctions of a specific differential operator connected to the Poincaré inequality. By construction, this basis is well suited for derivative-based global sensitivity analysis, which we explore both analytically and numerically.

Furthermore, we propose a new surrogate model for stochastic simulators. Taking the random function view of a stochastic simulator, we approximate its trajectories by sparse PCE and perform Karhunen-Loève expansion on them. The latter is a well-known spectral representation for a random field which separately characterizes its spatial and stochastic variation. The joint distribution of the random coefficients is inferred using the marginal-copula framework. The resulting surrogate model is able to approximate marginal distributions, mean, and covariance function of the stochastic simulator, and can generate new trajectories.

## Zusammenfassung

Computersimulationen sind ein wertvolles Werkzeug für die Modellierung und Erforschung realer Phänomene und Prozesse. Allerdings sind Simulationen, wie alle Modelle, von Ungewissheit betroffen, die durch fehlende Informationen oder durch die natürliche Variabilität ihrer Parameter, Anfangsbedingungen oder Eingabewerte verursacht wird. Dies führt zu Ungewissheit in der Modellantwort, welche quantifiziert werden muss, damit anschliessende Schlussfolgerungen und Entscheidungen verlässlich sind. Um die beträchtlichen Kosten von Ungewissheitsanalysen für teure Computermodelle zu verringern, werden letztere häufig durch Ersatzmodelle (engl. surrogates) angenähert, d.h. durch Funktionen mit expliziter analytischer Form, die mithilfe weniger Modellauswertungen erstellt und kostengünstig ausgewertet werden können.

Einige Computermodelle sind nur durch ihre Eingabeparameter von Ungewissheit betroffen: für eine gegebene Kombination von Eingabewerten erzeugen sie immer dieselbe Ausgabe. Solche Modelle heissen deterministische Simulatoren. Im Gegensatz dazu werden Modelle mit inhärenter Stochastizität stochastische Simulatoren genannt. Diese erzeugen bei jeder Auswertung eine andere Ausgabe, selbst wenn die Eingabewerte immer gleich sind. Mit anderen Worten, sie verhalten sich wie Zufallsfelder (engl. random fields), wobei der Raum der Eingabewerte als Indexmenge fungiert.

In dieser Arbeit untersuchen wir spektrale Ersatzmodelle, eine Klasse globaler nicht-intrusiver Methoden, welche das Computermodell mithilfe einer orthonormalen Basis eines geeigneten Funktionenraums darstellen. Wir konzentrieren uns auf dünnbesetzte (engl. sparse) Darstellungen, d.h. solche, die nur eine geringe endliche Anzahl Basisfunktionen verwenden. Dünnbesetzte Darstellungen erhält man üblicherweise mithilfe von Regression mit Nebenbedingungen, welche die Dünnbesetztheit begünstigen, wobei oft Ideen aus dem Gebiet der komprimierten Erfassung (engl. compressed sensing) verwendet werden.

Für deterministische Simulatoren haben wir uns speziell mit der beliebten Methode der dünnbesetzten polynomiellen Chaos-Expansion (PCE) beschäftigt, welche eine polynomielle Basis verwendet, die orthonormal bezüglich der Eingabeverteilung ist. Wir erstellen eine umfassende Literaturübersicht und führen eine Vergleichsstudie vielversprechender Methoden auf mehreren Modellen unterschiedlicher Dimension und Komplexität durch. Die Ergebnisse werden auf eine neue Weise zusammengeführt und visualisiert, um zuverlässige Empfehlungen zur Methodenwahl in der Praxis zu erhalten.

Wir untersuchen auch die relativ neue Methode der Poincaré-Chaos-Expansion, die eine im Allgemeinen nicht polynomielle Basis verwendet, welche aus Eigenfunktionen eines Differenzialoperators mit Verbindungen zur Poincaré-Ungleichung besteht. Diese Basis ist per Definition gut für die gradientenbasierte globale Sensitivitätsanalyse geeignet, was wir sowohl analytisch als auch numerisch untersuchen.

Ausserdem entwickeln wir ein neues Ersatzmodell für stochastische Simulatoren. Unter Verwendung der Zufallsfunktionen-Sicht eines stochastischen Simulators nähern wir dessen Trajektorien mit dünnbesetzten PCE an und berechnen die Karhunen-Loève-Expansion derselben. Letztere ist eine bekannte spektrale Darstellung für Zufallsfelder, welche die räumliche und stochastische Variabilität getrennt modelliert. Die multivariate Verteilung der zufälligen Koeffizienten wird

mithilfe von Marginalverteilungen und Copulas bestimmt. Das resultierende Ersatzmodell kann Marginalverteilungen sowie Mittelwert- und Kovarianzfunktion des stochastischen Simulators annähern und neue Trajektorien erzeugen.

## Acknowledgments

Life is random and uncertain. Whether or not someone has the opportunity to achieve something is influenced by many random events and decisions. Of course, my decision to pursue a PhD in the group of Prof. Bruno Sudret at ETH Zürich was a conscious and determined one, but there were lots of chances and lucky coincidences leading up to it. Certainly, many people had (and are having) a great and deterministic impact on my life, changing it to the better. I feel extraordinarily lucky and grateful to have them in my life.

First and foremost, I want to express my sincere gratitude to my supervisor Prof. Bruno Sudret. Thank you for accepting me into your group and for giving me the opportunity to work with you on these exciting problems. Thank you for your guidance, encouragement, and trust throughout these years, and for all the opportunities (writing, teaching, presentations, collaborations) and feedback you provided that allowed me to develop my skills and grow professionally. Working with you and in your group has been an exceptionally positive experience, and I am very grateful for it.

I was very lucky to have Dr. Stefano Marelli as my second supervisor. Thank you Stefano for the patient support and sound advice in all matters of PhD life, for the (sometimes heated, always super interesting) scientific discussions and valuable feedback, and for always having an "offenes Ohr" and being extremely generous and helpful whenever needed. I feel very lucky that I had the opportunity to work with you and learn from you during these years.

Furthermore, I want to thank Prof. Sebastién Da Veiga and Prof. Olivier Roustant for serving on my doctoral committee. Thank you very much for the time and energy you put into reading my thesis and into the very interesting, insightful discussion during my defense. To be honest, I would not have minded discussing with you even longer!

I would like to thank Prof. Olivier Roustant and Prof. Fabrice Gamboa for our inspiring collaboration. Thank you for giving me the opportunity to stay a month with you in Toulouse, and for the time and energy you took for our discussions. It is an honor to work with you!

I had the pleasure of spending my PhD years with a wonderful group of smart, fun and friendly colleagues. I immensely enjoyed our lunch discussions, coffee breaks, dinners, and all our other shared adventures. Thank you Adéla, Anderson, Andréia, Biswarup, Chiara, Christos, Damar, Diego, Emiliano, Fritz, Katerina, Marco, Margaretha, Max, Moustapha, Nicole, Paul, Philippe, René, Roberto, Rui, Stefano, Styfen, Tong, and Xujia! I especially want to thank Xujia, my PhD sibling: I am so happy that we were hired for the same project. Thank you for our many interesting discussions and shared highs and lows. I very much enjoyed going on this journey with you! Paul, my officemate for more than three years, thank you so much for integrating me into the group and to the Civil Engineering world in general, and for all the fun times and useful advice! Chiara, thank you for your moral and culinary support during the last stressful months—it was a pleasure to share the office with you!

Furthermore, I would like to thank Dr. Hélène Schernberg for taking the time to share her valuable experience and thoughtful advice regarding science and life choices with me. This has helped and continues helping me immensely.

During my studies, I had the honor of working with Prof. Martin Rumpf and his group who guided me on my first small steps in mathematics research—thank you for this enjoyable experience that inspired me to pursue a PhD! Furthermore, I would like to thank my teacher Ralf Wambach, who nearly 20 years ago patiently started encouraging me to join mathematics camps and competitions. Without him, I would almost surely have followed a different trajectory.

Of course, work needs to be balanced with free time, and I am grateful to all my wonderful friends in Switzerland and abroad for providing this balance. Thank you for distracting me with dinners, hikes, gardening, skiing, Ultimate Frisbee, or phone calls whenever the work thoughts became to heavy, for keeping in touch and sharing the ups and downs, and generally for all the good times we had during all the last years!

I feel very grateful to have the best family one could wish for. Thank you to my parents and sisters for believing in me and always providing the right loving kind of understanding, encouragement, and advice.

Finally, thank you Jerri for your constant and patient support, the good food you cook and your wonderful presence in my life.

# Contents

CHAPTER 1

# Overview

## 1.1 Motivation and context

Computer technology has made incredible progress. Just 80 years ago, only a handful people in the world were concerned with electronic computing machines. The first computers of that time were big devices filling whole rooms, and to enter a program into such a machine, the operators had to physically manipulate its wires and switches. One generation ago, when my parents wrote their dissertations using typewriters, computer technology had already advanced a lot. Powerful operating systems and programming languages had been developed, the first personal computers had become available, and around the time I was born, the world wide web started out as a project of a few especially tech-savvy people. Nowadays, the majority of people in our society owns at least a smartphone or a personal computer, not speaking of the various other types of smart devices from watches to cars that are becoming increasingly prevalent. Many of us use the internet every day, even depend on it to do their work and stay connected with friends and family, and a large number of companies and government institutions would not be able to operate anymore without computers. Computers are everywhere.

In addition to the well-known developments that came with computers, such as automation, big data, and anything related to the internet from Wikipedia to social media, there is another domain in which computers critically influence our everyday lives. It is perhaps less familiar to the lay-person, but no less important: namely, the domain of *computer simulation*. This broad term describes the replication of real-world processes and phenomena by mathematical models solved by computer calculations, exploiting the fact that computers can perform billions of calculation steps and process huge amounts of data per second. To name some prominent examples, computer simulations play an essential role in weather forecasting, climate science, and the aerospace industry. Recently, computer simulations of the spreading of infectious diseases have attracted media attention due to the COVID-19 pandemic.

Yet computer simulations, also called *computer models*, are used in many more fields and for

many more purposes. Nowadays, there is hardly a scientific field that does not enrich its theoretical and experimental research with insights and predictions gained from computer models. In industry, products and processes are optimized using computer simulation. Car manufacturers optimize the safety and cost of critical components. Insurance companies use computer models to assess risks and adjust their contract pricing. Even in politics, computer simulations can be used to aid policy decisions. Much of our modern, comfortable standard of living, counting on affordable products and services and reliable infrastructure, is facilitated by computer simulations that have helped explore and optimize the respective system of interest.

In all of the cases described above, computer simulations are crucial because they can give us information that could not otherwise be obtained: real-world experiments would be unethical, too expensive or impossible. At the same time, while it would theoretically be possible to perform the required calculations by hand, this would take many (hundreds of) years.

Of course, to be able to rely on the predictions and responses of a computer simulation, we need to be able to trust that the values it returns are reasonable. This concerns the mathematical equations used for the model, the algorithm used to solve them, and the implementation in computer code: the equations should describe the phenomenon well, the algorithm should be suitable, and the implementation should be free of bugs.

However, even if the equations, the algorithm, and the implementation are confirmed to be adequate, there are further sources of inaccuracy or uncertainty that can influence the predictions of a computer simulation. Every model depends on some parameters and input values that describe the properties and characteristics of the specific phenomenon of interest: e.g., in a space flight simulation, such parameters would include the weight, speed, and dimensions of the spacecraft. In an epidemics simulation, relevant parameters would be the infection and recovery rates as well as the size of the potentially affected population. However, the natural world is so complex that we can hardly ever know these parameter values exactly. In other words, the parameters are affected by uncertainty.

To illustrate this, let us consider an example from structural engineering, namely the compressive strength of concrete. Concrete is one of the most widely used building materials of our time, and its strength is a crucial parameter in any simulation of a concrete structure. Perhaps surprisingly (at least to non-engineers), it turns out that it is virtually impossible to produce two batches of concrete that have exactly the same strength. Concrete is made of cement, water, and a granular component called aggregate, and its strength depends on the properties and amounts of all these ingredients. The same company can produce concrete with the same recipe and still end up with concrete having a widely varying strength – an uncertainty that is usually dealt with by using generous safety margins in structural design. Likewise, if the concrete strength was an input parameter to a computational model, this uncertainty would need to be taken into account. The uncertainty in the input parameters to a computational model is called *parametric uncertainty*.

Does the uncertainty in the input parameters affect the output of the computer model, and if so, how severely? In what way are our subsequent conclusions and decisions affected by this uncertainty? Answering these questions is the main focus of the field of *uncertainty quantification* (UQ). More precisely, typical questions tackled in UQ include the following:

- How uncertain is the simulation prediction as a consequence of the uncertainty in the input? (*Uncertainty propagation*)
- Which input uncertainties have the largest influence on the output uncertainty, and which ones are unimportant? (*Sensitivity analysis*)
- What is the probability that the system attains a state that is not considered safe anymore (damaged or failed)? (*Reliability analysis*)
- Which parameters are optimal for a certain objective, if the system is affected by uncertainty? (*Robust optimization*)
- Which (range of) input parameters makes the model output fit best to given observations? (*Bayesian calibration*)
- Which set of parameter combinations provides the most information about the behavior of the model? (*Optimal experimental design*)

Clearly, questions from this list arise in almost any research field that uses computational models. To answer these questions, we need to combine knowledge and ideas from various scientific areas, which makes UQ an interdisciplinary science at the intersection of fields such as computational science, numerical and functional analysis, statistics, computer science, and whichever engineering or applied science field that is interested in answering the above-mentioned questions.

To understand the behavior of a computational model under uncertainty, a single model run is not enough – it must be run many times in different configurations to explore its behavior. The most widely-known uncertainty propagation method, *Monte Carlo simulation*, does exactly this: the computer model is run for a large number of different randomly generated parameter configurations, and the resulting spread of responses is interpreted as the output uncertainty. This method is universal and can be applied readily for computer simulations that only involve a few calculations and can be run in a fraction of a second. However, the typical computer simulation in current applied science and engineering research or in industrial applications is much more resource-intensive, which often makes the Monte Carlo approach overly expensive or even infeasible. Such an expensive computer model often involves the simulation of several components, each of which requires solving complex mathematical equations over a long time scale or with a fine spatial resolution. Performing a single simulation might take hours or days. Even though the computational power of modern computing facilities is increasing steadily due to technological progress, this is compensated for by the increasingly complex state-of-the-art simulations and the associated ever-growing computational demands. Examples of such complex simulations from civil engineering are the response of high-rise buildings to earthquakes, or the power production of an off-shore wind turbine that is subject to varying wind and wave conditions.

In these situations, where it is infeasible to run the computational model sufficiently many times to perform a thorough uncertainty analysis, a convenient solution is to take a detour through a *surrogate model*. A surrogate model is a mathematical function that approximates the input-output relationship of the original computational model. Being a model of a model, it is also sometimes called *metamodel*; other synonyms are *response surface* or *emulator*. It is often trained based on a relatively small number of samples from the original model, and can

be evaluated at a fraction of its cost. Subsequently, the inexpensive surrogate can be used for uncertainty analysis in place of the expensive original model.

If no knowledge of the original computational model except for a number of input-output pairs for training is required, we call the surrogate *non-intrusive* and the computational model a *black-box*. Non-intrusive surrogates for black-box models are the topic of this thesis. More specifically, we will investigate several instances of a specific class of surrogate models called *spectral expansions*, which combine a strong mathematical foundation with good performance in practical problems. The specific objectives of our research are explained in the following section.

## 1.2 Research objectives

Before we present the research questions investigated in this thesis, we must briefly introduce the two classes of computational models that we will consider, namely *deterministic* and *stochastic simulators*.

When investigating parametric uncertainty, it is convenient to treat the computer model as a *deterministic simulator*: there is no uncertainty within the computational model. Only the input parameters are assumed to be affected by uncertainty, and this uncertainty is modeled separately. Surrogating such a deterministic simulator can also be viewed as a function approximation task: an expensive deterministic function is to be approximated by another cheap-to-evaluate one, based on a few evaluations of the former. An introduction to surrogates for deterministic models is provided in Section 2.2.

However, there also exist computer experiments for which it is infeasible or undesirable to completely separate all the uncertainty from the computational model. For example, for stochastic differential equation models that involve a driving white noise term, it is usually not meaningful to treat the latter as a (high-dimensional) input variable, which would be necessary to make the model deterministic. Such models that are affected by additional stochasticity on top of the explicitly modeled randomness in the input variables are called *stochastic simulators*. Running such a simulator repeatedly with identical input parameter values can still result in varying responses because of the additional stochasticity. We will describe several examples of stochastic simulators in Section 2.3.

The overarching goal of this thesis is to develop cost-effective surrogate models for deterministic and stochastic simulators, and to investigate their performance and limitations.

Our methodology includes two main components: *spectral expansions* and *sparsity*. *Spectral expansions* are representations of a function in terms of an orthonormal basis of a suitable function space, and we will consider three instances: *polynomial chaos expansion* (PCE), *Poincaré chaos expansion*, and *Karhunen-Loève expansion*. The first and the last ones are well-known since the seminal work by Ghanem and Spanos (1991) establishing stochastic computation in engineering. The second one is a recent development (Roustant et al., 2017, 2020a). All details about these spectral expansions can be found in Section 2.4 below.

To represent a function in terms of a basis, it is expressed as a weighted sum of basis functions. If most of the weights are zero, i.e., if only a few basis functions are involved in the sum, we call the representation *sparse*. Sparsity is a useful concept that can achieve considerable savings in computational cost. In the domain of signal processing, it has led to the development of the fruitful field of *compressed sensing* (Donoho, 2006; Candès et al., 2006). It has also successfully been applied in the field of spectral expansions, notably for polynomial chaos expansions, resulting in *sparse PCE* (Blatman and Sudret, 2008, 2010, 2011; Doostan and Owhadi, 2011; Mathelin and Gallivan, 2012). Sparsity and compressed sensing are discussed in Section 2.5.

Conceptually, the work presented in this thesis can be divided into three parts, each part addressing a different topic in the area of surrogate modeling with sparse spectral methods:

1. **Benchmark and literature review of methods for computing sparse polynomial chaos expansions (PCE).**

   (a) The sparse PCE literature is vast, and for practitioners and newcomers it can be difficult to identify the most recent and relevant methodological developments, often resulting in suboptimal choices for their applications. Can we provide a comprehensive and accessible overview of ideas and concepts?

   (b) Which of the many sparse PCE computation techniques results in the best surrogate? How does this depend on the characteristics of the model that is to be surrogated?

   These questions are addressed in the two journal papers Lüthen et al. (2021) and Lüthen et al. (2022a) reported here in Chapters 3 and 4:

   - N. Lüthen, S. Marelli, B. Sudret (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif. 9*(2), 593–649. DOI: 10.1137/20M1315774
   - N. Lüthen, S. Marelli, B. Sudret (2022). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification 12*(3), 49–74. DOI: 10.1615/Int.J.UncertaintyQuantification.2021036153

2. **Regression-based sparse Poincaré chaos expansions.**
   What are the advantages of using a different, non-polynomial basis of the function space for the spectral expansion? Which properties does this basis have? How can derivative information be utilized for spectral surrogate modeling and sensitivity analysis?
   These questions are addressed in the submitted journal paper Lüthen et al. (2022c) provided in Chapter 5:

   - N. Lüthen, O. Roustant, F. Gamboa, B. Iooss, S. Marelli, B. Sudret (2022). Global sensitivity analysis using derivative-based sparse Poincaré chaos expansions. DOI: 10.48550/arXiv.2107.00394

3. **Surrogate modeling for stochastic simulators.**
   Sparse PCEs show good performance for deterministic simulators. How can their efficiency be exploited to build a surrogate model for stochastic simulators?
   This is addressed in the submitted journal paper Lüthen et al. (2022b) provided in Chapter 6:

   - N. Lüthen, S. Marelli, B. Sudret (2022). A spectral surrogate model for stochastic simulators computed from trajectory samples.
     DOI: 10.48550/arXiv.2207.05653

## 1.3 Outline

The thesis is organized as follows. Following this introductory chapter, we describe the current state of the art of surrogate modeling for deterministic and stochastic simulators with a focus on spectral methods and sparsity in Chapter 2. Chapters 3 to 6 contain the journal papers and preprints that resulted from our PhD research. The resulting findings and associated insights as well as connections to other research areas are discussed in Chapter 7.

CHAPTER 2

# Review of the state of the art

In this chapter, we describe the current state of the art of the most relevant concepts and methods in uncertainty quantification that constitute the foundation for the research developed during this PhD (Chapters 3 to 6). We first give an overview of probabilistic uncertainty quantification and the associated notation in Section 2.1. Then, we review the relevant literature for surrogate modeling for deterministic (Section 2.2) and stochastic simulators (Section 2.3). In both cases, our focus lies on a certain class of methods called *spectral*, which we explain in detail in Section 2.4. Another useful concept in surrogate modeling, complementing spectral methods well, is *sparsity*, which we discuss in Section 2.5. We then give an overview of sensitivity analysis methods in Section 2.6. Finally, in Section 2.7 we present available software implementations of uncertainty quantification methods.

## 2.1 Probabilistic uncertainty quantification

Mathematically, a deterministic computational model can be expressed as a function $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ that maps from the set $\mathcal{D}$ of admissible inputs to the set $\mathcal{Y}$ of possible outputs (or *responses*). Usually, it is assumed that the input domain $\mathcal{D}$ is a subset of $\mathbb{R}^d$, where $d$ is the dimension of the input space, i.e., the number of scalar parameters. In this thesis, we assume that the output range $\mathcal{Y}$ is a subset of $\mathbb{R}$, i.e., the model is *real-valued*.[1]

When performing uncertainty quantification, the first crucial step is to define and implement the computational model. In the workflow sketched in Figure 2.1 (adapted from Sudret (2007)), this is displayed as Step A. In particular, we need to define the output quantity of interest for the uncertainty analysis: it is often advantageous not to use the full set of output values the computer simulation provides, but instead a meaningful aggregate value or a set of values related to the specific question of interest.

[1] In general, the response of the model can also be vector-valued or even functional in space or time, which is often addressed by using linear (Nagel et al., 2020; Perrin et al., 2021) or nonlinear (Giovanis and Shields, 2018; Kontolati et al., 2022a) dimension reduction methods in the output space.

The second step (Step B in Figure 2.1) in the UQ workflow is to find a suitable mathematical representation of the uncertainty in the input parameters. While there exist several approaches to do so, by far the most common framework and the one used in this thesis is *probability theory* (Grigoriu, 2002).[2] We model the input of a deterministic simulator as a $d$-dimensional random vector $\boldsymbol{X}$ with finite variance and joint probability density function $f_{\boldsymbol{X}}$, where $d$ is typically in the order of ten to a hundred.[3] Here and in the following, uppercase letters denote random variables, whereas lowercase letters denote real numbers (e.g., realizations of a random variable). Boldface letters denote vectors or matrices.

Since the input is a random vector, the output $Y = \mathcal{M}(\boldsymbol{X})$ is a random variable. We assume that the model output $Y$ has finite variance, which is equivalent to the model $\mathcal{M}$ being in the function space

$$L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = \left\{ h : \mathcal{D} \to \mathbb{R} \;\middle|\; \int_{\mathcal{D}} h(\boldsymbol{x})^2 f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} < \infty \right\}, \tag{2.1}$$

i.e., being square-integrable with respect to the density $f_{\boldsymbol{X}}$. The uncertainty analysis depicted in Figure 2.1 as Step C is then conducted for the random variable $Y$. The goal of *uncertainty propagation* is to characterize the moments or even the full distribution of $Y$. *Sensitivity analysis*, introduced in more detail in Section 2.6, aims at ranking the input parameters by their importance for the variability of the model output. Further analyses could be performed such as reliability analysis, reliability-based design optimization, or Bayesian calibration of parameters from data (see also the list of UQ questions in Section 1.1); however, as they are not discussed in this thesis, they are omitted from Figure 2.1.

To reduce the computational cost of the uncertainty analysis for the original model $\mathcal{M}$, a surrogate model $\tilde{\mathcal{M}}$ might be constructed, which can provide an approximate answer to the questions of Step C. This is illustrated in Figure 2.1 in the form of an additional purple box. The construction and use of a surrogate model can be seen as a detour which aims at reducing the overall computational cost, potentially at the expense of accuracy. We discuss surrogate models for deterministic models in Section 2.2.

So far, we described the mathematical setup for deterministic simulators. In contrast, a *stochastic simulator* is a computer model $\mathcal{M}$ for which the mapping from the input variables to the response is not deterministic. As before, we assume that the $d$-dimensional input random vector $\boldsymbol{X}$ has finite variance. However, due to residual stochasticity in the model, now the output $Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x})$ is a random variable even for fixed $\boldsymbol{x} \in \mathcal{D}$. In other words, $\mathcal{M}$ can be modeled as a *random field*, i.e., a collection of random variables indexed by the set of inputs:

$\mathcal{M} : \mathcal{D} \times \Omega \to \mathbb{R}$

- For each $\boldsymbol{x} \in \mathcal{D}$, $\mathcal{M}(\boldsymbol{x}, \cdot) : \Omega \to \mathbb{R}$ is a random variable
- For each $\omega \in \Omega$, $\mathcal{M}(\cdot, \omega) : \mathcal{D} \to \mathbb{R}$ is a function (deterministic simulator)

[2] The uncertainty affecting a system is sometimes further classified as *epistemic*, i.e., due to incomplete knowledge, and *aleatory*, i.e., due to natural variability of the parameter in question. While some researchers apply different methods to the two types of uncertainty, using e.g. imprecise probability for epistemic uncertainty (Faes et al., 2021), we model both types probabilistically. Note also that the classification can depend on the point of view, as illustrated by the example of the compressive strength of concrete in Section 1.1.

[3] For an overview of the state-of-the-art treatment of very high-dimensional input in UQ, we refer to Kontolati et al. (2022b).

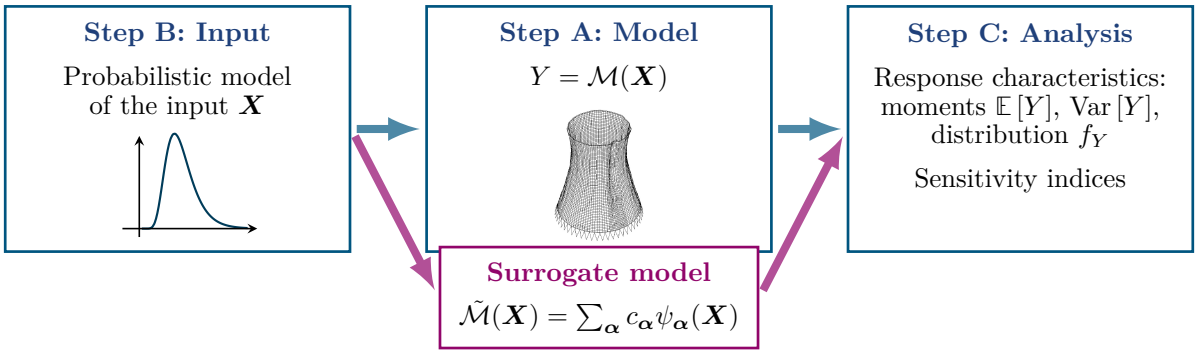



***Figure 2.1:*** *A diagram illustrating the three steps of an uncertainty analysis, adapted from Sudret (2007). An additional purple box shows the surrogate model (here illustrated with the functional form of PCE), which after its construction can be used in place of the original model to perform the UQ analysis. While there are more areas of UQ that could be included here, such as reliability analysis or Bayesian inference, this diagram focuses on the topics discussed in this thesis.*

Here, $\omega \in \Omega$ is a random event responsible for the stochasticity of the simulator. It is not explicitly modeled and instead considered to be an unobservable, latent parameter whose value is not known. In the case of computer simulations, it is usually possible to fix $\omega$ by fixing the seed of the pseudo-random number generator, which results in the same sequence of random numbers being generated. However, the seed is in general not a useful parametrization of the latent stochasticity and should not be treated as an input variable. We will discuss stochastic simulators, random fields, and the associated surrogate models in Section 2.3.

## 2.2 Surrogate modeling for deterministic simulators

We consider the case of real-valued deterministic simulators $\mathcal{M} : \mathcal{D} \to \mathcal{Y} \subset \mathbb{R}$. Uncertainty analysis is concerned with the properties of the induced random variable $Y = \mathcal{M}(\boldsymbol{X})$. There are two possible approaches to constructing a surrogate model for $Y$:

- The response random variable $Y$ could be approximated, aiming for equality in distribution: $\tilde{Y} \overset{d}{=} Y$.
- The mapping $\mathcal{M}$ could be approximated, aiming for almost sure equality $\tilde{\mathcal{M}}(\boldsymbol{X}) \overset{\text{a.s.}}{=} Y = \mathcal{M}(\boldsymbol{X})$.

The second approach is nowadays much more widespread. One reason for this might be that when constructing surrogate models for black-box models, we usually assume to have at our disposal a number of input-output pairs generated by the model. It is therefore natural to use this information to approximate the mapping itself instead of only the resulting distribution.

In the context of deterministic simulators, a surrogate model $\tilde{\mathcal{M}}$ is a mathematical function that approximates the input-output relationship of the computational model, which is considered a black-box. The surrogate model is computed based on a set of *training data* consisting of input-output pairs, also called *experimental design*, of the original deterministic simulator.[4]

[4]Due to their construction from data, such surrogate models are sometimes also called *data-fit* or *data-driven* surrogates (Robinson et al., 2008; Asher et al., 2015) to distinguish them from both reduced-order models, which

As opposed to real-world experiments, deterministic computer simulations are not affected by noise. Still, instead of interpolating the training data, most surrogate modeling methods approximate it, i.e., they determine a function that minimizes a suitable distance to the data points under some assumptions and constraints. Depending on the functional form of the surrogate, approximation can have a smoothing effect, i.e., it might capture the low-frequency components of the data and disregard the high-frequency ones. It is well-known that polynomial interpolation can be unstable unless specific experimental designs are used, which is known as Runge's phenomenon (Xiu, 2010). More generally, interpolating the training data can lead to *overfitting*, i.e., the surrogate might lose the ability to generalize to unseen points, whereas approximation is often more stable.

The various families of surrogate models differ in their functional form and in the number of parameters they involve. Determining the exact functional form and the parameter values fitting the provided data best is often called *training*, in reference to machine learning terminology. As usual, there is a trade-off between flexibility of the model and requirements on the amount of data: a function with many free parameters and a flexible functional form is able to approximate a wide variety of models, but it requires a lot of training data to avoid overfitting. On the other hand, a function making many structural assumptions and involving only a few parameters can be trained based on a small set of data, but it might be inexact if the structural assumptions do not match the true shape of the model. When knowledge about model characteristics is available, it can be used to select a surrogate model with suitable structure.

We give an overview of common surrogate models for deterministic simulators in Section 2.2.1. Then, in Section 2.2.2 we highlight some connections between surrogate modeling and (supervised) machine learning techniques. Finally, we discuss in Section 2.2.3 how the performance of a surrogate model can be assessed.

### 2.2.1 Overview of common surrogate models

In this section we give a short overview of the families of surrogate models that are typically used in UQ for deterministic simulators.[5]

*Polynomial chaos expansions* (PCE) approximate the original model by a weighted sum of multivariate polynomials, which are orthonormal with respect to the density $f_{\boldsymbol{X}}$ of the input. Under certain assumptions on $f_{\boldsymbol{X}}$, the orthonormal polynomials constitute a basis for $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ defined in Eq. (2.1) and are therefore suitable for approximating square-integrable models. One of the advantages of PCE is that moments and certain kinds of sensitivity indices can be computed analytically from its functional form (see Section 2.6). PCE is one of the main surrogate modeling techniques used and investigated in this thesis, and is explained in further detail in Section 2.4.1.1.

are simplified models utilizing knowledge about the physics and equations of the original model (Benner et al., 2015), and hierarchical (multifidelity) surrogates. However, as we explain in more detail in Section 2.2.2, the typical setup in UQ does not rely on data alone but assumes to have access to both the input distribution and the computational model from which the data is constructed (see also Section 2.1).

[5]The use of surrogates is not restricted to the field of UQ. Surrogate models are used also for other purposes such as optimization (Robinson et al., 2008; Forrester and Keane, 2009; Razavi et al., 2012).

*Kriging*, or Gaussian process modeling, is another popular surrogate technique in UQ. It has its origins in mining and geostatistics (Krige, 1951; Matheron, 1963) and was introduced for general computational models by Sacks et al. (1989). In the machine learning community, it is also known as *Gaussian process regression* (Rasmussen and Williams, 2006). In Kriging, we construct a Gaussian stochastic process that is conditioned on the given experimental design. Depending on the formulation, the resulting process can be interpolating or approximating. The choices of covariance function and associated hyperparameters govern the regularity of the realizations. The final prediction, i.e. the surrogate, is usually the mean of the inferred process. Being a stochastic process, Kriging provides confidence bounds for the prediction, which can be used for goal-oriented *active learning*, i.e., for the sequential enrichment of the experimental design with additional informative points. Kriging is popular because of its suitability for active learning and because its covariance function is a *kernel*, which is a powerful mathematical concept allowing to work even with non-numeric data. A drawback of Kriging is its unfavorable scaling with the number of experimental design points: to compute a Kriging metamodel, a dense covariance matrix needs to be inverted, which becomes prohibitively expensive for large experimental designs. Many of the approaches for surrogate modeling of stochastic simulators in the literature rely on Kriging in some way (see Section 2.3.3).

As a hybrid between the two types of approaches, polynomial chaos-Kriging or *PC-Kriging* has been developed (Kersaudy et al., 2015; Schöbi et al., 2015). Here, a Kriging metamodel is constructed whose trend function is a PCE, thus combining the global behavior of PCE with the local correction capabilities of Kriging.

Several other types of surrogate models have been proposed, including low-rank tensor approximations, also known as canonical decompositions (Nouy, 2010; Chevreuil et al., 2015; Konakli and Sudret, 2016), support vector machines (Smola and Schölkopf, 2004; Bourinet et al., 2011), regression trees (Breiman et al., 1984) and other localized regression methods (Marelli et al., 2021b), radial basis functions (Buhmann, 2000), and artificial neural networks (Hajela and Berke, 1992; Tripathy and Bilionis, 2018).

There are numerous papers reviewing and comparing surrogate modeling methods for different purposes (sensitivity analysis, optimization, etc.) and different fields of applications (Forrester and Keane, 2009; Razavi et al., 2012; Asher et al., 2015; Le Gratiet et al., 2017; Chatterjee et al., 2019; Becker, 2020; Cheng et al., 2020). Unsurprisingly, no surrogate modeling method is consistently superior to all others. The decision for a specific type of surrogate model depends on the characteristics of the computational model such as dimensionality and regularity, on the amount and properties of the available data, and on the purpose of the analysis (uncertainty propagation, reliability, sensitivity, optimization, etc.).

Since no single method performs best in all cases, several authors propose combining the predictions of several surrogates, or choosing one out of many based on model selection criteria (Viana et al., 2009; Sagi and Rokach, 2018; Liu et al., 2020b; Cheng and Lu, 2020; Parisi et al., 2021). This concept of aggregating surrogate models is known as *ensemble modeling*. For an overview of common model selection criteria, see Section 2.2.3.

Some surrogate models provide more than just an approximation to the model: for example, they might additionally yield analytical estimates of the response moments of the model, sensitivity

indices, or confidence bounds for their predictions. However, since evaluations of surrogate models are cheap, it is always possible to compute moments numerically using Monte Carlo simulation. Bootstrap can often be used to obtain confidence bounds (Hastie et al., 2001; Marelli and Sudret, 2018).

### 2.2.2 Surrogate modeling versus supervised machine learning

Readers familiar with machine learning (Bishop, 2006) will have recognized many methods and concepts mentioned in the previous section, which gives rise to the question: what is the difference between surrogate modeling in the field of UQ and *supervised learning* in the field of machine learning? The goal of supervised learning is, for a given set of input-output data pairs, to learn an suitable map that is able to make predictions for unseen data points. If the domain of the output data is a discrete set (e.g., when the outputs are categorical), the learning task is commonly called *classification*, while in the continuous case it is called *regression*. In this section, we summarize some of the relevant differences and similarities between black-box surrogate modeling and supervised machine learning regression, and point out interesting connections between the fields of uncertainty quantification and machine learning.

Mathematically, the setup of the problems looks similar. Both black-box surrogate modeling and supervised machine learning regression rely on a set of training data, and make a number of structural assumptions in order to approximate the input-output (or in machine learning terminology, feature-response) relationship using a mathematical model.

However, they differ in the data-generating process and the size of the training set. In UQ, we usually assume that both the computational model and a probabilistic model of its input are available (as explained in Section 2.1). Due to the cost of the computational model, only a few model evaluations can be afforded; however, it is possible to choose the experimental design points in order to collect especially informative model evaluations. The evaluations of the deterministic computational model are not affected by noise. To compensate for the relatively small data sets, UQ methods typically make rather strict structural assumptions.

In contrast, in machine learning the data-generating process is usually not available, but only the data itself is given, and it might be noisy. The amount of data is typically large (a situation described with the well-known term “big data”), which permits the use of very expressive models such as neural networks that have many free parameters. A challenge in machine learning is the choice of network architecture as well as the training process, both of which involve numerous hyperparameters that need to be chosen carefully, an operation known as *tuning*.

Machine learning methods typically need to deal with thousands (if not millions) of features, for example, when the features are the pixels of an image in a classification task. Also, the features are often not real numbers, but might be categorical variables, images or even text or speech. While many classical UQ techniques were initially developed for rather low-dimensional numerical input in the order of a few tens of continuous variables, recent UQ research is increasingly concerned with input spaces of very high dimension (Lataniotis et al., 2020; Kontolati et al., 2022b) and with the treatment of categorical or discrete inputs (Lauvernet and Helbert, 2020; Roustant et al., 2020b; Zhu and Sudret, 2021a; Rohmer et al., 2022).

Some methods, like Gaussian process regression (Kriging) (Rasmussen and Williams, 2006), support vector machines (Vapnik, 1995) and principal component analysis (Pearson, 1901; Jolliffe, 2002) have been used for years by both the UQ and the machine learning communities. Artificial neural networks have been used for structural engineering and for uncertainty quantification (Hajela and Berke, 1992; Tripathy and Bilionis, 2018), while Torre et al. (2019) and Timpe et al. (2020) have explored the use of PCE for machine learning applications. As a surrogate modeling method that is not purely black-box, the so-called *physics-informed neural networks* have received a lot of attention in recent years for the solution of (stochastic) differential equations (Raissi et al., 2019; Zhu et al., 2019; Yang et al., 2020; Karniadakis et al., 2021). Here, the solution is represented by a neural network, and the differential equations are embedded into the loss function used for training.

The connections between UQ and machine learning go beyond the data-driven identification of input-output maps. While neural networks are under certain assumptions universal approximators (Cheng and Titterington, 1994) and perform well for many practical problems when enough data is available, in their pure form they also have some disadvantages: it is often not obvious to understand why a neural network reaches a certain conclusion or what factors influence its response, nor how confident the network is in its prediction, all questions that by definition are central to the field of UQ. Therefore, recent machine learning research is increasingly concerned with questions of fairness, explainability and confidence (Sundararajan et al., 2017; Abdar et al., 2021; Bénesse et al., 2022) and also uses traditional UQ techniques for this purpose (Liu et al., 2020a; Fel et al., 2021).

### 2.2.3 Surrogate model assessment and model selection

In order to be useful, a surrogate model must accurately approximate the input-output relationship of the computational model. Therefore, we need to be able to evaluate how well a given surrogate is performing. Furthermore, surrogate models often depend on a number of tunable hyperparameters that need to be chosen according to such a performance criterion during training.

A global measure of the approximation quality is the *mean-squared error* between the model $\mathcal{M}$ and the surrogate model $\tilde{\mathcal{M}}$ defined by

$$\epsilon = \mathbb{E}_{\boldsymbol{X}}\left[\left(\mathcal{M}(\boldsymbol{X}) - \tilde{\mathcal{M}}(\boldsymbol{X})\right)^2\right]. \tag{2.2}$$

Here the expectation is with respect to $\boldsymbol{X}$, so that the squared difference is weighted by the input density $f_{\boldsymbol{X}}$.

Denote by $\mathcal{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^N \subset \mathcal{D}$ the experimental design used to train the surrogate model $\tilde{\mathcal{M}}$, usually sampled from $f_{\boldsymbol{X}}$. A simple approximation to $\epsilon$ is the *empirical error*

$$\epsilon_{\text{emp}} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{M}(\boldsymbol{x}^{(i)}) - \tilde{\mathcal{M}}(\boldsymbol{x}^{(i)}))^2. \tag{2.3}$$

However, judging the approximation quality of the surrogate from the empirical error, i.e., from its performance on the same points that were used to train it, will in general underestimate the true prediction error.

A better approximation to $\epsilon$ is the *validation error*

$$\epsilon_{\text{val}} = \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} (\mathcal{M}(\boldsymbol{x}_{\text{val}}^{(i)}) - \tilde{\mathcal{M}}(\boldsymbol{x}_{\text{val}}^{(i)}))^2, \tag{2.4}$$

where $\mathcal{X}_{\text{val}} = \{\boldsymbol{x}_{\text{val}}^{(i)}\}_{i=1}^{N_{\text{val}}}$ is a large set of points sampled from $f_{\boldsymbol{X}}$ independently of $\mathcal{X}$.

If such a large set of additional model evaluations is not available, which is usually the case for all but toy problems, the error can be approximated by *cross-validation* (Hastie et al., 2001) based on $\mathcal{X}$ as follows: the experimental design is split into several parts, called folds. Each fold is in turn treated as validation set, while the remaining points of the experimental design are used to construct the surrogate model. Finally, the errors computed on the left-out sets are averaged, and serve as an estimate for the validation error. If the number of folds is equal to the size of the experimental design, the method is known as *leave-one-out* (LOO) cross-validation (Hastie et al., 2001).

Cross-validation is a general error assessment technique that is not only used to judge the performance of the final surrogate, but is also often used already during the training process for selecting hyperparameters, or – in the case of iterative methods – for deciding when to stop the iterations. These decisions are also called *model selection*, because we have to decide between different surrogate models (of the same type) arising from different choices of hyperparameters.

For methods which aim at fitting a random quantity to given data, such as Kriging which infers a stochastic process (Rasmussen and Williams, 2006), or weak polynomial chaos approximation which fits a distribution (Xiu, 2010), an alternative discrepancy measure is the *likelihood*, which is the probability of observing the training data given a certain choice of parameters. A model with many free parameters is typically more flexible and can achieve a larger likelihood than one with fewer parameters. Therefore, model selection based on the likelihood alone can lead to overfitting. To prevent this, model selection criteria such as Akaike's information criterion (AIC) (Akaike, 1974) and the Bayesian information criterion (BIC) (Schwarz, 1978) additionally take the number of free parameters into account. It has been shown that AIC is asymptotically equivalent to cross-validation (Stone, 1977).

It should be noted that the repeated application of the same criterion for training and selection among a large number of models can also lead to overfitting (Rao et al., 2008; Arlot and Celisse, 2010; Cawley and Talbot, 2010).

## 2.3 Surrogate modeling for stochastic simulators

The classical framework of UQ shown in Figure 2.1 assumes that all stochasticity influencing the system can be modeled probabilistically in the form of random input variables. The computational model itself is then deterministic and can be surrogated by the deterministic function approximation methods described in Section 2.2. However, there are cases in which it is not desired or even not possible to completely separate the stochasticity from the computational model. The model contains residual randomness which causes the response to be a random quantity, even for fixed input parameters. In this case, we call the model a *stochastic simulator*. There are two main cases from which stochastic simulators arise.

### 2.3.1 Examples of stochastic simulators

In some instances, the stochasticity is an intrinsic part of the computational model. For example, consider the stochastic susceptible-infected-recovered (SIR) model developed in epidemiology to study the spread of an infectious disease (Britton, 2010). Here, the population consists of individuals that are either susceptible to a disease, infected by it, or recovered and therefore immune. Two numbers, the rates of infection and recovery, govern the stochastic evolution of the disease with respect to time. Due to the stochasticity of infection and recovery, the evolution might look different each time the simulation is run, even if the population size and the rate parameters are kept at the same values. More generally, any stochastic differential equation with a random forcing term is a stochastic simulator of this class. In these models, the stochasticity drives the simulation and therefore cannot be modeled separately: the model is intrinsically a stochastic simulator.

Stochastic simulators also arise in cases when the complete characterization of the stochasticity would in principle be possible but is not desired, e.g., if the latter is too high-dimensional or too complicated to be modeled explicitly. For example, when designing a wind turbine for a new geographical site, the structural safety of the turbine over a period of many years must be assessed for the specific wind conditions at the respective location. This is done by employing a wind generation model together with a deterministic aero-servo-elastic simulation of the wind turbine structure (Dimitrov et al., 2018; Slot et al., 2020).

The wind generation model generates *wind boxes*, i.e., realizations of a vector-valued random field defined in both time and space. A wind box is often summarized by a few statistics such as mean wind speed, turbulence, inflow angle etc., which together are called *wind climate*. For each wind climate, an arbitrary number of wind boxes can be generated that match these values. If we completely separated the wind box generation from the multi-physics simulation of the turbine, the latter would be a deterministic simulator; however, the wind fields are too high-dimensional to be modeled explicitly using the classical methodology described in Section 2.2.[6] Therefore, it can be desirable to treat the generation of wind boxes from wind conditions as a part of the computer model, resulting in a stochastic simulator which, for given wind climate, returns a different response each time the simulation is run. The same reasoning shows that the simulation of earthquake damage to a structure can be regarded as a stochastic simulator (Zhu et al., 2022). In these cases, the stochasticity comes from latent variables that are not explicitly modeled probabilistically as input variables.

### 2.3.2 Stochastic simulators as random fields

As explained in Section 2.1, stochastic simulators can be seen as random fields for which the space of explicitly modeled parameters $\mathcal{D}$ assumes the role of the index set, and the latent stochasticity is represented as an event from an abstract space $\Omega$.

[6] A conceptually quite different approach, namely autoregressive models, has recently been proposed for constructing custom surrogates for such high-dimensional time-dependent random input (Dimitrov et al., 2022).

The random field $\mathcal{M} : \mathcal{D} \times \Omega \to \mathbb{R}$ has the properties that

– For each $\boldsymbol{x} \in \mathcal{D}$, $Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x}, \cdot) : \Omega \to \mathbb{R}$ is a random variable; (2.5)

– For each $\omega \in \Omega$, $\mathcal{M}(\cdot, \omega) : \mathcal{D} \to \mathbb{R}$ is a deterministic function, called *trajectory*. (2.6)

We call Eq. (2.5) the *random variable view* of the random field, and Eq. (2.6) the *random function view*. These two perspectives lead to different types of surrogate models, as we will see below.

We assume that $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) \times L^2(\Omega, \mathcal{F}, P)$ where $(\Omega, \mathcal{F}, P)$ is the probability space of the stochastic event $\omega \in \Omega$.

Just like deterministic computer models, stochastic simulators can only be evaluated pointwise, i.e., for a finite number of experimental design points $\boldsymbol{x}^{(i)} \in \mathcal{D}$ and stochastic events $\omega^{(j)} \in \Omega$. Repeated evaluations for the same location $\boldsymbol{x} \in \mathcal{X}$, each time using a different $\omega^{(j)}, j = 1, \ldots, R$, are called *replications*. For many computational models, it is possible to fix the sequence of random numbers used for the computations by fixing the seed of the random number generator, which allows the (pointwise) evaluation of the simulator along trajectories even if the underlying stochastic event $\omega$ is not known.

To fully characterize a general random field, it is necessary to specify the collection of all its finite-dimensional distributions (Grigoriu, 2002, Chapter 3.5)

$$\{F_{Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_n}} : n \geq 1, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{D}\}, \tag{2.7}$$

where $F_{Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_n}}(y_1, \ldots, y_n) = \mathbb{P}\left(Y_{\boldsymbol{x}_1} \leq y_1 \wedge \ldots \wedge Y_{\boldsymbol{x}_n} \leq y_n\right)$. The univariate distributions $F_{Y_{\boldsymbol{x}}}$ are called *marginals* or *marginal distributions*. Extending the concept of moments of random variables to random fields, the deterministic *mean function* of $\mathcal{M}$ is given by $\mu(\boldsymbol{x}) = \mathbb{E}\left[Y_{\boldsymbol{x}}\right]$. If $\mu(\boldsymbol{x}) = 0$, the random field is called *centered*. The *(auto-)covariance function* of $\mathcal{M}$ is defined by

$$C(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Cov}\left[Y_{\boldsymbol{x}}, Y_{\boldsymbol{x}'}\right]. \tag{2.8}$$

To enable the theory presented in Section 2.4.2 below, we assume that $C$ is continuous on $\mathcal{D} \times \mathcal{D}$.

While Eq. (2.7) is needed for general random fields, there are a few special cases that can be characterized by a smaller number of properties, most notably the family of *Gaussian processes*, for which all finite-dimensional joint distributions are multivariate Gaussian distributions. A Gaussian process is uniquely defined by its mean and covariance function. Its conditional distributions are again multivariate Gaussians, which motivates the popular surrogate modeling technique Kriging a.k.a. Gaussian process modeling (Section 2.2.1). While Gaussian random fields are computationally convenient, random fields encountered in real-world problems (and in particular, stochastic simulators) are often not Gaussian. One obvious argument is that Gaussian variables are unbounded, whereas physical quantities are almost always bounded (Grigoriu, 2002, p. 118).

### 2.3.3 Overview of surrogate modeling approaches

Most surrogate modeling approaches for stochastic simulators developed to date have focused on the *random variable view* (Eq. (2.5)), i.e., on their interpretation as a collection of random

variables indexed by the set of inputs. The goal of these approaches, which we describe in Section 2.3.3.1, is to approximate the random variables $\mathcal{M}(\boldsymbol{x}, \cdot)$ in distribution, usually relying on the regularity of the mapping with respect to $\boldsymbol{x}$. In other words, they approximate the marginal distributions $F_{Y_{\boldsymbol{x}}}$, without attempting to match any of the higher-order joint distributions (i.e., $n > 1$ in Eq. (2.7)). The resulting surrogate can be used to generate pointwise samples, which will approximately follow the correct marginal distributions.

By construction, this class of surrogates is in general not able to generate trajectories. To obtain a surrogate model that has this feature, we must take the *random function view* (Eq. (2.6)), and data sampled from trajectories must be available. In the context of stochastic simulators, the literature on such surrogate models is scarce. However, in the broader context of inferring general random fields from given samples, there are a number of methods, which we discuss in Section 2.3.3.2.

#### 2.3.3.1 Surrogates in distribution

We first give an overview of surrogate models that characterize the response random variable in distribution. This can be seen as conditional distribution estimation, aiming at characterizing the quantity $\mathbb{P}(Y|\boldsymbol{X} = \boldsymbol{x})$.

To obtain an explicit expression for the statistics or the full distribution of $\mathcal{M}(\boldsymbol{x}, \cdot)$ at each location in the input domain $\mathcal{D} \ni \boldsymbol{x}$, various parametric approaches have been developed, which often rely on deterministic surrogate models to emulate these statistics or other auxiliary quantities (such as expansion coefficients) over $\mathcal{D}$. (Joint) Gaussian process models have been proposed to address scalar statistics of the random response variables, such as mean and variance (Iooss and Ribatet, 2009; Ankenman et al., 2010; Marrel et al., 2012; Binois et al., 2018). Other approaches emulate quantiles using *quantile Kriging* (Plumlee and Tuo, 2014) or other quantile regression methods (Torossian et al., 2020). Assuming that the full marginal distributions at a set of experimental design points $\mathcal{X} \subset \mathcal{D}$ are available, Moutoussamy et al. (2015) propose to represent the probability density in terms of a suitable functional basis. Similarly, Browne et al. (2016) propose to represent the quantile function of the stochastic simulator output in terms of a suitable basis, and emulate the parameters throughout the domain using Gaussian processes.

Another parametric choice is the *generalized lambda model* (Zhu and Sudret, 2020) that utilizes the generalized lambda distribution (Freimer et al., 1988), a parametric family of distributions with only four parameters that is able to closely approximate many of the classical unimodal parametric families (Gaussian, uniform, exponential, lognormal etc.). Zhu and Sudret (2020) apply this distribution to emulate the response densities based on replications, and extrapolate to the whole domain using PCE, using a joint computation scheme that simultaneously fits the response densities and the PCE models to the given data. The generalized lambda model has further been refined to avoid the need for replications by employing maximum conditional likelihood estimation (Zhu and Sudret, 2021b,c).

Replacing the generalized lambda distribution by a PCE approximation of the conditional random variables, Zhu and Sudret (2022) recently developed the replication-free *stochastic PCE method*. This approach uses PCE with two additional variables (a latent and a noise variable)

to emulate the stochastic simulator, inspired by the weak PCE methodology based on maximum likelihood estimation (Xiu, 2010).

A classical nonparametric method for conditional density estimation, likewise not requiring replications, is the well-known *kernel density estimation* (Fan and Gijbels, 1996; Hall et al., 2004).

Finally, in the field of machine learning, generative models like *variational autoencoders* (Kingma and Welling, 2014) and *generative adversarial networks* (Goodfellow et al., 2014) are designed to learn a target distribution from data and to sample from it, thus being a kind of surrogate in distribution. Similarly, *Bayesian neural networks* (MacKay, 1992; Goan and Fookes, 2020), whose weights are independent Gaussian random variables, are able to approximate conditional distributions. Bayesian methods such as Markov Chain Monte Carlo or variational inference are used to infer the parameters of the weight densities from the given data.

#### 2.3.3.2 Surrogates using the random function view

Whereas the methods presented in the previous section are able to accurately characterize the marginal response distributions, they do not model the joint distributions of higher order. Of course, unless we are in the specific case of a Gaussian random field, modeling all the finite-dimensional distributions in Eq. (2.7) is infeasible. If we still want to capture some of the higher-order information about the random field, we need to make use of its structure, i.e., of the properties of its covariance function (Eq. (2.8)) and the induced regularity of its trajectories. In other words, we need to utilize the random function view of random fields shown in Eq. (2.6). The overall goal is to find a representation of the random field that can be sampled from, both pointwise and trajectory-wise.

In the literature, two classes of methods for the representation of random fields can be distinguished. The first is the construction of a representation for given properties such as marginal distributions, covariance function, or spectral density.[7] These properties are assumed to be known analytically, and the goal is to generate samples from a random field with these properties. Note that general non-Gaussian random fields are not completely characterized by their marginals and covariance function (Grigoriu, 2002). We will give a short overview of such methods in the following.

The second class, which is most relevant in the context of surrogating stochastic simulators, is the inference of a representation from data, i.e., from pointwise samples of the random field along trajectories. This data is generated by a real-world process (such as an earthquake) or by a stochastic computational model, both of which behave like a random field. However, it might be impossible or too expensive to use these processes or models directly for generating a large amount of samples, and therefore a surrogate needs to be constructed. These methods are discussed below.

[7] The *spectral density* exists only for weakly stationary random processes, i.e., for processes with constant mean and an autocorrelation function $r(\boldsymbol{\tau}) = \mathbb{E}\left[Y_{\boldsymbol{x}} Y_{\boldsymbol{x}+\boldsymbol{\tau}}\right]$ that only depends on the difference between two locations. For a one-dimensional centered process, it is defined by $s(\nu) = \frac{1}{2\pi}\int_{-\infty}^{\infty} r(\tau)\exp(-i\nu\tau)\mathrm{d}\tau$, i.e., it is the Fourier transform of the autocorrelation function (Grigoriu, 2002).

**Simulation of non-Gaussian random fields from properties.** A simple way to discretize a random field is to represent it by a random vector (i.e., a finite number of random variables) on a mesh defined in the index set. The value of the random field at other locations is then computed from the random vector. Sudret (2007) gives an overview of such methods, including the midpoint method (Der Kiureghian and Ke, 1988) and spatial averaging (Vanmarcke and Grigoriu, 1983). However, Sudret and Der Kiureghian (2000) point out that these methods are rather inefficient.

Alternatively, *series expansion methods* exploit the structure of the random field in order to represent it using "spatial" functions (i.e., functions defined over the index set) and random coefficients (Sudret, 2007). Grigoriu (2010) classifies the series expansion methods into *linear models with dependent parameters* and *nonlinear models*, also known as *translation processes*.

Linear series expansion methods represent the random field as a series of spatial functions $h_k(\boldsymbol{x})$ weighted by random coefficients $\xi_k(\omega)$, with the general form (Grigoriu, 2006, 2010)

$$\mathcal{M}(\boldsymbol{x}, \omega) = \mu(\boldsymbol{x}) + \sum_{k=0}^{\infty} \xi_k(\omega) h_k(\boldsymbol{x}). \tag{2.9}$$

The *spectral representation method* expands a weakly stationary random process in terms of trigonometric polynomials (Fourier expansion) (Deodatis and Shinozuka, 1991; Grigoriu, 2006).[8] *Orthogonal series expansion* (OSE) expands a random field onto any orthonormal basis for the spatial component (Zhang and Ellingwood, 1994). Finally, *Karhunen-Loève expansion* (KLE) expands a general random field onto an optimal orthonormal spatial basis, which is computed as the solution of an integral eigenvalue problem involving the covariance function (Karhunen, 1946; Loève, 1978). We will explain KLE in detail in Section 2.4.2. In all these cases, the random coefficients $\xi_k$ of the resulting expansion are in general dependent, and failing to account for this dependence can result in improper characterizations of the random field (Grigoriu, 2010).

Note that the linear series expansion in Eq. (2.9) transforms the random field, which (taking the random variable view of Eq. (2.5)) is an uncountably infinite collection of correlated random variables, into a countable collection of random variables $\{\xi_k\}_{k=0}^{\infty}$ together with spatial functions $\{h_k\}_{k=0}^{\infty}$ capturing the correlation.

For the special case of Gaussian random fields, the *expansion optimal linear estimation* (EOLE) method (Li and Der Kiureghian, 1993) provides a compromise between pointwise and series representation, by representing the random field by a random vector defined on a mesh, and interpolating between the mesh points using optimal linear estimation, relying on the Gaussian assumption just like Kriging. The random vector can be further compressed by using a form of dimension reduction closely related to KLE (see Section 2.4.2).

*Translation processes* make use of the fact that simulating a Gaussian process $G$ is comparatively easy, e.g. by EOLE or KLE, since such a process is fully characterized by its mean and covariance function, and all conditional distributions are Gaussian (Grigoriu, 2006). For given marginals and covariance function or spectral density of the general non-Gaussian random field of interest,

[8] Strictly speaking, this interpretation only holds for periodic processes. For general processes, the analysis is slightly more involved (Grigoriu, 2006).

the goal of translation process algorithms is to find a mapping $\mathcal{T}$ such that the induced random field $\mathcal{T}(G)$ has the requested properties (Grigoriu, 1984; Yamazaki and Shinozuka, 1988; Grigoriu, 1998; Deodatis and Micaletti, 2001; Sakamoto and Ghanem, 2002; Poirion and Puig, 2010; Shields et al., 2011; Kim and Shields, 2015). Here, the difficulty is that the marginals and covariance function might be inconsistent, i.e., there might be no such mapping $\mathcal{T}$ that achieves the desired properties (Grigoriu, 2010).

**Inference of random fields from data.** The methods described above all assume that the marginals and the covariance function or the spectral density of the random field of interest are known analytically. Now, we turn to methods that aim to characterize a random field solely based on a set of samples. Many of these methods rely on the spectral method *Karhunen-Loève expansion* (KLE), which is described in detail in Section 2.4.2 below.

One popular approach is based on the repeated evaluation of the random field at the points of a given mesh. In other words, the random field is reduced to a high-dimensional random vector with one component per mesh point, similar to the pointwise representation described above. First, discrete KLE (i.e., principal component analysis (PCA); see Section 2.4.2) is performed on the realizations of the high-dimensional random vector to reduce its dimensionality. Then, the random coefficients are modeled, often by PCE (Ghanem and Spanos, 1991; Soize, 2010). The PCE coefficients are determined, e.g., by maximum likelihood (weak PCE) (Desceliers et al., 2006; Perrin et al., 2012) or Bayesian inference (Ghanem and Doostan, 2006). Das et al. (2009) propose two methods to approximate the experimentally determined multivariate joint distributions of the random coefficients by PCE based on conditional distributions and the Rosenblatt transform (Rosenblatt, 1952). To further model the uncertainty in the stochastic model induced by the finite size of the sample set, a number of papers explores the inference of random instead of deterministic PCE coefficients (Ghanem and Doostan, 2006; Das et al., 2008; Arnst et al., 2010; Soize, 2010).

The extension of the inferred high-dimensional random vector (defined on the mesh) to the continuous index set (where the original random field is defined) is often not discussed, because it is considered sufficient to know the random field on a fine mesh. Among the few publications explicitly addressing this topic, Desceliers et al. (2006) suggest to use the interpolation functions of the finite element method. Das et al. (2009) generally refer to this aspect as "a task of interpolation or/and approximation technique". In the specific context of stochastic simulators, Azzi et al. (2019) propose to use PCE or Kriging for this extension.

A variant of the KLE-PCE methodology described above is the intrusive method proposed by Doostan et al. (2007), which was later extended to the non-intrusive case by Raisee et al. (2015) and Abraham et al. (2018). Starting from a PCE expansion for each of the random variables of a coarse mesh computed from given replications, this method first computes a KLE on the coarse mesh ("optimal expansion"). The resulting random variables of the KLE, modeled as PCE, are then used to derive spatial functions on a finer mesh.

In the context of earthquake engineering, in a series of papers Zentner and Poirion (2012); Poirion and Zentner (2013, 2014) reverse this construction: first, they interpolate the trajectories based on the given data and express them in terms of an orthonormal basis. Then they perform KLE

on the interpolated trajectories, which – due to the orthonormality of the basis – reduces to a discrete eigenvalue problem. The random coefficients are modeled by kernel density estimation, taking bivariate dependence into account (Poirion and Zentner, 2014).

Azzi et al. (2019) propose another approach based on KLE, which first infers a continuous representation of the covariance function based on the sample covariance matrix computed from replications at mesh points. Then, the eigenfunctions of KLE are computed on a finer mesh.

Because KLE is linear in the random coefficients (see Eq. (2.9)), the complexity of non-Gaussian random fields potentially requires the random coefficients to follow intricate joint distributions (see also Section 2.4.2). To avoid having to infer complicated joint distributions, Sarma et al. (2008) and Ma and Zabaras (2011) instead suggest using *kernel PCA*, a nonlinear extension of KLE. The data is first mapped to the so-called *feature space* using a nonlinear transformation, and then KLE is applied, with the hope that the data is (close to) linear in the feature space. A difficulty of kernel PCA is the pre-image problem: the obtained realizations in the feature space must be mapped back to the original space of the data. Sarma et al. (2008) use a polynomial kernel while Ma and Zabaras (2011) use a Gaussian kernel. Both assume that in the feature space, the random coefficients of the linear expansion (Eq. (2.9)) are independent.

In the context of stochastic differential equations with stochastic forcing as well as random input parameters, Navarro Jimenez et al. (2017) suggest to model the resulting random field by PCE with random coefficients (i.e., choosing PCE basis functions for $h_k$ in Eq. (2.9)). The stochastic coefficients are determined by non-intrusive pseudospectral projection. They do not discuss resampling the random field, because the purpose of their construction is sensitivity analysis.

## 2.4 Spectral methods

The topic of this thesis is surrogate modeling with spectral methods. The word *spectral* usually hints to the connection to an eigenvalue problem, since the *spectrum* of an operator denotes the set of its eigenvalues. Solving an eigenvalue problem yields, in addition to the eigenvalues, eigenvectors or eigenfunctions that are orthogonal in the respective space where the problem is set. Under appropriate conditions, the eigenvectors or -functions constitute a basis for this space.

A *spectral expansion* is then the representation of an element from that space in terms of such an orthogonal basis. More generally, this expression is used for any representation as long as the respective basis is orthogonal.

### 2.4.1 Chaos expansions

In UQ, a spectral expansion in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ is also often called *chaos expansion*, in reference to an early contribution by Wiener (1938) who introduced them for stochastic processes using Hermite polynomials.[9]

[9]Therefore, the use of the term *chaos expansion* in UQ predates the mathematical field of chaos theory by several decades (Xiu, 2010). Chaos expansions as introduced here do not belong to the field of chaos theory.

The general mathematical framework of chaos expansions is as follows. Together with the inner product

$$\langle g, h \rangle = \int_{\mathcal{D}} g(\boldsymbol{x}) h(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{2.10}$$

the space of square-integrable functions $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ defined in Eq. (2.1) is a Hilbert space.[10] Interpreted probabilistically, a model $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ has finite variance under $\boldsymbol{X}$, i.e., $\mathrm{Var}_{\boldsymbol{X}}[\mathcal{M}(\boldsymbol{X})] < \infty$.

Let $\{\psi_k\}_{k=0}^{\infty}$ be an orthonormal basis for $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$, i.e.,

1. for all $i, j$, $\langle \psi_i, \psi_j \rangle = \delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases}$ ; and
2. $\{\psi_k\}_{k=0}^{\infty}$ is dense in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$.

Then the chaos expansion of $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ in terms of this basis is given by

$$\mathcal{M}(\boldsymbol{x}) = \sum_{k=0}^{\infty} c_k \psi_k(\boldsymbol{x}), \tag{2.11}$$

where

$$c_k = \langle \mathcal{M}, \psi_k \rangle \tag{2.12}$$

is the projection of $\mathcal{M}$ onto $\psi_k$.

It follows that the random response vector $Y = \mathcal{M}(\boldsymbol{X})$ can be expressed as a series of random variables $Z_k = \psi_k(\boldsymbol{X})$ by

$$Y = \sum_{k=0}^{\infty} c_k Z_k. \tag{2.13}$$

From now on, we assume that the components of the random vector $\boldsymbol{X}$ are independent.[11] Therefore, the density $f_{\boldsymbol{X}}$ factorizes into $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{d} f_{X_i}(x_i)$. A basis of $L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = L^2_{f_{X_1}}(\mathcal{D}_1) \otimes \ldots \otimes L^2_{f_{X_d}}(\mathcal{D}_d)$ can be obtained from one-dimensional bases $\{\psi_k^{(i)}\}_{k=0}^{\infty}$ of $L^2_{f_{X_i}}(\mathcal{D}_i)$ by a standard tensor product construction: an element $\psi_{\boldsymbol{\alpha}}$ of the multivariate basis is given by

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{i=1}^{d} \psi_{\alpha_i}^{(i)}(x_i), \tag{2.14}$$

where the so-called *multi-index* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ collects the indices of the respective univariate basis elements.

We will describe two particular types of chaos expansion below, namely *polynomial chaos expansions* (PCE) in Section 2.4.1.1 and *Poincaré chaos expansions* in Section 2.4.1.2. Beyond these, a chaos expansion in terms of a basis consisting of B-splines has recently been proposed (Rahman, 2020; Eckert et al., 2020). While rarely used in UQ, another instance that fits into the chaos expansion framework is the well-known *Fourier expansion*, whose orthonormal basis consists of sines and cosines.

[10] A *Hilbert space* is a complete metric space with an inner product that induces the metric. It is a powerful concept in functional analysis and applied mathematics. Under certain conditions it admits a countable orthonormal basis, which is essential for function approximation.

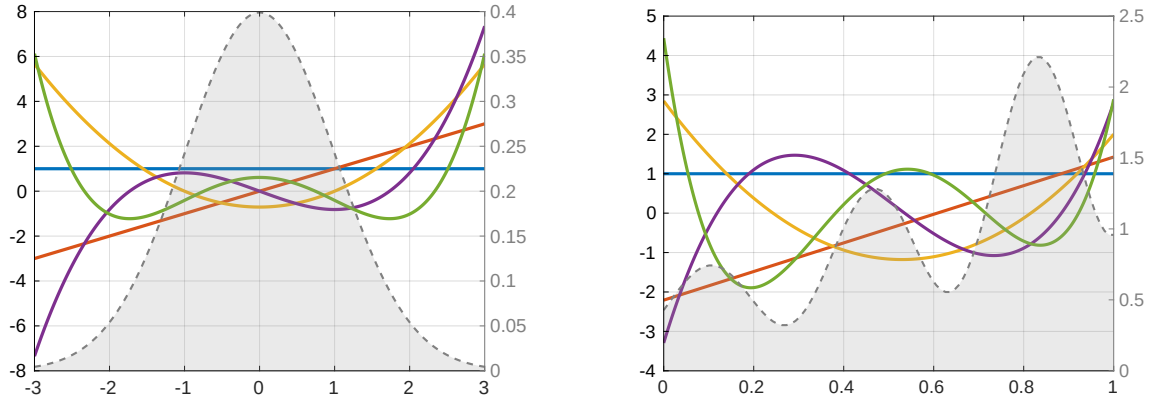
[11] We discuss the case of dependent input variables in Section 2.4.1.1 for PCE and in Section 2.6.3 for sensitivity analysis.

#### 2.4.1.1 Polynomial chaos expansion

*Polynomial chaos expansion* denotes the expansion of a model $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ in terms of a multivariate orthonormal polynomial basis obtained as the tensor product of univariate orthonormal polynomial bases (Eq. (2.14)). In the following, we describe the construction of the latter.

Let $f_X$ be the probability density function of a random variable $X$ defined on $\mathcal{D} \subset \mathbb{R}$. An orthonormal polynomial basis for $L^2_{f_X}(\mathcal{D})$ exists if and only if the distribution function $F_X$ is uniquely defined by the sequence of its moments (Ernst et al., 2012). This is the case for most of the standard parametric distributions such as uniform, Gaussian, Beta or Gamma distributions, but not for the lognormal distribution. A sufficient condition for existence is that all moments are finite and the support of the distribution is compact (Ernst et al., 2012).

For the classical parametric distribution families, the associated orthonormal polynomials are well-known, and their properties have been studied for centuries (Szegö, 1939; Gautschi, 2004). When $X$ is Gaussian, the associated orthonormal polynomials are the (probabilists') *Hermite polynomials*. A uniformly distributed $X$ is associated to the *Legendre polynomials*. A Beta distribution generates the *Jacobi polynomials*, and a Gamma distribution the *Laguerre polynomials*. In Figure 2.2, we illustrate the first five orthonormal polynomials associated to two different densities.



*(a)* Gaussian *(b)* Multimodal density

***Figure 2.2:*** *The first five orthonormal polynomials associated to two different densities: Left, Hermite polynomials associated to the Gaussian distribution. On the right, arbitrary polynomials associated to a custom (non-classical) multimodal density. The basis functions are plotted in color, while the densities are shown in gray.*

Originally, PCE was proposed in terms of Hermite polynomials by Ghanem and Spanos (1991) corresponding to the Gaussian distribution. The approach was generalized by Xiu and Karniadakis (2002) to distributions from the Askey scheme, i.e., the parametric families mentioned above. However, PCE can even be computed for arbitrary distributions (including discrete ones), as long as they admit an orthonormal polynomial basis, which is known under the name *arbitrary PCE* (Soize and Ghanem, 2004; Wan and Karniadakis, 2006; Oladyshkin and Nowak, 2012; Torre et al., 2019).

To construct an orthonormal polynomial basis for a given distribution, there are several options. For the three classical polynomial families Hermite, Jacobi (which includes Legendre as special case), and Laguerre, one way is to solve the associated eigenvalue problem which involves a differential operator (Bochner, 1929). An alternative for any type of distribution is orthonormalization of the set of monomials $\{1, x, x^2, x^3, \ldots\}$ using the Gram-Schmidt algorithm.

A third, often more stable, method is the *Stieltjes procedure* (Gautschi, 1982, 2004): Orthonormal polynomials obey an equation known as the three-term recurrence rule

$$\sqrt{\beta_{n+1}}\psi_{n+1}(x) = (x - \alpha_n)\psi_n(x) - \sqrt{\beta_n}\psi_{n-1}(x), \tag{2.15}$$

starting with $\psi_{-1}(x) = 0$ and $\psi_0(x) = 1$, where $\alpha_n$ and $\beta_n$ are given by

$$\alpha_n = \frac{\langle x\,\psi_n, \psi_n\rangle}{\langle \psi_n, \psi_n\rangle}, \tag{2.16}$$

$$\beta_n = \frac{\langle \psi_n, \psi_n\rangle}{\langle \psi_{n-1}, \psi_{n-1}\rangle}. \tag{2.17}$$

The coefficients $\alpha_n$ and $\beta_n$ are known analytically for the classical orthonormal families. For other cases, they can be computed by numerical integration.

A univariate orthonormal polynomial basis constructed by Eq. (2.15) has exactly one element of each degree $k \in \mathbb{N}_0$. The degree is used as the index of the elements in the basis.

If an orthonormal polynomial basis does not exist, such as in the case of the lognormal distribution, we can first map the given input to another distribution and then use the associated polynomials for the PCE. Such a mapping is called *isoprobabilistic transform*. For example, in the case of a lognormal variable $X$ a transformation to Gaussian variables is easily achieved by taking the logarithm: $Z = \mathcal{T}(X) = \log(X)$ is Gaussian. The PCE is then computed for the transformed model $\tilde{\mathcal{M}}(Z) = \mathcal{M}(\mathcal{T}^{-1}(Z))$.

In principle, this procedure can be used for any two continuous random variables $X$ and $Z$ using the probability transform $\mathcal{T} = F_Z^{-1} \circ F_X$. However, especially when one of the variables is bounded and the other unbounded, the transform can introduce significant nonlinearity into the composite model $\tilde{\mathcal{M}} = \mathcal{M} \circ \mathcal{T}^{-1}$, which in turn requires high-degree polynomials to be approximated well (Lei et al., 2019; Torre et al., 2019).

The case of dependent input variables is an active area of research. If the components of $\boldsymbol{X}$ are not independent, it is still possible to construct a multivariate polynomial basis that is orthonormal with respect to the distribution of $\boldsymbol{X}$, e.g., by applying the Gram-Schmidt procedure (Torre et al., 2019; Jakeman et al., 2019; Lei et al., 2019); however, it is not unique but depends on the ordering of terms. Alternatively, the dependent random vector $\boldsymbol{X}$ can be transformed to an independent one using a suitable probability transform (Lebrun and Dutfoy, 2009) before computing the PCE, which likewise is a non-unique operation depending on the ordering of the variables. Torre et al. (2019) demonstrated that for the purpose of global approximation, it can be advantageous to use a basis that is orthonormal to the marginals only, ignoring the dependence structure.

#### 2.4.1.2 Poincaré chaos expansion

The *Poincaré chaos expansion* has recently been proposed by Roustant et al. (2020a), expanding on the results of an earlier paper by Roustant et al. (2017) with the goal of computing Poincaré constants for univariate distributions.

In one dimension, the *Poincaré constant* $C_P(f_X)$ associated to the probability density $f_X$ of a random variable $X$ is the smallest possible constant $C$ with

$$\mathbb{E}_X\left[h(X)^2\right] \leq C\, \mathbb{E}_X\left[(h'(X))^2\right] \tag{2.18}$$

for all $h \in H^1_{f_X}$ with $\mathbb{E}_X\left[h(X)\right] = 0$, where $H^1_{f_X}$ is the Sobolev space of weakly differentiable functions, i.e., $H^1_{f_X} = \{h \in L^2_{f_X} : h' \in L^2_{f_X}\}$. The Poincaré inequality (2.18) bounds the variance of any function $h$ by its expected squared derivative. The constant $C_P(f_X)$ is informative about the underlying distribution: it is a measure of multimodality and can be related to the convergence rate of Markov chains (Pillaud-Vivien et al., 2020).

Roustant et al. (2017) introduced a computation procedure for Poincaré constants of univariate measures $f_X \mathrm{d}x$ fulfilling the following assumption:

**Assumption 1 (Assumption on $f_X$).** *Assume that $f_X$ is supported on a bounded interval $(a, b)$ and that $f_X(x) = e^{-V(x)}$ with $V$ continuous and piecewise $C^1$ on $[a, b]$.*

Under this assumption, the Poincaré constant can be obtained by solving the following eigenvalue problem:

$$\begin{aligned} L\psi_k = \psi_k'' - V'\psi_k' &= -\lambda_k \psi_k, \\ \psi_k'(a) = \psi_k'(b) &= 0, \end{aligned} \tag{2.19}$$

which can be expressed in weak form as

$$\text{for all } h \in H^1_{f_X} \qquad \langle h', \psi_k' \rangle = \lambda_k \langle h, \psi_k \rangle . \tag{2.20}$$

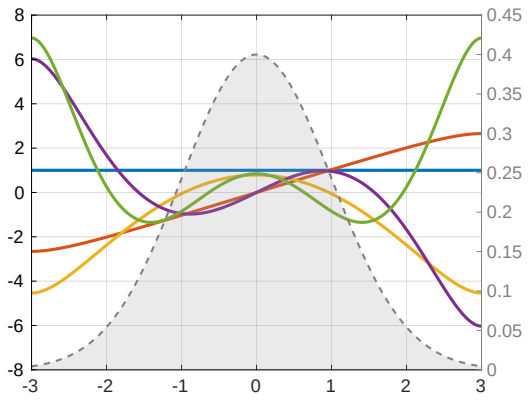
The solutions $(\lambda_k, \psi_k)$ have the following properties:

- After normalization, the eigenfunctions $(\psi_k)_{k\geq 0}$ form an orthonormal basis of $L^2_{f_X}$.
- The eigenvalues satisfy $0 = \lambda_0 < \lambda_1 < \ldots \to \infty$.
- For the first eigenpair, $\lambda_0 = 0$ and $\psi_0(x) = 1$.
- For the second eigenpair, $C_P(f_X) = \frac{1}{\lambda_1}$, and $\psi_1$ attains equality in Eq. (2.18).
- If $f_X \in C^m$, then $\psi_k \in C^{m+1}$.
- $\psi_k$ has $k$ roots.

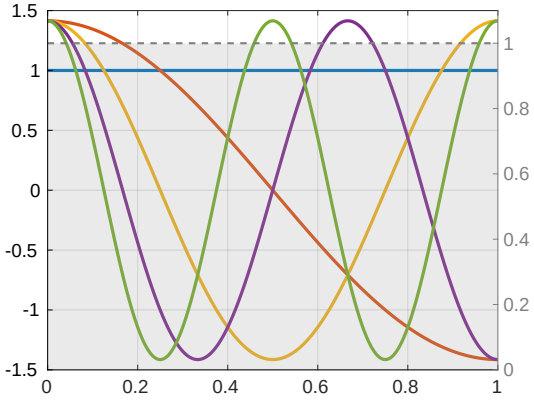
We show examples for univariate Poincaré bases in Figure 2.3. The Poincaré basis is in general not polynomial, except for the special case of the Gaussian distribution, for which the associated Poincaré basis coincides with the Hermite polynomials. The Poincaré basis associated to the uniform distribution can be shown to consist of cosine functions.

The eigenvalue problem in Eq. (2.19) can be solved numerically e.g. by one-dimensional finite elements. Since by Assumption 1 the measure is supported on a bounded interval, for most purposes it is enough to use an evenly spaced grid with sufficiently fine resolution.

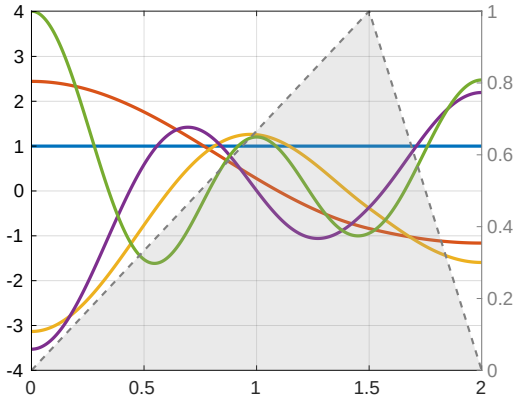
As a consequence of Eq. (2.20), the Poincaré basis has several interesting properties related to derivatives. This is explored in our paper Lüthen et al. (2022c) available in Chapter 5.



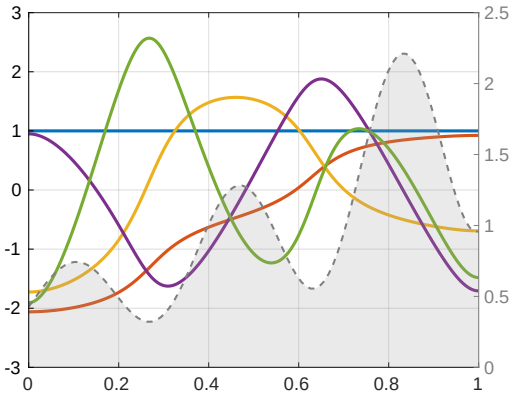
*(a) Truncated Gaussian (trunc. to $[-3, 3]$)*



*(b) Uniform*



*(c) Triangular*



*(d) Arbitrary*

***Figure 2.3:*** *The first five basis functions of Poincaré bases associated to different densities: Truncated Gaussian, uniform, triangular, and a custom density. The basis functions are plotted in colors, while the densities are shown in gray. We observe that higher-order functions oscillate more. The first eigenfunction is constant, and the functions have zero derivative at the boundary since they observe the homogeneous Neumann boundary condition (Eq. (2.19)).*

### 2.4.1.3 Truncation

In order to compute the chaos expansion in Eq. (2.11) for a given model, the coefficients of the expansion need to be determined. In general, infinitely many coefficients are needed for an exact representation, which is of course impossible in practice. Therefore, the expansion is truncated to a finite number of terms characterized by a multi-index set $\mathcal{A} \subset \mathbb{N}^d$. The most widely used truncation for PCE is the *total-degree* truncation of order $p \in \mathbb{N}$, where the total degree of each retained basis element does not exceed $p$:

$$\mathcal{A}^p = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \sum_{i=1}^{d} \alpha_i \leq p\}. \tag{2.21}$$

For the non-polynomial Poincaré chaos expansion, the word “degree” refers to the index of the univariate basis function or, equivalently, to the number of its roots.

More details about other possible truncation strategies and adaptive choices of truncation sets can be found in our publication Lüthen et al. (2022a), which is provided in Chapter 4.

#### 2.4.1.4 Computation of the coefficients

Since the most popular chaos expansion in the field of UQ is the PCE, most of the UQ literature about coefficient computation discussed in this section is dealing with orthonormal polynomial bases, but many results can directly be applied or be generalized easily to other types of chaos expansions.

Historically, PCE were first applied in an *intrusive* context with the goal of solving stochastic differential equations, i.e., differential equations with random parameters (Ghanem and Spanos, 1991). In the *stochastic Galerkin method*, polynomial functions of one or several random variables serve as a basis for the random part of the solution, in the same way as the (finite element) functions on a spatial grid might serve as a basis for the spatial part of the solution. The stochastic Galerkin method is called *intrusive* because in order to perform an uncertainty analysis for a given deterministic differential equation, the solution scheme itself needs to be modified to incorporate the additional basis functions for the random part.

However, because the modification of existing code is inconvenient (and impossible for certain industrial applications, as the source code of the used computer models is often not available), most research efforts nowadays focus on *non-intrusive* PCE, which are computed purely from input-output data pairs of the computational model. In the following, we briefly discuss the three main non-intrusive approaches: projection, interpolation, and regression.

*Projection* relies on Eq. (2.12) to compute each coefficient of the expansion by projecting the model onto the respective basis function. The integrals can be evaluated using numerical integration, e.g. Gauss quadrature, with methods such as *non-intrusive spectral projection* (Le Maître et al., 2002) or *sparse pseudospectral projection* (Constantine et al., 2012; Conrad and Marzouk, 2013). While the number of model evaluations for naive tensor product quadrature scales as $(p+1)^d$, sparse quadrature based on the Smolyak scheme (Gerstner and Griebel, 2003; Conrad and Marzouk, 2013) can alleviate this to some degree (Eldred et al., 2008).

Whereas projection uses the pointwise model evaluations indirectly for computing approximations to the integrals in Eq. (2.12), another class of methods directly *interpolates* the model in the given pointwise evaluations. In other words, for a set of *experimental design* points $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^d$ the goal is to find the coefficient vector $\boldsymbol{c}$ such that

$$\mathcal{M}(\boldsymbol{x}^{(i)}) = \mathcal{M}_{\boldsymbol{X}}^{\mathrm{PCE}}(\boldsymbol{x}^{(i)}; \boldsymbol{c}) \text{ for all } i = 1, \dots, N, \tag{2.22}$$

where the notation $\mathcal{M}_{\boldsymbol{X}}^{\mathrm{PCE}}(\,\cdot\,;\boldsymbol{c})$ denotes the PCE adapted to the random vector $\boldsymbol{X}$ and using the coefficient vector $\boldsymbol{c}$. This is sometimes called *stochastic collocation* (Xiu and Hesthaven, 2005; Babuška et al., 2007).[12] The choice of points $\mathcal{X}$ crucially influences the stability of the

[12] There is some confusion around the term "stochastic collocation" in PCE because it has been used by various authors to refer to different classes of methods. Narayan and Zhou (2015) use this term for both interpolation and regression-based methods. It is used to refer to least-squares regression by e.g. Hosder et al. (2007); Eldred et al. (2008); Narayan et al. (2017). Xiu (2009) also counts pseudospectral projection as stochastic collocation method, to contrast it with stochastic Galerkin methods.

interpolation procedure (Babuška et al., 2007; Narayan and Jakeman, 2014; Narayan and Zhou, 2015; Loukrezis et al., 2019).

Alternatively, the coefficients can be computed by *regression*, which aims to find an approximate solution to Eq. (2.22). One possibility is *ordinary least-squares regression* (OLS) (Isukapalli, 1999; Berveiller et al., 2006). Here, the model as well as the basis functions are evaluated on the experimental design $\mathcal{X}$, yielding the *response vector* $\boldsymbol{y} = (\mathcal{M}(\boldsymbol{x}^{(1)}), \dots, \mathcal{M}(\boldsymbol{x}^{(N)}))^T \in \mathbb{R}^N$ and the *regression matrix* $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ with entries $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{x}^{(i)})$ for an enumeration $\mathcal{A} = \{\boldsymbol{\alpha}_j\}_{j=1}^P$. To obtain the coefficients, the OLS problem

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 \tag{2.23}$$

is solved, which has the closed-form solution $\hat{\boldsymbol{c}} = (\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^T\boldsymbol{y}$. The inversion requires $N \geq P$, where $P = \binom{d+p}{d}$ for a total-degree basis of degree $p$ in $d$ dimensions (Eq. (2.21)). To avoid overfitting, a heuristic recommendation in the engineering community is to choose $N \geq 2P$ or $\geq 3P$ (Hosder et al., 2007; Fajraoui et al., 2017).[13]

The continuous version of the OLS problem, using the notation of Eq. (2.22), is

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}_{\boldsymbol{X}}^{\mathrm{PCE}}(\boldsymbol{X}; \boldsymbol{c})\right)^2\right]. \tag{2.24}$$

The analytical solution to this problem is given by Eq. (2.12), which makes the continuous version of OLS equivalent to the exact projection-based computation of the coefficients.

An alternative regression-based method that is often more sample-efficient than OLS is given by *sparse regression*, which we will explain in detail in Section 2.5.2.

### 2.4.2 Karhunen-Loève expansion

The chaos expansion described in the previous section was defined for deterministic models $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ – or, equivalently, for the induced random variables $Y = \mathcal{M}(\boldsymbol{X})$. Now, we turn to random fields $\{Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x}, \cdot)\}_{\boldsymbol{x} \in \mathcal{D}}$ as defined in Section 2.3.2, which are collections of random variables indexed over an index set $\mathcal{D}$.

#### 2.4.2.1 The standard case

Let $\mu(\boldsymbol{x})$ be the mean function and $C(\boldsymbol{x}, \boldsymbol{x}')$ be the covariance function of the random field (see Section 2.3.2). Assume that $\mathcal{D}$ is closed and bounded, $C$ is continuous, and that $\mathcal{M}(\boldsymbol{x}, \cdot)$ has finite variance for all $\boldsymbol{x} \in \mathcal{D}$.

Then the *Karhunen-Loève expansion* (KLE) of the random field $\mathcal{M}$ is given by the following representation (Kosambi, 1943; Karhunen, 1946; Loève, 1978; Ghanem and Spanos, 1991):

$$\mathcal{M}(\boldsymbol{x}, \omega) = \mu(\boldsymbol{x}) + \sum_{k=1}^{\infty} \sqrt{\lambda_k}\, \xi_k(\omega)\, \phi_k(\boldsymbol{x}) \tag{2.25}$$

[13] In fact, rigorous theoretical analyses of least squares methods for orthonormal polynomials have resulted in the following more demanding bounds: in one dimension, $\frac{N}{\log N} \geq c_1 P^2$ for OLS (Migliorati et al., 2014); in $d$ dimensions, $N \geq c_2 P \log P$ (Hampton and Doostan, 2015a) or $\frac{N}{\log N} \geq c_3 P$ for certain weighted least-squares methods (Cohen and Migliorati, 2017); where in all cases, $c_i$ is a constant that does not depend on $N$ or $P$.

where $\{\phi_k\}_{k=1,2,\dots}$ is an orthonormal basis of $L^2(\mathcal{D})$, $\lambda_1 \geq \lambda_2 \geq \dots \to 0$ is a decreasing sequence of real numbers, and $\{\xi_k\}_{k=1,2,\dots}$ is a countable family of zero mean, unit variance, uncorrelated random variables. The quantities $(\lambda_k, \phi_k)$ arise as solutions of the integral eigenvalue problem

$$\int_{\mathcal{D}} C(\boldsymbol{x}, \boldsymbol{x}')\phi_k(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}' = \lambda_k \phi_k(\boldsymbol{x}), \tag{2.26}$$

which constitutes a homogeneous Fredholm integral equation of the second kind (Atkinson, 1996). The KL-random variables (KL-RV) $\xi_k$ are the result of the projection of $\mathcal{M}$ onto the basis:

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}} \int_{\mathcal{D}} \left(\mathcal{M}(\boldsymbol{x}, \omega) - \mu(\boldsymbol{x})\right) \phi_k(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{2.27}$$

Note that both $\{\phi_k\}_{k=1}^{\infty}$ and $\{\xi_k\}_{k=1}^{\infty}$ are orthonormal bases, one capturing the "spatial" variation of $\mathcal{M}$ over $\mathcal{D}$ (in terms of $\boldsymbol{x}$), the other capturing the stochastic variation of $\mathcal{M}$ (in terms of $\omega$). In that sense, Eq. (2.25) can be called a diagonalization of the random field (Venturi, 2011).

The covariance function can be expressed as the uniformly converging series (Mercer's theorem) (Grigoriu, 2002)

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sum_{k=1}^{\infty} \lambda_k \phi_k(\boldsymbol{x})\phi_k(\boldsymbol{x}'). \tag{2.28}$$

It follows that the average variance of the random field over the domain $\mathcal{D}$ is equal to $\sum_{k=1}^{\infty} \lambda_k$. The series in Eq. (2.28) only depends on the spatial basis functions: the KL random variables do not enter this expression.

#### 2.4.2.2 Non-Gaussian random fields

For a Gaussian random field, the KL random variables $\xi_k$ are standard Gaussian and independent. However, KLE also exists for non-Gaussian fields. The non-Gaussianity is governed by the joint distribution $f_{\boldsymbol{\xi}}$, which might in general be complex and difficult to model or infer, leading some authors to conclude that KLE is mainly useful for Gaussian random fields (Grigoriu, 2006; Betz et al., 2014).

Still, several approaches for modeling the (joint) non-Gaussian distribution of KL random variables have been proposed. For given marginals and covariance function, Phoon et al. (2002, 2005) propose an algorithm which iteratively updates the distributions of the KL-RV in order to match the requested marginals. Other authors suggest to model the KL-RV by PCE using an iterative algorithm for the coefficients (Dai et al., 2019) or the method of fractional moments (Zhang et al., 2022). When only samples are given, the method of choice is often kernel density estimation (Grigoriu, 2010; Poirion and Zentner, 2013, 2014).

#### 2.4.2.3 Extended KLE

While in Eqs. (2.25) to (2.27), KLE is defined for a closed and bounded index set $\mathcal{D}$ and uses unweighted integrals, it can also be generalized to the weighted Hilbert space $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ with probability density $f_{\boldsymbol{X}}$. It is then called *extended KLE* (Iemma et al., 2006).

#### 2.4.2.4 Numerical computation

In practice, the infinite expansion in Eq. (2.25) must be truncated, resulting in

$$\mathcal{M}(\boldsymbol{x},\omega) \approx \mu(\boldsymbol{x}) + \sum_{k=1}^{M} \sqrt{\lambda_k}\, \xi_k(\omega)\, \phi_k(\boldsymbol{x}). \tag{2.29}$$

$K$ is usually chosen so that the fraction of *explained variance* satisfies

$$\frac{\sum_{k=1}^{M} \lambda_k}{\sum_{k=1}^{\infty} \lambda_k} > 1 - \epsilon \tag{2.30}$$

for a small threshold parameter $\epsilon > 0$. The basis $\{\phi_k\}_{k=1,2,\dots}$ is optimal in the sense that for any $M$, Eq. (2.29) has the smallest truncation error among all possible expansions in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$.

In most cases, the integral eigenvalue problem in Eq. (2.26) does not have an analytical solution. The numerical solution is challenging because the integral is potentially high-dimensional and the quantities of interest are not only vectors but functions. Numerous methods for solving Eq. (2.26) have been proposed, such as Galerkin methods (Ghanem and Spanos, 1991; Gutiérrez et al., 1992; Schwab and Todor, 2006), the Nyström method (Atkinson, 1996; Betz et al., 2014), expansion optimal linear estimation (EOLE) (Li and Der Kiureghian, 1993), orthogonal series expansion (OSE) (Zhang and Ellingwood, 1994), and degenerate kernel methods (Atkinson, 1996; Betz et al., 2014). We refer to the comprehensive review by Betz et al. (2014) for more details.

#### 2.4.2.5 Functional PCA

KLE can be seen as the continuous equivalent of the well-known dimension reduction method *principal component analysis* (PCA) (Jolliffe, 2002), which for a given random vector (or for realizations of a random vector) determines a new orthogonal coordinate system that is optimal for a truncated expansion. KLE does the same for random functions, and is therefore also known as *functional PCA* (Besse and Ramsay, 1986; Jolliffe, 2002; Ramsay and Silverman, 2005).

## 2.5 Sparsity and compressed sensing

Many numerical methods are affected by the so-called *curse of dimensionality*. This term denotes the explosion of the computational cost that often occurs when solving problems with a large number of parameters, which is referred to as their *dimension*. For example, the size of a PCE basis of total degree $p$ as defined in Eq. (2.21), which in $d$ dimensions is given by $P = \binom{p+d}{p}$, grows fast with increasing $d$. Solving a system of linear equations is another example: the number of required operations scales with $N^3$, where $N$ is the number of degrees of freedom (e.g., the number of nodes in a finite element mesh with increasingly fine resolution).

Due to the curse of dimensionality, many high-dimensional problems may exceed the capacity even of today's powerful supercomputers. While the capabilities of computers continue to grow due to technological progress, the curse of dimensionality will always be an issue as long as the

algorithm in question scales worse than the improvement in technology. The only solution here is to alleviate the curse of dimensionality by devising better algorithms.

Such a tool, able to mitigate the curse of dimensionality for certain applications, is given by *sparsity*. Let $\boldsymbol{v} \in \mathbb{R}^P$ be a vector, sometimes also called *signal*.[14] For example, $\boldsymbol{v}$ could be the vector of coefficients for a representation of an object in a suitable basis. The signal $\boldsymbol{v}$ is called *sparse* if most of its entries are equal to zero. More precisely, it is called *$s$-sparse* if it does not have more than $s$ nonzero entries. A closely related concept is *compressibility*: a signal is called *compressible* if most of its entries are close to zero and only a few of them are large in magnitude. In other words, a compressible signal is well-approximated by a sparse signal.

### 2.5.1 The setting of compressed sensing

The concept of sparsity has had a big impact on the field of signal processing with the introduction and rigorous analysis of *compressed sensing* (Candès et al., 2006; Donoho, 2006; Candès and Wakin, 2008), which is sometimes also known as *compressive sensing* or *compressive sampling*. Traditionally, signal processing was bound by the Nyquist rate: to reconstruct a time-dependent signal with a frequency of at most $B$ Hz from measurements, at least $2B$ samples at constant rate are necessary. The groundbreaking result of compressed sensing is the following: under the assumption that the representation of the signal in some suitable basis is $s$-sparse, and using a specific type of measurement acquisition, the signal can be reconstructed with much less than $2B$ samples – notably without knowing where in the sparse representation the $s$ nonzero entries occur.

To make these notions more precise, let $\boldsymbol{c} \in \mathbb{R}^P$ be the sparse signal of interest.[15] It is measured by the *sensing matrix* $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$, which results in the vector of measurements $\boldsymbol{y} = \boldsymbol{\Psi c} \in \mathbb{R}^N$, where usually $N < P$. In other words, $\boldsymbol{y}$ is a linear combinations of the columns of $\boldsymbol{\Psi}$, weighted by the entries of $\boldsymbol{c}$. Since $N < P$, the resulting linear system of equations $\boldsymbol{\Psi c}' = \boldsymbol{y}$ is underdetermined. Equivalently, in the compressed sensing literature this situation is also often referred to as *overcomplete basis*, i.e., there are many more elements in the basis (columns in $\boldsymbol{\Psi}$) than necessary to explain the measurements $\boldsymbol{y}$.

The goal is to *reconstruct* the sparse signal from the measurements, i.e., to find the sparsest solution $\boldsymbol{c}$ of the equation $\boldsymbol{\Psi c}' = \boldsymbol{y}$. Mathematically, this can be written using the $\ell^0$-"norm"[16] $\|\boldsymbol{c}\|_0 = \sum_i \mathbf{1}_{c_i \neq 0}$ counting the number of nonzero entries in $\boldsymbol{c}$ as follows:

$$\boldsymbol{c} = \arg \min_{\boldsymbol{c}' \in \mathbb{R}^P} \|\boldsymbol{c}'\|_0 \text{ subject to } \boldsymbol{\Psi c}' = \boldsymbol{y}. \tag{2.31}$$

The only way to solve this so-called *$\ell^0$-minimization problem* exactly is by a combinatorial search through all possible nonzero patterns for $\boldsymbol{c}$. However, this is infeasible for large problem sizes.

[14] This term is due to the origins of this theory in the field of signal processing.

[15] We assume for simplicity that the signal of interest $\boldsymbol{c}$ is sparse in $\mathbb{R}^P$. More generally, we can assume that $\boldsymbol{c}$ is sparse when expressed in a certain basis $\boldsymbol{\Phi} \in \mathbb{R}^{P \times P}$, i.e., $\boldsymbol{c} = \boldsymbol{\Phi a}$ with $\boldsymbol{a}$ sparse. Then, the compressed sensing methodology will be applied to find $\boldsymbol{a} = \arg\min_{\boldsymbol{a}' \in \mathbb{R}^P} \|\boldsymbol{a}'\|_0$ subject to $\boldsymbol{\Psi \Phi a}' = \boldsymbol{y}$.

[16] The $\ell^0$-"norm" is not actually a norm because the triangle inequality is not fulfilled.

The convex relaxation of this problem is given by $\ell^1$-*minimization*, where $\|\boldsymbol{c}\|_0$ is replaced by the $\ell^1$-norm $\|\boldsymbol{c}\|_1 = \sum_i |c_i|$:

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}' \in \mathbb{R}^P} \|\boldsymbol{c}'\|_1 \text{ subject to } \boldsymbol{\Psi}\boldsymbol{c}' = \boldsymbol{y}. \tag{2.32}$$

Under certain assumptions on $\boldsymbol{\Psi}$ and on the sparsity of $\boldsymbol{c}$, the solution of Eq. (2.32) coincides with the solution of Eq. (2.31). At the same time, Eq. (2.32) is much easier to solve because this problem is convex. An exemplary mathematical result of compressed sensing, making precise what assumptions are needed, is given in Section 2.5.2 below. Making compressed sensing even more powerful, the same methodology can be used to compute best $s$-sparse approximations $\boldsymbol{c}_s$ to compressible signals $\boldsymbol{c}$, achieving a trade-off between the accuracy of the solution $\boldsymbol{c}_s$ and the number $N$ of required measurements.

Considering signals that are sparse in some basis has two main advantages:

1. It improves the interpretability of the solution: it is immediately clear which basis elements are needed to explain the signal.
2. Due to sparsity, reconstruction is possible even in the case $N < P$, when the linear system of equations is underdetermined, which alleviates the curse of dimensionality.

Compressed sensing, and more generally sparsity, has had significant impact not only in the field of signal processing but also in many other fields, from biomedicine to information security (Kougioumtzoglou et al., 2020). Specifically, the field of engineering mechanics has benefitted from compressed sensing techniques, as discussed in the extensive review by Kougioumtzoglou et al. (2020). We will discuss sparsity in the context of engineering models in more detail in Section 2.5.3.

In the field of UQ, sparsity has been utilized in particular for sparse regression-based PCE, starting in 2008 with a method proposed by Blatman and Sudret (2008). We give an overview of the literature on sparse regression-based PCE in Section 2.5.4.

We now cite one exemplary result from the compressed sensing literature to illustrate the type of mathematical guarantee obtained by compressed sensing.

### 2.5.2 An exemplary result of compressed sensing

The following results are from Candès and Plan (2011), adapted to our notation. Let $\boldsymbol{c} \in \mathbb{R}^P$ be an $s$-sparse signal. Let $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ be the so-called *sensing matrix* (to be multiplied with the signal) and assume that each of its rows were sampled i.i.d. from a vector-valued distribution $F$ with the following two properties:

1. *Isotropy*: For a row $\Psi \sim F$, it holds that $\mathbb{E}\left[\Psi^T \Psi\right] = \mathbf{1}_P$ (with $\mathbf{1}_P$ the $P \times P$ identity matrix).
2. *Incoherence*: For the *coherence parameter* $\mu(F)$ defined by

$$\mu(F) = \sup_{\Psi \sim F} \max_{k=1,\dots,P} |\Psi_k|^2, \tag{2.33}$$

it holds that $\mu(F) < \infty$, where $\Psi_k$ is the $k$-th entry of a random row vector $\Psi \sim F$.[17]

[17] If such a bound does not exist (e.g., for the Gaussian example $F = \mathcal{N}(0, \mathbf{1}_P)$), a probabilistic bound can be defined so that $\mu(F)$ is with high probability an upper bound.

For example, the isotropic distribution $F = \mathcal{N}(\mathbf{0}, \mathbf{1}_P)$ results in a sensing matrix $\boldsymbol{\Psi}$ whose entries are i.i.d. standard Gaussian random variables. Its coherence can be shown to be $\mu(F) = 6 \log P$ (Candès and Plan, 2011).

PCE provides another example of an isotropic sensing matrix: assuming that the experimental design $\mathcal{X}$ has been sampled i.i.d. from the input distribution $f_{\boldsymbol{X}}$, a PCE regression matrix $\boldsymbol{\Psi}$ containing the basis function evaluations at these locations is isotropic, because the basis functions are orthonormal w.r.t. $f_{\boldsymbol{X}}$: $\mathbb{E}_{\boldsymbol{X}}\left[\psi_k(\boldsymbol{X})\psi_l(\boldsymbol{X})\right] = \delta_{kl}$. We will talk more about the application of compressed sensing theory to PCE in Section 2.5.4 below.

Now, consider the $\ell^1$-minimization problem in Eq. (2.32). Candès and Plan (2011) obtain the following result:

**Theorem 1 (Noiseless incoherent sampling (Candès and Plan, 2011)).** *Let $\boldsymbol{c}$ be a fixed but otherwise arbitrary $s$-sparse vector in $\mathbb{R}^P$ and pick any scalar $\beta > 0$. Then with probability at least $1 - \frac{5}{P} - e^{-\beta}$, $\boldsymbol{c}$ is the unique minimizer to Eq. (2.32) with $\boldsymbol{y} = \boldsymbol{\Psi}\boldsymbol{c}$ provided that [the number $N$ of rows of $\boldsymbol{\Psi}$, each sampled from $F$, fulfills]*

$$N \geq C_\beta \, \mu(F) \, s \, \log(P). \tag{2.34}$$

*More precisely, $C_\beta$ may be chosen as $C_0(1+\beta)$ for some positive numerical constant $C_0$.*

If we wanted to solve $\boldsymbol{\Psi}\boldsymbol{c} = \boldsymbol{y}$ by inversion of $\boldsymbol{\Psi}$, we would need at least $N = P$ points. In contrast, the bound in Eq. (2.34) grows only logarithmically in $P$, up to the factor $\mu(F)$ that depends on the way the sensing matrix $\boldsymbol{\Psi}$ is constructed. This makes the coherence parameter $\mu(F)$ an important property that crucially influences the efficiency of the procedure. A number of contributions in the PCE literature computes and optimizes the coherence for the distribution $F$ induced by the PCE regression matrix (Doostan and Owhadi, 2011; Yan et al., 2012; Hampton and Doostan, 2015a,b; Shin and Xiu, 2016; Jakeman et al., 2017; Narayan et al., 2017); see also Section 2.5.4 below.

The cited result relies on the coherence parameter $\mu(F)$ to guarantee recovery. Bounds similar to Eq. (2.34) have been obtained for other properties of the sensing matrix $\boldsymbol{\Psi}$, for example the *spark* (Donoho and Elad, 2003), the *mutual coherence* (Donoho et al., 2006), and the *restricted isometry property* (RIP) (Candès et al., 2008).

While the above result is for noiseless recovery, i.e., $\boldsymbol{\Psi}\boldsymbol{c} = \boldsymbol{y}$, similar results exist for the case that $\boldsymbol{c}$ is not exactly $s$-sparse but only compressible, or that $\boldsymbol{y}$ is corrupted by noise. For this case, Candès and Plan (2011) propose to solve the so-called *LASSO problem*[18]

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}' \in \mathbb{R}^P} \left\| \boldsymbol{\Psi}\boldsymbol{c}' - \boldsymbol{y} \right\|_2^2 + \lambda \left\| \boldsymbol{c}' \right\|_1 . \tag{2.35}$$

This results in bounds on the norms $\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_1$ and $\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2$ which involve the best $s$-sparse approximation to the true solution $\boldsymbol{c}$, while requiring the same number of samples as the noiseless case in Eq. (2.34). $\lambda$ is a hyperparameter governing the trade-off between accuracy and sparsity, chosen as $\lambda = 10\sqrt{P}$ by Candès and Plan (2011). While Eq. (2.35) looks similar to *ridge regression*, which uses the $\ell^2$-norm for the constraint in the second summand and can be solved

[18] LASSO stands for "least absolute shrinkage and selection operator".

analytically, the LASSO problem does not have a closed-form solution (Hastie et al., 2001, Ch. 3.4.2).[19]

Besides Eqs. (2.31) and (2.35), there are many other problem formulations and algorithms that provably encourage sparse solutions. For example, a formulation equivalent to Eq. (2.35) is *basis pursuit denoising* (BPDN) given by

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}'} \|\boldsymbol{c}'\|_1 \text{ s.t. } \|\boldsymbol{\Psi}\boldsymbol{c}' - \boldsymbol{y}\|_2 \leq \sigma. \tag{2.36}$$

Instead of measuring sparsity with the $\ell^1$-norm $\|\boldsymbol{c}\|_1 = \sum_i |c_i|$, other sparsity-enhancing formulations make use e.g. of a combination of different norms such as $\ell^1 - \ell^2$ minimization (Yin et al., 2015) or elastic net (Hastie et al., 2001), or of Bayesian formulations utilizing a sparsity-encouraging prior on the coefficients (Tipping, 2001; Tipping and Faul, 2003; Babacan et al., 2010). A plethora of algorithms has been developed for solving these formulations, from gradient-descent based methods such as spectral projected gradient-$\ell^1$ (SPGL1) for (2.36) (van den Berg and Friedlander, 2008) to stepwise regression algorithms such as orthogonal matching pursuit (OMP) (Pati et al., 1993; Tropp and Gilbert, 2007) or least-angle regression (LARS) (Efron et al., 2004) to iterative algorithms for sparse Bayesian learning (Tipping and Faul, 2003; Wipf and Rao, 2004; Ji et al., 2008; Babacan et al., 2010), to name just a few. General surveys of compressed sensing algorithms are available, e.g., in Bruckstein et al. (2009); Qaisar et al. (2013); Zhang et al. (2015); Arjoune et al. (2017).

### 2.5.3 Sparsity in engineering models

In computational as well as experimental engineering, it is common to deal with sparse data, i.e., only a small number of point measurements of the system of interest in a few configurations is available (Kougioumtzoglou et al., 2020). This is the case, for example, in structural health monitoring where a small number of sensors on a structure is used to determine the state of the system (Bao et al., 2011). The sparse set of data must be used together with some structural assumptions to draw conclusions about the system of interest.

As elaborated in the extensive review by Kougioumtzoglou et al. (2020), compressed sensing has successfully been applied to several engineering subfields, from structural health monitoring to uncertainty modeling and uncertainty propagation. With our focus on surrogate modeling, this thesis belongs to the third subfield. In particular, we will use sparse PCE and its close relative, sparse PoinCE, to surrogate deterministic simulators, while for stochastic simulators we will employ a method that uses sparse PCE as one of its building blocks (see Chapters 3 to 6).

The successful application of compressed sensing requires a suitable basis in which the signal of interest is sparse. As evident from the success of sparse PCE for engineering models (see, e.g., Blatman and Sudret (2008); Chatterjee et al. (2019); Hariri-Ardebili and Sudret (2020)), the polynomial chaos basis seems to be well suited for this task. A tentative explanation for

[19] In the special case that $\boldsymbol{\Psi}$ has orthogonal columns and is overdetermined, the solution to Eq. (2.35) can be obtained analytically by shrinking and truncating the coefficients of the OLS solution in Eq. (2.23) (soft thresholding) (Hastie et al., 2001, Ch. 3.4). However, this formula is not applicable in the underdetermined case, in which the OLS solution cannot be computed.

this observation could be that many quantities of interest computed from engineering models are effectively smooth functions of their input parameters, or arise from physical processes that have (close to) linear, exponential or low-frequency oscillatory behavior and are thus well approximated by polynomials.

Furthermore, due to independence, the orthonormal polynomial basis of PCE is constructed as a tensor product of univariate bases (Section 2.4.1). By construction, most of the resulting terms only involve a subset of all variables. The number of basis functions in a total-degree basis is large, but often only a few of these functions are actually needed to represent the model. A few heuristic arguments can be found in the literature to explain this observation:

- If the model is high-dimensional, it is often the case that only a few of its input parameters are important. Becker (2020) relates this to the well-known *Pareto principle* (also known as "80/20 rule"): a small number of variables is able to explain a large amount of variation. In other words, models often have *low effective dimensionality*.
- The response of a system is often dominated by main effects (i.e., terms involving only a single variable) and low-order interactions between variables, an observation called *sparsity-of-effects principle* (Montgomery, 2004).

However, it should be noted that not all engineering models are sparse in the PCE basis. Most are only compressible, i.e., the magnitude of their PCE coefficients decays fast. At the same time, there are also examples that are not well approximated by polynomials, e.g., functions such as likelihoods with highly localized behavior, non-differentiable functions arising e.g. from taking maxima, or discontinuous models. For such models, a possible solution are adaptive domain partitioning schemes in which local surrogates are constructed (Le Maître et al., 2004; Wan and Karniadakis, 2006; Moustapha and Sudret, 2019; Marelli et al., 2021b).

We will numerically explore the sparsity of engineering models in terms of the non-polynomial Poincaré basis (Section 2.4.1.2) in Chapter 5.

### 2.5.4 Sparse regression for polynomial chaos expansions

Shortly after the breakthrough in compressed sensing around 2006 (Donoho, 2006; Candès et al., 2006), these ideas also arrived in the UQ community working on PCE. The first papers suggest stepwise regression algorithms to build a sparse solution step by step (Blatman and Sudret, 2008, 2010). Subsequently, the $\ell^1$-minimization formulation and the associated rigorous algorithms were applied to PCE, yielding *sparse PCE* (Blatman and Sudret, 2011; Doostan and Owhadi, 2011; Mathelin and Gallivan, 2012) which continues to receive considerable attention from both the engineering and the applied mathematics communities.

Regression-based PCE perfectly fits into the framework of compressed sensing. For an experimental design $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\} \subset \mathcal{D}$ sampled from the input distribution $f_{\boldsymbol{X}}$, the regression matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ is obtained by evaluating the orthonormal polynomial basis functions at the experimental design points, resulting in entries $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{x}^{(i)})$. The columns of $\boldsymbol{\Psi}$ are by design orthogonal in expectation: in the terminology of Section 2.5.2, $\boldsymbol{\Psi}$ is isotropic. The sparse signal to be "recovered" is the vector of coefficients $\boldsymbol{c}$ that defines the PCE through

$\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$. The measurements are the model evaluations collected into the response vector $\boldsymbol{y} = (\mathcal{M}(\boldsymbol{x}^{(1)}), \dots, \mathcal{M}(\boldsymbol{x}^{(N)}))^T$.

As introduced in Eq. (2.23), the ordinary least squares (OLS) regression problem is to find $\boldsymbol{c}$ with $\boldsymbol{c} = \arg\min_{\boldsymbol{c}'\in\mathbb{R}^P} \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}'\|_2^2$. This problem has the unique solution $\boldsymbol{c}_{\mathrm{OLS}} = \left(\boldsymbol{\Psi}^T\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^T\boldsymbol{y}$ if the regression matrix $\boldsymbol{\Psi}$ has rank $P$, which requires $N \geq P$ (Berveiller et al., 2006).

In the case of PCE, a sparse solution is desired not because sparse solutions are better approximations but for computational reasons: a sparse solution of sufficient accuracy can often be obtained with considerably fewer model evaluations than a dense solution would require. However, if enough data was available, a denser solution might provide an overall better approximation.

To find sparse coefficient vectors $\boldsymbol{c}$ for given data, the PCE community has developed numerous approaches ranging from methods that closely follow recent developments in compressed sensing (Blatman and Sudret, 2011; Hampton and Doostan, 2015b; Jakeman et al., 2017; Alemazkoor and Meidani, 2018a) to heuristic algorithms which achieve sparsity by iteratively adding or removing nonzero coefficients from the vector (Blatman and Sudret, 2008, 2010; Shao et al., 2017; Alemazkoor and Meidani, 2017). Not all methods that are currently in use can be traced back to a specific article introducing it to the PCE community. There are a number of methods that have been used for years without an explicit introduction, e.g., orthogonal matching pursuit (OMP) (Pati et al., 1993; Tropp and Gilbert, 2007), spectral projected gradient-$\ell^1$ (SPGL1) (van den Berg and Friedlander, 2008), and subspace pursuit (Dai and Milenkovic, 2009).

Furthermore, as explained in Section 2.5.2, the properties of the regression matrix are important for the quality of the result. The regression matrix is induced by the sampling of the experimental design, whose choice can therefore influence the final result. Again, the literature on sampling schemes is vast. Currently used and investigated schemes include, among others, Latin hypercube sampling (LHS) (McKay et al., 1979; Shields and Zhang, 2016), D-optimal sampling (Kiefer and Wolfowitz, 1959; Diaz et al., 2018), and coherence-optimal sampling (Hampton and Doostan, 2015b). Some of these methods introduce weights into the regression problem.

Further contributions to the sparse PCE literature include preconditioning schemes (Alemazkoor and Meidani, 2018b), basis adaptivity (Jakeman et al., 2015; Hampton and Doostan, 2018), rotation of the input coordinates (Tipireddy and Ghanem, 2014; Yang et al., 2018; Tsilifis et al., 2019; Papaioannou et al., 2019), and sequential sampling of the experimental design (Fajraoui et al., 2017; Hampton and Doostan, 2018; Diaz et al., 2018).

For a thorough and extensive literature review, we refer to our publications Lüthen et al. (2021) and Lüthen et al. (2022a) provided in Chapters 3 and 4.

## 2.6 Sensitivity analysis

An important part of understanding the model behavior under uncertainty is the analysis of the relative importance of each of the input variables, which is known as *sensitivity analysis* (Saltelli et al., 2008). There are numerous approaches for sensitivity analysis from local to global methods, ranging from correlation-based methods suited mainly for linear models to the more

general but more costly variance-based or density-based approaches. For extensive reviews and comparisons of the available methods, see Iooss and Lemaître (2015); Borgonovo and Plischke (2016); Borgonovo (2017); Becker (2020); Da Veiga et al. (2021).

To assess the global sensitivity of a real-valued nonlinear model with independent input variables, one of the most widespread approaches is the variance-based *Sobol' sensitivity analysis*, which we describe in detail in Section 2.6.1. Sobol' indices determine the contribution of each input parameter (and of each possible combination of parameters) to the variance of the output.

In the case that partial derivatives of the computational model are available, the *derivative-based global sensitivity measures* (DGSM) can be computed. These indices are defined as the $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$-norm of the partial derivatives over the input space. We present this method in Section 2.6.2.

Further notable developments in global sensitivity analysis include the following. *Shapley indices* were originally introduced in game theory to attribute the "created value" of a team to each of the team members. In UQ, they can be applied for sensitivity analysis using the "team" of input variables and a suitable definition for the created value, e.g., the unnormalized closed Sobol' index (Owen, 2014; Owen and Prieur, 2017). The density-based *Borgonovo indices* measure the expected $L^1$-difference between the unconditional output density and the density conditioned on the $i$-th variable (Borgonovo, 2007). These were generalized by Da Veiga (2015) by replacing the $L^1$-difference by other dissimilarity measures between distributions. Recently, Da Veiga (2021) proposed two types of moment-independent indices based on kernel mean embeddings and the maximum mean discrepancy measure, or the Hilbert-Schmidt independence criterion (HSIC), respectively. Both indices allow an ANOVA-like decomposition analogous to Eq. (2.38) and can be seen as density-based generalizations of Sobol' indices.

In Sections 2.6.1 and 2.6.2, we present two global methods in more detail, namely Sobol' indices and derivative-based global sensitivity indices (DGSM), which will be needed in Chapter 5. In Section 2.6.3 we discuss which sensitivity analysis methods are applicable in the more general cases when the input variables are dependent or when the output is not scalar and real-valued. The latter includes the case of stochastic simulators.

### 2.6.1 Hoeffding-Sobol' decomposition and Sobol' indices

Assuming that the input random variables $X_i, i = 1, \dots, d$ are independent, any function $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ can be decomposed uniquely as a sum of terms of increasing complexity

$$\begin{aligned}\mathcal{M}(\boldsymbol{x}) = m_0 + \sum_{1\le i\le d} m_i(x_i) + \sum_{1\le i<j\le d} m_{i,j}(x_i, x_j) \\ + \sum_{1\le i<j<k\le d} m_{i,j,k}(x_i, x_j, x_k) + \cdots + m_{1,\dots,d}(x_1, \dots, x_d),\end{aligned} \tag{2.37}$$

where the terms satisfy $\int m_I(\boldsymbol{x}_I) f_{X_k}(x_k) \mathrm{d}x_k = 0$ for all $I \subset \{1, \dots, d\}$ and $k \in I$. This implies that the terms are mutually orthogonal, i.e., $\int m_I(\boldsymbol{x}_I) m_J(\boldsymbol{x}_J) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 0$ for $I \neq J$. This decomposition is known as *Hoeffding-Sobol' decomposition* or *ANOVA decomposition* (Hoeffding, 1948; Efron and Stein, 1981; Sobol', 1993).

ANOVA stands for "analysis of variance", which refers to the following decomposition resulting from taking the variance on both sides of Eq. (2.37):

$$\mathrm{Var}\left[\mathcal{M}(\boldsymbol{X})\right] = \sum_{1\leq i\leq d} \mathrm{Var}\left[m_i(X_i)\right] + \sum_{1\leq i<j\leq d} \mathrm{Var}\left[m_{i,j}(X_i,X_j)\right] + \cdots + \mathrm{Var}\left[m_{1,\ldots,d}(X_1,\ldots,X_d)\right]. \tag{2.38}$$

Since all these terms are non-negative, it is clearly visible which amount of the output variance is explained by variable $X_i$ alone: this quantity is called *first-order Sobol' index* and is given by

$$S_i^1 = \frac{\mathrm{Var}\left[m_i(X_i)\right]}{\mathrm{Var}\left[\mathcal{M}(\boldsymbol{X})\right]}. \tag{2.39}$$

The total influence of a variable $X_i$ on the output variance is given by the *total Sobol' index*

$$S_i^T = \frac{1}{\mathrm{Var}\left[\mathcal{M}(\boldsymbol{X})\right]} \sum_{J:i\in J} \mathrm{Var}\left[m_J(\boldsymbol{X}_J)\right], \tag{2.40}$$

gathering all the contributions by terms that involve variable $X_i$. Due to the normalization, $0 \leq S_i^1 \leq 1$ and $0 \leq S_i^T \leq 1$. It holds that $S_i^1 \leq S_i^T$, and the difference between these two indices quantifies the interaction effect involving $X_i$. If $S_i^1$ is large, $X_i$ is important. On the other hand, $S_i^T$ close to zero implies that $X_i$ does not have a large effect on the variance of the output; in other words, setting $X_i$ to a constant value will not have a large impact on the output uncertainty.

Sobol' indices can be computed by Monte Carlo-based methods (Homma and Saltelli, 1996; Jansen, 1999; Sobol', 2001; Janon et al., 2014; Becker, 2020). An alternative is to compute them from a surrogate model (Iooss and Lemaître, 2015; Le Gratiet et al., 2017). For this, polynomial chaos expansions are especially suitable, because the tensor product basis functions directly relate to the terms of the Hoeffding-Sobol' decomposition as follows (Sudret, 2008). Let $\mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$. Then

$$m_I(\boldsymbol{X}_I) = \sum_{\substack{\boldsymbol{\alpha}:\alpha_i>0 \text{ for } i\in I,\\ \alpha_j=0 \text{ for } j\notin I}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \tag{2.41}$$

and it follows that the Sobol' indices can be computed from the PCE coefficients as

$$S_i^1 = \frac{1}{D} \sum_{\substack{\boldsymbol{\alpha}:\alpha_i>0,\\ \alpha_j=0 \text{ for } j\neq i}} c_{\boldsymbol{\alpha}}^2, \tag{2.42}$$

$$S_i^T = \frac{1}{D} \sum_{\boldsymbol{\alpha}:\alpha_i>0} c_{\boldsymbol{\alpha}}^2 \tag{2.43}$$

with the *total variance* $D = \mathrm{Var}\left[\mathcal{M}(\boldsymbol{X})\right] = \sum_{\boldsymbol{\alpha}\neq\boldsymbol{0}} c_{\boldsymbol{\alpha}}^2$.

More generally, the formulas for the first-order and total Sobol' indices Eqs. (2.42) and (2.43) hold for any tensorized spectral expansion whose associated univariate bases each contain the constant function as first element.

### 2.6.2 Derivative-based global sensitivity measure (DGSM)

When the partial derivatives of the computational model are available, a global sensitivity index can be obtained by integration. The so-called *derivative-based sensitivity measure* (DGSM)

index of $\mathcal{M}$ with respect to $X_i$ (Sobol and Gresham, 1995; Kucherenko et al., 2009; Sobol' and Kucherenko, 2009) is defined by

$$\nu_i = \mathbb{E}\left[\left(\frac{\partial \mathcal{M}}{\partial x_i}(\boldsymbol{X})\right)^2\right] = \int_{\mathbb{R}^d}\left(\frac{\partial \mathcal{M}}{\partial x_i}(\boldsymbol{x})\right)^2 f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \left\|\frac{\partial \mathcal{M}}{\partial x_i}\right\|^2. \tag{2.44}$$

This can be seen as a continuous version of the Morris sensitivity indices (Morris, 1991; Lamboni et al., 2013). DGSM can efficiently be computed by Monte Carlo simulation (Kucherenko et al., 2009). An alternative is the computation through PCE, which is possible analytically for certain polynomial families (Sudret and Mai, 2015).

### 2.6.3 Dependent input and non-scalar output

Most of the methods described above assume that the input variables are independent and that the output is real-valued and continuous. We now briefly discuss extensions to the more general cases of dependent input and non-scalar output. In particular, the latter includes sensitivity analysis for stochastic simulators.

When the input variables are dependent, several methods, including the popular Sobol' indices, are not applicable or give answers that are difficult to interpret. Efficiently conducting meaningful sensitivity analysis for dependent variables is an active area of research, and several solutions have been proposed from local polynomial regression (Da Veiga et al., 2009) over modified variance-based indices utilizing random variable transformations (Mara and Tarantola, 2012; Mara et al., 2015) to Shapley indices (Owen, 2014; Owen and Prieur, 2017; Da Veiga, 2021). For a comprehensive overview of the current state of the art in this field, we refer to Da Veiga et al. (2021).

For the treatment of time- or space-dependent vector-valued or functional output, Nagel et al. (2020) and Perrin et al. (2021) reduce the dimensionality of the output by principal component analysis (PCA) before applying Sobol' analysis to the principal components. Nagel et al. (2020) analyze the evolution of the resulting Sobol' indices with time, while Perrin et al. (2021) aggregate the indices using the generalized sensitivity index of Lamboni et al. (2011) and Gamboa et al. (2014). Density-based methods like the Borgonovo indices or the dissimilarity measure generalization by Da Veiga (2015) can take multivariate output into account by integrating the density measure over the whole multivariate output range. Finally, the kernel-based methods proposed by Da Veiga (2021) can be applied to output of any type (categorical, vector-valued, functional) provided that suitable kernels are used.

Sensitivity analysis can also be performed for stochastic simulators (Section 2.3). As discussed by Zhu and Sudret (2021c), Sobol' indices can be generalized to stochastic simulators in several ways. The first is to consider both the explicit and latent variables as input variables, resulting in classical Sobol' indices for the extended input space (Iooss and Ribatet, 2009). The second way is to consider a statistic of the output random variable, which effectively reduces the stochastic simulator to a deterministic one, and to perform Sobol' analysis for this statistic (Iooss and Ribatet, 2009; Azzi et al., 2020). Finally, Sobol' analysis can be applied to each of the deterministic trajectories of the stochastic simulator obtained by fixing the latent variables.

The resulting indices are random variables that depend on the latent variables (Hart et al., 2017; Navarro Jimenez et al., 2017).

Kernel-based methods provide another alternative for the sensitivity analysis of stochastic simulators. Using the random variable view of Eq. (2.5), a stochastic simulator can be seen as a deterministic model with random variable output. Therefore, with a suitable choice of kernel for probability distributions, the recently proposed kernel-based indices by Da Veiga (2021) can be applied.

## 2.7 Software for uncertainty quantification

There is an increasing awareness in the computational science community that understanding the influence of uncertainty on the model output is an important part of the quality assurance of computational models, and that UQ methods can give valuable insights about the model itself. UQ is an active field of research, and a vast amount of efficient and reliable techniques have been and continue being developed to address the questions from Section 1.1. This also poses a challenge: modelers and practitioners, while being experts in their respective field of applied science or engineering, cannot always be aware of the most recent developments and recommendations in UQ, and might perform outdated, inaccurate or inefficient analyses (Saltelli et al., 2019).

In view of this, it is necessary to test the developed methods on practical problems, and to benchmark them to provide recommendations. At the same time, it is important to make the recent developments available to the public in form of software that is thoroughly tested and easy to use, so that not every researcher has to implement the UQ method of interest (which might be mathematically or computationally quite involved) from scratch. Fortunately, numerous such software projects have been developed by experts in the field of UQ, written in various programming languages and with different focus areas and target groups. Among others, there are the Python libraries UQPy (Olivier et al., 2020), UQ[Py]Lab (Lataniotis et al., 2021), and PyApprox (Jakeman, 2022); the C++/Python libraries OpenTURNS (Andrianov et al., 2007) and UQTk (Sargsyan et al., 2022); the C++ toolkit Dakota (Adams et al., 2014); the R sensitivity package (Iooss et al., 2021); the R/Shiny platform Lagun (Da Veiga et al., 2022); and the Matlab-based software packages Cossan (Patelli et al., 2014) and UQLab (Marelli and Sudret, 2014).

For the research conducted in the context of this thesis, we have made extensive use of UQLab. UQLab is a general-purpose UQ software developed by the Chair of Risk, Safety and Uncertainty Quantification at ETH Zürich. It is based on Matlab (Mat, 2020) and comprises numerous modules covering all steps of the UQ framework (Section 2.1), notably the modeling of the input, surrogate models like sparse PCE and Kriging, as well as sensitivity analysis, but also further modules such as Bayesian inversion, random fields, or reliability analysis, just to name a few. Selected results of our research related to sparse PCE solvers (Chapter 3) were already made available in the recent release UQLab Version 1.4.0 (Marelli et al., 2021a).

## Summary and outlook on the following chapters

This concludes our review of the state of the art in uncertainty quantification, in which we specifically focused on the topics relevant for this thesis, i.e., surrogate modeling by spectral methods, sparsity, and sensitivity analysis. In the subsequent Chapters 3 to 6, we report our research results which build on the presented material as follows.

In Chapters 3 and 4, we survey and classify the literature on sparse regression-based PCE, whose basics we reviewed in Section 2.4.1 (PCE) and Section 2.5 (sparse regression). Furthermore, we perform an extensive benchmark of the most promising methods using UQLab (Section 2.7).

In Chapter 5, we analytically and numerically investigate the properties of sparse Poincaré chaos expansions, relying on the theory presented in Section 2.4.1.2 (Poincaré chaos), Section 2.5 (sparse regression), and Section 2.6 (sensitivity analysis).

Finally, in Chapter 6 we propose a new sparse spectral surrogate model for stochastic simulators, which belongs to the class of surrogate models taking the random function view as explained in Section 2.3.3.2. We use sparse polynomial chaos expansion (Sections 2.4.1 and 2.5) and Karhunen-Loève expansion (Section 2.4.2) to construct a surrogate that is able to approximate non-Gaussian stochastic simulators.

CHAPTER 3

# Sparse polynomial chaos expansions: Literature survey and benchmark

This chapter contains the postprint of

> Lüthen, N., Marelli, S., Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif.* *9*(2), 593–649.[1]

as published in the *SIAM/ASA Journal of Uncertainty Quantification*, differing from the published paper only in terms of layout and formatting.

**Author contributions.** **N. Lüthen:** Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization. **S. Marelli:** Conceptualization, Methodology, Writing - Review & Editing, Supervision. **B. Sudret:** Conceptualization, Methodology, Writing - Review & Editing, Supervision, Funding Acquisition.

## Abstract

Sparse polynomial chaos expansions (PCE) are a popular surrogate modelling method that takes advantage of the properties of PCE, the sparsity-of-effects principle, and powerful sparse regression solvers to approximate computer models with many input parameters, relying on only few model evaluations. Within the last decade, a large number of algorithms for the computation of sparse PCE have been published in the applied math and engineering literature. We present an extensive review of the existing methods and develop a framework for classifying

[1]First published in SIAM/ASA Journal of Uncertainty Quantification in Volume 9, Number 2, 2021, published by the Society for Industrial and Applied Mathematics (SIAM). Copyright © by SIAM and ASA. Unauthorized reproduction of this article is prohibited.

the algorithms. Furthermore, we conduct a unique benchmark on a selection of methods to identify which approaches work best in practical applications. Comparing their accuracy on several benchmark models of varying dimensionality and complexity, we find that the choice of sparse regression solver and sampling scheme for the computation of a sparse PCE surrogate can make a significant difference, of up to several orders of magnitude in the resulting mean-squared error. Different methods seem to be superior in different regimes of model dimensionality and experimental design size.

## 3.1 Introduction

Computer modelling is used in nearly every field of science and engineering. Often, these computer codes model complex phenomena, have many input parameters, and are expensive to evaluate. In order to explore the behavior of the model under uncertainty (e.g., uncertainty propagation, parameter calibration from data or sensitivity analysis), many model runs are required. However, if the model is costly, only a few model evaluations can be afforded, which often do not suffice for thorough uncertainty quantification. In engineering and applied sciences, a popular work-around in this situation is to construct a surrogate model. A surrogate model is a cheap-to-evaluate proxy to the original model, which typically can be constructed from a relatively small number of model evaluations and approximates the input-output relation of the original model well. Since the surrogate model is cheap to evaluate, uncertainty quantification can be performed at a low cost by using the surrogate model instead of the original model. Therefore, surrogate modelling aims at constructing a metamodel that provides an accurate approximation to the original model while requiring as few model evaluations as possible for its construction.

In this article, we focus on *nonintrusive regression-based sparse polynomial chaos expansions* (PCE), which is a popular surrogate modelling technique, and within the last decade is has received attention from the communities of applied mathematics and engineering. PCE express the computational model in terms of a basis of polynomials orthonormal with respect to the input random variables (Xiu and Karniadakis, 2002) and work well for globally smooth problems, which are common in many engineering applications. In addition to being a surrogate model, PCE are also often used for uncertainty propagation and sensitivity analysis, since moments and Sobol' sensitivity indices can be computed analytically (Sudret, 2008). *Nonintrusive* PCE treat the model as a black box (unlike intrusive PCE commonly used for solving stochastic PDEs). It is often advantageous to compute a *sparse PCE*, which is an expansion for which most coefficients are zero. This can be justified by the *sparsity-of-effects principle* and by *compressibility*: The sparsity-of-effects principle is a heuristic stating that most models describing physical phenomena are dominated by main effects and interactions of low order (Montgomery, 2004). Furthermore, PCE of real-world models are usually either sparse or at least compressible, meaning that the PCE coefficients, sorted by magnitude, decay quickly. Additional advantages of sparse expansions are given in Section 3.2.2.

Within the last decade, a large number of articles has been published on the topic of regression-based sparse PCE, each containing promising improvements on how to perform sparse PCE but

often lacking a thorough comparison to previously published methods. In this work, we survey the state-of-the-art literature, develop a general framework into which the various approaches can be fit, and carry out a numerical benchmark of a selection of methods to assess which of the many sparse PCE methods perform best on a representative set of realistic benchmark models.

The paper is structured as follows. Section 3.2 contains the description of our framework for classifying the sparse PCE literature as well as the extensive literature review. Section 3.3 contains the benchmark description and the numerical results. Finally, conclusions are drawn in section 3.4. More detailed descriptions of selected sparse solvers and experimental design techniques are given in the Appendices using unified notation.

## 3.2 Framework and literature survey for sparse polynomial chaos expansions

### 3.2.1 Regression-based polynomial chaos expansions

Let $\boldsymbol{X}$ be a $d$-dimensional random vector on a domain $\mathcal{D} \subset \mathbb{R}^d$ with independent components and probability density function $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{d} f_{X_i}(x_i)$. Let $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ be the space of all scalar-valued models with finite variance under $f_{\boldsymbol{X}}$, i.e., $L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = \{h : \mathcal{D} \to \mathbb{R} \mid \mathrm{Var}_{\boldsymbol{X}}[h(\boldsymbol{X})] < +\infty\}$. Under certain assumptions on the input distribution $f_{\boldsymbol{X}}$ (Xiu and Karniadakis, 2002; Ernst et al., 2012), there exists a polynomial orthonormal basis $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^d\}$ for $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$. Since the components of $\boldsymbol{X}$ are assumed to be independent, the basis elements are products of univariate orthonormal polynomials and are characterized by the multi-index $\boldsymbol{\alpha} \in \mathbb{N}^d$ of polynomial degrees in each dimension.

We consider a particular model $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ and denote by $Y = \mathcal{M}(\boldsymbol{X})$ the corresponding output random variable. $Y$ can be represented exactly through an infinite expansion in $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^d\}$. In practice, however, not all infinitely many coefficients can be computed, and we are interested in a truncated expansion

$$Y = \mathcal{M}(\boldsymbol{X}) \approx \mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \tag{3.1}$$

whose accuracy depends on the choice of the finite set $\mathcal{A} \subset \mathbb{N}^d$ (i.e., on the basis elements used for the expansion) as well as on the coefficients $c_{\boldsymbol{\alpha}}$. Several truncation techniques are described in Section 3.2.4.

To compute the coefficients, one well-known and practical approach is regression (Isukapalli, 1999; Berveiller et al., 2006)[2]. The basic regression approach is *ordinary least squares* (OLS). Let $\{\boldsymbol{x}^{(k)}\}_{k=1}^{N} \subset \mathcal{D}$ be a sample of the input space called *experimental design* (ED). Let $\boldsymbol{y} = (y^{(1)}, \dots, y^{(N)})^T$ be the vector of model responses with $y^{(k)} = \mathcal{M}(\boldsymbol{x}^{(k)})$. Define the matrix

[2]Other, earlier approaches for computing the coefficients are stochastic Galerkin and stochastic collocation methods (Ghanem and Spanos, 1991; Xiu and Hesthaven, 2005; Shen et al., 2020). For a comparison of their performance to regression-based PCE, see, e.g., Berveiller (2005); Hosder et al. (2007); Doostan and Owhadi (2011); Mathelin and Gallivan (2012).

of basis function evaluations $\boldsymbol{\Psi}$ with entries $\Psi_{ij} = \psi_j(\boldsymbol{x}^{(i)})$, where the basis functions are enumerated in an arbitrary way. Denoting the number of basis functions with $P$, we see that the *regression matrix* $\boldsymbol{\Psi}$ is an $N \times P$-matrix. Then, the OLS regression problem can be written as

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}\in\mathbb{R}^P} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2 \,. \tag{3.2}$$

For a unique and robust solution, a heuristic number of model evaluations is $N \approx 2P, 3P$ (Hosder et al., 2007; Fajraoui et al., 2017), which can be infeasible for high-dimensional or high-degree PCE approximations.

### 3.2.2 Sparse PCE

Sparse coefficient vectors are determined through *sparse regression*, which, in addition to a good regression fit, requires that the solution be sparse. This constraint on sparsity is realized e.g. by adding as a regularization term the $\ell^0$-"norm" or the $\ell^1$-norm of the coefficient vector to the OLS formulation of (3.2) (see Appendix 3.B for more details). Many sparse regression methods used in PCE were originally developed in the context of compressive sensing (Donoho, 2006; Candès et al., 2006). For an introduction to the concepts and ideas of compressive sensing, see, e.g., Candès and Wakin (2008); Bruckstein et al. (2009); Kougioumtzoglou et al. (2020).

Unlike OLS, compressive sensing methods allow one to use fewer design points $N$ than basis functions $P$ and still recover the true sparse solution, or find a good sparse approximation to it. This and its robustness to noise, both of which are induced by the sparsity constraint, are the main reasons why sparse PCE are preferred to full PCE in practical settings when the number of model evaluations necessary for OLS-based PCE would be infeasible to compute. Note that while the use of sparse PCE for engineering models can be justified by compressibility and the sparsity-of-effects heuristic (section 3.1), the main goal in sparse PCE is to compute a good surrogate model from a few model evaluations and not to find the sparsest possible expansion. The assumption of sparsity is used as a tool for finding robust solutions to underdetermined systems of linear equations.

The first publications on sparse regression-based PCE proposed greedy forward-backward selection algorithms (Blatman and Sudret, 2008, 2010) and introduced the LARS algorithm for sparse PCE (Blatman and Sudret, 2011). On the mathematical side, Doostan and Owhadi (2011) analyzes convergence properties for sparse Legendre PCE when the design points are sampled from the uniform distribution. Another early work is Mathelin and Gallivan (2012) demonstrating that sparse PCE are less costly and more accurate than PCE based on Smolyak sparse grids. Since then, a large number of articles has been published on the topic of sparse PCE suggesting new methods for specific aspects of the sparse regression procedure. In the following, we present a framework into which the existing literature can be fit. The framework provides an overview of the available choices and enables a structured comparison of their impact on the performance of the resulting sparse PCE. Naturally, some new combinations of methods arise that have not yet been considered in the literature.

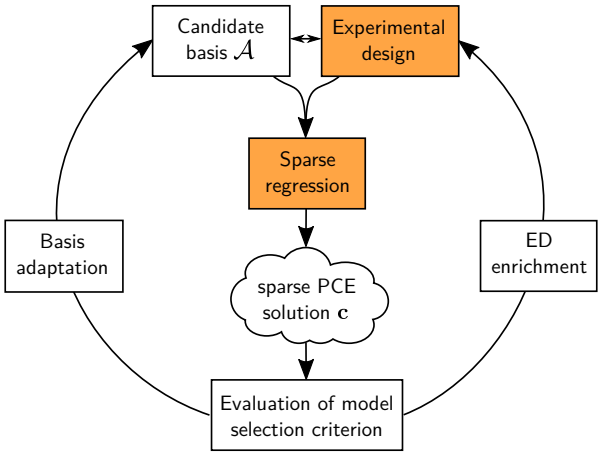



***Figure 3.1:*** *Framework for computing sparse PCE. For each component of the framework, a number of methods has been proposed in the literature. In the first part (section 3.2) of this paper, we review the literature for each of the components. Details on selected methods are given in Appendices 3.A and 3.B. In the second part (section 3.3), we conduct a benchmark of selected methods for the components marked in orange, performing a single iteration of the framework. Iterative basis adaptation and experimental design enrichment are not considered in this work and are left for future benchmarks.*

### 3.2.3 Framework: Classifying the literature on sparse PCE

Here, we present the framework we developed in order to gain an overview of the extensive literature proposing new methods for computing sparse PCE. Figure 3.1 shows a sketch of this framework. To compute a sparse PCE, the first step is to choose a set $\mathcal{A}$ of candidate polynomials for the expansion (Section 3.2.4) as well as an experimental design (Section 3.2.5). The experimental design defines the locations of the model evaluations. Once the model evaluations are obtained, the sparse solution can be computed by applying a sparse regression solver (see Section 3.2.6). This solver often depends on a number of hyperparameters that have to be selected carefully in order to get good results. Then, a suitable model selection criterion is evaluated (Section 3.2.7). If the obtained solution is satisfactory, the process can be stopped. Otherwise, the basis can be adapted (usually augmented; see Section 3.2.4), and/or the experimental design can be enriched (see Section 3.2.5.4). This process is repeated until the value of the model selection criterion either is satisfactory or cannot be reduced.

In addition to the components shown in Figure 3.1, there are methods (we call them *enhancements*) that aim at generally improving the solution to the sparse regression problem by, e.g., adapting the input space or preconditioning the regression matrix. They are discussed in Section 3.2.8.

### 3.2.4 Choice of basis and basis adaptation

The approximation quality of a truncated PCE hinges on the polynomial functions available for building the surrogate model, which are characterized by the associated set of multi-indices $\mathcal{A} \subset \mathbb{N}^d$. We call the finite set of polynomials $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}\}$ included in the current truncated PCE model *basis* and call its members *candidate polynomials* or *candidate basis functions*.

A sparse PCE algorithm will find nonzero coefficients only for a subset $\mathcal{A}^{\text{active}} \subset \mathcal{A}$ of the basis functions, which we call *active basis functions*. On the one hand, $\mathcal{A}$ should include enough candidate polynomials to facilitate a good approximation. On the other hand, unnecessary basis functions decrease the ratio $N/P$ of model evaluations to unknown coefficients and deteriorate the regression matrix. Therefore, it is beneficial to carefully select the polynomials to be included in the expansion.

The choice of basis is often motivated by the *sparsity-of-effects principle*, a heuristic guideline stating that most real-world models are well approximated by terms of low degree and low interaction order. The following are popular ways to construct a basis:

- *Total-degree* A total-degree basis of degree $p$ is defined by $\mathcal{A}^p = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_1 \leq p\}$.
- *Hyperbolic truncation* Let $p$ be fixed. Define the q-norm-truncated basis

$$\mathcal{A}^{p,q} = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_q \leq p\} \tag{3.3}$$

  with $q \in (0, 1]$ (Blatman and Sudret, 2011) and the quasi-norm $\|\boldsymbol{x}\|_q = \left(\sum_{i=1}^d |x_i|^q\right)^{\frac{1}{q}}$. For $q = 1$, $\mathcal{A}^{p,1}$ is the total-degree basis of order $p$. For smaller $q$, this truncation scheme excludes terms with high interaction order while keeping univariate polynomials up to degree $p$.
- *Interaction order* The interaction order of the basis can be restricted by defining

$$\mathcal{A}^{p,r} = \{\boldsymbol{\alpha} \in \mathcal{A}^p : \#\{i : \alpha_i \neq 0, i = 1, \ldots, d\} \leq r\} \tag{3.4}$$

  (Blatman and Sudret, 2008; Marelli and Sudret, 2019). This is useful for reducing the number of basis functions especially in high dimensions and when it is known (e.g., for physical reasons) that only a certain number of variables might interact.

Instead of using a fixed basis $\mathcal{A}$, it can be beneficial to employ an iterative scheme which starts from a small set of basis functions (low-dimensional, low-order) and, after computing a sparse solution, repeatedly adapts the basis by including a set of the most promising candidate polynomials and possibly removing others. This is called *basis adaptivity* (not to be confused with Gaussian adaptation (Tipireddy and Ghanem, 2014); see also Section 3.2.8).

A simple instance of basis adaptivity is *degree adaptivity* (Blatman and Sudret, 2010), which is based on total-degree bases. The procedure starts with a basis of low total degree and iteratively increases the total degree of the basis. Finally, a model selection criterion is used to select the best basis and associated sparse solution. Similarly, q-norm and interaction order, or a combination of all three, can also be used to design a basis adaptation scheme (Blatman and Sudret, 2010, 2011). This basis adaptivity is solution-agnostic in the sense that it does not use any information from the solutions computed in previous runs for the augmentation of the basis. Another solution-agnostic method is the dimension- and order-incrementing algorithm of Alemazkoor and Meidani (2017). Two methods that adapt the basis based on the active terms of the previous sparse solution are forward neighbor basis adaptivity (Sargsyan et al., 2014; Jakeman et al., 2015) and anisotropic degree basis adaptivity (Hampton and Doostan, 2018). These approaches keep the size of the basis small by strictly controlling which functions are added to the basis, often starting with a constant surrogate model and adding dimensions

only when necessary. A discussion and benchmark of basis-adaptive methods for sparse PCE is available in Lüthen et al. (2022a).

### 3.2.5 Experimental design

Generally, experimental design techniques aim to select points in order to achieve certain goals related to exploration of the space (space-filling design), or to achieve certain properties of the regression matrix such as (in expectation) orthonormal columns, small determinant, small condition number, etc. For regression-based PCE, there are several main classes of experimental design techniques:

- *Sampling based on the input distribution.* The samples are drawn from the input distribution. Techniques like Latin hypercube sampling (LHS) can be used to improve the space-filling properties.
- *Sampling from a different distribution* (also called *induced sampling* (Guo et al., 2020)). A different distribution and associated basis are constructed that have better properties than the input distribution and its basis.
- *Choosing points according to an optimality criterion from a candidate set.* Certain properties of the regression matrix are optimized by choosing the design points from a suitable candidate set.

Note that none of these sampling methods considers the evaluations $\boldsymbol{y}$ of the computational model. For model-aware sampling techniques (active and supervised learning), see Section 3.2.5.4.

The following sections contain an overview of available sampling methods for sparse PCE. Selected experimental design techniques are described in more detail in Appendix 3.A. Note that due to the large amount of literature on experimental design for sparse PCE, our review cannot be exhaustive. There are many more approaches available, including the deterministic Weil points (Zhou et al., 2014), sparse grids (Perkó et al., 2014), randomized or subsampled quadrature points (Berveiller et al., 2006; Tang and Iaccarino, 2014; Guo et al., 2017), etc.

#### 3.2.5.1 Sampling based on the input distribution

The most basic sampling method is Monte Carlo (MC) sampling, where the points are sampled independently from the input distribution (Doostan and Owhadi, 2011; Hampton and Doostan, 2015b). LHS (McKay et al., 1979) aims at distributing the design points in a more space-filling way than MC sampling, using a stratification of the input quantile space in each dimension. LHS is known to filter main effects; i.e., it reduces the variance of linear regression estimators when the quantity of interest is dominated by terms of interaction order one (Shields and Zhang, 2016). LHS with sample decorrelation can further reduce the variance (Owen, 1994). LHS can also be used together with a criterion such as maximin distance (maximize the minimal distance between the design points in quantile space) (Pronzato and Müller, 2012), where several LHS designs are generated and the one that optimizes the criterion is returned.

A generalization of LHS that combines it with stratified sampling is *Latinized partially stratified sampling* (Shields and Zhang, 2016), which filters both main effects and low-order interaction terms and has been shown to consistently outperform LHS in high-dimensional cases.

Other space-filling/low-discrepancy methods are Sobol' sequences (Sobol', 1967) and Halton sequences (Halton, 1960), which are deterministic but appear to be quasi-random and space-filling in low dimensions.

#### 3.2.5.2 Sampling from a different distribution

Several methods consider the *coherence* parameter of a basis, defined by

$$\mu(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\}) = \sup_{\boldsymbol{x}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|^2, \tag{3.5}$$

which can be used to bound the number of samples needed for accurate recovery by $\ell^1$-minimization (Candès and Plan, 2011; Hampton and Doostan, 2015b). For Hermite and Legendre polynomial bases, the coherence parameter of a total-degree basis grows exponentially with the total degree $p$ (Rauhut and Ward, 2012; Yan et al., 2012; Hampton and Doostan, 2015b).

To construct a *coherence-optimal design*, a new probability distribution and its associated orthonormal basis are constructed that achieve minimal coherence (Hampton and Doostan, 2015b,a). The new basis can be derived from the original PCE basis by multiplying each member by a weight function. Coherence-optimal samples can be drawn by Markov Chain Monte Carlo (MCMC) (Hampton and Doostan, 2015b) or by rejection sampling (see section 3.A.2.3). A related sampling scheme is obtained by constructing a new probability distribution and associated orthonormal basis which have improved but not optimal coherence; however, the distribution is constructed to belong to some classical family and is therefore straightforward to sample. This is called *asymptotic sampling* (Hampton and Doostan, 2015b,a) and results in a Chebyshev distribution for uniform input, and in a uniform distribution (within a ball of degree-dependent radius) for Gaussian input. Numerical experiments confirm the expected performance gain of coherence-optimal sampling over both MC and asymptotic sampling, and of asymptotic sampling over MC in the case of low dimension $d$ and high total degree $p$. For high-dimensional problems with low degree, MC often performs better than asymptotic sampling (Hampton and Doostan, 2015b).

The so-called *Christoffel sparse approximation (CSA)* (Jakeman et al., 2017; Narayan et al., 2017; Cohen and Migliorati, 2017) is a related approach which constructs a new orthonormal basis that minimizes the quantity

$$\tilde{\mu}(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\}) = \sup_{\boldsymbol{x}\in\mathcal{D}} \left( \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{\alpha}\in\mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|^2 \right)^{\frac{1}{2}}. \tag{3.6}$$

As for coherence-optimal sampling, the new basis can be derived from the original basis by multiplying each member by a weight function, which results in a weighted regression problem. The corresponding probability distribution is chosen to be the so-called *weighted pluripotential equilibrium measure*, which for bounded distributions is the Chebyshev distribution. For

one-dimensional Gaussian input, this measure is a symmetric Beta distribution with degree-dependent bounds.

Note that all three sampling methods described in this section introduce weights and therefore modify the objective function into a weighted regression problem $\boldsymbol{W}\boldsymbol{\Psi}\boldsymbol{c} \approx \boldsymbol{W}\boldsymbol{y}$. Since the objective function belongs to the scope of the solver, these methods cannot be considered as pure sampling methods in the sense of being completely independent of the solver.

#### 3.2.5.3 Choosing points according to an optimality criterion from a candidate set

The following methods choose points from a candidate set in order to optimize properties of the regression matrix. Candidate points can be sampled, e.g., using MC, LHS (Fajraoui et al., 2017), coherence-optimal sampling (Diaz et al., 2018; Alemazkoor and Meidani, 2018a), or Christoffel sampling (Shin and Xiu, 2016). Note that some of these methods introduce weights, resulting in a weighted regression problem. The candidate set can have a large influence on the resulting design.

- *D-optimal sampling* aims at maximizing the determinant $D(\boldsymbol{\Psi}) = \det(\frac{1}{N}\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{\frac{1}{P}}$ of the information matrix (Kiefer and Wolfowitz, 1959). Note that $D(\boldsymbol{\Psi}) = 0$ if $N < P$. Maximizing the D-value is connected to minimizing the variance of the coefficients of the PCE estimate (Zein et al., 2013). Algorithms for D-optimal designs include greedy augmentation (Dykstra, 1971), exchange techniques (Fedorov, 2013; Cook and Nachtsheim, 1980; Nguyen and Miller, 1992; Zein et al., 2013), maxvol (Mikhalev and Oseledets, 2018), gradient descent (Zankin et al., 2018), and rank-revealing QR decomposition (RRQR)/subset selection (Diaz et al., 2018; Gu and Eisenstat, 1996). The advantage of the last method is that it can also be applied for wide matrices $\boldsymbol{\Psi} \in \mathbb{R}^{N\times P}$ where $N < P$.
- *S-optimal sampling* (also called "quasi-optimal" in Shin and Xiu (2016a)) selects samples from a large pool of candidate points so that the PCE coefficients computed using the selected set are as close as possible to the coefficients computed from the whole set of candidate points (Shin and Xiu, 2016a). The S-value is defined by[3]

$$S(\boldsymbol{\Psi}) = \left(\frac{\sqrt{\det \boldsymbol{\Psi}^T\boldsymbol{\Psi}}}{\prod_{i=1}^{P} \|\Psi_i\|_2}\right)^{\frac{1}{P}} \tag{3.7}$$

  where $\Psi_i$ denotes the $i$th column of the regression matrix. Its maximization has the effect of maximizing the column orthogonality of the regression matrix while at the same time maximizing the determinant of the information matrix (Shin and Xiu, 2016a). Note that $S(\boldsymbol{\Psi}) = 0$ if $N < P$. An S-optimal experimental design can be computed using a greedy exchange algorithm (Shin and Xiu, 2016a,?; Fajraoui et al., 2017).
- *Near-optimal sampling* simultaneously minimizes the two matrix properties *mutual coherence* and *average cross-correlation* (Alemazkoor and Meidani, 2018a), both of which quantify the correlation between normalized columns of the regression matrix (see section 3.A.3.3 for the definitions of these properties). A near-optimal design can be built by

[3]This definition assumes that the columns of the matrix $\boldsymbol{\Psi}_{\text{cand}}$, containing the evaluations of all candidate points, are mutually orthogonal.

a greedy algorithm (Alemazkoor and Meidani, 2018a). Note that for near-optimal sampling, it is not necessary that $N \geq P$, since this method does not rely on the determinant of the information matrix.

#### 3.2.5.4 Sequential enrichment of the experimental design

Instead of sampling the whole experimental design at once, it has been proposed to use *sequential enrichment*. Starting with a small experimental design, additional points are chosen based on the last computed sparse PCE solution or on an augmented basis. In the context of machine learning, sequential sampling is also known as *active learning* (Settles, 2012). Sequential enrichment has been proposed in the context of S-optimal sampling (Fajraoui et al., 2017), D-optimal sampling (Diaz et al., 2018), and coherence-optimal sampling (Hampton and Doostan, 2018). Zhou et al. (2019) suggest an enrichment strategy based on approximations to the expected quadratic loss function, i.e., the mean-squared error. Ji et al. (2008) and Seeger and Nickisch (2008) propose choosing points that minimize the differential entropy of the posterior distribution of the coefficients (using a Bayesian regression setting). In all cases, numerical examples show that the sequential strategy generally leads to solutions with a smaller validation error compared to nonsequential strategies. Due to the complexity of the topic and the already large extent of our benchmark, this strategy, albeit promising, is not explored further in this paper.

### 3.2.6 Solution of the minimization problem

There are many formulations of the regression problem that lead to a sparse solution, such as $\ell^0$-minimization, $\ell^1$-minimization (basis pursuit denoising (BPDN), LASSO), $\ell^1 - \ell^2$ minimization, Bayesian methods, etc. (see also Appendix 3.B). Based on these formulations, a vast number of sparse solvers has been proposed in the compressed sensing literature; see, e.g., Carron (2013) and the surveys of Qaisar et al. (2013); Zhang et al. (2015); Arjoune et al. (2017). We focus here on solvers that have been proposed in the context of sparse PCE. Of course, it is straightforward to use any other sparse solver to compute a sparse PCE.

The following solvers have been proposed in the sparse PCE literature:

- *Convex optimization solvers.* $\ell^1$-minimization in its various formulations is a (constrained) convex optimization problem. Least angle regression (LARS) (Efron et al., 2004; Blatman and Sudret, 2011; Marelli and Sudret, 2019) is an iterative method that adds regressors one by one according to their correlation with the current residual, and updates the coefficients following a least angle strategy. With the LARS-LASSO modification, which allows for backwards elimination of regressors, LARS is able to generate the whole LASSO path (Efron et al., 2004). Unmodified LARS can also be classified as a greedy method. SPGL1 (van den Berg and Friedlander, 2008; Van den Berg and Friedlander, 2015) solves the BPDN formulation by solving a succession of LASSO instances using the spectral projected gradient (SPG) method. Other solvers belonging to this class are e.g. the solvers implemented in $\ell_1$magic (Candès and Romberg, 2005) and SparseLab (Donoho et al., 2007).

- *Greedy methods* are variants of stepwise regression where the regressors are added to the model one by one according to some selection criterion, aiming at finding a heuristic solution to the intractable $\ell^0$-minimization formulation. Orthogonal matching pursuit (OMP) (Tropp and Gilbert, 2007; Doostan and Owhadi, 2011; Marelli and Sudret, 2019) is a classical forward selection algorithm in which orthonormalized regressors are added to the model one by one according to their correlation with the residual, and the coefficients are computed by least-squares. Baptista et al. (2019) suggests extensions to OMP such as parallelization, randomization and a modified regressor selection procedure. Subspace pursuit (SP) (Dai and Milenkovic, 2009; Diaz et al., 2018; Diaz, 2018) is an iterative algorithm that repeatedly uses least squares on a subset of regressors. LARS (Efron et al., 2004; Blatman and Sudret, 2011) without the LASSO modification (allowing for removal of regressors) can also be classified as a greedy method. Another greedy method is ranking-based sparse PCE (Tarakanov and Elsheikh, 2019) which employs batch updating, coordinatewise gradient descent of the elastic net formulation, and a correlation- and stability-based ranking procedure for the regressors. Many more greedy stepwise regression techniques have been proposed, utilizing various selection criteria, solvers, and stopping criteria. An overview of methods following this scheme is given in section 3.B.6.
- *Bayesian compressive sensing (BCS)* (a.k.a. sparse Bayesian learning) is a class of methods that use a Bayesian setting to find a sparse solution. They impose a sparsity-inducing prior on the coefficients, whose parameters are again considered to be random variables with a hyperprior (Tipping, 2001; Ji et al., 2008; Sargsyan et al., 2014; Tsilifis et al., 2020). The solution is typically the maximum a posteriori estimate of the coefficients and can be computed e.g. by differentiation (Tipping, 2001), expectation-maximization (Figueiredo, 2003; Wipf and Rao, 2004), expectation-propagation (Seeger and Nickisch, 2008), variational inference (Tsilifis et al., 2020; Bhattacharyya, 2020), or a fast approximate algorithm (Faul and Tipping, 2002; Tipping and Faul, 2003). An extension called FastLaplace with an additional layer of hyperparameters has been proven to attain even sparser solutions (Babacan et al., 2010; Babacan, 2011). A greedy algorithm using the Bayesian setting to select the regressors is the greedy Bayesian Kashyap information criterion (KIC)-based algorithm (Shao et al., 2017).
- *Iteratively reweighted methods.* Iteratively reweighted $\ell^1$-minimization uses the coefficients computed in a previous iteration to construct a weighted $\ell^1$-minimization problem (Candès et al., 2008; Yang and Karniadakis, 2013). Cheng and Lu (2018b) suggest an iterative reweighted method with D-MORPH regression (Li and Rabitz, 2010) as its computational core, which is a technique that follows a certain path, defined by a quadratic objective function, on the manifold of solutions to the underdetermined system.

Each of the solvers mentioned above features one or more hyperparameters whose values must be calibrated. This is usually done by cross-validation. Popular choices are leave-one-out (LOO) cross-validation (accelerated for least-squares solutions) (Blatman and Sudret, 2010, 2011), LOO cross-validation with a modification factor for small sample sizes (Chapelle et al., 2002; Blatman and Sudret, 2011), and $k$-fold cross-validation (Doostan and Owhadi, 2011; Jakeman et al., 2015; Huan et al., 2018).

Selected sparse regression solvers are described in more detail in Appendix 3.B.

### 3.2.7 Model selection criterion

To decide whether to continue iterating in the framework or stop the process, we need to assess how well the current sparse solution performs. Our main quantity of interest is the *generalization error*, which quantifies the mean-square accuracy of the surrogate. It is given by

$$E_{\text{gen}} = \mathbb{E}_{\boldsymbol{X}}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}^{\text{PCE}}(\boldsymbol{X})\right)^2\right] \tag{3.8}$$

where $\mathcal{M}$ is the computational model, $\boldsymbol{X}$ is the random input vector, and $\mathcal{M}^{\text{PCE}}$ is the sparse PC surrogate.

The generalization error can be approximated by the *validation error*, which is the MC estimate of (3.8) on a validation set $\{(\boldsymbol{x}_{\text{val}}^{(i)}, y_{\text{val}}^{(i)}) : \boldsymbol{x}_{\text{val}}^{(i)} \sim_{\text{i.i.d.}} f_{\boldsymbol{X}}, y_{\text{val}}^{(i)} = \mathcal{M}(\boldsymbol{x}_{\text{val}}^{(i)}), i = 1, \dots, N_{\text{val}}\}$. To make the validation error independent of the scaling of the model, it is convenient to use the *relative mean-squared error* defined by

$$\text{RelMSE} = \frac{\sum_{i=1}^{N_{\text{val}}}(y_{\text{val}}^{(i)} - \mathcal{M}^{\text{PCE}}(\boldsymbol{x}_{\text{val}}^{(i)}))^2}{\sum_{i=1}^{N_{\text{val}}}(y_{\text{val}}^{(i)} - \bar{y})^2} \tag{3.9}$$

where $\bar{y} = \frac{1}{N_{\text{val}}}\sum_{i=1}^{N_{\text{val}}} y_{\text{val}}^{(i)}$.

The best surrogate model $\mathcal{M}^{\text{PCE}}$ (defined by $\mathcal{A}$ and $\boldsymbol{c}$) is the one that has the smallest generalization error. In practical applications, the generalization error typically cannot be computed, and a large validation set is not available due to computational constraints. Instead, we define a *model selection criterion* that acts as a proxy for the generalization error. A typical stopping criterion in the PCE framework of Figure 3.1 is the observation that the model selection criterion no longer improves. The following model selection criteria have been proposed in the sparse PCE literature:

- *k-fold cross-validation (CV)* (Hastie et al., 2001; Jakeman et al., 2015; Hampton and Doostan, 2018), which approximates the validation error by building a surrogate several times on different subsets of the data, and evaluating the error on the remaining data points.
- *Leave-one-out (LOO) cross-validation* (Hastie et al., 2001; Blatman and Sudret, 2010, 2011), which is $N$-fold cross-validation (where $N$ is the size of the experimental design). For PCE approximations computed by OLS, there exists an efficient formula to evaluate the LOO error (Blatman and Sudret, 2011, Appendix D).
- *Modified LOO* (Blatman and Sudret, 2011), which uses a correction factor for the LOO which was derived for the empirical error for OLS with small sample size (Chapelle et al., 2002). The correction factor depends on the experimental design and the active basis functions.
- *Kashyap information criterion (KIC)* (Shao et al., 2017; Zhou et al., 2019c), an approximation to the Bayesian model evidence, which is the likelihood of observations given the model.

- *Sparsity* (Alemazkoor and Meidani, 2017), which uses the idea that a larger basis should lead to a sparser solution when the necessary basis functions enter the model, unless the ratio of basis functions to model evaluations becomes too large.

A model selection criterion is also often used to determine the hyperparameter of the sparse solver (see Section 3.2.6).

If cross-validation is used to select the solver hyperparameter, this estimate of the validation error is often too optimistic due to model selection bias. Instead of reusing this estimate for model selection, it is better to perform an outer loop of $k$-fold or LOO cross-validation, a procedure called *double cross-validation* or cross-model validation (Baumann and Baumann, 2014; Liu et al., 2020b).

### 3.2.8 Further enhancements of sparse PCE

There are many enhancements to the simple scheme for sparse PCE presented in Figure 3.1. The following methods have been suggested to improve the accuracy of the solution and reduce the number of model evaluations needed:

- Alemazkoor and Meidani (2018b) construct a preconditioning matrix for a given regression matrix which reduces the mutual coherence while avoiding deterioration of the signal-to-noise ratio.
- Huan et al. (2018) suggest a technique called stop-sampling, which guides the decision of whether to obtain more samples (sequential ED enrichment) by observing the decrease of the CV error.
- In the case when prior information about the magnitude of the coefficients is available, Peng et al. (2014) use this information to construct a weighted regression problem which allows a more accurate solution with fewer points (similar to iteratively reweighted $\ell^1$-minimization).
- Liu et al. (2020b) use resampled PCE, which is a technique for improving the PCE solution by aggregating the results of several solver runs on different subsets of the data. Only the terms that are chosen most often by the solvers are retained in the final solution.
- Several methods exist to reduce the dimension of the input space before computing the sparse PCE. Unsupervised methods are principal component analysis (PCA) and kernel PCA (Lataniotis, 2019). "Basis adaptation" methods (referring to a basis of the input random space) determine a suitable rotation of the input space, often assumed to be independent standard Gaussian, into new coordinates which permit a sparser representation in fewer coordinates (see Tipireddy and Ghanem (2014); Yang et al. (2018); Tsilifis et al. (2019) and others). A related technique is nonlinear PCE-driven partial least squares (PLS) (Papaioannou et al., 2019; Zhou et al., 2020), which reduces the input dimension by identifying directions in the input space that are able to explain the output well in terms of a sum of one-dimensional PCEs.

## 3.3 Numerical results

### 3.3.1 Benchmark design

While the number of methods for computing sparse PCE is large, to the best of the authors' knowledge there is no comprehensive benchmark study on this topic. Most publications only compare the newly developed method to one or two baseline methods. An overview of publications containing comparisons of sparse PCE methods is presented in Appendix 3.D.

Since the number of possible combinations of sampling schemes, sparse regression solvers, basis adaptation schemes, model evaluation criteria etc. is huge (see Section 3.2.3 and thereafter), we restrict our benchmark as follows to some of the most promising and best-known methods:

- We consider the sampling schemes MC, LHS, coherence-optimal, and D-optimal. LHS is used together with a maximin criterion to improve the space-filling property (using the MATLAB function `lhsdesign`). D-optimal designs are constructed from a coherence-optimal candidate set[4] using the subset selection/RRQR algorithm, which allows for the construction of D-optimal experimental designs with size $N$ smaller than the number of regressors $P$ (Diaz et al., 2018). There is no such algorithm for S-optimal sampling, which is why we do not consider the latter in this benchmark. We do not consider Sobol' sequences, since they have been shown to be outperformed by LHS in sparse PCE applications (Fajraoui et al., 2017). Near-optimal sampling can realistically be used only for rather small bases ($P \in \mathcal{O}(100)$), since its algorithm scales as $\mathcal{O}(MP^2)$, with $M = 10P$ as suggested by (Diaz et al., 2018) (see below). We use it with a coherence-optimal candidate set for two models with small basis.
- We consider the sparse regression solvers LARS, OMP, subspace pursuit (SP), FastLaplace (which we call here BCS), and SPGL1. Each of these solvers involves at least one hyperparameter, whose range is chosen according to reasonable guesses. For LARS and OMP, the hyperparameter is the number $K$ of selected regressors and its range is $[1, \min\{P, N-1\}]$. For SP, $K$ must fulfill $2K \leq \min\{P, N\}$. For BCS and SPGL1, the hyperparameter $\sigma$ is chosen from the range $\sigma^2 \in N \cdot \hat{\mathrm{Var}}[\boldsymbol{y}] \cdot [10^{-16}, 10^{-1}]$ which resembles a suitable range of possible relative MSE values. The hyperparameter values of LARS and OMP are determined by modified LOO cross-validation, while the hyperparameters of SP, BCS, and SPGL1 are determined by $k$-fold cross-validation (Section 3.2.6). In addition, we consider a variant of SP which uses LOO cross-validation instead of $k$-fold cross-validation, which we name $SP_{LOO}$.
- We only consider the nonadaptive setting, in which both the basis and the size of the experimental design are fixed before the sparse PCE is computed.
- For each model, we define a reasonable range of 5–7 experimental design sizes. Each experiment is repeated 30–50 times to account for statistical uncertainty. The experimental

[4] We have also conducted all benchmark experiments with D-optimal designs constructed from LHS candidate sets, but we do not display these results, because we found that in most cases, D-opt(LHS) sampling performs (significantly) worse than most other sampling schemes, and often worse than its candidate set LHS. This matches with the results of Fajraoui et al. (2017) who observed this in a sequential enrichment setting and with the LARS solver.

designs are generated anew for each repetition and each ED size. All solvers are tested on the same ED realizations.

- The coherence-optimal candidate sets from which the D-optimal designs are selected have size $M = 10P$ as in (Diaz et al., 2018). For computational reasons, they are not sampled completely anew for each replication, but are rather drawn uniformly at random without replacement from a larger set of size $2M = 20P$ as in (Diaz et al., 2018).
- Since we are interested in sparse PCE for the purpose of surrogate modelling, our main quantity of interest is the relative mean-squared error[5][6] (RelMSE) as defined in (3.9). We investigate the RelMSE for several models, sparse solvers, and experimental design techniques. Typically, the practical interest lies in small experimental designs.
- Since the experimental design is random, the resulting validation error is a random variable. We visualize the data with boxplots. When comparing the performance of different methods, we consider the median performance and the spread of the resulting validation error. However, often there can be considerable overlap of validation errors between methods.

### 3.3.2 Software

For the implementation of the benchmark, we use the general-purpose uncertainty quantification software UQLab (Marelli and Sudret, 2014). UQLab supports the integration of other software packages.[7] We utilize the following code:

- UQLab for MC sampling and LHS (Marelli and Sudret, 2014).
- `DOPT_PCE` for D-optimal sampling (subset selection/RRQR) and subspace pursuit (Diaz et al., 2018; Diaz, 2018).
- An in-house developed rejection-based implementation of coherence-optimal sampling.
- An in-house implementation of near-optimal sampling based on the description by Alemazkoor and Meidani (2018a).
- UQLab for the solvers LARS and OMP (Marelli and Sudret, 2014).
- `spgl1-1.9` for SPGL1 (van den Berg and Friedlander, 2008; Van den Berg and Friedlander, 2015).
- `FastLaplace` for the hierarchical implementation "FastLaplace" of BCS (Babacan et al., 2010; Babacan, 2011).

[5]Note that some authors such as Doostan and Owhadi (2011); Hampton and Doostan (2015b); Diaz et al. (2018); Alemazkoor and Meidani (2018a) choose to normalize instead by $\sum_{\boldsymbol{x}\in\mathcal{X}_{\text{val}}} \mathcal{M}(\boldsymbol{x})^2$ or use the unnormalized mean-squared error (Shin and Xiu, 2016). To assess the recovery of sparse vectors just as in compressed sensing, some consider the error in the coefficient vector instead of the error in the model approximation (Alemazkoor and Meidani, 2018a).

[6]Since a typical application of PCE is the computation of moments and Sobol' indices, the error in these quantities is another possible performance measure. However, globally accurate prediction as considered in this paper is more challenging that the prediction of moments and Sobol' indices, which are accurate if the largest-in-magnitude coefficients are estimated accurately. If a globally accurate surrogate model can be constructed, typically also the moments and Sobol' indices are accurate.

[7]A description of how to use custom sparse solvers and sampling schemes in the UQLab framework can be found in the supplementary material.

### 3.3.3 Benchmark: Considered models

Our benchmark is performed on a selection of 11 computational models of varying complexity and input dimensionality, which are typical benchmark models in the context of sensitivity and reliability analysis. An overview of these models is given in Table 3.1. For details on the models, we refer the reader to the respective publications supplied in the last column of the table. While of course not representative of all possible classes of engineering models, we believe that this sample provides a good testing ground for the comparative performance among different approaches for computing sparse PCE.

***Table 3.1:*** *Overview of the 11 computational models used in our benchmark. Italic font denotes finite element (FE) models, all other models are analytical. For each model, a static total-degree basis with hyperbolic truncation defined by $p$ and $q$ is used. The values are chosen to fulfill $P \approx \frac{10}{3} N_{max}$, where $N_{\max}$ is the largest tested experimental design size. The values for $p$ in parentheses for the Ishigami and borehole models refer to the smaller basis used in Section 3.3.6. The column "Reference" provides the relevant literature in which the models and their probabilistic inputs are described in detail.*

| Model | Dimension | Input distributions | Basis | $N_{\max}$ | Reference |
|---|---|---|---|---|---|
| Ishigami function | 3 | uniform | $p = 14$ (12), $q = 1$ | 200 | (Blatman and Sudret, 2011) |
| Undamped oscillator | 6 | Gaussian | $p = 5$, $q = 1$ | 150 | (Echard et al., 2013) |
| Borehole function | 8 | Gaussian, lognormal, uniform | $p = 5$ (4), $q = 1$ | 300 | (Harper and Gupta, 1983) |
| Damped oscillator | 8 | lognormal | $p = 5$, $q = 1$ | 400 | (Dubourg, 2011) |
| Wingweight function | 10 | uniform | $p = 4$, $q = 1$ | 300 | (Forrester et al., 2008) |
| *Truss model* | 10 | lognormal, Gumbel | $p = 4$, $q = 1$ | 300 | (Blatman and Sudret, 2011) |
| Morris function | 20 | uniform | $p = 8$, $q = 0.5$ | 400 | (Blatman and Sudret, 2010b) |
| *Structural frame model* | 21 | lognormal, Gaussian; dependent input variables | $p = 8$, $q = 0.5$ | 400 | (Blatman and Sudret, 2010) |
| 2-*dim diffusion model* | 53 | Gaussian | $p = 4$, $q = 0.5$ | 500 | (Konakli and Sudret, 2016) |
| 1-*dim diffusion model* | 62 | Gaussian | $p = 4$, $q = 0.5$ | 500 | (Fajraoui et al., 2017) |
| 100D function | 100 | uniform | $p = 4$, $q = 0.5$ | 1400 | UQLab example[8] |

In addition to analyzing aggregated performance on all 11 models, we investigate the behavior of

**Table 3.2:** *Borehole function: Input random variables and their distributions*

| Variable | Distribution | Description |
|---|---|---|
| $r_w$ | $\mathcal{N}(0.10, 0.0161812)$ | borehole radius |
| $L$ | $\mathcal{U}([1120, 1680])$ | borehole length |
| $K_w$ | $\mathcal{U}([9855, 12045])$ | borehole hydraulic conductivity |
| $T_u$ | $\mathcal{U}([63070, 115600])$ | transmissivity of upper aquifer |
| $T_l$ | $\mathcal{U}([63.1, 116])$ | transmissivity of lower aquifer |
| $H_u$ | $\mathcal{U}([990, 1110])$ | potentiometric head of upper aquifer |
| $H_l$ | $\mathcal{U}([700, 820])$ | potentiometric head of lower aquifer |
| $r$ | Lognormal$([7.71, 1.0056])$ | radius of influence |

the methods in detail on a subset of four *spotlight models*, each of which possesses characteristic properties that might influence the approximation quality of sparse PCE methods: the *Ishigami function* is low-dimensional and highly compressible in the PCE basis but requires a high-degree basis to be approximated accurately. The *borehole function* is smooth and nonlinear and therefore is an example for a well-behaved engineering model. A *two-dimensional diffusion model*, a stochastic heat diffusion PDE in two physical dimensions, is high-dimensional, not analytical, and the magnitude of its expansion coefficients decays only slowly. Finally, the $100D$ *function* is high-dimensional, analytical, and compressible.

The Ishigami model is the well-known three-dimensional, highly nonlinear, smooth analytical function

$$f(X_1, X_2, X_3) = \sin(X_1) + a \sin^2(X_2) + b X_3^4 \sin(X_1) \tag{3.10}$$

taking uniform input $\boldsymbol{X} \sim \mathcal{U}([-\pi, \pi]^3)$. A typical choice is $a = 7, b = 0.1$. For this function, any sparse solver should be able to find a sparse solution.

The borehole function simulates the water flow through a borehole between two aquifers (Harper and Gupta, 1983). It is an eight-dimensional nonlinear function which, despite having an analytical form, is not trivial to approximate. It is defined by

$$B(r_w, L, K_w, T_u, T_l, H_u, H_l, r) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}. \tag{3.11}$$

Its input random variables and their distributions are provided in Table 3.2.

The two-dimensional heat diffusion model (Konakli and Sudret, 2016) is defined by the partial differential equation (PDE)

$$-\nabla \cdot (\kappa(\boldsymbol{x}) \nabla T(\boldsymbol{x})) = Q \mathbb{1}_A(\boldsymbol{x}) \qquad \text{in } \Omega = [-0.5, 0.5]^2 \tag{3.12}$$

with boundary conditions $T = 0$ on the top boundary and $\nabla T \cdot \boldsymbol{n} = 0$ on the left, lower, and right boundaries of the square domain $\Omega$, where $\boldsymbol{n}$ denotes the outer unit normal (see Konakli and Sudret (2016) for an illustration of the setup). Here, the source is in $A = [0.2, 0.3]^2$ with strength $Q = 500$. The output quantity of interest is the average temperature in $B = [-0.3, -0.2]^2$. The diffusion coefficient $\kappa(\boldsymbol{x})$ is modelled by a lognormal random field with mean $\mu_\kappa = 1$ and standard deviation $\sigma_\kappa = 0.3$. The autocorrelation function of the underlying Gaussian random field is an isotropic squared-exponential with length scale $l = 0.2$. The random field $\kappa(\boldsymbol{x})$ is discretized using the EOLE method (Li and Der Kiureghian, 1993) with $d = 53$ terms, which

comprises 99% of its variance. The solution to an individual heat diffusion problem is computed using an in-house finite element code (Konakli and Sudret, 2016). The input comprises $d = 53$ independent standard normal random variables.

Finally, the so-called 100D function is an analytical model of the form

$$f(\boldsymbol{X}) = 3 - \frac{5}{d}\sum_{i=1}^{d} iX_i + \frac{1}{d}\sum_{i=1}^{d} iX_i^3 + \frac{1}{3d}\sum_{i=1}^{d} i \ln\left(X_i^2 + X_i^4\right) + X_1X_2^2 + X_2X_4 - X_3X_5 + X_{51} + X_{50}X_{54}^2 \tag{3.13}$$

taking uniform inputs $X_i \sim \mathcal{U}([1,2]), i \neq 20$, and $X_{20} \sim \mathcal{U}([1,3])$. We use $d = 100$. This function was designed for sensitivity analysis: the first-order sensitivity indices of the input variables are generally nonlinearly increasing with their index, with certain variables having especially high sensitivity. The model also contains four interaction terms. It is an example from UQLab.[8]

For each of the models, we use a fixed basis for the benchmark. In general, the best total degree $p$ and the hyperbolic truncation $q$ are a priori unknown. We heuristically choose $q = 1$ for low-dimensional models ($d \leq 10$) and $q = 0.5$ for high-dimensional models ($d \geq 20$). The degree $p$ is chosen so that the number of basis functions $P$ is approximately $\frac{10}{3}N_{\max}$, where $N_{\max}$ is the largest number of experimental design points for the specific benchmark. This choice is based on the reasoning that for an experimental design of size $N$, sparse solvers like LARS often select an active basis of size $\approx \frac{N}{3}$, and that the candidate basis might be 10 times larger than the final active basis to be sufficiently rich. We focus on rather small experimental designs, since our goal is not to investigate the convergence of the methods as $N \to \infty$ (which has been demonstrated elsewhere), but to decide which methods are most efficient for small $N$. This results in the choice of values for $p$, $q$, and $N_{\max}$ displayed in Table 3.1.

### 3.3.4 Results: Comparison of solvers

First, we use a fixed sampling scheme (LHS) to compare the performance of the six solvers LARS, OMP, subspace pursuit (SP) using $k$-fold cross-validation, Subspace Pursuit using LOO cross-validation ($\text{SP}_{\text{LOO}}$), FastLaplace (BCS), and SPGL1 on all 11 benchmark models described above.

In Figure 3.2 we display boxplots (50 replications) of relative MSE against experimental design size for all six solvers for the four spotlight models. For the remaining seven models, the corresponding boxplots (30 replications) of relative MSE against experimental design size are provided in Figure 3.C.1 in Appendix 3.C. In the plots, the lines as well as the dot inside the white circle denote the median of the relative MSE. We make the following observations:

- For the smallest experimental designs, all solvers perform similarly poorly: there is not enough information in the ED to construct an accurate surrogate model. For larger experimental designs, there can be considerable differences between the solvers' generalization errors of up to several orders of magnitude, which demonstrates that the solvers do not use the available information in identical ways.

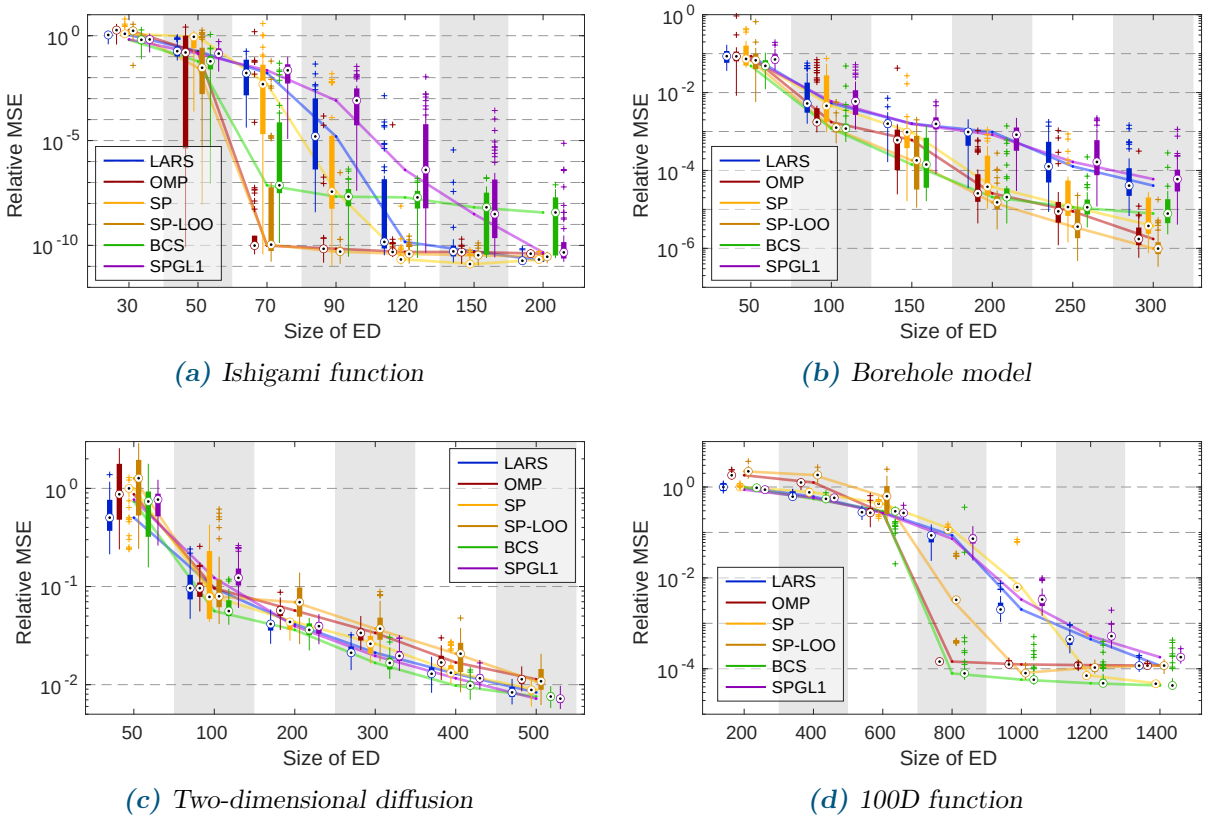
[8] https://www.uqlab.com/sensitivity-high-dimension

- BCS, $\mathrm{SP}_{\mathrm{LOO}}$, and OMP are often among the best solvers. BCS performs especially well for smaller experimental design sizes. In the case of the Ishigami model, it seems to plateau earlier than the other solvers. It also tends to find sparser solutions than the other solvers (not shown in plots), which might explain these observations: sparse solutions are advantageous when only limited data is available, but at the same time they carry the risk of ignoring important terms. For large ED sizes and the two highly compressible models (Ishigami and 100D function), BCS has a larger spread than the other solvers, possibly because the sparsity-enforcing procedure does not always include all of the important terms. In contrast, the greedy solver OMP returns rather dense solutions, which seem to generalize well. $\mathrm{SP}_{\mathrm{LOO}}$ performs well in general for low-dimensional models.
- SP does not perform well for small ED sizes, but for large ED sizes it sometimes outperforms the other solvers. Together with BCS, it tends to find sparser solutions than the other solvers (not shown in plots).
- LARS and SPGL1 often achieve a similar generalization error, which is often larger, sometimes significantly, than that of the other solvers. SPGL1 tends to return rather dense solutions (not shown in plots), which might not generalize as well as other solutions.
- Some models characterized by relatively poor compressibility (e.g. diffusion and frame) show comparable performance among all solvers. This is expected, as in such cases the sparsity assumption is a rather weak proxy for solution quality.
- The exceptions to the general observations outlined above are the damped oscillator and the Morris function, for which the solver performance is reversed, with LARS and SPGL1 among the best solvers, and OMP and $\mathrm{SP}_{\mathrm{LOO}}$ among the worst. In these cases, however, none of the methods achieves satisfactory accuracy within the available computational budget.

To objectively assess the performance of the methods, we now aggregate the results across models. Since all solvers are tested on the same set of experimental designs, we can determine the ranking of solvers for each experimental design (50 replications $\times$ $6-7$ ED sizes for the four spotlight models, and 30 replications $\times$ $5-7$ ED sizes for each of the seven additional models, resulting in 2620 PCEs) and count how often each solver achieved each rank. This is displayed in Figure 3.3 in the form of stacked bar plots, where the counts are given as percentages. The counts have been normalized by the number of replications and ED sizes used for each model, so that each of the models contributes equally to the final percentages.

This ranking alone, however, does not provide a complete picture; e.g., a solver ranked last can be off by orders of magnitude or barely worse than the best-performing one. Therefore, we added an additional set of triangle markers detailing for how many of the EDs the respective solver returned a result that was within two, five, or 10 times the smallest relative MSE attained by any of the six solvers on the same ED. For example, the red triangle in the top row of Figure 3.3a indicates that in ca. 30% of the runs, the solution returned by LARS had a validation error that was at most twice as large as that of the best solution on the same ED.

We have grouped the analysis results separately for the 6 low-dimensional ($d \leq 10$) and the 5 high-dimensional ($d \geq 20$) models, because we observed that dimension had a significant impact on the rankings, and this information is readily available even for black-box models. We also




*(a) Ishigami function* *(b) Borehole model*

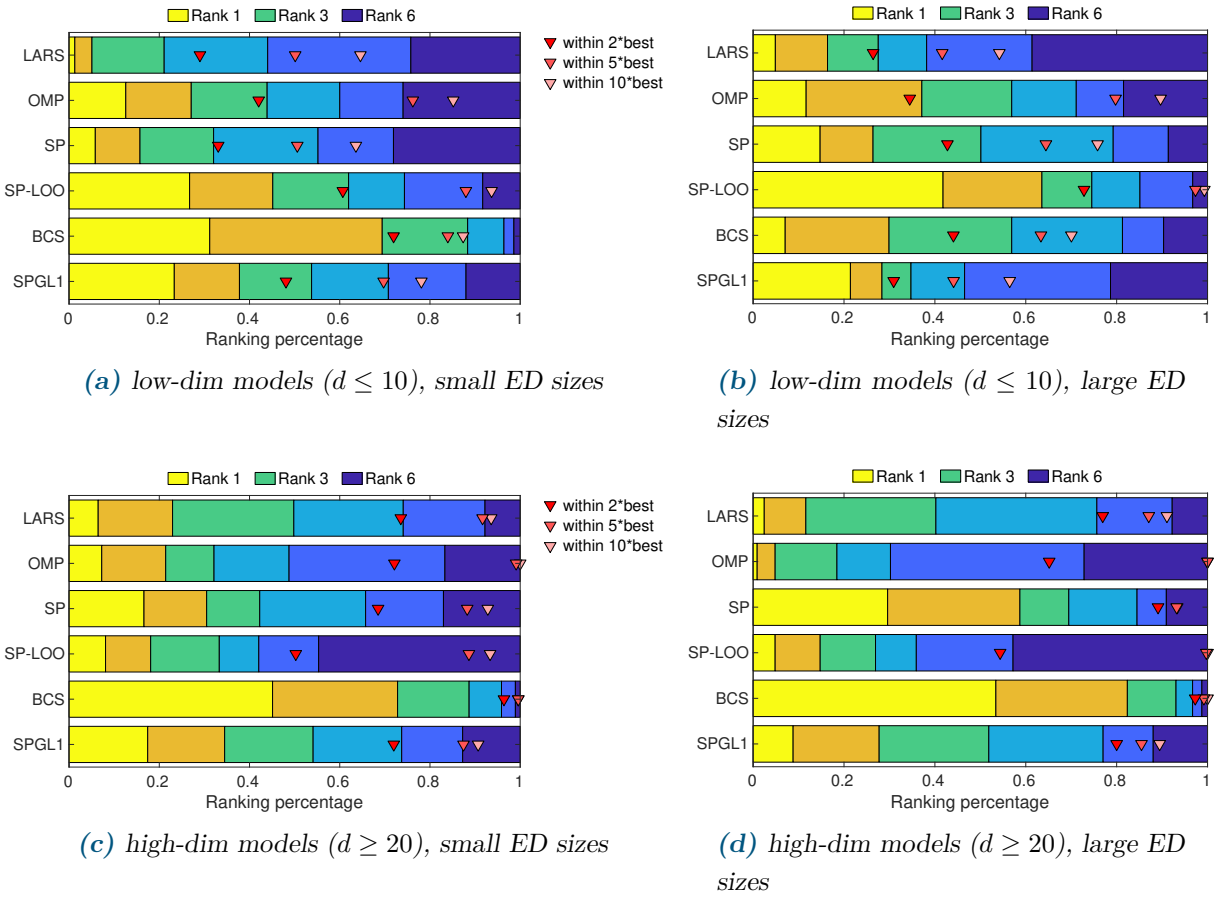
*(c) Two-dimensional diffusion* *(d) 100D function*

***Figure 3.2:*** *Boxplots of relative MSE against ED size from the solver benchmark for the four spotlight models. Results for six sparse solvers and LHS design. For the corresponding plots for the seven remaining models, see Figure 3.C.1 in Appendix 3.C.*

analyze small and large ED sizes separately, where the first half of considered ED sizes (3–4) are regarded small, and the second half large.

We make the following observations:

- *Low-dimensional models, small ED sizes*: BCS is the best solver most often (31% of runs) and also most often within two times the smallest error (72% of runs). However, it comes within one order of magnitude of the smallest relative MSE (a property which we here call *robustness*) in only 87% of runs, while $\mathrm{SP}_{\mathrm{LOO}}$ achieves this in 94% of runs. In the two former metrics, $\mathrm{SP}_{\mathrm{LOO}}$ comes second. LARS and SP perform worst, followed by OMP and SPGL1. Here and in the following, we observe that OMP and $\mathrm{SP}_{\mathrm{LOO}}$ often result in quite robust solutions.
- *Low-dimensional models, large ED sizes*: $\mathrm{SP}_{\mathrm{LOO}}$ outperforms the other solvers in all respects. It provides the smallest relative MSE of all solvers in 42% of runs, is in 73% of runs within two times of the smallest relative MSE, and is even in 99% of runs within one order of magnitude of the smallest relative MSE. For the other solvers, the statistics confirm the observations highlighted by the spotlight models: LARS and SPGL1 overall do not perform well. SP and BCS come close to the best solution quite often, but are less robust, whereas OMP is robust but often not as close to the best solution.
- *High-dimensional models*: For the small as well as the large ED sizes, we see that BCS




*(a)* *low-dim models ($d \leq 10$), small ED sizes*

*(b)* *low-dim models ($d \leq 10$), large ED sizes*

*(c)* *high-dim models ($d \geq 20$), small ED sizes*

*(d)* *high-dim models ($d \geq 20$), large ED sizes*

***Figure 3.3:*** *Aggregated results for the solver benchmark (Section 3.3.4), separately for low-dimensional (a),(b) and high-dimensional models (c),(d). Left: small ED sizes. Right: large ED sizes. For each model and experimental design, the ranking of the six solvers is determined. The stacked bar plots visualize how often each solver achieved the respective rank. The triangle markers in hues of red additionally demonstrate in how many runs the obtained relative MSE was within a factor of $\{2, 5, 10\}$ of the smallest relative MSE achieved on this experimental design.*

performs exceptionally well. It is the best solver in 45% (53%) of runs and comes within two times the smallest relative MSE in even 96% (97%) of runs. Both BCS and OMP attain in *all* cases a relative MSE within 10 times the smallest relative MSE. SP performs better for large rather than small ED sizes. While $\text{SP}_{\text{LOO}}$ performed best for the low-dimensional models, here it shows poor performance. Note that all solvers come within 10 times the smallest relative MSE in more than 89% of all runs, showing that the choice of solver has a smaller impact for high-dimensional than for low-dimensional models.

### 3.3.5 Results: Comparison of sampling schemes together with solvers

We pair the five solvers LARS, OMP, SP, $\text{SP}_{\text{LOO}}$, and BCS[9] with the sampling schemes MC, LHS, coherence-optimal, and D-optimal based on a coherence-optimal candidate set. We use the abbreviations coh-opt and D-opt(coh-opt) for the latter two. Since $\text{SP}_{\text{LOO}}$ performed poorly

[9]Since SPGL1 did not perform well in the previous section, and is quite slow, we do not include it further in this benchmark.

for high-dimensional models in the solver benchmark of the previous section, we do not consider it here for the high-dimensional models.

We run the benchmark for the low-dimensional models Ishigami, undamped oscillator, borehole, damped oscillator, and wingweight function, and for the high-dimensional models Morris function, two-dimensional diffusion, one-dimensional diffusion, and 100D function. The truss model has Gumbel input, for which we (as of now) cannot construct a coherence-optimal sample. The same holds for the structural frame model with its dependent input.

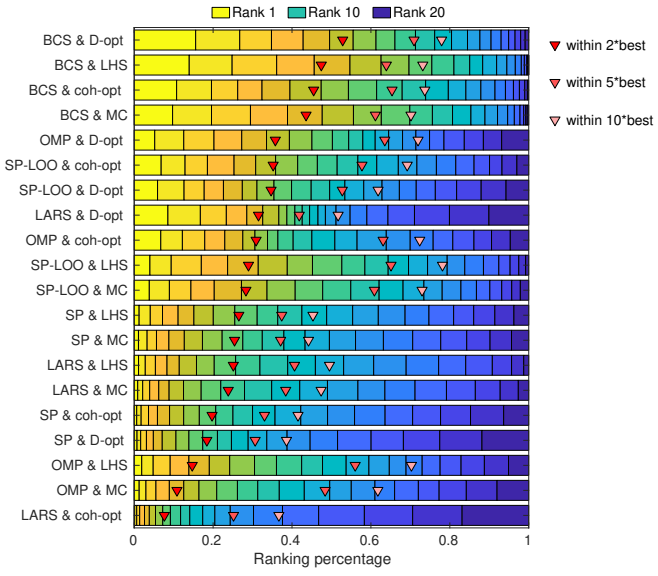
Detailed boxplots of the relative MSE against ED size for the spotlight models, showing how each solver performed when paired with the sampling schemes, can be found in Appendix 3.C, Figures 3.C.2–3.C.5. For the sake of readability, in this section we only show results that are aggregated over models, separately for the low- and high-dimensional cases. For every model and repetition index, we determine the relative ranking of the 20 (16) combinations (5 (4) solvers $\times$ 4 sampling strategies). We also determine which of the combinations came within a factor of $\{2, 5, 10\}$ of the smallest relative MSE among this set (robustness). Then, we count how often each combination achieved each rank, and how often each combination achieved a relative MSE within a factor of the smallest relative MSE. The results are displayed in Figure 3.4 in the form of stacked bar plots for the ranks, with triangle markers denoting the percentage of robust runs. The combinations are sorted by the percentage of runs in which they achieved a relative MSE within two times the smallest relative MSE, because we find that this metric is a good compromise between performance and robustness. We analyze small and large ED sizes separately, where the first half of considered ED sizes (3–4) are regarded as small and the second half as large.

Note that, as opposed to the aggregated results in Figure 3.3, where the solvers are compared on the same experimental designs, here the comparison is done on *different* experimental designs, which are matched randomly.[10] Our results in Figure 3.4 are bootstrapped four times using random permutations of the replication index, corresponding in total to 250 replications, to minimize the influence of this randomness (which is in any case not large, as can be seen from permutation tests).

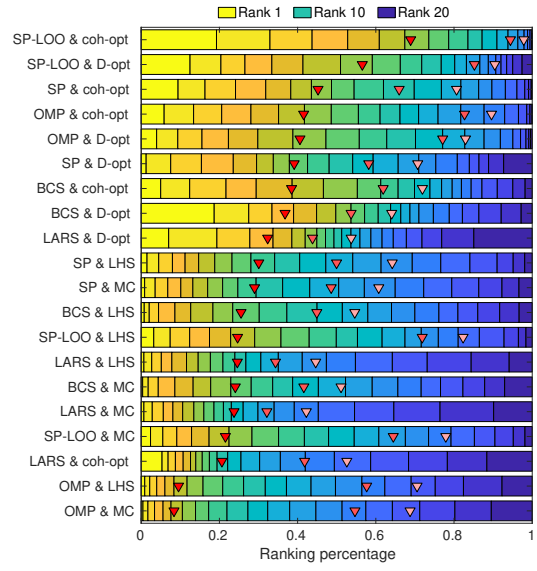
From Figure 3.4 and Figures 3.C.2–3.C.5, we make the following observations:

- There can be considerable differences in the performance of different combinations of solvers and sampling schemes. The differences are larger for low-dimensional models, visible in the spread of triangle markers in Figures 3.4a and 3.4b. For high-dimensional models, many combinations find an error that is close to the smallest error, which can be seen from the red triangle markers at high percentages in Figures 3.4c and 3.4d, and from the clustered boxplots in Figures 3.C.4 and 3.C.5, (e) and (f).
- MC and LHS perform comparably, and for the high-dimensional models almost identically.

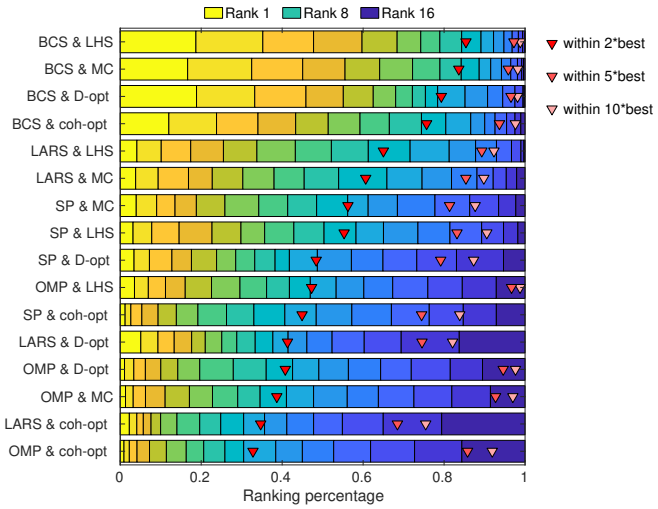
[10] The relative MSE of each combination is interpreted as a random variable $E_{\text{solver}}^{\text{sampling}}$, where the randomness is induced by the randomness in the experimental design. E.g., $E_{\text{BCS}}^{\text{LHS}}$ is the random variable of relative MSE attained by BCS applied to an LHS design of specified size. The reference error ("smallest relative MSE") $E^*$ is a random variable as well, defined as the minimum over *one realization* of each combination of methods: $E^* = \min_{s \in \text{sampling}, t \in \text{solvers}} E_s^t$. The plots in Figure 3.4 are therefore read as follows: e.g., in the low-dimensional, small ED size case (Figure 3.4a): $\mathbb{P}\left(E_{\text{BCS}}^{\text{D-opt}} = E^*\right) = 0.16, \quad \mathbb{P}\left(E_{\text{BCS}}^{\text{D-opt}} \leq 10E^*\right) = 0.78$.



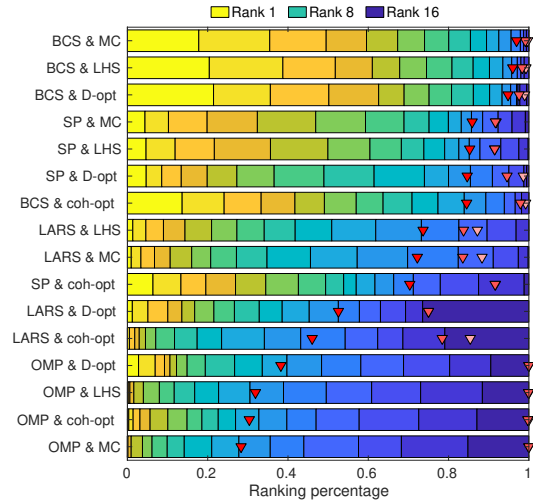

*(a)* $d \leq 10$, *small ED sizes*




*(b)* $d \leq 10$, *large ED sizes*




*(c)* $d \geq 20$, *small ED sizes*




*(d)* $d \geq 20$, *large ED sizes*

***Figure 3.4:*** *Aggregated results for the five low-dimensional models Ishigami, undamped oscillator, borehole, damped oscillator, and wingweight (top), and for the four high-dimensional models Morris function, structural frame, two-dimensional diffusion, and 100D function (bottom). Separately for small (a),(c) and large (b),(d) experimental designs. For the low-dimensional (high-dimensional) case, we investigate five (four) solvers and four sampling schemes, resulting in 20 (16) combinations. For each model and repetition, the ranking of all the combinations is determined (note that as opposed to Figure 3.3, here the comparison is done on different EDs, which are matched randomly. Results are bootstrapped four times by random permutations to increase robustness). The stacked bar plots visualize how often each combination achieved the respective rank. The triangle markers in hues of red additionally demonstrate in how many runs the obtained relative MSE was within a factor of* $\{2, 5, 10\}$ *of the smallest relative MSE achieved in this comparison. The combinations are sorted by the percentage of runs in which they achieved a relative MSE within two times the smallest relative MSE of the respective random pairing (red triangle marker). Plots of relative MSE against ED size for the spotlight models can be found in Appendix 3.C, Figures 3.C.2–3.C.5.*

For the low-dimensional models, LHS sampling has, in most cases, median error and variability that are the same as or smaller than MC. This is consistent with the literature (Shields and Zhang, 2016). These observations are confirmed by the plots in Figure 3.4: for almost every solver, the combination with LHS is slightly better than the corresponding one with MC in each of the metrics.

- The advanced sampling schemes coh-opt and D-opt(coh-opt) show a clear advantage over MC and LHS sampling for low-dimensional models and large experimental designs (consistent with theoretical considerations and numerical experiments (Hampton and Doostan, 2015b)). For low-dimensional models and small experimental designs, they show mixed performance; for high-dimensional models, they perform the same as or worse than LHS and MC.

  It is known that coh-opt sampling leads to a greater improvement over MC sampling for low-dimensional, high-degree expansions than for high-dimensional, low-degree expansions (Hampton and Doostan, 2015b; Alemazkoor and Meidani, 2018a). Note also that all numerical experiments in the literature testing coh-opt sampling were performed in $d \leq 30$ dimensions, using only models with uniform input, or manufactured sparse PCE, i.e., polynomial models with an exactly sparse representation (Hampton and Doostan, 2015b,a; Alemazkoor and Meidani, 2018a; Diaz et al., 2018).

  Both coh-opt and D-opt are sampling methods that aim to improve properties of the regression matrix. They are adapted to the candidate basis. If the candidate basis is large and contains many regressors that are not needed in the final sparse expansion, this adaptation might even deteriorate the solution.
- BCS is one of the best-performing solvers, almost regardless of sampling scheme. The exceptions are low-dimensional models with large experimental designs, where $\mathrm{SP}_{\mathrm{LOO}}$ with coh-opt sampling outperforms all other solvers. This might be related to BCS plateauing earlier than other solvers (see Figure 3.2a and 3.2b). It seems BCS is preferable whenever the information content is low (small ED sizes or high-dimensional models).
- OMP and $\mathrm{SP}_{\mathrm{LOO}}$ are generally quite robust (within one order of magnitude of the best solution). However, OMP often does not come close to the best solution, especially when paired with LHS or MC. BCS is more robust for high-dimensional models than for low-dimensional models. LARS and SP show mixed performance, with LARS being one of the least robust solvers.
- Aggregating the results for each sampling scheme separately (not shown), we observe that the behavior of the solvers is very similar in terms of ranking and robustness to the behavior observed on LHS (Figure 3.3), suggesting that the ranking of solvers is mostly independent of the sampling scheme.

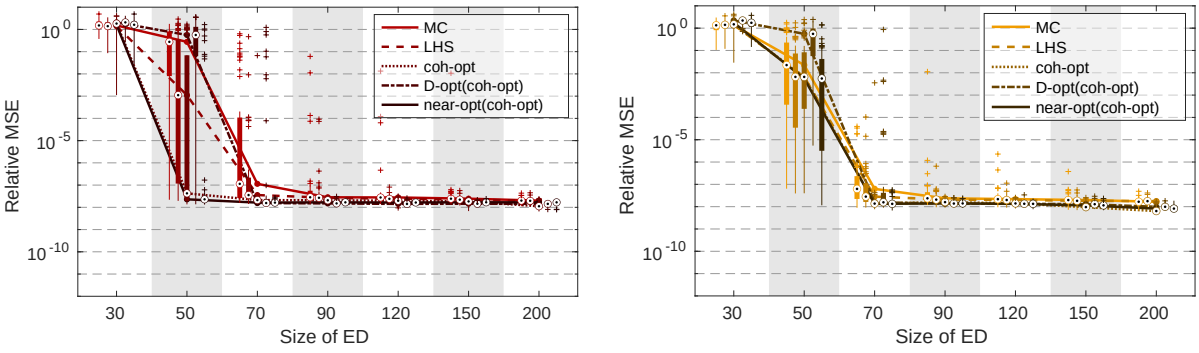
Note that the results in Figures 3.C.2, 3.C.3, and 3.C.5 exhibit plateauing for larger sample sizes. This indicates that the maximal accuracy achievable with this set of basis functions has been reached. Using a larger basis might lead to more accurate solutions, if it contains an important regressor that was previously missing. However, note that a larger basis can also lead to less accurate solutions: when the experimental design size is held fixed while a larger basis is used, the ratio of experimental design points to basis functions is smaller, and the properties of the regression matrix might deteriorate.

### 3.3.6 Results: Comparison of sampling schemes together with solvers, using a smaller candidate basis

We repeat the experiments from the previous section for the Ishigami and borehole models, using a smaller candidate basis for which near-optimal sampling is feasible. The tested solvers are LARS, OMP, SP, $SP_{LOO}$, and BCS. We use the sampling schemes MC, LHS, coh-opt, D-opt(coh-opt) and near-opt(coh-opt). Boxplots of relative MSE against ED size are shown in Figures 3.5 and 3.6. For the sake of conciseness, we only show the combinations involving OMP and $SP_{LOO}$. The remaining plots are provided in Appendix 3.C, Figure 3.C.6.

We observe the following:

- Since the basis is smaller, the relative MSE reaches a plateau already for smaller experimental design sizes.
- Most qualitative observations regarding solver and sampling performance are the same as in the previous section, where a larger basis was used.
- Near-optimal sampling often achieves the same or a slightly smaller error than coh-opt sampling, which is consistent with (Alemazkoor and Meidani, 2018a). In many cases, near-optimal sampling achieves the smallest median error. For the Ishigami model, near-optimal sampling additionally exhibits small variability, while for the borehole model, it has a rather large spread, i.e., several outliers.



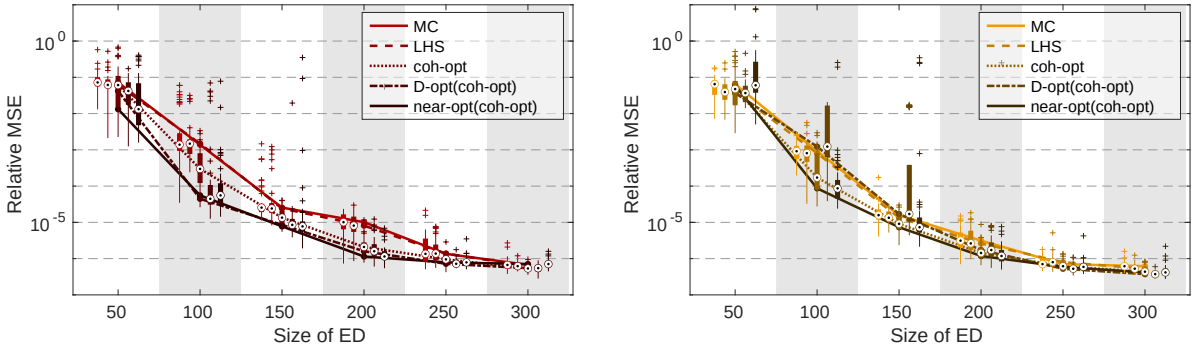

*(a) OMP* *(b) $SP_{LOO}$*

***Figure 3.5:*** *Results for the Ishigami model with a smaller basis ($d = 3, p = 12, q = 1$). Results for two sparse solvers and five experimental design schemes. 50 replications. For the remaining plots, see Figure 3.C.6 in Appendix 3.C.*

## 3.4 Discussion and conclusions

In this paper, we investigated sparse PCE methods with the goal of computing accurate surrogate models based on a few model evaluations.

We presented a literature survey and a framework describing the general computation procedure for sparse PCE. We have seen that the existing literature on sparse PCE can be fit into this framework and that methods developed for different components of the framework can naturally be combined.




*(a) OMP* *(b) $SP_{LOO}$*

***Figure 3.6:*** *Results for the borehole model with a smaller basis ($d = 8, p = 4, q = 1$). Results for two sparse solvers and five experimental design schemes. 50 replications. For the remaining plots, see Figure 3.C.6 in Appendix 3.C.*

In order to give recommendations to practitioners who want to use sparse PCE surrogates for their applications, we performed a numerical benchmark based on 11 example functions which are intended to resemble real-world engineering problems presenting different challenges. We tested several popular sparse solvers and sampling schemes on a fixed set of basis functions, using a range of experimental design sizes and 30–50 replications, and made the following observations:

- The choice of sampling scheme and sparse regression solver can make a difference of up to several orders of magnitude for the relative MSE. Mostly, the rankings of solvers and sampling schemes seem to be independent of one another: an experimental design that works best for one solver will also perform well with other solvers, and the ranking of solvers looks similar independent of which sampling scheme is used. Both solvers and sampling schemes make a greater difference for low-dimensional models.
- For low-dimensional models and small ED sizes, the solver BCS performs best, regardless of sampling scheme (with D-opt(coh-opt) being slightly preferable), while the solver $\mathrm{SP}_{\mathrm{LOO}}$ (a variant of SP) appears to be especially robust.
- For low-dimensional models and large ED sizes, $\mathrm{SP}_{\mathrm{LOO}}$ together with coherence-optimal sampling outperforms all other combinations.
- For low-dimensional models, and when the basis is small enough to make it feasible, near-optimal sampling outperforms all other sampling schemes, regardless of the solver.
- For high-dimensional models, BCS is by far the best solver. All solvers perform better when paired with LHS; in other words, no advanced sampling scheme appears competitive compared to LHS for such problems, whatever the solver used.

The benchmark results demonstrate that in costly, real-world applications it is worth choosing the sparse PCE training strategy carefully, since the methods can make a substantial difference in the quality of the resulting surrogate. While a more accurate surrogate model is generally desirable, in industrial applications it might have a higher impact for purposes such as optimization, rather than, e.g., sensitivity analysis.

Our conclusions are based on a number of benchmark models, which we consider representative of engineering models in terms of dimensionality and complexity. Naturally, however, no selection

of models can cover the whole space of engineering models. Further work would be required to understand the connection among model properties, basis choice, experimental design size, and sparse PCE techniques like solvers and sampling schemes.

All results were obtained using a fixed basis based on a heuristic choice (see Section 3.3.3). Generally, when the optimal degree of the basis is unknown, degree adaptivity (based on a cross-validation error) can be a useful strategy. Due to time and space constraints, this was not investigated in the present work. Adaptivity critically depends on the availability of an accurate error estimator. Some of the best solvers in this study (i.e., OMP and BCS) tend to underestimate the generalization error (not shown in the plots), which might be a drawback in the setting of adaptive degree selection and might change the effect and ranking of solvers and sampling methods. For a detailed discussion and benchmark of basis-adaptive schemes, we refer the reader to Lüthen et al. (2022a).

As evident from the extensive literature on the topic, sparse PCE is an already well-established technique, as well as an active field of methodological research. Recent innovations include Bayesian techniques for sparse PCE and the identification of suitable rotated coordinates for the expansion. Such innovative ideas are expected to lead to further improvements in the computation of sparse PCE, which will in turn benefit applications as well as all advanced methods that use sparse PCE as one of their building blocks (see, e.g., Schöbi et al. (2015); Chatterjee et al. (2019); Zhu and Sudret (2020); Marelli et al. (2021b)).

PCE is a popular metamodelling tool in the engineering community, and many different methods are available. Up to now, the choice of which of the many PCE methods to apply was mostly left to chance or the personal experience of the practitioner. In our benchmark, we explored a significant set of methods that have received attention in the past few years. We hope that this work can serve as a basis for further benchmarking efforts, in order to identify which of the many available methods are most suitable for real-world problems. These might include sequential enrichment of experimental design, Gaussian adaptation of the input space, stepwise regression algorithms and many other ideas for sparse solvers, as well as methods for extremely high-dimensional problems.

Our benchmark code is available on request. The solvers BCS and $\mathrm{SP}_{\mathrm{LOO}}$ will be made available in the 1.4.0 release of UQLab. For a description of how to add custom sampling schemes and sparse solvers for PCE to UQLab, we refer the reader to the supplementary material accompanying this paper. To facilitate easier benchmarking of PCE techniques on a large number of examples in a standardized setup, we are actively engaging in designing and developing a benchmarking platform for surrogate modelling methods similar to the UCI machine learning repository[11] or the structural reliability platform RPrepo[12] where data sets, models, and methods can be made available for testing and benchmarking.

## Acknowledgments

We thank Dr. Emiliano Torre for helpful discussions.

[11] https://archive.ics.uci.edu

[12] https://rprepo.readthedocs.io/en/latest/reliability_problems.html

## References

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. *J. Comput. Phys.*, 332:461 – 474.

Alemazkoor, N. and Meidani, H. (2017). Divide and conquer: An incremental sparsity promoting compressive sampling approach for polynomial chaos expansions. *Comput. Methods Appl. Mech. Engrg.*, 318:937–956.

Alemazkoor, N. and Meidani, H. (2018a). A near-optimal sampling strategy for sparse recovery of polynomial chaos expansions. *J. Comput. Phys.*, 371:137–151.

Alemazkoor, N. and Meidani, H. (2018b). A preconditioning approach for improved estimation of sparse polynomial chaos expansions. *Comput. Methods Appl. Mech. Engrg.*, 342:474–489.

Arjoune, Y., Kaabouch, N., El Ghazi, H., and Tamtaoui, A. (2017). Compressive sensing: Performance comparison of sparse recovery algorithms. In *2017 IEEE CCWC*, pages 1–7. IEEE.

Babacan, D. (2011). MATLAB code for the TIP 2010 paper "Bayesian Compressive Sensing using Laplace Priors". http://www.dbabacan.info/software.html. [Online; accessed 28-August-2019].

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2010). Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.*, 19(1):53–63.

Baptista, R., Stolbunov, V., and Nair, P. B. (2019). Some greedy algorithms for sparse polynomial chaos expansions. *J. Comput. Phys.*, 387:303–325.

Baumann, D. and Baumann, K. (2014). Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation. *J. Cheminf.*, 6(1):47.

Berchier, M. (2015). Orthogonal matching pursuit for sparse polynomial chaos expansions. ETH Zürich. Semester project.

Berveiller, M. (2005). *Stochastic finite elements : intrusive and non intrusive methods for reliability analysis*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *Eur. J. Comput. Mech.*, 15(1–3):81–92.

Bhattacharyya, B. (2020). Global sensitivity analysis: A Bayesian learning based polynomial chaos approach. *J. Comput. Phys.*, 415:109539.

Birgin, E. G., Martínez, J. M., and Raydan, M. (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211.

Blatman, G. and Sudret, B. (2008). Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *C. R. Mécanique*, 336(6):518–523.

Blatman, G. and Sudret, B. (2010a). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Prob. Eng. Mech.*, 25:183–197.

Blatman, G. and Sudret, B. (2010b). Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 95:1216–1229.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys*, 230:2345–2367.

Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81.

Candès, E. and Romberg, J. (2005). $\ell$1-MAGIC. https://statweb.stanford.edu/~candes/software/l1magic/. [Online; accessed 22-January-2020].

Candès, E., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509.

Candès, E. J. and Plan, Y. (2011). A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Inform. Theory*, 57(11):7235–7254.

Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Process. Mag.*, 25(2):21–30.

Candès, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted $\ell_1$ minimization. *J. Fourier Anal. Appl.*, 14(5-6):877–905.

Carron, I. (2013). Compressive Sensing: The Big Picture. https://sites.google.com/site/igorcarron2/cs. [Online; accessed 22-January-2020].

Chapelle, O., Vapnik, V., and Bengio, Y. (2002). Model selection for small sample regression. *Machine Learning*, 48(1):9–23.

Chatterjee, T., Chakraborty, S., and Chowdhury, R. (2019). A critical review of surrogate assisted robust design optimization. *Arch. Comput. Method. E.*, 26(1):245–274.

Cheng, K. and Lu, Z. (2018a). Adaptive sparse polynomial chaos expansions for global sensitivity analysis based on support vector regression. *Comput. Struct.*, 194:86–96.

Cheng, K. and Lu, Z. (2018b). Sparse polynomial chaos expansion based on D-MORPH regression. *Appl. Math. Comput.*, 323:17–30.

Cohen, A. and Migliorati, G. (2017). Optimal weighted least-squares methods. *SMAI J. Comp. Math.*, 3:181–203.

Cook, R. D. and Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22(3):315–324.

Dai, W. and Milenkovic, O. (2009). Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inform. Theory*, 55(5):2230–2249.

Diaz, P. (2018). DOPT_PCE. https://github.com/CU-UQ/DOPT_PCE. [Online; accessed 22-January-2020].

Diaz, P., Doostan, A., and Hampton, J. (2018). Sparse polynomial chaos expansions via compressed sensing and D-optimal design. *Comput. Methods Appl. Mech. Engrg.*, 336:640–666.

Donoho, D., Drori, I., Stodden, V., Tsaig, Y., and Shahram, M. (2007). SparseLab - Seeking Sparse Solutions to Linear Systems of Equations. http://sparselab.stanford.edu/. [Online; accessed 22-January-2020].

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306.

Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.*, 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.

Dutta, S. and Gandomi, A. H. (2020). Design of experiments for uncertainty quantification based on polynomial chaos expansion metamodels. In *Handbook of Probabilistic Models*, pages 369–381. Elsevier.

Dykstra, O. (1971). The augmentation of experimental data to maximize $[X'X]$. *Technometrics*, 13(3):682–688.

Echard, B., Gayton, N., Lemaire, M., and Relun, N. (2013). A combined importance sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliab. Eng. Syst. Saf.*, 111:232–240.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Stat.*, 32:407–499.

Ernst, O., Mugler, A., Starkloff, H.-J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.*, 46(02):317–339.

Fajraoui, N., Marelli, S., and Sudret, B. (2017). Sequential design of experiment for sparse polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):1061–1085.

Faul, A. C. and Tipping, M. E. (2002). Analysis of sparse Bayesian learning. In *Advances in neural information processing systems*, pages 383–389.

Fedorov, V. V. (2013). *Theory of optimal experiments*. Elsevier.

Figueiredo, M. A. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1150–1159.

Figueiredo, M. A. and Nowak, R. D. (2001). Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior. *IEEE Trans. Image Process.*, 10(9):1322–1331.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. Wiley.

Ghanem, R. G. and Spanos, P. (1991). *Stochastic finite elements – A spectral approach*. Springer Verlag, New York. (Reedited by Dover Publications, Mineola, 2003).

Gu, M. and Eisenstat, S. C. (1996). Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869.

Guo, L., Narayan, A., Liu, Y., and Zhou, T. (2020). Sparse approximation of data-driven polynomial chaos expansions: An induced sampling approach. *Commun. Math. Res.*, 36(2):128–153.

Guo, L., Narayan, A., Zhou, T., and Chen, Y. (2017). Stochastic collocation methods via $\ell_1$ minimization using randomized quadratures. *SIAM J. Sci. Comput.*, 39(1):A333–A359.

Hadigol, M. and Doostan, A. (2018). Least squares polynomial chaos expansion: A review of sampling strategies. *Comput. Methods Appl. Mech. Engrg.*, 332:382–407.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2(1):84–90.

Hampton, J. and Doostan, A. (2015a). Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. *Comput. Methods Appl. Mech. Engrg.*, 290:73–97.

Hampton, J. and Doostan, A. (2015b). Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. *J. Comput. Phys.*, 280:363–386.

Hampton, J. and Doostan, A. (2017). COH-OPT. https://github.com/CU-UQ/COH-OPT. [Online; accessed 22-January-2020].

Hampton, J. and Doostan, A. (2018). Basis adaptive sample efficient polynomial chaos (BASE-PC). *J. Comput. Phys.*, 371:20–49.

Harper, W. V. and Gupta, S. K. (1983). Sensitivity/uncertainty analysis of a borehole scenario comparing Latin hypercube sampling and deterministic sensitivity approaches. Technical Report No. BMI/ONWI-516, Battelle Memorial Inst. - Office of Nuclear Waste Isolation, Columbus, OH (USA).

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer, New York.

Hong, Y. P. and Pan, C.-T. (1992). Rank-revealing QR factorizations and the singular value decomposition. *Math. Comput.*, 58(197):213–232.

Hosder, S., Walters, R., and Balch, M. (2007). Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In *48th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, page 1939.

Hu, C. and Youn, B. D. (2011). Adaptive-sparse polynomial chaos expansion for reliability analysis and design of complex engineering systems. *Struct. Multidisc. Optim.*, 43:419–442.

Hu, R. and Ludkovski, M. (2017). Sequential design for ranking response surfaces. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):212–239.

Huan, X., Safta, C., Sargsyan, K., Vane, Z. P., Lacaze, G., Oefelein, J. C., and Najm, H. N. (2018). Compressive sensing with cross-validation and stop-sampling for sparse polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.*, 6(2):907–936.

Isukapalli, S. S. (1999). *Uncertainty analysis of transport-transformation models*. PhD thesis, The State University of New Jersey.

Jakeman, J. D., Eldred, M. S., and Sargsyan, K. (2015). Enhancing $\ell_1$-minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.*, 289:18–34.

Jakeman, J. D., Narayan, A., and Zhou, T. (2017). A generalized sampling and preconditioning scheme for sparse approximation of polynomial chaos expansions. *SIAM J. Sci. Comput.*, 39(3):A1114–A1144.

Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Trans. Signal Process.*, 56(6):2346–2356.

Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *Ann. Math. Stat.*, 30(2):271–294.

Konakli, K. and Sudret, B. (2016). Global sensitivity analysis using low-rank tensor approximations. *Reliab. Eng. Sys. Safety*, 156:64–83.

Kougioumtzoglou, I. A., Petromichelakis, I., and Psaros, A. F. (2020). Sparse representations and compressive sampling approaches in engineering mechanics: A review of theoretical concepts and diverse applications. *Prob. Eng. Mech.*, 61:103082.

Lataniotis, C. (2019). *Data-driven uncertainty quantification for high-dimensional engineering problems*. PhD thesis, ETH Zurich.

Li, C. C. and Der Kiureghian, A. (1993). Optimal discretization of random fields. *J. Eng. Mech.*, 119(6):1136–1154.

Li, G. and Rabitz, H. (2010). D-MORPH regression: application to modeling with unknown parameters more than observation data. *J. Math. Chem.*, 48(4):1010–1035.

Liu, Z., Lesselier, D., Sudret, B., and Wiart, J. (2020a). Surrogate modeling based on resampled polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 202:107008.

Liu, Z., Lesselier, D., Sudret, B., and Wiart, J. (2020b). Surrogate modeling of indoor down-link human exposure based on sparse polynomial chaos expansion. *Int. J. Uncertainty Quantification*, 10(2):145–163.

Lüthen, N., Marelli, S., and Sudret, B. (2022). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification*, 12(3):49–74.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. 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.

Marelli, S., Wagner, P.-R., Lataniotis, C., and Sudret, B. (2021). Stochastic spectral embedding. *Int. J. Uncertainty Quantification*, 11(2):25–47.

Mathelin, L. and Gallivan, K. (2012). A compressed sensing approach for partial differential equations with random input data. *Commun. Comput. Phys.*, 12(4):919–954.

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*, 2:239–245.

Mikhalev, A. and Oseledets, I. V. (2018). Rectangular maximum-volume submatrices and their applications. *Linear Algebra Appl.*, 538:187–211.

Montgomery, D. C. (2004). *Design and analysis of experiments*. John Wiley and Sons, New York.

Narayan, A., Jakeman, J., and Zhou, T. (2017). A Christoffel function weighted least squares algorithm for collocation approximations. *Math. Comput.*, 86(306):1913–1947.

Needell, D. and Tropp, J. A. (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. A.*, 26(3):301–321.

Nguyen, N.-K. and Miller, A. J. (1992). A review of some exchange algorithms for constructing discrete D-optimal designs. *Comput. Stat. Data An.*, 14(4):489–498.

Owen, A. B. (1994). Controlling correlations in Latin hypercube samples. *J. Am. Stat. Assoc.*, 89(428):1517–1522.

Papaioannou, I., Ehre, M., and Straub, D. (2019). PLS-based adaptation for efficient PCE representation in high dimensions. *J. Comput. Phys.*, 387:186–204.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proc. of 27th Asilomar Conf. on Signals, Systems and Computers*, pages 40–44. IEEE.

Peng, J., Hampton, J., and Doostan, A. (2014). A weighted $\ell_1$-minimization approach for sparse polynomial chaos expansions. *J. Comput. Phys.*, 267:92–111.

Perkó, Z., Gilli, L., Lathouwers, D., and Kloosterman, J. L. (2014). Grid and basis adaptive polynomial chaos techniques for sensitivity and uncertainty analysis. *J. Comput. Phys.*, 260:54–84.

Pronzato, L. and Müller, W. G. (2012). Design of computer experiments: space filling and beyond. *Stat. Comput.*, 22(3):681–701.

Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., and Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *J. Commun. Netw.*, 15(5):443–456.

Rauhut, H. and Ward, R. (2012). Sparse Legendre expansions via $\ell_1$-minimization. *J. Approx. Theory*, 164(5):517–533.

Sargsyan, K., Safta, C., Najm, H., Debusschere, B., Ricciuto, D., and Thornton, P. (2014). Dimensionality reduction for complex models via Bayesian compressive sensing. *Int. J. Uncertainty Quantification*, 4(1):63–93.

Schöbi, R., Sudret, B., and Wiart, J. (2015). Polynomial-chaos-based Kriging. *Int. J. Uncertainty Quantification*, 5(2):171–193.

Seeger, M. W. and Nickisch, H. (2008). Compressed sensing and Bayesian experimental design. In *Proc. of the 25th Int. Conf. on Machine Learning*, pages 912–919. ACM.

Seshadri, P., Narayan, A., and Mahadevan, S. (2017). Effectively subsampled quadratures for least squares polynomial approximations. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):1003–1023.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Shao, Q., Younes, A., Fahs, M., and Mara, T. (2017). Bayesian sparse polynomial chaos expansion for global sensitivity analysis. *Comput. Meth. Appl. Mech. Eng.*, 318:474–496.

Shen, D., Wu, H., Xia, B., and Gan, D. (2020). Polynomial chaos expansion for parametric problems in engineering systems: A review. *IEEE Syst. J.*

Shields, M. D. and Zhang, J. (2016). The generalization of Latin hypercube sampling. *Reliab. Eng. Syst. Saf.*, 148:96–108.

Shin, Y. and Xiu, D. (2016a). Nonadaptive quasi-optimal points selection for least squares linear regression. *SIAM J. Sci. Comput.*, 38(1):A385–A411.

Shin, Y. and Xiu, D. (2016b). On a near optimal sampling strategy for least squares polynomial regression. *J. Comput. Phys.*, 326:931–946.

Sobol', I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Comput. Maths. Math. Phys.*, 7:86–112.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Saf.*, 93:964–979.

Tang, G. and Iaccarino, G. (2014). Subsampled Gauss quadrature nodes for estimating polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.*, 2(1):423–443.

Tarakanov, A. and Elsheikh, A. H. (2019). Regression-based sparse polynomial chaos for uncertainty quantification of subsurface flow models. *J. Comput. Phys.*, 399:108909.

Tipireddy, R. and Ghanem, R. (2014). Basis adaptation in homogeneous chaos spaces. *J. Comput. Phys.*, 259:304–317.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1(Jun):211–244.

Tipping, M. E. and Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics*.

Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53(12):4655–4666.

Tsilifis, P., Huan, X., Safta, C., Sargsyan, K., Lacaze, G., Oefelein, J. C., Najm, H. N., and Ghanem, R. G. (2019). Compressive sensing adaptation for polynomial chaos expansions. *J. Comput. Phys.*, 380:29–47.

Tsilifis, P., Papaioannou, I., Straub, D., and Nobile, F. (2020). Sparse Polynomial Chaos expansions using variational relevance vector machines. *J. Comput. Phys.*, 416:109498.

van den Berg, E. and Friedlander, M. P. (2008). Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912.

Van den Berg, E. and Friedlander, M. P. (2015). SPGL1 - A Matlab solver for sparse optimization. https://friedlander.io/spgl1/. [Online; accessed 22-January-2020].

Wipf, D. P., Palmer, J., and Rao, B. D. (2004). Perspectives on sparse Bayesian learning. In *Advances in neural information processing systems*, pages 249–256.

Wipf, D. P. and Rao, B. D. (2004). Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.*, 52(8):2153–2164.

Xiu, D. and Hesthaven, J. S. (2005). High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644.

Yan, L., Guo, L., and Xiu, D. (2012). Stochastic collocation algorithms using $\ell_1$-minimization. *Int. J. Uncertainty Quantification*, 2(3).

Yang, X. and Karniadakis, G. E. (2013). Reweighted $\ell_1$ minimization method for stochastic elliptic differential equations. *J. Comput. Phys.*, 248:87–108.

Yang, X., Li, W., and Tartakovsky, A. (2018). Sliced-inverse-regression–aided rotated compressive sensing method for uncertainty quantification. *SIAM/ASA J. Uncertain. Quantif.*, 6(4):1532–1554.

Yin, P., Lou, Y., He, Q., and Xin, J. (2015). Minimization of $\ell_{1-2}$ for compressed sensing. *SIAM J. Sci. Comput.*, 37(1):A536–A563.

Zankin, V. P., Ryzhakov, G. V., and Oseledets, I. V. (2018). Gradient descent-based D-optimal design for the least-squares polynomial approximation. *arXiv preprint arXiv:1806.06631*.

Zein, S., Colson, B., and Glineur, F. (2013). An efficient sampling method for regression-based polynomial chaos expansion. *Commun. Comput. Phys.*, 13(4):1173–1188.

Zhang, Z., Xu, Y., Yang, J., Li, X., and Zhang, D. (2015). A survey of sparse representation: algorithms and applications. *IEEE Access*, 3:490–530.

Zhao, H., Gao, Z., Xu, F., Zhang, Y., and Huang, J. (2019). An efficient adaptive forward–backward selection method for sparse polynomial chaos expansion. *Comput. Methods Appl. Mech. Engrg.*, 355:456–491.

Zhao, W. and Bu, L. (2019). Global sensitivity analysis with a hierarchical sparse metamodeling method. *Mech. Syst. Signal Pr.*, 115:769–781.

Zhou, T., Narayan, A., and Xu, Z. (2014). Multivariate discrete least-squares approximations with a new type of collocation grid. *SIAM J. Sci. Comput.*, 36(5):A2401–A2422.

Zhou, Y., Lu, Z., and Cheng, K. (2019a). Sparse polynomial chaos expansions for global sensitivity analysis with partial least squares and distance correlation. *Struct. Multidiscip. O.*, 59(1):229–247.

Zhou, Y., Lu, Z., Cheng, K., and Ling, C. (2019b). An efficient and robust adaptive sampling method for polynomial chaos expansion in sparse Bayesian learning framework. *Comput. Methods Appl. Mech. Engrg.*, 352:654–674.

Zhou, Y., Lu, Z., Cheng, K., and Shi, Y. (2019c). An expanded sparse Bayesian learning method for polynomial chaos expansion. *Mech. Syst. Signal Pr.*, 128:153–171.

Zhou, Y., Lu, Z., Hu, J., and Hu, Y. (2020). Surrogate modeling of high-dimensional problems via data-driven polynomial chaos expansions and sparse partial least square. *Comput. Methods Appl. Mech. Engrg.*, 364:112906.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *Int. J. Uncertainty Quantification*, 10(3):249–275.

# Appendix

## 3.A Details on experimental design sampling techniques

It depends on certain properties of the regression matrix $\mathbf{\Psi}$ whether or not sparse regression techniques are able to find the true sparse solution of a linear system of equations (assuming that it exists). In the context of polynomial chaos, the entries of the regression matrix are the basis polynomials evaluated at the design points. The basis polynomials are determined by the distribution of the input random variables and the choice of the index set $\mathcal{A}$, while the design points $\{\boldsymbol{x}^{(j)}\}_{j=1}^{N}$ can be chosen freely from the input space to optimize properties of the resulting regression matrix.

In the following, we present sampling schemes that have been proposed in the literature for the computation of sparse PCE. Some of the schemes come with theoretical results about their performance for sparse PCE, others have heuristic justification or have guarantees for least-squares regression. They can be broadly grouped into three categories:

- Sampling according to the input distribution
  - MC (Doostan and Owhadi, 2011; Hampton and Doostan, 2015b)
  - LHS (McKay et al., 1979)
- Sampling from a modified distribution (induced sampling)
  - asymptotic (Hampton and Doostan, 2015b)
  - coherence-optimal (Hampton and Doostan, 2015b)
  - Christoffel sparse approximation (Jakeman et al., 2017)
- Optimizing matrix properties
  - D-optimal (Diaz et al., 2018)
  - S-optimal (Shin and Xiu, 2016a; Fajraoui et al., 2017)
  - near-optimal (Alemazkoor and Meidani, 2018a)

Some of the sampling schemes are nontrivial or costly to evaluate, or even not available for all input distributions. However, the bottleneck in surrogate modelling for practical applications is typically the repeated evaluation of the model, which justifies the use of a complex sampling scheme if it allows better approximation with fewer samples.

### 3.A.1 Sampling according to the input distribution

This class of sampling methods consists of all methods that are oblivious to the choice of truncation set $\mathcal{A}$ and whose main objective is to distribute design points evenly in the quantile space. Heuristically, the more uniformly the points are distributed in the quantile space, the more information about the model is captured in the model evaluations, since no region of the input domain is forgotten.

LHS is one technique for achieving a space-filling design. For each component of the input random vector $\boldsymbol{X}$, the corresponding quantile space is divided into $N$ intervals. In each interval, one point is sampled uniformly at random. Then, the points for each dimension are combined randomly into vectors and finally transformed into the input space using an isoprobabilistic transform. LHS can be shown to reduce the variance of linear regression estimates when the main effects are dominant, i.e., when the most important terms have interaction order one (Shields and Zhang, 2016). LHS can be combined with heuristic criteria such as the maximin distance strategy, where among several random LHS designs the one with the largest minimal pairwise distance between points is chosen, to further improve on the space-filling property.

Stratified sampling is a related sampling technique in which the input space is divided into disjoint regions, called strata, from which points are sampled and weighted according to the probability mass of their stratum. Stratified sampling reduces the variance of statistical estimators (McKay et al., 1979). There exists a range of methods between stratified sampling and LHS, called partially stratified sampling, which are able to reduce the variance of statistical estimators when interaction terms are dominant (Shields and Zhang, 2016). The authors propose an additional method called Latinized partially stratified sampling (LPSS) which combines LHS and stratified sampling with the aim tif minimizing the variance of the resulting estimator. It is especially beneficial when there is prior knowledge about which variable groups interact, and it has been used for several problems with input dimension $d = 100$.

MC sampling, i.e., sampling from the input distribution, is a special case of the coherence-based theory detailed in Section 3.A.2.1 below, and bounds on the coherence and the associated number of points needed for recovery can be derived (Doostan and Owhadi, 2011; Rauhut and Ward, 2012; Yan et al., 2012; Hampton and Doostan, 2015b).

### 3.A.2 Sampling from a different distribution

The ability of sparse regression to recover the true sparse solution (if it exists; otherwise it recovers the best sparse approximation) largely depends on the regression matrix. In the case of PCE, the entries of this matrix are the evaluations of the basis polynomials at the experimental design points. The points can be chosen in a way that improves the recovery properties of the matrix.

Several approaches exist in which the $\ell^1$-minimization problem is modified into a weighted problem and samples are drawn not from the input distribution, but from a suitable modified distribution. The idea of these approaches is as follows. Define a weight function $w(x) : \Omega \to \mathbb{R}$ in a suitable way, which will be explained later. For an ED $\{\boldsymbol{x}^{(k)}\}_{k=1}^{N}$, define the diagonal matrix

$\boldsymbol{W} = \text{diag}(w(\boldsymbol{x}^{(1)}), \ldots, w(\boldsymbol{x}^{(N)}))$. Then the following modified system is solved:

$$\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \text{ s.t. } \|\boldsymbol{W}\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{W}\boldsymbol{y}\|_2 \leq \epsilon. \tag{3.14}$$

Depending on $w(\boldsymbol{x})$, this modification can improve or deteriorate the solution $\boldsymbol{c}$. Of course, the weight function is chosen to improve the solution. The matrix $\boldsymbol{W}\boldsymbol{\Psi}$ can also be interpreted as the evaluation of a modified basis $\{\tilde{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = w(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\}_{\boldsymbol{\alpha}\in\mathcal{A}}$. To ensure orthonormality of the columns of $\boldsymbol{W}\boldsymbol{\Psi}$, the design points are drawn from a suitably modified input distribution $f_{\tilde{\boldsymbol{X}}}$.

### 3.A.2.1 Coherence, isotropy, and weighted orthonormal systems

In this section, we define concepts that are the basis for guarantees on accuracy and stability for different sampling distributions. We mainly follow the exposition in Hampton and Doostan (2015b).

In the setting of PCE, the *coherence* of an orthonormal system $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathcal{A}}$ is defined by

$$\mu(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\}) = \sup_{\boldsymbol{x}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|^2. \tag{3.15}$$

For distributions for which this quantity would be $\infty$, such as a Gaussian distribution, see the remark below.

A second important concept is *isotropy* (Candès and Plan, 2011): a random matrix, whose rows are chosen randomly following some distribution $\boldsymbol{a} \sim F_{\boldsymbol{a}}$, is isotropic if it holds that $\mathbb{E}\left[\boldsymbol{a}^T\boldsymbol{a}\right] = \mathbb{1}$. In the case of PCE, $F_{\boldsymbol{a}}$ is induced by propagating the input distribution $F_{\boldsymbol{X}}$ through the basis functions. By construction, the regression matrix of standard PCE is isotropic if the ED is sampled from the input distribution. Under the assumption that the regression matrix $\boldsymbol{\Psi}$ is isotropic, the number of samples needed for perfect recovery of sparse solutions in the noiseless case is proportional to $\mu(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\})s\log(P)$ with high probability (Candès and Plan, 2011), where $s$ is the sparsity of the solution vector and $P = |\mathcal{A}|$ is the number of basis functions. A similar result holds in the noisy case.

Thus, an orthonormal system $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathcal{A}}$ with low coherence $\mu(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\})$ requires fewer samples for perfect recovery. The goal of coherence-optimal sampling is to find a weighted system $\{\tilde{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = w(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\}_{\boldsymbol{\alpha}\in\mathcal{A}}$ that achieves $\mu(\mathcal{A}, \{\tilde{\psi}_{\boldsymbol{\alpha}}\}) < \mu(\mathcal{A}, \{\psi_{\boldsymbol{\alpha}}\})$ and is orthonormal with respect to some distribution $\tilde{f}_{\boldsymbol{X}}$.

The ideas of isotropy and coherence were applied to PCE by Hampton and Doostan (2015b), who construct an isotropic regression matrix with improved coherence as follows. Let $B : \mathcal{D} \to \mathbb{R}$ be the tight upper bound for the polynomial basis,

$$B(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|. \tag{3.16}$$

Let $G : \mathcal{D} \to \mathbb{R}$ be a loose upper bound with $G(\boldsymbol{x}) \geq B(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{D}$. ($G$ is useful because using a simple expression for the upper bound can in some cases result in $\tilde{f}_{\boldsymbol{X}}$ being a well-known distribution that can be sampled from easily.) Define a new probability distribution $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ by

$$\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x}) = c^2 G(\boldsymbol{x})^2 f_{\boldsymbol{X}}(\boldsymbol{x}), \tag{3.17}$$

where $c = \left(\int_\Omega f_{\boldsymbol{X}}(\boldsymbol{x})G(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x}\right)^{-\frac{1}{2}}$ is the normalizing constant. Then, with the weight function

$$w(\boldsymbol{x}) = \frac{1}{cG(\boldsymbol{x})}, \tag{3.18}$$

the set of functions $\{\tilde{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = w(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\}_{\boldsymbol{\alpha}\in\mathcal{A}}$ is an orthonormal system with respect to the distribution $\tilde{f}_{\boldsymbol{X}}$. This follows directly from the orthonormality of $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathcal{A}}$ with respect to $f_{\boldsymbol{X}}$. Furthermore, if $G = B$, the coherence $\mu(\mathcal{A}, \{\tilde{\psi}_{\boldsymbol{\alpha}}\})$ is minimal.

**Remark** Some polynomial bases (e.g. Hermite polynomials) do not have a finite upper bound. It is still possible to obtain similar results by considering a smaller domain $\mathcal{S} \subset \mathcal{D}$ on which the upper bound is finite and the isotropy is still approximately fulfilled. The modified probability distribution is then $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x}) = c^2 G(\boldsymbol{x})^2 f_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{1}_{\mathcal{S}}(\boldsymbol{x})$.

#### 3.A.2.2 Sampling using a loose upper bound ("asymptotic sampling")

In the case of Legendre and Hermite polynomials, and using a certain loose upper bound $G(\boldsymbol{x}) \geq \max_{\boldsymbol{\alpha}\in\mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|$, analytical expressions for distributions with improved coherence can be obtained (Hampton and Doostan, 2015b).

In the case of Legendre polynomials on $[-1, 1]^d$, a loose upper bound on the polynomials is given by $G(\boldsymbol{x}) \propto \prod_{i=1}^d (1 - x_i^2)^{-\frac{1}{4}}$, which leads to the Chebyshev distribution $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^d \frac{1}{\pi\sqrt{1-x_i^2}}$ and to the weight function $w(\boldsymbol{x}) = \prod_{i=1}^d (1 - x_i^2)^{\frac{1}{4}}$.

In the case of Hermite polynomials for standard Gaussian variables, a loose upper bound on the polynomials is given by $G(\boldsymbol{x}) \propto \exp(\frac{1}{4}\|\boldsymbol{x}\|_2^2)$, and the subset $\mathcal{S}$ is chosen to be the $d$-dimensional ball with radius $\sqrt{2}\sqrt{2p+1}$. This leads to a uniform distribution $\tilde{f}_{\boldsymbol{X}}$ on $\mathcal{S}$ and to the weight function $w(\boldsymbol{x}) = \exp(-\frac{1}{4}\|\boldsymbol{x}\|_2^2)$.

Additionally, asymptotic distributions for Laguerre polynomials (corresponding to the Gamma distribution) and for Jacobi polynomials (Beta distribution) have been implemented in the software package COH-OPT (Hampton and Doostan, 2017).

For Legendre polynomials, asymptotic sampling has a smaller coherence than standard sampling in the case $d < p$ (asymptotically). In the case $d > p$, which is more common in applications, standard sampling has (asymptotically) a smaller coherence. According to theory, the sampling scheme with smaller coherence should exhibit better recovery rates. This is confirmed numerically (Hampton and Doostan, 2015b, section 5.1). For Hermite polynomials, the same observation is made.

#### 3.A.2.3 Coherence-optimal sampling

The choice $G = B$ leads to the minimum possible coherence $\mu(\mathcal{A}, \{\tilde{\psi}_{\boldsymbol{\alpha}}\})$ (Hampton and Doostan, 2015b, Theorem 4.5). $B$ is simple to evaluate for a single point $\boldsymbol{x} \in \mathcal{D}$, but its functional form is in general not known. Therefore, Hampton and Doostan (2015b) suggest sampling $\tilde{f}_{\boldsymbol{X}} \propto B^2 f_{\boldsymbol{X}}$ using Markov chain Monte Carlo (MCMC) sampling with proposal distribution equal to the input distribution in the case $d \geq p$ and equal to the asymptotic distribution in the case $d < p$.

The resulting (unnormalized) weights are $w(\boldsymbol{x}) = \frac{1}{B(\boldsymbol{x})}$. As expected from theory, numerical examples indicate that coherence-optimal sampling achieves better recovery and a smaller error in various norms than both standard and asymptotic sampling (Hampton and Doostan, 2015b). Coherence-optimal sampling can be shown to have good properties also when used as a sampling scheme for least-squares regression (Hampton and Doostan, 2015a).

MATLAB code for MCMC-based coherence-optimal sampling is available (Hampton and Doostan, 2015a,b). However, MCMC-based coherence-optimal sampling can be very slow for high-dimensional input. An alternative is rejection-based coherence-optimal sampling. Here, samples $\boldsymbol{x}_{\text{cand}}$ are generated from a proposal distribution $f_{\text{prop}}$, which has the property that there is a $\gamma \in \mathbb{R}$ such that $\gamma f_{\text{prop}}(\boldsymbol{x}) \geq \tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$. Uniform random numbers $u \sim_{\text{i.i.d.}} \mathcal{U}([0,1])$ are generated. A proposed point $\boldsymbol{x}_{\text{cand}}$ is accepted if $u \leq \frac{\tilde{f}(\boldsymbol{x}_{\text{cand}})}{\gamma f_{\text{prop}}(\boldsymbol{x}_{\text{cand}})}$. This is the implementation used in this benchmark. We use a product proposal density whose marginals are determined by the input marginals, the dimension of the problem, and the degree of the expansion: we choose uniform proposal marginals for uniform input marginals. For Gaussian input marginals, we use Gaussian proposal marginals if $d \geq p$; otherwise, we use the corresponding asymptotic distribution. As usual, lognormal input is mapped to Gaussian random variables before sampling (Blatman and Sudret, 2011).

Note that for Gaussian input, coherence-optimal and asymptotic sampling have a significantly larger spread than input sampling, as can be seen from Figure 3.A.2. Their support is the ball of radius $r = \sqrt{2}\sqrt{2p+2}$ (as implemented in (Hampton and Doostan, 2017)). This can potentially cause problems in engineering applications, for which simulations may be less accurate when the input parameters are far from typical operating conditions.

#### 3.A.2.4 Christoffel sparse approximation

A similar weighted sampling scheme is Christoffel sparse approximation (Narayan et al., 2017; Jakeman et al., 2017; Cohen and Migliorati, 2017). Those authors propose to use the weight function

$$w(\boldsymbol{x}) = \left( \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} |\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})|^2 \right)^{-\frac{1}{2}} \tag{3.19}$$

which leads to a modified basis that has pointwise minimal average squared basis magnitude (compare to (3.18) with $G = B$). This quantity (3.6) is a measure similar to coherence (3.5) and is used by Hampton and Doostan (2015a) and Cohen and Migliorati (2017) together with the induced probability measure to obtain convergence results for weighted least-squares regression. Narayan et al. (2017) and Jakeman et al. (2017) choose as probability distribution the so-called *weighted pluripotential equilibrium measure* (possibly degree-dependent), which asymptotically coincides with $\tilde{f}(\boldsymbol{x}) = c^2 w(\boldsymbol{x})^2 f(\boldsymbol{x})$ when the total degree of the truncated basis $p \to \infty$. However, the modified basis is not orthonormal with respect to this measure, which leads to weaker theoretical recovery results. Theoretical results are available only for the univariate case. In numerical examples, the method performs well for low-dimensional high-degree cases and often very similarly to asymptotic sampling. In high dimensions, it performs worse than input sampling (i.e., MC). It has not been compared to coherence-optimal sampling.

### 3.A.3 Choosing points according to an optimality criterion from a candidate set

The following methods aim to improve the properties of the regression matrix by choosing the "best" design points from a large candidate set. The methods differ in the criterion defining what are the "best" points. Most presented algorithms are greedy or heuristic and are actually only able to find a suboptimal design (local optimum). The choice of the candidate set obviously influences the quality of the resulting design. In the literature, candidate sets were sampled from MC (Diaz et al., 2018), LHS (Fajraoui et al., 2017), coherence-optimal sampling (Diaz et al., 2018; Alemazkoor and Meidani, 2018a), or Christoffel sparse approximation (Shin and Xiu, 2016). In the case of coherence-optimal sampling or Christoffel sparse approximation, the resulting optimized sample inherits the weights. It also often preserves the spread of the candidate set, as can be seen in Figure 3.A.2.

#### 3.A.3.1 D-optimal sampling

D-optimal design of experiments (Kiefer and Wolfowitz, 1959; Dykstra, 1971) aims at maximizing the determinant of the so-called *information matrix* $\frac{1}{N}\boldsymbol{\Psi}^T\boldsymbol{\Psi} \in \mathbb{R}^{P\times P}$. The D-value is defined as

$$D(\boldsymbol{\Psi}) = \det(\boldsymbol{\Psi}^T\boldsymbol{\Psi}). \tag{3.20}$$

Sometimes the determinant of the inverse information matrix is minimized (Nguyen and Miller, 1992), or the $P$th root is taken for normalization purposes (Diaz et al., 2018). The maximization of this determinant is connected to the minimization of the variance of the PCE coefficient estimate (Nguyen and Miller, 1992; Zein et al., 2013). Note that $D(\boldsymbol{\Psi}) = 0$ if $N < P$.

There exists a large selection of methods for constructing D-optimal experimental designs. For an overview of methods for constructing designs following alphabetic optimality criteria (such as A-, D-, or E-optimality), see (Hadigol and Doostan, 2018, Section 4.5).

Here, we only discuss D-optimal sampling based on rank-revealing QR decomposition (RRQR) (Diaz et al., 2018), since this is the technique used in our benchmark. We decided to use RRQR-based D-optimal sampling because it can be used even in the case $N < P$ when other D-optimal methods fail due to singularity of the information matrix. Note that RRQR is not guaranteed to find a design with maximal D-value but only a local optimum (Diaz et al., 2018, Section 3.4).

Let $\boldsymbol{\Psi}_{\text{cand}} \in \mathbb{R}^{M\times P}$ be the regression matrix evaluated at a set of $M$ candidate points. The goal is to select $N \leq M$ points from this candidate set with the property that the D-value of the resulting regression matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N\times P}$ is as large as possible. Since in the case of sparse PCE often $N < P$, which leads to $D(\boldsymbol{\Psi}) = 0$, another strategy is necessary. The RRQR decomposition, also known as pivoted QR, aims at permuting the columns of the original matrix in a way that ensures the R-matrix of the associated QR decomposition is as well-behaved[13] as possible. This is useful for inexpensively determining the numerical rank of a matrix (Hong and Pan, 1992; Gu and Eisenstat, 1996). RRQR has a strong connection to SVD and to the

[13] $\boldsymbol{R} = \begin{pmatrix} A_k & B_k \\ 0 & C_k \end{pmatrix}$ where $A_k$ is well-conditioned and $\|C_k\|_2$ is small

selection of submatrices of maximal determinant (Hong and Pan, 1992). Gu and Eisenstat (1996) propose a pivoted QR decomposition where pivots are chosen to maximize the determinant of the resulting quadratic submatrix of R. The exchange of rows is based on a formula relating the determinant of the quadratic submatrix of R before and after the row exchange by a simple factor (Gu and Eisenstat, 1996, Lemma 3.1). This algorithm can be used together with SVD to perform subset selection to construct an initial experimental design from a large set of candidate samples (Seshadri et al., 2017; Diaz et al., 2018). Here, first an SVD of the matrix $\boldsymbol{\Psi}_{\text{cand}}^T$ is computed. Then RRQR is applied to the transpose of the matrix consisting of the first $N$ right singular vectors. The resulting permutation matrix is used to determine the points to be chosen from the candidate set.

### 3.A.3.2 Quasi-optimal sampling based on the S-value

Here, the idea is to select samples from a pool of candidate points so that the PCE coefficients obtained using the selected set are as close as possible to the coefficients that would be obtained if the whole set of candidate points was used (Shin and Xiu, 2016a). Under the assumption that the columns of the matrix $\boldsymbol{\Psi}_{\text{cand}}$ are mutually orthogonal, the S-value is defined by

$$S(\boldsymbol{\Psi}) = \left( \frac{\sqrt{\det \boldsymbol{\Psi}^T \boldsymbol{\Psi}}}{\prod_{i=1}^{P} \|\Psi_i\|_2} \right)^{\frac{1}{P}}, \tag{3.21}$$

where $\Psi_i$ denotes the $i$th column of the regression matrix $\boldsymbol{\Psi}$. Its maximization has the (heuristic) effect of maximizing the column orthogonality of the regression matrix while at the same time maximizing the determinant of the information matrix (Shin and Xiu, 2016a). It holds that $S(\boldsymbol{\Psi}) \in [0, 1]$ due to Hadamard's inequality. If $N < P$, $S(\boldsymbol{\Psi}) = 0$. If $N \geq P$, $S(\boldsymbol{\Psi}) = 1$ if and only if the columns of $\boldsymbol{\Psi}$ are mutually orthogonal. There exists an update formula for the S-value when the regression matrix is augmented by one row, which thus avoids the repeated calculation of determinants.

Shin and Xiu (2016a) suggest a greedy algorithm that in every iteration augments the current matrix by an additional row which maximizes the S-value of the resulting matrix among all candidate rows. When the current number of rows in the matrix $\boldsymbol{\Psi}$ is smaller than the number of columns, the procedure can be adapted to avoid $S(\boldsymbol{\Psi}) = 0$. We do not include it in our benchmark because it is not well suited for situations where there are more basis polynomials than design points, which is the case in sparse PCE without experimental design enrichment. However, in a sequential enrichment context (Fajraoui et al., 2017) and for least-squares regression (Shin and Xiu, 2016), this algorithm performs well.

### 3.A.3.3 Near-optimal sampling

The coherence parameter (3.15) gives a bound on the recovery rate, but it is not the only criterion that has been studied with respect to recovery accuracy. Two other matrix properties related to recovery accuracy are mutual coherence and average cross-correlation. Both of them consider the correlation between normalized columns of the regression matrix, i.e., their scalar product. They are scalar measures of how "orthonormal" the columns of a rectangular matrix

$\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ with $N < P$ are. The heuristic idea is that columns should point in as different directions as possible, so that the multiplication with sparse coefficient vectors, which results in a linear combination of a subset of the columns, is "as unique as possible". This facilitates the recovery of the true sparse solution (assuming that it exists).

The *mutual coherence* is defined by

$$\mu(\boldsymbol{\Psi}) = \max_{i \neq j} \frac{|\Psi_i^T \Psi_j|}{\|\Psi_i\|_2 \, \|\Psi_j\|_2}, \tag{3.22}$$

where $\Psi_i$ denotes the $i$th column of the regression matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$. The mutual coherence is the worst-case cross-correlation between any two columns of the matrix. It is zero for orthonormal matrices and positive for $N < P$.

The *average (squared) cross-correlation* is defined by

$$\gamma(\boldsymbol{\Psi}) = \frac{1}{P(P-1)} \left\| \mathbb{1}_P - \tilde{\boldsymbol{\Psi}}^T \tilde{\boldsymbol{\Psi}} \right\|_F^2 = \frac{1}{P(P-1)} \sum_{i \neq j} \frac{|\Psi_i^T \Psi_j|^2}{\|\Psi_i\|_2^2 \, \|\Psi_j\|_2^2} \tag{3.23}$$

where $\tilde{\boldsymbol{\Psi}}$ is the column-normalized version of $\boldsymbol{\Psi}$, and $\Psi_i$ denotes the $i$th column of the regression matrix. The norm is the Frobenius-norm, taking the sum of squares of all matrix entries, and the factor $P(P-1)$ is the number of column pairs.

Alemazkoor and Meidani (2018a) suggest simultaneously optimizing mutual coherence and average cross-correlation by using the greedy procedure described in Algorithm 1 below: In each iteration, the current regression matrix is augmented by one row. This row corresponds to that point $\boldsymbol{x}_j$ from the large pool of candidate points which minimizes the (normalized) distance of $(\mu'_j, \gamma'_j) \in \mathbb{R}^2$ to the "utopia point" $(\min(\boldsymbol{\mu}'), \min(\boldsymbol{\gamma}'))$ among all candidate points.

**Algorithm 1** Near-optimal sampling (Alemazkoor and Meidani, 2018a).

1: Sample a large number $M$ of candidate points from the coherence-optimal distribution and compute candidate rows arranged in a matrix $\boldsymbol{\Psi}_{\text{cand}}$
2: Initialize $\boldsymbol{\Psi}_{\text{opt(1)}}$ to be a random row from $\boldsymbol{\Psi}_{\text{cand}}$
3: **for** $i = 2 \dots N$ **do**
4: **for** $j = 1 \dots M$ **do**
5: $\boldsymbol{\Psi}_{\text{temp}} = \text{row-concatenate}(\boldsymbol{\Psi}_{\text{opt(i-1)}}, \Psi^{(j)}_{\text{cand}})$
6: $\mu'_j = \mu(\boldsymbol{\Psi}_{\text{temp}})$ and $\gamma'_j = \gamma(\boldsymbol{\Psi}_{\text{temp}})$
7: **end for**
8: $\boldsymbol{\mu}' = (\mu'_1, \dots, \mu'_M)$ and $\boldsymbol{\gamma}' = (\gamma'_1, \dots, \gamma'_M)$
9: $j^* = \arg\min_j \left( \frac{\mu'_j - \min(\boldsymbol{\mu}')}{\max(\boldsymbol{\mu}') - \min(\boldsymbol{\mu}')} \right)^2 + \left( \frac{\gamma'_j - \min(\boldsymbol{\gamma}')}{\max(\boldsymbol{\gamma}') - \min(\boldsymbol{\gamma}')} \right)^2$
10: $\boldsymbol{\Psi}_{\text{opt(i)}} = \text{row-concatenate}(\boldsymbol{\Psi}_{\text{opt(i-1)}}, \Psi^{(j^*)}_{\text{cand}})$
11: **end for**

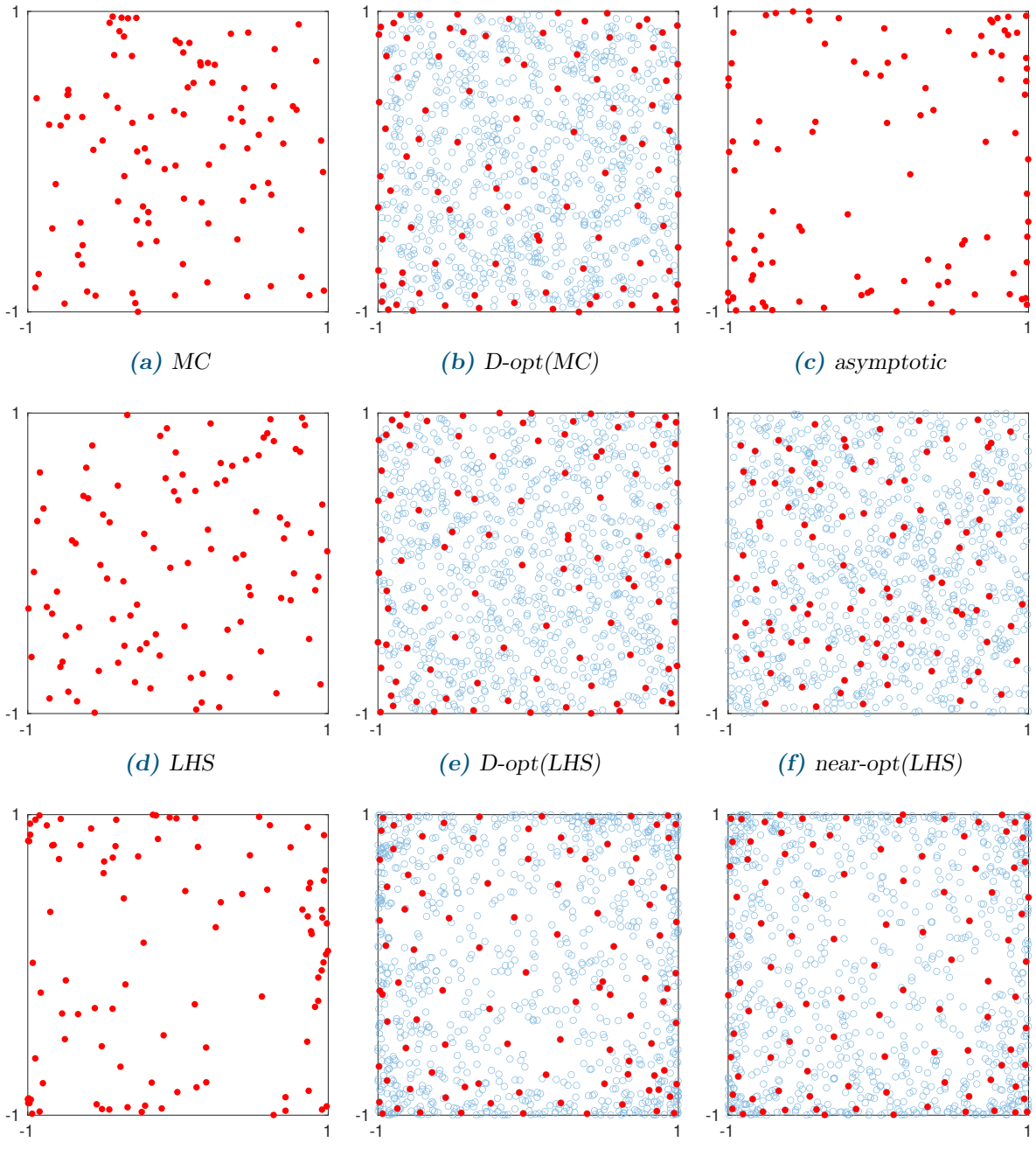
The algorithm is called near-optimal because it is a greedy algorithm, finding only a local optimum, and because optimized mutual coherence and average cross-correlation are only hinting at, but not guaranteeing, good recovery accuracy (Alemazkoor and Meidani, 2018a). Its computational complexity is $\mathcal{O}(NMP^2)$, where $N$ is the size of the final experimental design, $M$ is the

number of candidate samples (chosen to be, e.g., proportional to $P$ (Diaz et al., 2018)), and $P$ is the number of regressors. This makes the algorithm prohibitively expensive in the case of large bases ($P$ in the order of thousands), which is why we do not use it for some of the benchmark examples.

### 3.A.4 Illustration of sampling schemes

In Figures 3.A.1 and 3.A.2, we show illustrations of experimental designs in $d = 2$ dimensions with $N = 100$ and $p = 12$ for selected sampling techniques. The candidate set has a size of $M = 1000$. Figure 3.A.1 presents experimental designs for uniform input in the interval $[-1, 1]$, while Figure 3.A.2 presents experimental designs for standard Gaussian input.

Note that in the standard Gaussian case, the asymptotic distribution, the coherence-optimal distribution, and the matrix-optimal distributions based on a coherence-optimal candidate set all have a very large spread that grows with the total degree of the basis. For degree $p = 12$, some points are seven standard deviations away from the mean. Engineering models are typically calibrated only for a certain region of the input domain corresponding to nonnegligible probability, and they may be less accurate (or even fail) outside of this region.

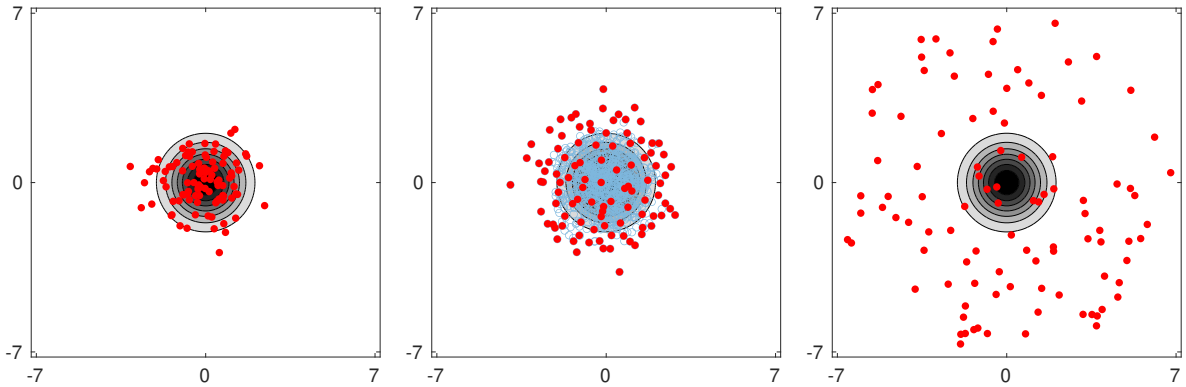



*(a)* MC *(b)* D-opt(MC) *(c)* asymptotic

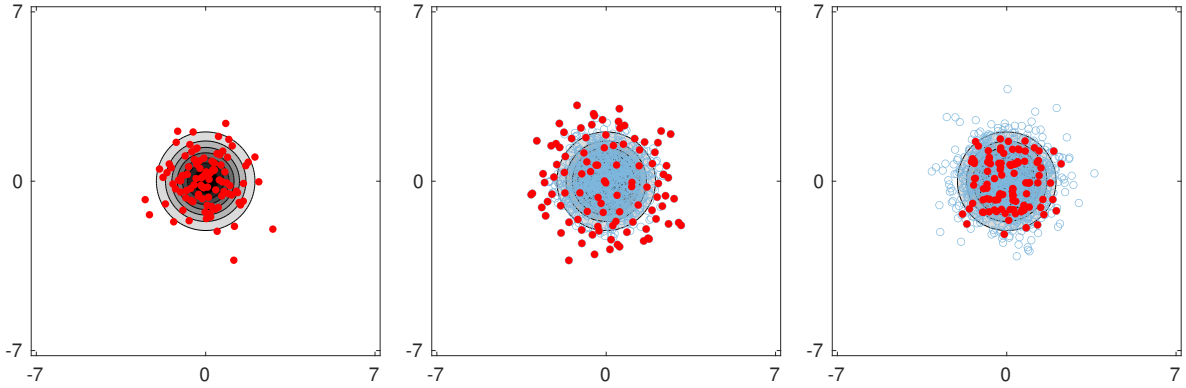
*(d)* LHS *(e)* D-opt(LHS) *(f)* near-opt(LHS)

*(g)* coh-opt *(h)* D-opt(coh-opt) *(i)* near-opt(coh-opt)

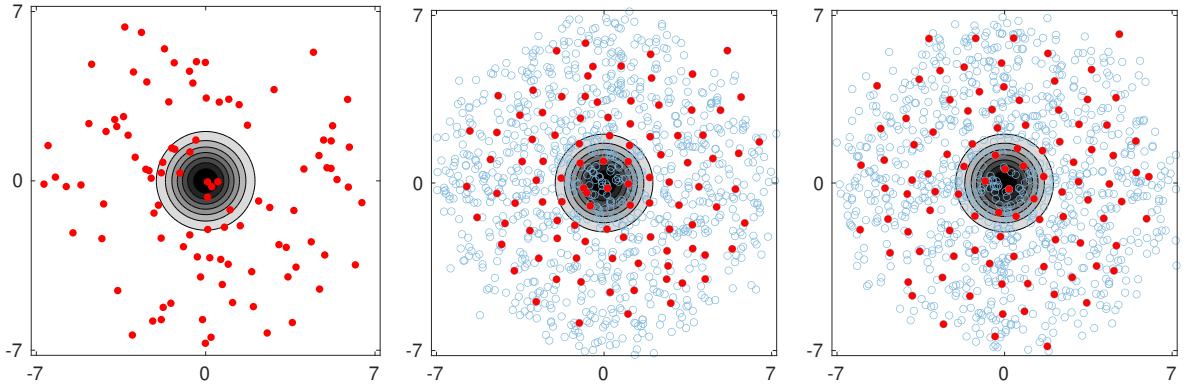
***Figure 3.A.1:*** *Visualization of experimental designs constructed for uniform input in* $[-1,1]^2$ *for degree* $p = 12$*. Red filled points denote the chosen experimental design, while blue circles denote the candidate set. Size of the ED:* $N = 100$*, size of the candidate set:* $M = 1000$*.*




*(a)* MC *(b)* D-opt(MC) *(c)* asymptotic




*(d)* LHS *(e)* D-opt(LHS) *(f)* near-opt(LHS)




*(g)* coh-opt *(h)* D-opt(coh-opt) *(i)* near-opt(coh-opt)

***Figure 3.A.2:*** *Visualization of the experimental design constructed for standard Gaussian input in $d = 2$ dimensions for degree $p = 12$. The gray surface plot illustrates the Gaussian probability density function. Red filled points denote the chosen experimental design, while blue circles denote the candidate set. Size of the ED: $N = 100$; size of the candidate set: $M = 1000$. The support of the asymptotic and the coherence-optimal distribution is the ball of radius $r = \sqrt{2}\sqrt{2p+2} \approx 7.2$. Note that engineering models may be less accurate in regions where the input distribution has negligible mass.*

## 3.B Details on sparse regression solvers

In this appendix, we describe the sparse solvers used in our benchmark in more detail: LARS, OMP, subspace pursuit, SPGL1 and FastLaplace (BCS). In addition, we present an overview of greedy stepwise regression solvers for sparse PCE.

There exist various formulations for the sparse regression problem. The typical form minimizes the $\ell^2$-norm of the empirical error under an additional constraint that is designed to enforce sparsity.

Sparsity is measured by the number of nonzero entries in a vector, formally denoted by $\|\boldsymbol{c}\|_0 = \sum_i \mathbf{1}_{\{c_i \neq 0\}}$ (even though this expression is not a norm). This results in the sparse regression problem

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{c}\|_0 \tag{3.24}$$

called $\ell^0$-minimization. The only way to solve this problem exactly is by a combinatorial search through all possible nonzero patterns for $\boldsymbol{c}$, which is infeasible for large problem sizes.

The convex relaxation of this problem is $\ell^1$-minimization, where $\|\boldsymbol{c}\|_0$ is replaced by $\|\boldsymbol{c}\|_1 = \sum_i |c_i|$. There are several equivalent formulations of the relaxed problem, namely

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{c}\|_1 \tag{3.25}$$

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \text{ s.t. } \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2 \leq \sigma \tag{3.26}$$

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2 \text{ s.t. } \|\boldsymbol{c}\|_1 \leq \tau \tag{3.27}$$

called Lagrangian formulation, basis pursuit denoising (BPDN), and least absolute shrinkage and selection operator (LASSO), respectively. It has been shown that under certain conditions, the solutions to $\ell^0$-minimization and $\ell^1$-minimization coincide (Bruckstein et al., 2009). However contrary to (3.24), formulations (3.25)–(3.27) are convex problems and allow a numerical solution with considerably smaller cost. The three formulations (3.25), (3.26), and (3.27) are equivalent in the sense that if $\hat{\boldsymbol{c}}$ is solution to one of the formulations, there exists a value of constraint parameter $\sigma, \tau$, or $\lambda$ so that $\hat{\boldsymbol{c}}$ is also a solution to the other formulations. However, the relationship between the parameters $\sigma, \tau, \lambda$ that makes the problems equivalent depends on $\boldsymbol{\Psi}$ and $\boldsymbol{y}$ and is not known in advance (van den Berg and Friedlander, 2008).

There exist other sparsity-enforcing formulations, such as $\ell^p$-norms (Bruckstein et al., 2009), $\ell^1 - \ell^2$-minimization (Yin et al., 2015), or elastic net (Tarakanov and Elsheikh, 2019). One example that we will describe is Bayesian compressive sensing, where a sparsity-enforcing prior is used for the coefficients of the PCE, resulting in a formulation that is equivalent to a sparse regression problem with a different kind of sparsity constraint, e.g., one related to the Student-t distribution (Tipping, 2001).

In the following descriptions of the algorithms, $\mathcal{A} \subset \mathbb{N}^d$ with various sub- or superscripts denotes a set of multi-indices, which by definition of PCE can be identified with a set of basis polynomials. With the notation of Section 3.2.1, $\boldsymbol{y} \in \mathbb{R}^N$ denotes the vector of model responses, $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ denotes the regression matrix of basis polynomials evaluated at the $N$ experimental design points, and $\boldsymbol{c} \in \mathbb{R}^P$ denotes the coefficients of a PCE. The residual is defined by $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}$, so that the norm of the residual is the empirical error.

The polynomials used for building the PCE are sometimes also called basis functions, regressors, or predictors. A sparse PCE is a PCE for which only some of the basis functions have nonzero coefficients: these basis functions are called *active*. We assume the regressors are normalized, so that the correlation between two regressors is equivalent to their inner product (and to the cosine of the angle between them).

### 3.B.1 Orthogonal matching pursuit (OMP)

Orthogonal matching pursuit (OMP), also called forward stepwise regression, is a classical greedy technique for finding approximate solutions to the $\ell^0$-minimization problem (3.24) (Pati et al., 1993; Bruckstein et al., 2009). Despite its being a heuristic method, under certain assumptions there are theoretical guarantees for the solutions returned by OMP (Tropp and Gilbert, 2007; Bruckstein et al., 2009). OMP is an iterative algorithm that starts out with an empty model and adds the regressors one by one to the set of active regressors. In each iteration, OMP selects the regressor that is most correlated with the current residual, adds it to the set of active regressors, and then updates the coefficients of all active regressors to make sure the new residual is orthogonal to all of them and has smallest possible norm. The updating of the coefficients can be done through an update formula (Berchier, 2015) or by computing the least-squares solution to the system of equations involving only the active regressors (Marelli and Sudret, 2019).

The technique is presented in Algorithm 2. The iterations are continued until $\min\{N, P\}$ basis functions are in the active set (then either all polynomials are selected, or there are not enough points in the experimental design to use least-squares anymore).

**Algorithm 2** Orthogonal matching pursuit (OMP) (Pati et al., 1993; Tropp and Gilbert, 2007; Marelli and Sudret, 2019).

1: Given a set of candidate basis functions $\mathcal{A}_{\text{cand}}$
2: Initialize all coefficients to zero: $\boldsymbol{c}_0 = 0$. Set $\mathcal{A}_0 = \emptyset$
3: Set the residual vector $\boldsymbol{r} := \boldsymbol{y}$
4: **for** $i = 1, \dots, m$ **do** ▷ $m \leq \min\{N, P\}$
▷ OMP can be stopped early when the error did not decrease anymore for a while
5: Find $\boldsymbol{\alpha}^* \in \mathcal{A}_{\text{cand}} \setminus \mathcal{A}_{i-1}$ with maximal correlation with the residual by solving

$$\boldsymbol{\alpha}^* = \arg \max_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{cand}} \setminus \mathcal{A}_{i-1}} |\boldsymbol{r}^T \boldsymbol{\psi}_{\boldsymbol{\alpha}}|$$

▷ The entries of vector $\boldsymbol{\psi}_{\alpha}$ are evaluations of the basis function $\psi_{\alpha}$ at the ED
6: $\mathcal{A}_i = \mathcal{A}_{i-1} \cup \{\boldsymbol{\alpha}\}$ ▷ Current set of active predictors
7: Compute the coefficients $\boldsymbol{c}_i$ by least-squares using only the active indices $\mathcal{A}_i$
▷ This can be done in $\mathcal{O}(iN)$ when maintaining a QR factorization (Tropp and Gilbert, 2007)
8: Update the residual $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{\Psi}_{\mathcal{A}_i} \boldsymbol{c}_i$
9: **end for**

OMP does not, per se, return a sparse solution. If a desired level of sparsity $K$ is known a priori, the algorithm can be stopped after $K$ iterations. Another possibility is to stop the algorithm as

soon as the residual norm is smaller than some error threshold (Bruckstein et al., 2009; Doostan and Owhadi, 2011; Jakeman et al., 2015), where the best error threshold is determined through cross-validation. A third possibility is to determine the best number of active basis functions through a model selection criterion, e.g. the LOO error (Marelli and Sudret, 2019). Since the coefficients are computed by OLS on the active basis, the LOO can be computed cheaply (Chapelle et al., 2002; Blatman and Sudret, 2011). Typically, for an increasing sequence of basis functions the LOO error first decreases (reduction of underfitting), then increases (overfitting). This can be utilized to terminate the algorithm early once the LOO error starts rising (early-stop criterion) (Marelli and Sudret, 2019).

The computational complexity of OMP is $\mathcal{O}(mNP)$ (Tropp and Gilbert, 2007; Dai and Milenkovic, 2009), where $m \leq \min\{N, P\}$ is the number of iterations. The computation of the correlations of the current residual with all regressors is $\mathcal{O}(NP)$ and has to be performed $m$ times. The computation of the least-squares solution in step $i$ can be done in $\mathcal{O}(iN)$, e.g., by maintaining a QR factorization of the information matrix (Tropp and Gilbert, 2007), or by using Schur's complement to update the information matrix inverse whenever a new regressor is added.

From the authors' experience, OMP often suffers from overfitting and can produce an unreliable LOO error estimate, which can be detrimental in basis-adaptive settings (see also (Lüthen et al., 2022a)).

OMP is available in many software packages, among them UQLab (Marelli and Sudret, 2014).

### 3.B.2 Least angle regression (LARS)

Least-angle regression (LARS, sometimes also abbreviated LAR) is a greedy technique that finds an approximate solution to the $\ell^1$-minimization problem (Efron et al., 2004). It is similar to OMP in that the algorithm starts out with an empty model and adds regressors one by one based on their correlation with the residual. However, unlike OMP, which updates the coefficients using least-squares (making the residual orthogonal to all active regressors in each step), LARS updates the coefficients in such a way that all active regressors have equal correlation with the residual. LARS can be interpreted as producing a path of solutions to (3.27), corresponding to increasing $\tau$. The coefficients are increased in the equiangular direction until a nonactive regressor has as much correlation with the residual as all the active regressors. This regressor is then added to the set of active regressors and the new equiangular direction is computed. The optimal stepsize between the addition of subsequent regressors can be computed analytically (Efron et al., 2004). This algorithm solves (3.27) approximately. A slightly modified version of LARS, called LARS-LASSO, removes regressors whenever the sign of their coefficient changes, and it has been proven to solve (3.27) (or its noiseless counterpart) exactly under certain conditions (Efron et al., 2004; Bruckstein et al., 2009).

The LARS technique is presented in Algorithm 3. It returns a sequence $\mathcal{A}_1 \subset \mathcal{A}_2 \subset \ldots \subset \mathcal{A}_m$ of sets containing indices of active basis functions, with $m = \min\{N, P\}$. Just like OMP, LARS can be stopped when a predefined sparsity $K$ is reached or when the norm of the residual $\|\boldsymbol{r}\|_2$ falls below a predefined error threshold.

**Algorithm 3** Least angle regression (LARS) (Efron et al., 2004; Blatman and Sudret, 2011).

Given a set of candidate basis functions $\mathcal{A}_{\text{cand}}$
Initialize all coefficients to zero: $\boldsymbol{c}_0 = 0$. Set $\mathcal{A}_0 = \emptyset$
Set the residual vector $\boldsymbol{r} := \boldsymbol{y}$
**for** $i = 1, \ldots, m$ **do** ▷ $m = \min\{N, P\}$
    Find $\boldsymbol{\alpha} \in \mathcal{A}_{\text{cand}} \setminus \mathcal{A}_{i-1}$ with maximal correlation with the residual
    ▷ For $i > 1$, $\boldsymbol{\alpha}_i$ is the element "responsible for" $\gamma_i$
    $\mathcal{A}_i = \mathcal{A}_{i-1} \cup \{\boldsymbol{\alpha}\}$ ▷ Current set of active predictors
    Compute $\boldsymbol{c}_i$ ▷ Equiangular direction for all $\boldsymbol{\alpha} \in \mathcal{A}_i$
    ▷ $\boldsymbol{c}_1$ is equal to the first selected predictor
    ▷ (Efron et al., 2004, Eq. 2.6)
    Compute $\gamma_i$ ▷ Optimal stepsize: using this, there is a new regressor that is as much correlated with $r$ as all regressors in $\mathcal{A}_i$ are
    ▷ (Efron et al., 2004, Eq. 2.13)
    Compute the new coefficients $\boldsymbol{c}_i = \boldsymbol{c}_{i-1} + \gamma_i \boldsymbol{c}_i$ ▷ Move the coefficients jointly into the direction of the least-squares solution until one of the other predictors in $\mathcal{A}_{\text{cand}} \setminus \mathcal{A}_i$ has as much correlation with the residual as the predictors in $\mathcal{A}_i$ (ensured by choice of $\gamma_i$ and $\boldsymbol{c}_i$)
    Update the residual $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{\Psi}_i \boldsymbol{c}_i$
**end for**

A modified version of LARS, called *hybrid LARS*, uses the equicorrelated approach to select the predictors, but computes the coefficients of the metamodel by least-squares (Efron et al., 2004; Blatman and Sudret, 2011). Once the LARS algorithm has finished and returned the sequence of basis sets $\mathcal{A}_1, \ldots, \mathcal{A}_m$, the corresponding coefficients are recomputed by least squares, $\boldsymbol{c}_i = \boldsymbol{c}_i^{\text{LSQ}}$, which ensures minimal empirical error for every metamodel $(\mathcal{A}_i, \boldsymbol{c}_i)$. Hybrid LARS facilitates another way to choose the best sparsity level: as for OMP, a model selection criterion (e.g. LOO) for each metamodel can be evaluated, and the best one is chosen. This procedure is detailed in Algorithm 4. Cheap OLS-based computation of LOO (Chapelle et al., 2002; Blatman and Sudret, 2011) and the early-stop criterion (Marelli and Sudret, 2019) can be applied as well.

**Algorithm 4** Hybrid LARS with LOO-CV (Blatman and Sudret, 2011; Marelli and Sudret, 2019)

Initialization as in LARS (Algorithm 3)
**for** $i = 1, \ldots, m$ **do**
    Run one step of LARS and obtain $(\mathcal{A}_i, \boldsymbol{c}_i)$ ▷ Algorithm 3
    Recompute the coefficient vector using least-squares on the selected basis $\mathcal{A}_i$ only, obtaining $\boldsymbol{c}_i^{\text{OLS}}$ (the coefficients corresponding to $\mathcal{A} \setminus \mathcal{A}_i$ are set to zero) ▷ Hybrid LARS
    Compute the LOO error $\epsilon_{\text{LOO}}(i)$ for $\boldsymbol{c}_i^{\text{OLS}}$ ▷ OLS-based LOO computation (Chapelle et al., 2002; Blatman and Sudret, 2011)
**end for** ▷ early stopping possible by monitoring the LOO error (Marelli and Sudret, 2019)
Return the metamodel $(\mathcal{A}_{i^*}, \boldsymbol{c}_{i^*})$ with $i^* = \arg\min_i \epsilon_{\text{LOO}}(i)$

As for OMP, the computational complexity of LARS (in the case $N < P$) is $\mathcal{O}(mNP)$, where $m \leq \min\{N, P\}$ is the number of iterations. This is due to matrix-vector multiplication and matrix inversion which have to be performed in every iteration. The latter can be computed in $\mathcal{O}(mN)$, when using techniques such as Schur's complement to update the information matrix inverse whenever a new regressor is added.

LARS is available in many software packages, e.g., as MATLAB implementation in UQLab (Marelli and Sudret, 2014).

### 3.B.3 Subspace pursuit (SP)

Another formulation of the $\ell^0$-minimization problem is

$$\min_{\boldsymbol{c}\in\mathbb{R}^P} \|\boldsymbol{\Psi c} - \boldsymbol{u}\|_2 \text{ s.t. } \|\boldsymbol{c}\|_0 = K \tag{3.28}$$

which is equivalent to (3.24) for a certain choice of $\lambda$.

Subspace pursuit (SP) seeks to identify a solution to (3.28) by iteratively and greedily enlarging and shrinking the set of active basis functions (Dai and Milenkovic, 2009). As with LARS and OMP, regressors are added to the set of active basis functions according to their correlation with the residual. However, the regressors are not added one by one, but batchwise. More precisely, SP maintains at all times an active basis of size $K$, where $K$ denotes the desired sparsity. In each iteration, it adds $K$ regressors at once and computes the coefficients of the active regressors by OLS. Then, it removes the $K$ regressors with the smallest-in-magnitude coefficients. This is continued until convergence. Under certain assumptions, there are theoretical guarantees for the solution that SP returns (Dai and Milenkovic, 2009). To make the augmentation of the basis and the OLS regression feasible, it must hold that $2K \leq \min\{N, P\}$.

The technique is described in Algorithm 5 for a fixed value of sparsity $K$. The residual of a vector and a regression matrix is defined as

$$\text{residual}(\boldsymbol{y}, \boldsymbol{\Psi}) = \boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger\boldsymbol{y} \tag{3.29}$$

where $\boldsymbol{\Psi}^\dagger$ denotes the pseudoinverse of $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^\dagger\boldsymbol{y} = \boldsymbol{c}$ is the least-squares solution to $\boldsymbol{\Psi c} \approx \boldsymbol{y}$ (the case of an overdetermined system). The algorithm returns a set $\mathcal{A}$ containing $K$ multi-indices.

For arbitrary sparse vectors, the computational complexity is $\mathcal{O}(N(P+K^2)K)$ (Dai and Milenkovic, 2009). For very sparse vectors with $K^2 \in \mathcal{O}(P)$, the complexity thus becomes $\mathcal{O}(NPK)$, comparable to the runtime of OMP. The number of iterations that the SP algorithm performs can be shown to be $\mathcal{O}(K)$ in general and even $\mathcal{O}(\log K)$ in certain cases (Dai and Milenkovic, 2009).

When the optimal sparsity level $K$ is unknown, it can be determined e.g. by cross-validation: Diaz et al. (2018) suggest running Algorithm 5 for a range of $N_K = 10$ different values for $K$ and choosing the one with the smallest 4-fold cross-validation error. In this paper, we propose to use leave-one-out cross-validation instead of 4-fold cross-validation, resulting in the SP variant $\text{SP}_{\text{LOO}}$.

**Algorithm 5** Subspace pursuit (SP) (Dai and Milenkovic, 2009).

1: Given desired sparsity $K \leq \min\{\frac{N}{2}, \frac{P}{2}\}$
2: Given the experimental design and the candidate basis $\mathcal{A}_{\text{cand}}$, compute the associated regression matrix $\boldsymbol{\Psi}$ and the right-hand-side $\boldsymbol{y}$
3: $\mathcal{A}^0 = \{K$ indices corresponding to the largest magnitude entries in $\boldsymbol{\Psi}^T\boldsymbol{y}\}$
▷ Scalar product of columns of $\boldsymbol{\Psi}$ with $\boldsymbol{y}$
4: $\boldsymbol{y}_{\text{res}}^0 = \text{residual}(\boldsymbol{y}, \boldsymbol{\Psi}_{\mathcal{A}^0})$ ▷ Residual of least-squares solution based on active basis
5: **for** $l = 1, 2, \ldots$ **do**
6: $\mathcal{S}^l = \mathcal{A}^{l-1} \cup \{K$ indices corresponding to the largest magnitude entries in $\boldsymbol{\Psi}^T\boldsymbol{y}_{\text{res}}^{l-1}\}$
▷ Augment by indices of full basis that correlate best with the residual
7: $\boldsymbol{c} = \boldsymbol{\Psi}_{\mathcal{S}^l}^{\dagger}\boldsymbol{y}$ ▷ Least-squares solution based on set $\mathcal{S}^l$ of size $2K$
8: $\mathcal{A}^l = \{K$ indices corresponding to the largest magnitude entries in $\boldsymbol{c}\}$
9: $\boldsymbol{y}_{\text{res}}^l = \text{residual}(\boldsymbol{y}, \boldsymbol{\Psi}_{\mathcal{A}^l})$ ▷ Residual of least-squares solution based on $\mathcal{A}^l$
10: **if** $\left\|\boldsymbol{y}_{\text{res}}^l\right\|_2 \geq \left\|\boldsymbol{y}_{\text{res}}^{l-1}\right\|_2$ **then** ▷ if new $K$-sparse approx. is worse than the previous one
11: STOP iteration and **return** $\mathcal{A}^{l-1}$.
12: **end if**
13: **end for**

*Remark: In line 10, the original publication (Dai and Milenkovic, 2009) uses "$>$" instead of "$\geq$", but we also want to stop when the set has converged.*

A related algorithm is CoSAMP (Needell and Tropp, 2009), which differs from SP mainly in the number of regressors added in each iteration.

Subspace pursuit is available as MATLAB implementation in the software package `DOPT_PCE` (Diaz et al., 2018; Diaz, 2018).

### 3.B.4 SPGL1

$\ell^1$-minimization is a convex problem, since both the objective function and the constraint are convex functions. Therefore, convex optimization methods can be used to find a solution. In this section, we describe the algorithm SPGL1 (van den Berg and Friedlander, 2008).

For a given value of $\tau$, formulation (3.27) (LASSO) can be solved by spectral projected gradient (SPG) descent (Birgin et al., 2000; van den Berg and Friedlander, 2008).[14] However, for real-world problems, we often do not know a priori an appropriate value for $\tau$. On the other hand, a sensible range of values for $\sigma$ in formulation (3.26) (BPDN) can typically be estimated based on the noise level in the data and the expected model fit. In the case of PCE metamodelling, $\sigma$ can be related to an estimate of the relative MSE through $\text{RelMSE} = \frac{\sigma^2}{N\hat{\text{Var}}[\boldsymbol{y}]}$, whose values are for engineering models typically between $10^{-10}$ and $10^0 = 1$.

The main idea of the solver SPGL1 is to solve BPDN through a detour over LASSO. Let $\boldsymbol{c}_\tau$ be

[14] SPG is a gradient-based optimization algorithm with several enhancements (Barzilai–Borwein spectral step length and the Grippo–Lampariello–Lucidi scheme of nonmonotone line search) and projection onto the feasible set $\Omega_\tau = \{\boldsymbol{c} \in \mathbb{R}^P : \|\boldsymbol{c}\|_1 \leq \tau\}$

the solution to LASSO for a given $\tau$. Define a function $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ by

$$\phi(\tau) := \|\boldsymbol{\Psi}\boldsymbol{c}_\tau - \boldsymbol{y}\|_2 \,. \tag{3.30}$$

Then the solution to BPDN with $\sigma := \phi(\tau)$ is $\boldsymbol{c}_\tau$. In other words, $\phi$ is the functional relationship between $\sigma$ and $\tau$ that makes the two formulations BPDN and LASSO equivalent for given $\boldsymbol{\Psi}$ and $\boldsymbol{y}$. $\phi$ is the Pareto front of LASSO and BPDN and shows the trade-off between the minimal achievable $\ell^1$-norm of the coefficients and the minimal $\ell^2$-norm of the corresponding residual. The Pareto front is convex, nonincreasing and differentiable with an analytically computable derivative (van den Berg and Friedlander, 2008).

To find a solution to BPDN with a given $\sigma$, LASSO is solved with SPG several times for a sequence of $\tau$ until one is found with $\phi(\tau) = \sigma$. The sequence of $\tau$ is created by performing Newton's root finding algorithm on the function $f(\tau) = \sigma - \phi(\tau)$.

Each SPG iteration has a computational complexity of $\mathcal{O}(NP + P \log P)$ (from matrix-vector multiplication and $\ell^1$-projection). Multiplying this with the number of SPG steps and the number of Newton steps yields the computational complexity of SPGL1.

This algorithm is available as MATLAB package `SPGL1` (van den Berg and Friedlander, 2008; Van den Berg and Friedlander, 2015).

In our numerical benchmarks computing sparse PCE for compressible models, SPGL1 was among the slowest solvers and often returned rather dense solutions.

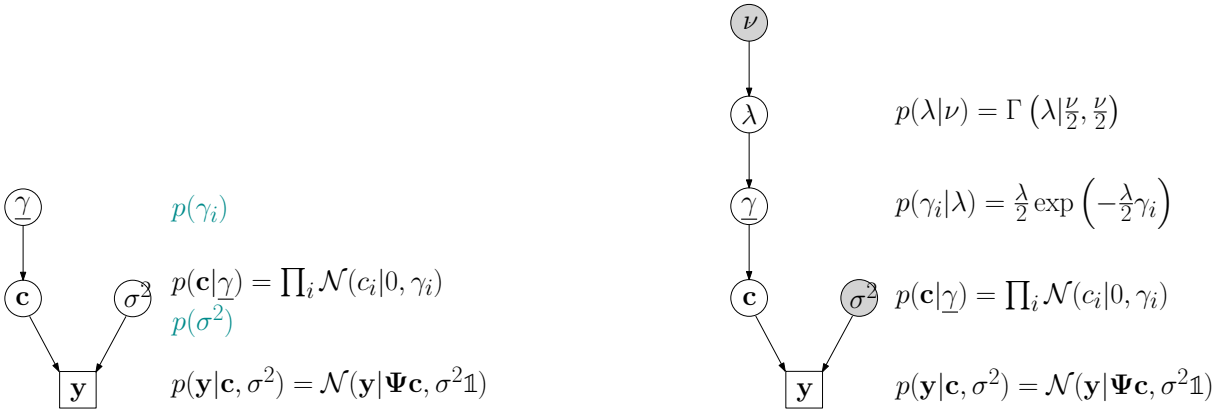
### 3.B.5 Sparse Bayesian learning

Methods from the class of Bayesian compressive sensing (BCS), also known as sparse Bayesian learning (SBL), embed the regression problem in a probabilistic framework (Tipping, 2001; Ji et al., 2008; Babacan et al., 2010; Sargsyan et al., 2014; Tsilifis et al., 2020). The goal is to compute, for a given model response vector $\boldsymbol{y}$ and a regression matrix $\boldsymbol{\Psi}$, the coefficient vector $\boldsymbol{c}^{\mathrm{MAP}}$ which maximizes the posterior distribution $p(\boldsymbol{c}|\boldsymbol{y})$. Another quantity of interest could be the most probable value $y^*$ at a new point $\boldsymbol{x}^*$ maximizing $p(y^*|\boldsymbol{y})$.

In BCS, it is assumed that the "measurements" $\boldsymbol{y}$ are generated by adding zero-mean, finite-variance noise to the evaluations of the true model. This noise is often assumed to be Gaussian white noise with standard deviation $\sigma$, which, for a given input $\boldsymbol{x}$, results in a Gaussian distribution for its output $\boldsymbol{y}$ with mean $\boldsymbol{\Psi}\boldsymbol{c}$ and covariance matrix $\sigma^2\mathbb{1}$ , i.e., $\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{x}, \sigma \sim \mathcal{N}(\boldsymbol{\Psi}\boldsymbol{c}, \sigma^2\mathbb{1})$. Note that in the case of PCE, this is generally not a valid assumption: when an important term is missing from the PCE model, the discrepancy between measurements and PCE model evaluations can be highly correlated, heteroscedastic, and non-Gaussian, and have nonzero mean. However, even though the assumptions might not be fulfilled, this framework can still be useful for finding sparse solutions.

The class of BCS algorithms comprises several methods that differ in the assumptions on the distributions of the various hyperparameters and in the (usually iterative, approximate) techniques for computing the posterior quantities.

In Figure 3.B.1a we present the general setup of sparse Bayesian learning. The measurements $\boldsymbol{y}$ are assumed to follow a Gaussian distribution as described above. The noise variance $\sigma^2$ is assumed to be a random variable whose distribution has to be specified (e.g. fixed, uniform, or inverse-Gamma). The coefficients $c_i$ are assumed to be random variables as well, drawn from a normal distribution with mean zero and variance $\gamma_i$, i.e., each weight has its own variance. $\boldsymbol{\gamma}$ is a so-called *hyperparameter*, parametrizing the distribution of a parameter.




***(a)*** *General setup of BCS* ***(b)*** *Setup by Babacan et al. (2010)*

***Figure 3.B.1:*** *Illustration of the general setup of BCS (a) and the hierarchical generalization of Babacan et al. (2010) (b). (a): The likelihood and the prior on the coefficients are usually Gaussian, but the choice of $p(\sigma^2)$ and $p(\boldsymbol{\gamma})$ differs between publications, as well as the resulting solution algorithm. (b): Babacan et al. (2010) makes a specific choice for $p(\boldsymbol{\gamma})$ and includes an additional layer of hyperparameters. Shaded variables are held fixed.*

So far, the described setup with fixed $\sigma^2$ and $\gamma_i$ would yield (weighted) ridge regression. The sparsity comes into play through an assumption on the distribution of the hyperparameter $\boldsymbol{\gamma}$. For specific choices of $p(\gamma_i)$, it can be shown that the resulting *effective prior* on the coefficients $p(\boldsymbol{c}) = \int p(\boldsymbol{c}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})\mathrm{d}\boldsymbol{\gamma}$ is a sparsity-encouraging distribution, i.e., one that has a sharp peak at zero, encouraging zero values, while at the same time a heavy tail, allowing for large coefficient values as well. Examples are the Laplace distribution and the generalized Student-t distribution (Wipf et al., 2004; Babacan et al., 2010; Figueiredo and Nowak, 2001).

Such sparsity-encouraging distributions are often intractable to use, because they do not allow for analytical computation of the desired values (such as the most likely coefficients given the data, or the prediction of the measurement value at a new point). However, feasible algorithms can be developed based on a suitable approximation step. Various frameworks for sparse Bayesian learning have been proposed whose setup follows the general structure of Figure 3.B.1a, but which differ in the choice of priors for the hyperparameters and employ different solution algorithms for the MAP estimate of these hyperparameters (Tipping, 2001; Faul and Tipping, 2002; Figueiredo, 2003; Tipping and Faul, 2003; Wipf and Rao, 2004; Ji et al., 2008; Seeger and Nickisch, 2008; Babacan et al., 2010; Sargsyan et al., 2014; Tsilifis et al., 2020; Bhattacharyya, 2020). The MAP estimate of the hyperparameters is inserted into the distribution for $\boldsymbol{c}$. Because $p(\boldsymbol{c}|\boldsymbol{\gamma})$ and $p(\boldsymbol{y}|\boldsymbol{c}, \sigma^2)$ are normal distributions, all subsequent computations can be carried out analytically (Tipping, 2001; Wipf et al., 2004; Wipf and Rao, 2004). The sparsity of $\boldsymbol{c}$ is

enforced because by the choice of $p(\boldsymbol{\gamma})$ and the other distributions, many of the components $\gamma_i^{\mathrm{MAP}}$ of $\boldsymbol{\gamma}^{\mathrm{MAP}}$ will actually be zero, forcing the corresponding $c_i$ to be zero as well.

BCS in the implementation of Babacan et al. (2010) was suggested for sparse PCE by Sargsyan et al. (2014). This approach employs an additional layer of hyperparameters as displayed in Figure 3.B.1b. The prior on the coefficient variances is an exponential distribution $p(\gamma_i|\lambda) = \mathrm{Exp}\left(\gamma_i|\frac{\lambda}{2}\right)$ with shared hyperparameter $\lambda$. The hyperparameter $\lambda$ follows a Gamma distribution $p(\lambda|\nu) = \Gamma(\lambda|\frac{\nu}{2}, \frac{\nu}{2})$ with hyperparameter $\nu$. $\nu \to 0$ implies $p(\lambda) \propto \frac{1}{|\lambda|}$ (improper prior) and $\nu \to \infty$ implies the certain value $\lambda = 1$. In practice, Babacan et al. (2010) find that $\nu = 0$ gives the best results. The prior on $\beta = \sigma^{-2}$ is a Gamma distribution $p(\beta) = \Gamma(\beta|a, b)$ with hyperparameters $a, b$. In practice, the algorithm does not estimate $\beta$ well, which is, however, crucial; therefore, it is set to a fixed value (e.g. $\beta^{-1} = 0.01\,\|\boldsymbol{y}\|_2^2$ in Babacan et al. (2010); in our benchmark, we use cross-validation to determine the best value for this parameter, similarly to the strategy for SPGL1). The objective function is the logarithm of the joint distribution $\mathcal{L}(\boldsymbol{\gamma}, \lambda, \beta) = \log p(\boldsymbol{y}, \boldsymbol{\gamma}, \lambda, \beta)$, which is an analytical expression. To maximize it, Babacan et al. (2010) adapt the fast approximate algorithm of Tipping and Faul (2003); Faul and Tipping (2002) to their generalized hierarchical setting. Here, the derivatives of the objective function with respect to the hyperparameters $\lambda, \beta$, and $\gamma_i, i = 1, \ldots, P$ are computed. This results in an iterative scheme where these parameters are optimized one at a time while the other ones are held fixed. The algorithm is explained in detail in Babacan et al. (2010, Algorithm 1) and has been implemented in MATLAB under the name FastLaplace (Babacan, 2011).

### 3.B.6 Greedy stepwise regression solvers

Many of the sparse regression solvers that have been proposed for computing sparse PCE belong to the class of greedy stepwise regression. Here, starting from an empty model, the regressors are added one by one according to a selection criterion (forward selection). Some methods also include a backward elimination step. Then the coefficients of the selected regressors are computed. The procedure is iterated until a stopping criterion is reached. Alternatively, several models are built and one is selected in the end using a model selection criterion. We summarize some greedy stepwise regression techniques proposed for sparse PCE, together with their choices for selection criterion, coefficient computation, and stopping criterion, in Table 3.B.1, including the well-known methods OMP and LARS.

New greedy methods in the fashion of Table 3.B.1 can easily be derived by pairing other methods for the regressor selection, the coefficient computation method, and the model selection criterion. Note that except for LARS and OMP, these greedy methods are heuristic (no theoretical guarantee of convergence) and often depend on a number of tuning parameters.

***Table 3.B.1:*** *A selection of greedy stepwise regression algorithms proposed for sparse PCE. F = forward selection, FB = forward selection and backward elimination.*

| Ref. | F/B | Regressor selection | Computation of the coefficients | Model selection / stopping criterion | Comment |
|---|---|---|---|---|---|
| OMP (Tropp and Gilbert, 2007) | F | correlation with residual | OLS | *several choices:* given number of regressors (Tropp and Gilbert, 2007); (modified) LOO (Marelli and Sudret, 2019); norm of residual (Baptista et al., 2019); threshold on moving average of LOO (Baptista et al., 2019) | theoretical guarantees exist (Tropp and Gilbert, 2007) |
| LARS (Efron et al., 2004; Blatman and Sudret, 2011) | F | correlation with residual | least angle strategy | *several choices:* coefficient of determination (Efron et al., 2004); (modified) LOO (Blatman and Sudret, 2011) | theoretical guarantees exist (Efron et al., 2004; Bruckstein et al., 2009) |
| Blatman and Sudret (2008) | FB | coefficient of determination | OLS | coefficient of determination | degree-adaptive |
| Blatman and Sudret (2010) | FB | coefficient of determination | OLS | LOO | degree-adaptive; with ED enrichment |
| Hu and Youn (2011) | F | testing bivariate interaction | stepwise moving least-squares | coefficient of determination | Only for interaction order $\leq 2$. ED-adaptive. Degree of interaction terms determined in inner loop. |
| Abraham et al. (2017) | FB | F: one-predictor regression criterion<br>B: confidence intervals of coefficients | OLS | given number of regressors or iterations | - |
| Shao et al. (2017) | F | partial correlation coefficient | variant of BCS (fixed priors) | KIC | prior depends on interaction order and degree of the regressor;<br>degree-adaptive |
| Cheng and Lu (2018a) | FB | variance contribution of coefficient | support vector regression (SVR) | coefficient of determination | - |
| AFBS (Zhao et al., 2019) | FB | correlation with residual | OLS | threshold for residual norm | Main difference with OMP: backward selection. Employs several elimination checks. |
| HSPLSR-PCE (Zhao and Bu, 2019) | FB | F: partial least-squares (PLS) (regressors partitioned into blocks)<br>B: soft thresholding | linear PLS on the regression matrix | modified (pseudo) LOO | Note that the input is assumed to be Gaussian, but the (nonlinear)<br>regressors do not follow a Gaussian distribution. |
| Zhou et al. (2019a) | F | (partial) distance correlation | PLS | coefficient of determination | adaptive in degree and interaction order |
| Zhou et al. (2019c) | FB | correlation with residual | BCS | KIC | nesting iterative BCS procedure with forward-backward-selection scheme |

## 3.C Additional results

In this appendix, we display additional results that complement the results shown in Sections 3.3.4–3.3.6. For a detailed description of the setup, we refer the reader to Section 3.3.

### 3.C.1 Comparison of sparse solvers

In Figure 3.C.1 we display the boxplots of relative MSE for the seven additional models presented in Table 3.1.

### 3.C.2 Comparison of sampling schemes together with solvers

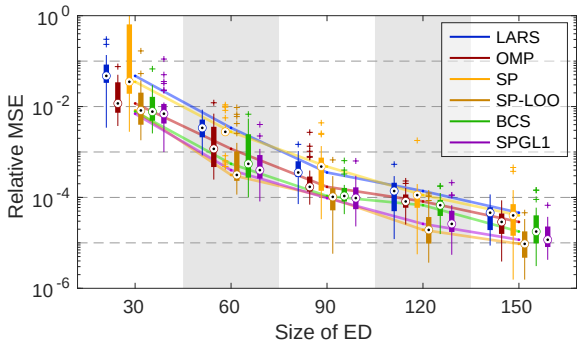
In Section 3.3.5, Figure 3.4 we showed aggregated results for the benchmark of solvers and sampling schemes. To give a more tangible impression of the data, in Figures 3.C.2–3.C.5 we display the boxplots of relative MSE against ED size for the four models Ishigami, borehole, two-dimensional diffusion, and 100D function. We show all combinations of solvers and sampling schemes, resulting in 16–20 combinations. Solvers are denoted by different colors. Sampling schemes are shown in varying shades and line styles. We also show the same results sliced at small and large ED sizes to compare the performance between solvers.

### 3.C.3 Comparison of sampling schemes together with solvers, using a smaller candidate basis

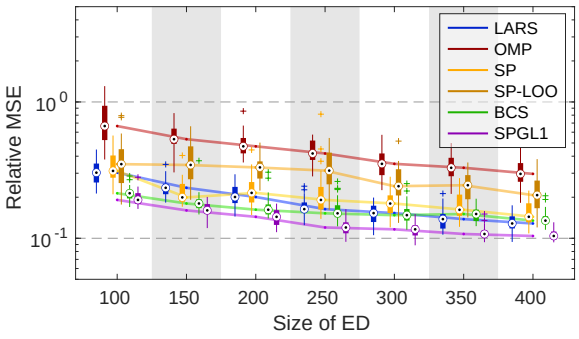
Due to space limitations, in Section 3.3.6 (Figures 3.5 and 3.6) we only showed results for two of the five solvers (OMP and $\mathrm{SP}_{\mathrm{LOO}}$). In Figure 3.C.6, we show boxplots of relative MSE against ED size for the three remaining solvers LARS, SP, and BCS.

## 3.D Benchmark studies

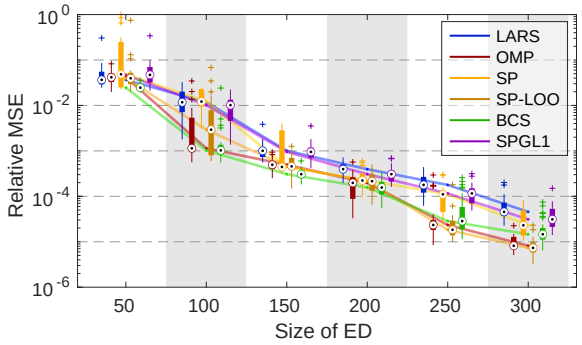
An overview of articles and benchmark studies comparing sparse PCE methods, including their main results, is given in Table 3.D.1.



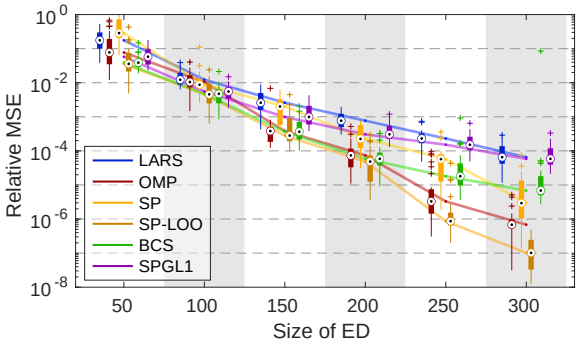

*(a) Undamped oscillator ($d = 6$)*



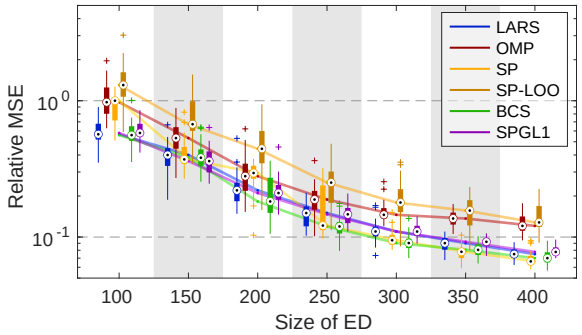

*(b) Damped oscillator ($d = 8$)*



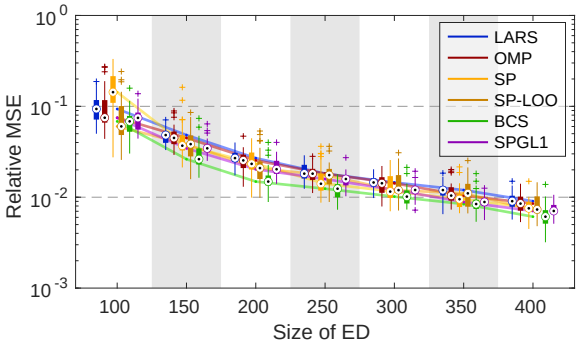

*(c) Wingweight function ($d = 10$)*



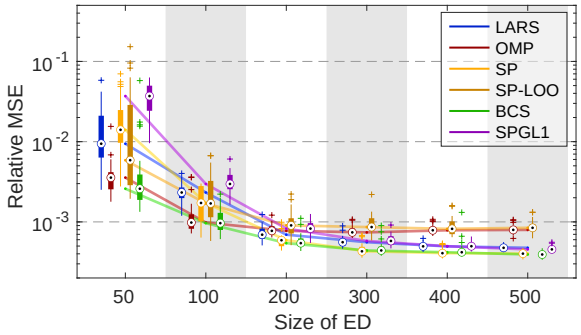

*(d) Truss model ($d = 10$)*




*(e) Morris function ($d = 20$)*

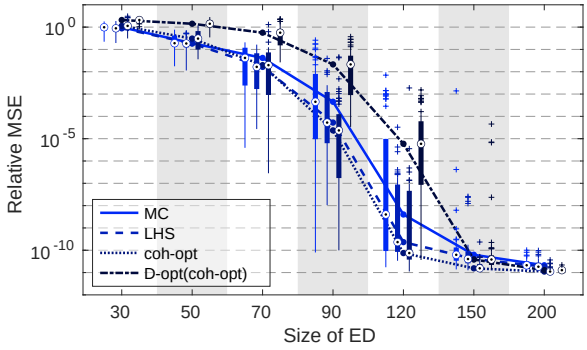



*(f) Structural frame model ($d = 21$)*

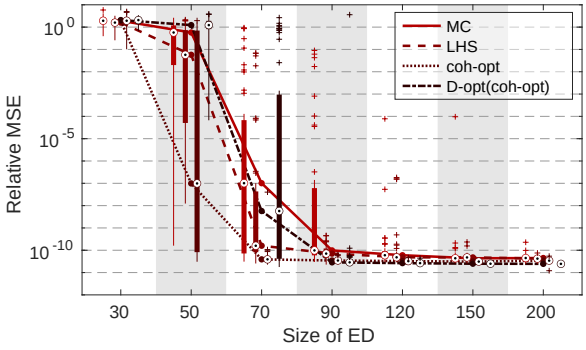



*(g) 1-dim diffusion model ($d = 62$)*

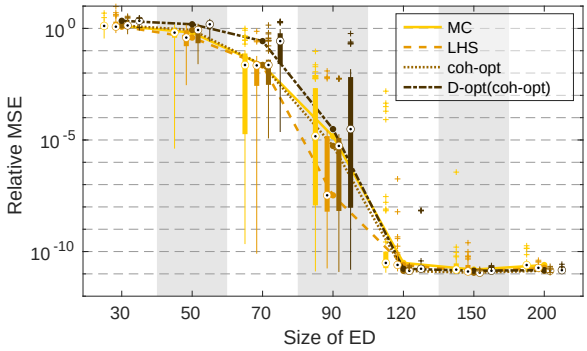
***Figure 3.C.1:*** *Results for seven additional models (see Table 3.1 in Section 3.3.4 for more details), complementing the results in Figure 3.2. Boxplots of relative MSE against experimental design for six sparse solvers and LHS design. Thirty replications. Note that the damped oscillator and the Morris function are very challenging for PCE: no solver achieves a relative MSE significantly smaller than 0.1, even when large EDs are used.*



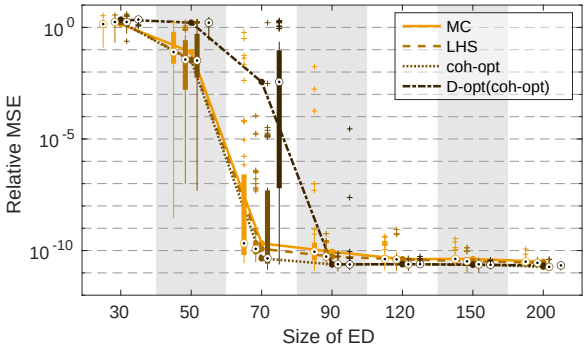

*(a) LARS*



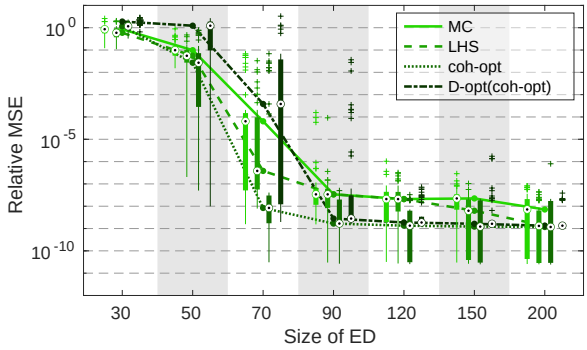

*(b) OMP*



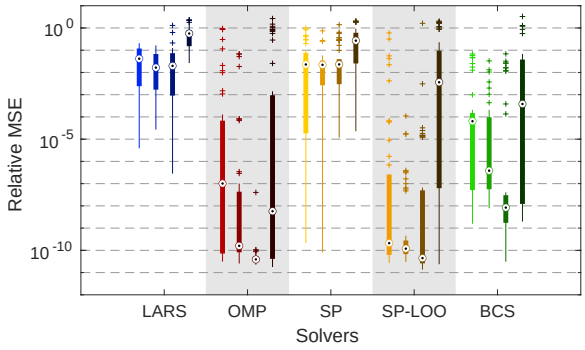

*(c) SP*



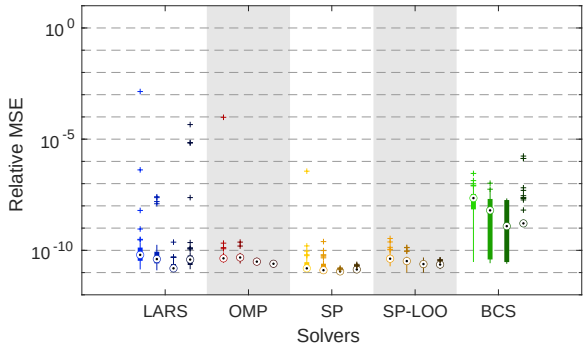

*(d) $SP_{LOO}$*




*(e) BCS*

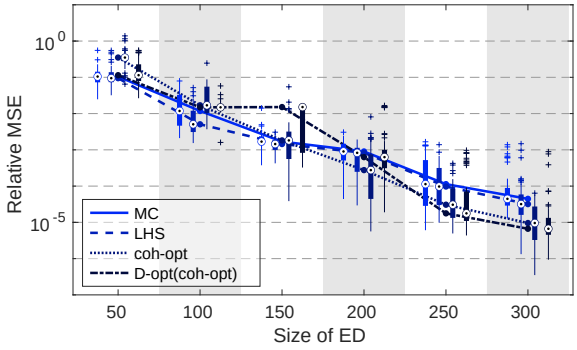



*(f) Small ED (70 points)*

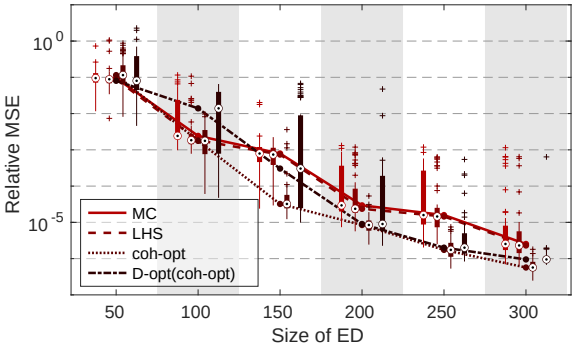



*(g) Large ED (150 points)*

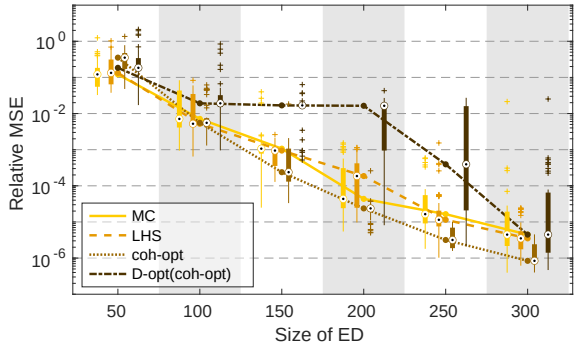
***Figure 3.C.2:*** *Boxplots of relative MSE from the benchmark of five solvers and four sampling schemes for the Ishigami model ($d = 3, p = 14, q = 1$). Solvers are coded by colors. Sampling schemes are shown in varying shades and line styles. In (f) and (g), we show the relative MSE of each of the solvers combined with each sampling scheme in the order MC–LHS–coh-opt–D-opt(coh-opt).*



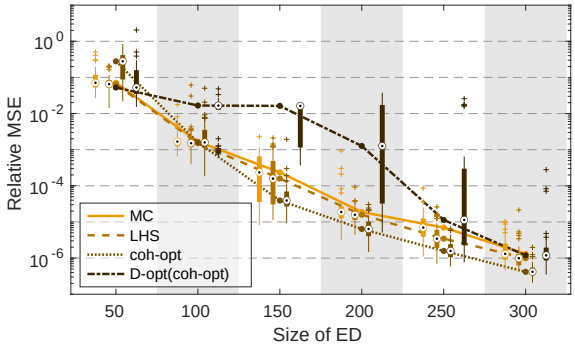

*(a) LARS*



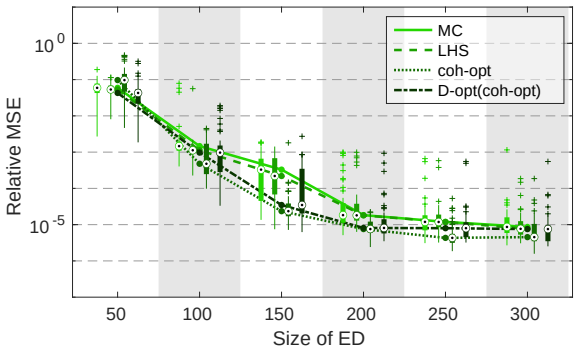

*(b) OMP*



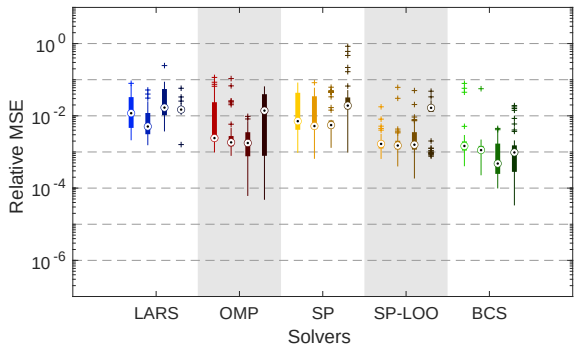

*(c) SP*



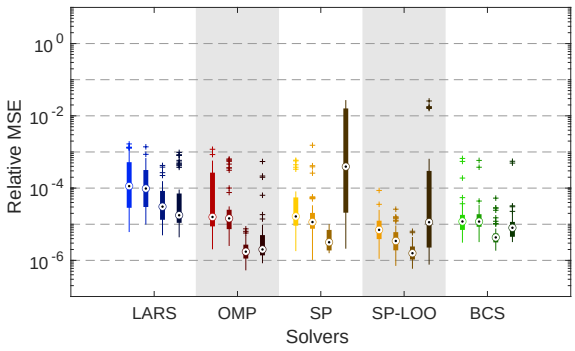

*(d) $SP_{LOO}$*




*(e) BCS*

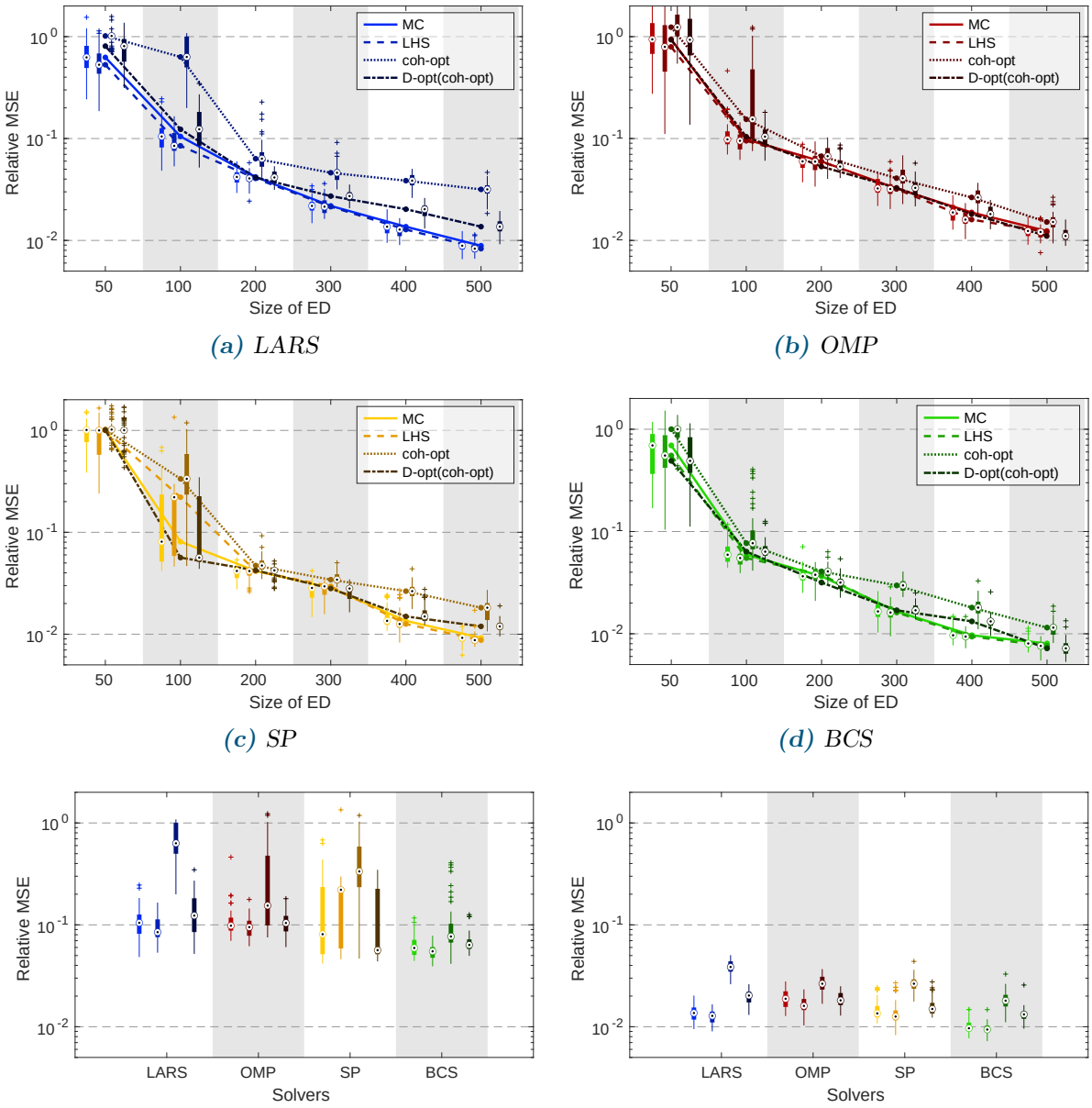



*(f) Small ED (100 points)*




*(g) Large ED (250 points)*

***Figure 3.C.3:*** *Boxplots of relative MSE from the benchmark of five solvers and four sampling schemes for the borehole model ($d = 8, p = 4, q = 1$). Solvers are coded by colors. Sampling schemes are shown in varying shades and line styles. In (f) and (g), we show the relative MSE of each of the solvers combined with each sampling scheme in the order MC–LHS–coh-opt–D-opt(coh-opt).*

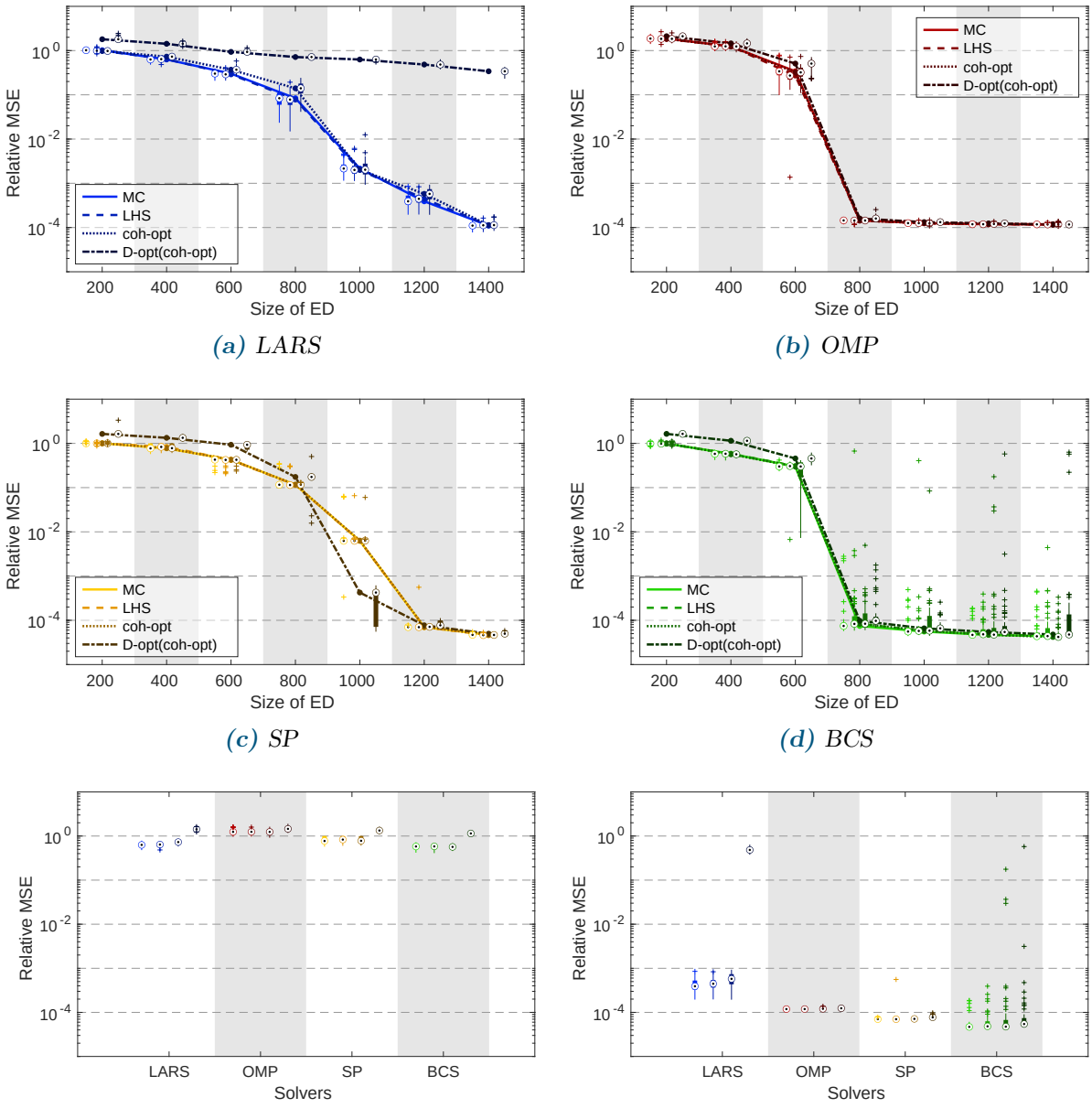



*(a) LARS* *(b) OMP*

*(c) SP* *(d) BCS*

*(e) Small ED (100 points)* *(f) Large ED (400 points)*

***Figure 3.C.4:*** *Boxplots of relative MSE from the benchmark of four solvers and four sampling schemes for the two-dimensional diffusion model ($d = 53, p = 4, q = 0.5$). Solvers are coded by colors. Sampling schemes are shown in varying shades and line styles. In (e) and (f), we show the relative MSE of each of the solvers combined with each sampling scheme in the order MC–LHS–coh-opt–D-opt(coh-opt).*

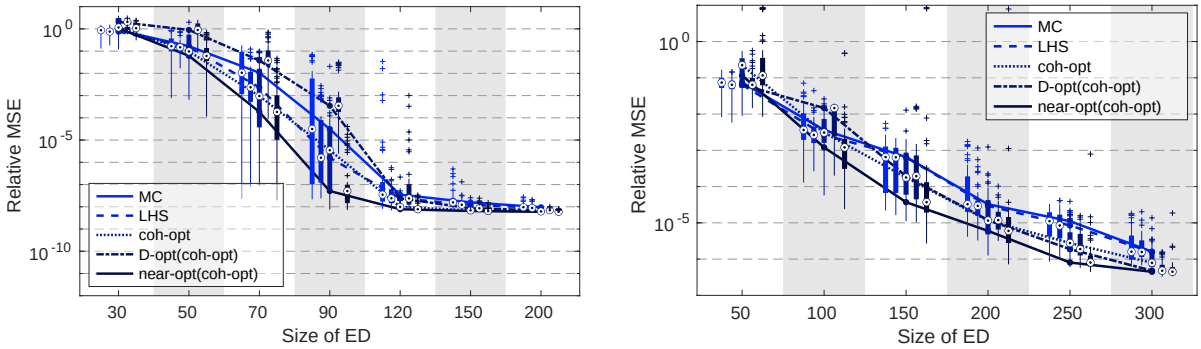



*(a) LARS* *(b) OMP*

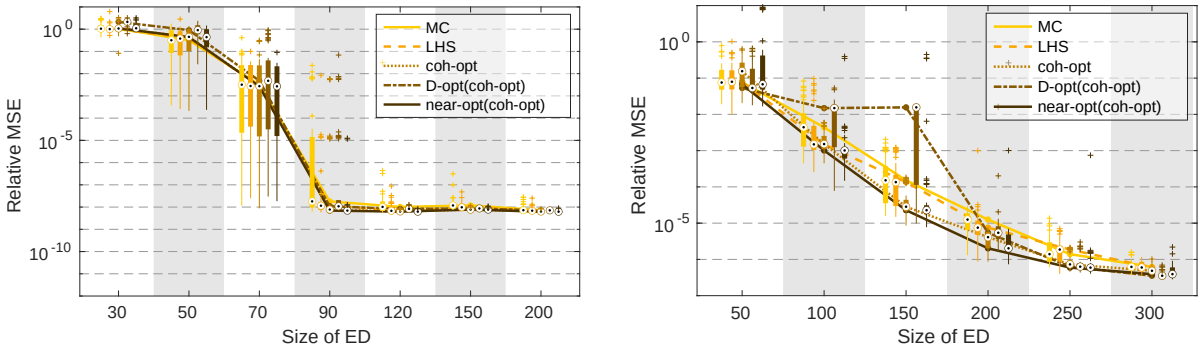
*(c) SP* *(d) BCS*

*(e) Small ED (400 points)* *(f) Large ED (1200 points)*

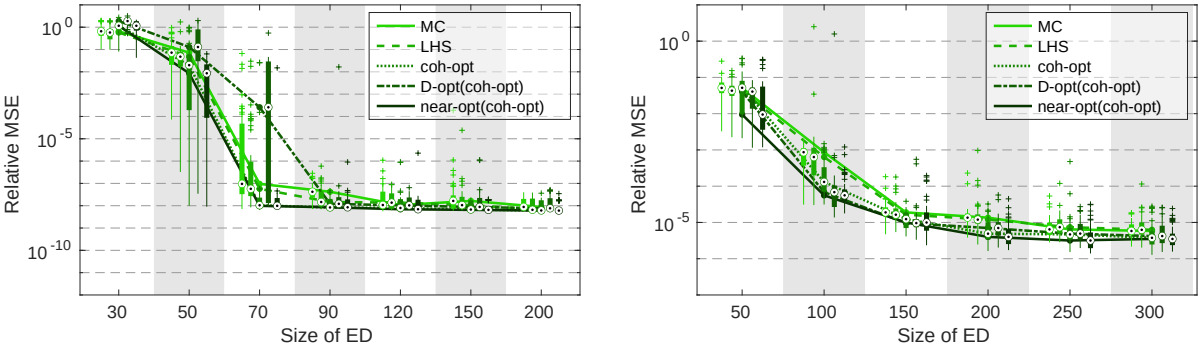
***Figure 3.C.5:*** *Boxplots of relative MSE from the benchmark of four solvers and four sampling schemes for the 100D function ($d = 100, p = 4, q = 0.5$). Solvers are coded by colors. Sampling schemes are shown in varying shades and line styles. In (e) and (f), we show the relative MSE of each of the solvers combined with each sampling scheme in the order MC–LHS–coh-opt–D-opt(coh-opt).*




*(a) Ishigami model, LARS* *(b) Borehole model, LARS*




*(c) Ishigami model, SP* *(d) Borehole model, SP*




*(e) Ishigami model, BCS* *(f) Borehole model, BCS*

***Figure 3.C.6:*** *Left column: results for the Ishigami model with a smaller basis ($d = 3, p = 12, q = 1$), complementing the plots in Figure 3.5. Right column: results for the borehole model with a smaller basis ($d = 8, p = 4, q = 1$), complementing the plots in Figure 3.6. Results for three sparse solvers and five experimental design schemes. Fifty replications.*

***Table 3.D.1:*** *Overview of some articles and benchmark studies comparing sparse PCE methods. The "best method" listed in the last column is the one delivering the smallest target error, as reported in the respective publications. Target quantities can be moments, Sobol' indices or the generalization error of the PCE surrogate. Abbreviations: '$\succ$' stands for 'better than'; for the acronyms of solvers and sampling schemes see Sections 3.2.5 and 3.2.6 or the respective publications; for the sampling schemes, the method given in parentheses indicates how the corresponding candidate set is created (e.g., D-opt(coh-opt) stands for D-optimal sampling based on a coherence-optimal candidate set). If the cited paper proposed a new method, this method is marked by a star (*).*

| Ref. | Type | Methods compared | Result |
|---|---|---|---|
| Hampton and Doostan (2015b,a) | sampling | MC, asymptotic*, and coh-opt* | coh-opt best |
| Fajraoui et al. (2017) | sampling | LHS, Sobol, D-opt(LHS), S-opt(LHS)*; sequential sampling* | sequential S-opt(LHS) best; D-opt(LHS) worst |
| Hadigol and Doostan (2018) | sampling (OLS) | MC, LHS, coh-opt, A-opt(coh-opt)*, D-opt(coh-opt)*, E-opt(coh-opt)* | D-opt(coh-opt) best for $p > d$; MC, LHS best for $p < d$ |
| Jakeman et al. (2017) | sampling | MC, asymptotic, CSA* | for $p > d$: CSA better than MC and asymptotic |
| Alemazkoor and Meidani (2018a) | sampling | MC, coh-opt, near-opt(coh-opt)* | near-opt(coh-opt) $\succ$ coh-opt $\succ$ MC |
| Diaz et al. (2018) | sampling | coh-opt, D-opt(coh-opt), sequential D-opt(coh-opt)* | seq. D-opt(coh-opt) $\succ$ D-opt(coh-opt) $\succ$ coh-opt |
| Dutta and Gandomi (2020) | sampling | MC, LHS, Sobol, Importance Sampling | LHS best |
| Hu and Ludkovski (2017) | solvers | OMP, SPGL1, BCS (Babacan et al., 2010) | BCS $\succ$ OMP $\succ$ SPGL1 |
| Huan et al. (2018) | solvers | l1_ls, SpaRSA, CGIST, FPC_AS, ADMM with default parameters | all showed similar performance; ADMM slightly advantageous |
| Liu et al. (2020b) | solvers | OMP, LARS, rPCE* | rPCE $\succ$ LARS $\succ$ OMP |
| Baptista et al. (2019) | solvers | OMP, SPGL1, and two variants* of OMP (modified regressor selection, randomization) | best: OMP, and OMP with modified regressor selection |
| Tarakanov and Elsheikh (2019) | solvers | OMP, LARS, Rank-PCE* | best: Rank-PCE |
| Zhou et al. (2019c) | solvers | LARS, BCS (Wipf and Rao, 2004), BCS (Ji et al., 2008), D-MORPH-reweighted Cheng and Lu (2018b), stepwise regression based on Bayesian ideas* | best: stepwise regression (based on 1 Sobol' design) |

CHAPTER 4

# Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications

This chapter contains the postprint of

> Lüthen, N., Marelli, S., Sudret, B. (2022). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification 12*(3), 49–74.[1]

published in the *International Journal for Uncertainty Quantification*, differing from the published paper only in terms of layout and formatting.

**Author contributions.** **N. Lüthen:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization. **S. Marelli:** Conceptualization, Methodology, Writing - Review & Editing, Supervision. **B. Sudret:** Conceptualization, Writing - Review & Editing, Supervision, Funding Acquisition.

## Abstract

Sparse polynomial chaos expansions (PCE) are an efficient and widely used surrogate modeling method in uncertainty quantification for engineering problems with computationally expensive models. To make use of the available information in the most efficient way, several approaches for so-called basis-adaptive sparse PCE have been proposed to determine the set of polynomial regressors ("basis") for PCE adaptively.

[1]First published in International Journal for Uncertainty Quantification in Volume 12, Number 3, 2022, published by Begell House, Inc. Copyright © 2022 by Begell House, Inc.

The goal of this paper is to help practitioners identify the most suitable methods for constructing a surrogate PCE for their model. We describe three state-of-the-art basis-adaptive approaches from the recent sparse PCE literature and conduct an extensive benchmark in terms of global approximation accuracy on a large set of computational models. Investigating the synergies between sparse regression solvers and basis adaptivity schemes, we find that the choice of the proper solver and basis-adaptive scheme is very important, as it can result in more than one order of magnitude difference in performance. No single method significantly outperforms the others, but dividing the analysis into classes (regarding input dimension and experimental design size), we are able to identify specific sparse solver and basis adaptivity combinations for each class that show comparatively good performance.

To further improve on these findings, we introduce a novel solver and basis adaptivity selection scheme guided by cross-validation error. We demonstrate that this automatic selection procedure provides close-to-optimal results in terms of accuracy, and significantly more robust solutions, while being more general than the case-by-case recommendations obtained by the benchmark.

## 4.1 Introduction

Surrogate modeling techniques are a popular tool in applied sciences and engineering, because they can significantly reduce the computational cost of uncertainty quantification analysis for costly real-world computational models. Here, the computational model is approximated by a cheaper-to-evaluate function, which is created based on a small number of model evaluations, the so-called experimental design. One well-known and popular surrogate modeling technique is polynomial chaos expansion (PCE), which approximates the output of a computational model by a spectral expansion in terms of an orthonormal polynomial basis in the input random variables (Xiu and Karniadakis, 2002). PCE is particularly well suited for surrogating smooth models in low to medium dimension, and for the efficient computation of moments and Sobol' indices (Sudret, 2008; Le Gratiet et al., 2017). Engineering models are often challenging due to their computational cost: complex models often depend on a large number of input parameters, but we can only afford a few tens or hundreds of model evaluations. This is the so-called low-data regime. Sparse PCE techniques, which aim to compute an expansion involving only few terms, have proven especially powerful and cost-efficient for real-world engineering problems such as, among many others, surrogate-assisted robust design optimization (Chatterjee et al., 2019), hybrid simulation for earthquake engineering (Abbiati et al., 2021), dam engineering (Guo et al., 2019; Hariri-Ardebili and Sudret, 2020), and wind turbine design (Slot et al., 2020). Note that real-world applications are typically not exactly sparse; however, sparse regression-based PCE is a useful tool to find good approximations at low computational cost. The aim in this context is not sparsity, but the accurate approximation of the computational model. In particular, we are interested in global approximation accuracy, as measured by the relative mean-squared error. Applications with specific requirements on accuracy, such as optimization or reliability analysis, might rely instead on more specialized surrogate techniques.

In the last decade, a large amount of literature on sparse PCE has been published, proposing methods that make sparse PCE more accurate, efficient and applicable to high-dimensional

problems. However, it is often not obvious how well these methods perform when compared to and combined with each other, especially on real-world engineering problems. In an attempt to fill this gap, the authors recently conducted a literature survey on sparse PCE and a classification of the available methods into a general framework, as well as a benchmark of selected methods on a broad class of computational models (Lüthen et al., 2021). This benchmark extensively compared sparse regression solvers and experimental design sampling schemes, using a fixed polynomial basis to focus on the effect of those two classes of methods. We found that the choice of sparse solver and sampling scheme can make a difference of up to several orders of magnitude in the resulting approximation error, and that different methods are preferable in different regimes of ED size. The performance of solvers and sampling schemes seemed to be mostly independent from one another.

The goal of the present paper is to build on and complement this earlier benchmark by exploring the promising field of *basis-adaptive sparse PCE*, in which the basis is iteratively augmented and refined. So far, novel basis-adaptive methods have been proposed in isolation and not been compared to one another. We want to help practitioners choose the most suitable methods for constructing a PCE surrogate for their applications by answering the following questions: (1) Is there a significant difference between different combinations of sparse solvers and basis adaptivity strategies, and does the proper choice matter in actual applications? (2) In the case that no combination clearly emerges as superior, is there a smart strategy to automatically select a good combination? To answer the first question, we describe three basis-adaptive schemes from the sparse PCE literature in detail, namely, degree and q-norm adaptivity (Blatman and Sudret, 2011; Marelli et al., 2021a), forward neighbor degree adaptivity (Jakeman et al., 2015), and anisotropic degree basis adaptivity (Hampton and Doostan, 2018). We then evaluate the performance and synergies of combinations of these basis-adaptive schemes with several sparse regression solvers in terms of validation error[2] on a library of 11 computational models of varying complexity, representative of a broad class of engineering models. These range from three- to 100-dimensional and include both analytical and numerical models. Since no combination of solver and basis adaptivity significantly outperforms the others, we address the second question by introducing an additional selection step, choosing one among several candidate PCEs computed by different combinations of methods on the same experimental design, using a cross-validation estimate for the generalization error.

The paper is structured as follows. In Section 4.2, we recall the definition of (sparse) PCE and the computing framework introduced in Lüthen et al. (2021). We discuss various estimators for the generalization error, sparse regression solvers, and basis adaptivity, and introduce automatic selection. The associated benchmark results for basis adaptivity and automatic selection are presented in Section 4.3. Finally, a discussion and summary of our results is provided in Section 4.4. Additional information and results can be found in the Appendix.

[2]We focus on global approximation accuracy instead of the computational cost of training, since the main cost in surrogate modeling for engineering applications is typically incurred by the model evaluations themselves. Indeed, the training of each of the surrogates presented in this work requires between a few seconds and a few minutes on a standard business laptop. Even orders of magnitude difference in the training costs of each surrogate do not matter as long as the resulting accuracy is appreciably better.

## 4.2 Sparse polynomial chaos expansions

Let $\boldsymbol{X}$ be a $d$-dimensional random vector with mutually independent components and probability density function $f_{\boldsymbol{X}}$. Denote by $\mathcal{D}_{\boldsymbol{X}}$ the domain of the random vector $\boldsymbol{X}$. Define the space $L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = \{h : \mathcal{D}_{\boldsymbol{X}} \to \mathbb{R} \mid \mathrm{Var}_{\boldsymbol{X}}[h(\boldsymbol{X})] < +\infty\}$ of all scalar valued models with finite variance under $f_{\boldsymbol{X}}$. Under very general conditions on the input distribution $f_{\boldsymbol{X}}$, there exists a polynomial basis $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^d\}$ of $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ (Xiu and Karniadakis, 2002; Ernst et al., 2012). Since the components of $\boldsymbol{X}$ are mutually independent, each polynomial basis function can be built as a product of univariate polynomials in $X_1, \ldots, X_d$ and characterized by the multi-index $\boldsymbol{\alpha} \in \mathbb{N}^d$ whose entries are equal to the degrees of the univariate terms.

For a computational model $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$, let $Y = \mathcal{M}(\boldsymbol{X})$ denote the output random variable. $Y$ can be cast as the following spectral expansion:

$$Y = \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}). \tag{4.1}$$

In practice, a finite, *truncated polynomial chaos expansion*

$$Y = \mathcal{M}(\boldsymbol{X}) \approx \mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \tag{4.2}$$

is computed, where $\mathcal{A} \subset \mathbb{N}^d$ is the truncation set defining the basis elements used in the expansion. The accuracy of the expansion depends on $\mathcal{A}$ and the coefficient values $(c_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathcal{A}} =: \boldsymbol{c} \in \mathbb{R}^P$, with $P = \mathrm{card}(\mathcal{A})$.

Among several methods for computing the coefficient vector $\boldsymbol{c}$ for a given truncation set, *sparse regression* is a particularly powerful and efficient method (Doostan and Owhadi, 2011; Blatman and Sudret, 2011). In this approach, the model is evaluated at a number of points $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\} \subset \mathcal{D}_{\boldsymbol{X}}$ called the *experimental design* (ED), yielding the vector of model responses $\boldsymbol{y} = (\mathcal{M}(\boldsymbol{x}^{(1)}), \ldots, \mathcal{M}(\boldsymbol{x}^{(N)}))^T$. Let $\{\boldsymbol{\alpha}_j\}_{j=1}^P$ be an arbitrary ordering of the multi-indices in the truncation set and define the regression matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ with entries $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{x}^{(i)})$. Sparse regression methods determine a vector $\boldsymbol{c}$ that minimizes the residual norm $\|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2$ under the constraint that it has only few nonzero entries i.e., it is *sparse*. This is usually achieved by regularized regression, resulting e.g. in the LASSO formulation

$$\hat{\boldsymbol{c}} = \min_{\boldsymbol{c} \in \mathbb{R}^P} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{c}\|_1 \leq \tau, \tag{4.3}$$

where $\tau$ is a parameter regulating the sparsity of $\boldsymbol{c}$. A PCE with a sparse coefficient vector $\boldsymbol{c}$ is called *sparse PCE*. Provided that the regression matrix fulfills certain properties (Candès and Wakin, 2008; Bruckstein et al., 2009; Candès and Plan, 2011), sparse regression can find robust solutions to underdetermined systems of linear equations, which means that the experimental design can be smaller than the number of unknown coefficients.

The quality of the solution depends on the choice of the basis $\mathcal{A}$, on the experimental design $\boldsymbol{\mathcal{X}}$, and on the method used for computing the coefficients. For each of these components, many different methods have been proposed in recent years, including iterative methods which adaptively determine $\mathcal{A}$, or the experimental design $\boldsymbol{\mathcal{X}}$. These methods were recently surveyed

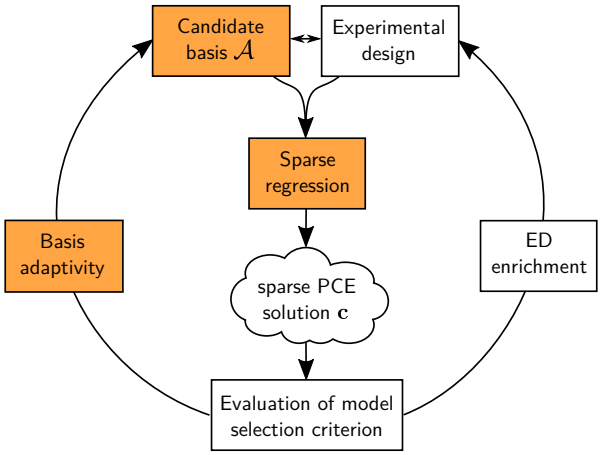



***Figure 4.1:*** *Framework for computing sparse PCE introduced by Lüthen et al. (2021), who conducted a literature survey and a benchmark for the central components "Experimental design" and "Sparse regression" (adapted from Lüthen et al. (2021)). In the present work, we discuss and benchmark the components marked in orange. In particular, we discuss the component "Basis adaptivity" and explore its relationship to sparse regression solvers.*

and classified into the framework shown in Fig. 4.1 (modified from Lüthen et al. (2021)). In the present contribution, we focus on the question of how to determine a suitable basis $\mathcal{A}$. To this end, we benchmark several basis-adaptive approaches and explore their interplay with sparse regression solvers.

### 4.2.1 Error estimation and model selection

Model selection is applied on several levels in the sparse PCE framework:

- Within the sparse solver to select its hyperparameter (see Section 4.2.2)
- Within the basis adaptivity scheme to select a basis (see Section 4.2.3)
- Finally, between solvers and basis adaptivity schemes, to automatically select a combination that is close to best (see Section 4.2.4).

Our main quantity of interest is the generalization error, in the form of the relative mean squared error normalized by the model variance

$$E_{\text{gen}} = \frac{\mathbb{E}_{\boldsymbol{X}}\left[(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}^{\text{PCE}}(\boldsymbol{X}))^2\right]}{\text{Var}_{\boldsymbol{X}}\left[\mathcal{M}(\boldsymbol{X})\right]}. \tag{4.4}$$

It can be approximated by the validation error in the form of the relative mean squared error (MSE)

$$\text{RelMSE} = \frac{\sum_{i=1}^{N_{\text{val}}}\left(\mathcal{M}(\boldsymbol{x}_{\text{val}}^{(i)}) - \mathcal{M}^{\text{PCE}}(\boldsymbol{x}_{\text{val}}^{(i)})\right)^2}{\sum_{i=1}^{N_{\text{val}}}\left(\mathcal{M}(\boldsymbol{x}_{\text{val}}^{(i)}) - \frac{1}{N_{\text{val}}}\sum_{j=1}^{N_{\text{val}}}\mathcal{M}(\boldsymbol{x}_{\text{val}}^{(j)})\right)^2} \tag{4.5}$$

computed on a large validation set $\{\boldsymbol{x}_{\text{val}}^{(i)}\}_{i=1}^{N_{\text{val}}} \sim_{\text{i.i.d.}} f_{\boldsymbol{X}}$. We only consider model selection criteria that are estimates of the generalization error.

To get an accurate estimate of the generalization error, without using an additional validation set, a widely used method is *cross-validation* (Vapnik, 1995). Here, the available data is repeat-

edly divided into two disjoint sets, one for computing the solution (training) and the other for evaluating the error (validation). Aggregating the error estimates from the different splits, we get a cross-validation estimate of the generalization error.

One cross-validation strategy is to divide the data randomly into $k$ disjoint, roughly equally large parts, and use each of the parts in turn as validation set. This is called *$k$-fold cross-validation*. If $k$ is chosen to be equal to the size of the experimental design, the strategy is known as *leave-one-out cross-validation* (LOO). This is closest to using all data points to compute a solution, since in each iteration, only one point is left out from the training set.

In general, LOO can be quite costly, since for an experimental design of size $N$, the method under consideration has to be applied $N$ times. A cheap approximation to the LOO error, which requires only one application of the method, is available for certain sparse regression solvers, namely for those which in their last step recompute the solution with ordinary least-squares (OLS) on the set of regressors with nonzero coefficient (called *active basis*) (Blatman and Sudret, 2010). In particular, this is the case for the solvers hybrid least angle regression (LARS) (Blatman and Sudret, 2011), orthogonal matching pursuit (OMP) (Pati et al., 1993; Jakeman et al., 2015), and subspace pursuit (SP) (Diaz et al., 2018) (see Section 4.2.2). For these solvers, the OLS-based LOO estimate coincides with the true LOO error if the following holds: regardless of which experimental design point is left out, the sparse regression solver consistently yields the same active basis.

Since the repeated use of LOO for model selection often results in the generalization error being underestimated, especially on small experimental designs, Blatman and Sudret (2011) proposed to use a modification factor originally developed for the empirical error (Chapelle et al., 2002), defined by

$$T = \frac{N}{N - P_{\text{active}}} \left(1 + \operatorname{tr}((\boldsymbol{\Psi}_{\text{active}}^T \boldsymbol{\Psi}_{\text{active}})^{-1})\right), \tag{4.6}$$

where $P_{\text{active}}$ denotes the number of nonzero coefficients (the corresponding basis functions are called *active*), and $\boldsymbol{\Psi}_{\text{active}}$ denotes the regression matrix containing only the active regressors. The product of the modification factor $T$ with the LOO error is called *modified LOO error*.

### 4.2.2 Sparse regression solvers

Various sparse regression solvers available for solving sparse PCE were described in detail in Lüthen et al. (2021). We give here only a short overview of the solvers used in our benchmark, and refer to Lüthen et al. (2021) for further details. These solvers are common choices in the sparse PCE literature.

- Hybrid least angle regression (LARS) (Blatman and Sudret, 2011): adding regressors one-by-one, following a least-angle strategy. The hybrid approach recomputes the final coefficient values by ordinary least squares (OLS) on the selected regressors.
- Orthogonal matching pursuit (OMP) (Pati et al., 1993; Jakeman et al., 2015): greedily adding orthogonal regressors one-by-one, computing their coefficients by OLS.
- Subspace pursuit (SP) (Diaz et al., 2018): searching iteratively for a solution with a certain $\ell^0$-norm by adding and removing regressors from the active set. Coefficients are computed

by OLS. We include two variants of SP in this benchmark, one which determines its hyperparameter by 5-fold cross-validation similar to the implementation in (Diaz, 2018), which we denote by $\mathrm{SP}_{k=5}$, and one where it is determined by OLS-based LOO, introduced in (Lüthen et al., 2021) and denoted by $\mathrm{SP}_{\mathrm{LOO}}$.
- Bayesian compressive sensing (BCS) (Babacan et al., 2010): using a Bayesian framework to enforce sparsity of the coefficients.
- SPGL1 (van den Berg and Friedlander, 2008): a convex optimization solver following the Pareto front of the residual-sparsity trade-off.

Each of the solvers features at least one hyperparameter whose value needs to be determined via cross-validation in order to get a good solution. For LARS, OMP, $\mathrm{SP}_{k=5}$, and $\mathrm{SP}_{\mathrm{LOO}}$, this hyperparameter is the number of active basis functions (nonzero coefficients) of the final solution. For BCS and SPGL1, it is the bound on the residual norm in the sparse regression formulation.

The benchmark in Lüthen et al. (2021) of sparse regression solvers on a non-adaptive polynomial basis showed that BCS and $\mathrm{SP}_{\mathrm{LOO}}$ generally outperform other sparse solvers for low-dimensional models, while for high-dimensional models, BCS is by far the best sparse solver.

### 4.2.3 Basis adaptivity

The sparse solver and the experimental design, which were benchmarked in (Lüthen et al., 2021), are not the only ingredients to a sparse PCE. The choice of the *candidate basis*, from which the sparse solver determines the *active basis* (i.e., the set of regressors with nonzero coefficient), is another important ingredient: if the candidate basis is chosen too small, important terms might be missing, which might lead to a large model error. On the other hand, if the candidate basis is too large, the ratio of the number of experimental design points to the number of coefficients is small, which causes some properties of the regression matrix to deteriorate and can result in a large approximation error.

Of course, it is not possible to know a-priori the best choice of the candidate basis. Basis-adaptive schemes start with an initial candidate basis and adapt it iteratively, adding or removing basis functions in each iteration according to various heuristics. The performance of the bases in the different iterations is evaluated using a model selection criterion, typically an estimate of the generalization error.

Several procedures for basis-adaptive sparse PCE have been proposed in the literature. We discuss three approaches in detail, namely

- degree and q-norm ("p&q") basis adaptivity, as implemented in UQLab (Marelli et al., 2021a), see Section 4.2.3.1;
- forward neighbor basis adaptivity (Jakeman et al., 2015), see Section 4.2.3.2; and
- anisotropic degree basis adaptivity (Hampton and Doostan, 2018), from the software BASE_PC (Hampton and Doostan, 2017), see Section 4.2.3.3.

We also briefly mention several other approaches found in the literature.

#### 4.2.3.1 Degree and q-norm ("p&q") adaptivity

A typical choice for a PCE basis is the *basis of total degree p* defined by the set of multi-indices

$$\mathcal{A}^p = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \sum_{i=1}^{d} \alpha_i \leq p\}. \tag{4.7}$$

Furthermore, *hyperbolic truncation* (Blatman and Sudret, 2011) uses the q-(quasi-)norm

$$\|\boldsymbol{x}\|_q = \left(\sum_{i=1}^{d} |x_i|^q\right)^{\frac{1}{q}} \tag{4.8}$$

with $q \in (0, 1]$ to truncate a total-degree basis further:

$$\mathcal{A}^{p,q} = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \|\boldsymbol{\alpha}\|_q \leq p\}. \tag{4.9}$$

Hyperbolic truncation has the effect of excluding some of the basis functions with high degree and high interaction order. This is particularly effective for high-dimensional problems.

A simple basis-adaptive scheme is *degree adaptivity* (Blatman and Sudret, 2011), which computes a number of PCEs on a sequence of total-degree candidate bases of increasing size, and returns the PCE minimizing a certain error estimate as final solution. Analogously, a q-norm adaptive scheme can be developed, and easily be combined with degree adaptivity (Marelli et al., 2021a), yielding *degree and q-norm (p&q) adaptivity*. Degree and q-norm adaptivity is solution-agnostic, i.e., it does not take the solution computed in the previous iteration into account.

#### 4.2.3.2 Forward neighbor basis adaptivity

Jakeman et al. (2015) suggest a basis-adaptive algorithm based on a graph view of the PCE regressors (see also Gerstner and Griebel (2003); Narayan et al. (2014)). Since the input random vector is assumed to have independent components, the basis functions have tensor product structure. The basis functions can be considered nodes of a directed graph constructed as follows (see also Fig. 4.2): two regressors are considered neighbors if their multi-index of degrees differs only in one dimension by 1, i.e., there is a directed edge from $\psi_{\boldsymbol{\alpha}}$ to $\psi_{\boldsymbol{\beta}}$ iff $\boldsymbol{\gamma} := \boldsymbol{\beta} - \boldsymbol{\alpha}$ is a multi-index with $\gamma_i = 1$ for one $i \in \{1, \ldots, d\}$ and $\gamma_j = 0, j \neq i$. *Forward neighbors* of a regressor are regressors reachable by an outgoing edge, and *backwards neighbors* are regressors connected by an incoming edge.

In the context of basis-adaptive PCE, this construction is used to determine a number of candidate regressors to be added to the currently active basis, starting from an initial basis of small total degree. Assume that an important high-degree or high-order regressor is not yet part of the candidate basis but some of its backwards neighbors are. The fact that it is missing should be visible in the coefficients of its backwards neighbors, which can be expected to have a significant nonzero coefficient to compensate for the fact that the important high-degree regressor is not yet part of the basis.

This heuristic is the foundation of the algorithm whose pseudocode is summarized in Algorithm 6. In each iteration, the current set of active basis functions is determined (restriction

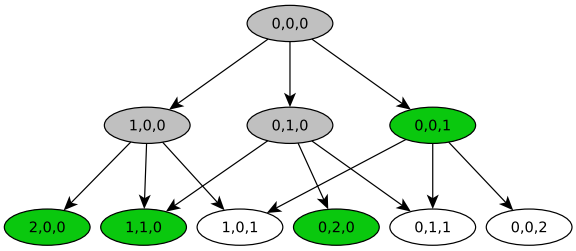



***Figure 4.2:*** *Illustration of forward neighbor basis adaptivity (Jakeman et al., 2015) in $d = 3$ dimensions. The regressors are seen as nodes in a graph, where a directed edge connects regressors whose multi-index differs by 1 in exactly one dimension. Assume that the set of active regressors $\mathcal{A}^{\text{active}}$ initially consists of the gray nodes. All admissible nodes are highlighted in green. Together with the gray nodes, they constitute the set $\mathcal{A}^{(1)}$ and would be added to the model in the next expansion step of forward neighbor basis adaptivity (Step 3 in Algorithm 6).*

step). All forward neighbors of these active basis functions are surveyed and added to the candidate basis if they are *admissible*, i.e., if all of their backwards neighbors are in the active set (expansion step). Jakeman et al. (2015) employ several expansion steps to prevent premature termination, and use cross-validation error estimates. We call this algorithm *forward neighbor basis-adaptive*.

**Algorithm 6** Forward neighbor basis adaptivity (Jakeman et al., 2015)

1: Initial PCE (basis chosen a-priori, typically small total-degree basis)
2: Restriction: retain only the active regressors $\mathcal{A}^{\text{active}}$
3: Expansion: let $\mathcal{A}^{(0)} = \mathcal{A}^{\text{active}}$. For $t = 1, \ldots, T$, obtain the set $\mathcal{A}^{(t)}$ by augmenting $\mathcal{A}^{(t-1)}$ by all its admissible forward neighbors.
4: Compute a PCE and its error estimate for each augmented basis $\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(T)}$.
5: Choose the PCE with the lowest error estimate among the $T$ candidates. Stop if this error estimate is larger than the previously obtained best error estimate. Else, continue iteration with restriction step (Step 2)

This algorithm is implemented in the software Dakota (Adams et al., 2014). We use our own Matlab-based implementation of the algorithm. Consistently with Jakeman et al. (2015), we set $T = 3$.

#### 4.2.3.3 Anisotropic degree basis adaptivity

Hampton and Doostan (2018) propose an algorithm called BASE-PC, which combines a basis-adaptive scheme with sequential enrichment of a coherence-optimal experimental design. The two steps are executed alternatingly: based on the current ED, a suitable basis is chosen; and according to the currently active basis functions, the ED is enriched and the weights are updated.

In the present work, we solely consider the basis adaptivity part of the BASE-PC algorithm. The main idea of this algorithm is to use an *anisotropic degree basis*, defined by a *degree vector* $\boldsymbol{p} \in \mathbb{N}^d$. A related idea was explored earlier by Blatman and Sudret (2009). Similarly to a

total-order basis, an anisotropic degree basis is defined by

$$\mathcal{A}^{\boldsymbol{p}} = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \sum_{i=1}^{d} \frac{\alpha_i}{p_i} \leq 1\}. \tag{4.10}$$

If all entries of $\boldsymbol{p}$ are the same, i.e., $p_1 = p_2 = \ldots = p_d = p$, this definition reduces to a total-order basis of degree $p$. The equation $\sum_{i=1}^{d} \frac{\alpha_i}{p_i} = 1$ defines a hyperplane that cuts the $i$-th coordinate axis at $\alpha_i = p_i$.

In each iteration, the algorithm determines the current anisotropic degree vector based on the currently active basis. A new, larger candidate basis is then constructed by considering the anisotropic degree basis corresponding to a uniformly increased anisotropic degree vector.

We use our own, slightly modified implementation of the basis adaptivity part of BASE-PC, as summarized in Algorithm 7. The most costly operation is the computation of the anisotropic-degree basis (line 6). Hampton and Doostan (2018) developed a specialized efficient algorithm to generate the multi-indices of an anisotropic degree basis, which we utilize in our implementation. We call Algorithm 7 *anisotropic degree basis-adaptive*.

**Algorithm 7** Anisotropic degree basis adaptivity (Hampton and Doostan, 2018)

1: Initial PCE (fixed basis of low order)
2: **for** $o = 1, \ldots, 10$ **do** ▷ outer loop
3: Restriction: denote by $\mathcal{A}^{\text{active}}$ the set of active regressors of the last selected PCE
4: **for** $i = 1, \ldots, 10$ **do** ▷ inner loop
5: Additional restriction: remove $\frac{i-1}{10}$ regressors from the set $\mathcal{A}^{\text{active}}$ starting with the ones with smallest-in-magnitude coefficients, obtaining a set $\mathcal{A}^i \subseteq \mathcal{A}^{\text{active}}$
6: Expansion: compute the dimension-wise maximal degree of the regressors in $\mathcal{A}^i$, denoted by degree vector $\boldsymbol{p}^{\max}$. Compute $\boldsymbol{p}^{\text{new}} = (p_1^{\max} + 1, p_2^{\max} + 1, \ldots, p_d^{\max} + 1)$ and the associated anisotropic-degree basis $\mathcal{A}^{i,\text{new}}$
7: Compute a PCE on the basis $\mathcal{A}^{i,\text{new}}$ and its error estimator $e^i$
8: If $e^i \geq e^{i-1}$, increase the so-called *strike counter* by 1. Break from the inner loop if the strike counter is $\geq 3$
9: **end for**
10: From the 10 candidate PCEs, select the PCE with the lowest error estimate
11: Break from the outer loop if the error estimate has increased three times in subsequent iterations (ExpansionEarlyStop; same idea as inner loop strike counter)
12: **end for**
13: Return the PCE with the lowest error estimate among all PCEs selected in the outer loop.

#### 4.2.3.4 Other basis adaptivity algorithms

We briefly summarize other approaches for basis adaptivity for sparse PCE. These approaches will not be investigated in our benchmark.

Many algorithms which use stepwise regression to build up a sparse basis regressor-by-regressor can be classified both as sparse regression solvers (as done in Lüthen et al. (2021)) and as basis-adaptive algorithms. As an example, the approach by Blatman and Sudret (2010) adds and

removes regressors one-by-one based on the induced change in LOO error, thereby building up a small set of active basis functions which is sparse in a larger total-degree basis. In this case, the candidate basis coincides with the active basis. Mai and Sudret (2015) use the "principle of heredity" together with LARS to determine additional bivariate interaction terms to be added to the model, once a univariate term is identified as relevant. We do not consider these approaches, since we are interested in algorithms modifying the candidate basis not only one regressor at a time, but at a larger scale.

Alemazkoor and Meidani (2017) propose a basis-adaptive algorithm relying on sparsity and step-by-step augmentation of the basis. In each step, the candidate basis is a total-order basis for a subset of input random variables, while the remaining input random variables are considered constant. The initial basis consists only of the constant term. In each iteration, either one of the constant dimensions is activated, or the total degree of the candidate basis is increased by one (active dimensions only), depending on the resulting error estimate or sparsity of the solution. The coefficients are solved by OLS until a pre-defined threshold of residual norm is reached. Then, the sparse regression solver SPGL1 is used, which identifies the sparsest solution whose residual norm is smaller than the given threshold. We do not consider this method due to its high computational cost and its strong tie to the solver SPGL1, making it less effective when paired with other sparse regression solvers.

Loukrezis et al. (2019) propose a basis-adaptive algorithm for interpolating PCE on Leja-based sparse grids. Their algorithm is a more cautious version of forward neighbor basis adaptivity (Section 4.2.3.2): after adding all admissible forward neighbors to the basis and computing their coefficients, all of the recently added regressors are removed again, except for the one that achieved the largest-in-magnitude coefficient. Thapa et al. (2020) suggest a basis-adaptive procedure that relies on total-order bases of increasing degree. In contrast with degree adaptivity (Section 4.2.3.1), the basis functions of degree $p+1$ are not added all at once, but in chunks of a certain size dependent on the dimension and the degree $p$. After adding a chunk of basis functions, the PCE is recomputed and regressors with a coefficient smaller than a certain threshold are removed from the basis. We do not consider these two approaches because they are similar to the previously presented approaches while being more costly.

### 4.2.4 Automatic selection of a sparse PCE solution from several candidate solutions

For realistic simulators used in engineering and applied sciences, evaluating the computational model is the expensive part of the surrogate modeling process. Once the model evaluations are obtained, all further computations are post-processing steps, and are computationally cheap in comparison to the model evaluations. Thus, it is feasible to apply several adaptive sparse PCE methods and choose the best one.

We therefore propose the use of an additional layer of selection which we call here *automatic selection*. It can be seen as a simple case of *ensemble modeling* (Sagi and Rokach, 2018) where a single metamodel ("strong learner") is chosen based on its cross-validation performance – or, in other words, model selection. For a given experimental design, we compute several sparse

PCEs through different combinations of sparse solvers and basis adaptivity schemes. From the resulting PCE solutions, we choose one based on the value of a suitable estimate of generalization error as model selection criterion. There are several possibilities. One possible choice is the estimator already used for selecting the hyperparameter of the solver and for basis adaptivity, i.e., (modified) LOO for LARS, OMP and $\text{SP}_{\text{LOO}}$, and $k$-fold cross-validation for $\text{SP}_{k=5}$, BCS and SPGL1. However, this estimator might not be consistent between solvers, which is however necessary for such an automatic selection. Furthermore, this estimate might be biased due to its repeated use.

A second class of estimators is given by the so-called *hybrid cross-validation error estimators*. The word *hybrid* is a reference to Efron et al. (2004), who created the hybrid version of least-angle regression (LARS) which uses LARS only to identify the set of active basis functions, and then recomputes the coefficients by ordinary least-squares (OLS) on this active set. To compute the hybrid leave-one-out (LOO) or hybrid $k$-fold cross-validation error estimate for any PCE solution, the same procedure is used: first, the active basis functions are identified using the whole experimental design. Then, the coefficients are recomputed several times using OLS, following the chosen cross-validation framework. This requires solving a linear system of equations $k$ times in the case of $k$-fold cross-validation. In case of LOO, only one linear system of equations needs to be solved (Blatman and Sudret, 2010). Furthermore, we can make use of the modification factor of Eq. (4.6) to compute the hybrid modified LOO error estimate.

As baseline cases, we will also select a solution 1) randomly from a set of generally well-performing methods, and 2) corresponding to the best combination identified in the benchmark of basis adaptivity schemes (Section 4.3.1).

## 4.3 Numerical results

In this benchmark, we compare the performance of various combinations of sparse regression solvers with basis-adaptive schemes. We use the following methods and associated implementations. The sparse solvers (wrapped to fit into our benchmark framework) are:

- Hybrid-LARS: UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- OMP: UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- $\text{SP}_{k=5}$: own implementation of Diaz et al. (2018), available in UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- $\text{SP}_{\text{LOO}}$: own adaptation of Diaz et al. (2018), available in UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- BCS: own implementation of FastLaplace (Babacan et al., 2010), available in UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- SPGL1: SPGL1 v1.9[3] (van den Berg and Friedlander, 2008; Van den Berg and Friedlander, 2015)

The basis adaptivity schemes are:

[3]Our benchmark is performed with SPGL1 v1.9. A new version of SPGL1, v2.1, is available since June 2020. In our tests, the new version (with "hybrid" mode) did not perform significantly better than the old version. The numerical results show therefore results for SPGL1 v1.9 with default parameters.

- p&q adaptivity: UQLab (Marelli et al., 2021a; Marelli and Sudret, 2014)
- forward neighbor basis adaptivity: own implementation of the algorithm based on the description in Jakeman et al. (2015)
- anisotropic degree basis adaptivity: own implementation of an algorithm adapted from the basis-adaptive part of BASE-PC (v1) (Hampton and Doostan, 2018), using the function `basis_anisotropic_total_order.m` from Hampton and Doostan (2017) to generate the multi-indices of an anisotropic degree basis (see Section 4.2.3.3)

To reduce the complexity of our benchmark, we choose Latin hypercube sampling with maximin distance optimization to generate the experimental design (ED) since Lüthen et al. (2021) demonstrated that the choices of sparse solver and sampling scheme are mostly independent from each other.

We use the following model selection methods:

- For the selection of the hyperparameters of the sparse regression solvers, we use
  - modified OLS-based LOO for the solvers LARS, OMP and $\text{SP}_{\text{LOO}}$
  - $k$-fold CV for the solvers $\text{SP}_{k=5}$ ($k = 5$), BCS ($k = 10$) and SPGL1 ($k = 10$)
- The basis adaptivity schemes use the same criterion as the respective solver uses.
- We investigate in Section 4.3.2 which criterion is suited best for the final model selection.

I.e., instead of prescribing fixed values for the hyperparameters, we let the parameters be determined adaptively. In this sense, we give each method equal opportunity to produce the best possible solution (assuming that the CV error is a good proxy for solution quality).

For our benchmark, we use 11 benchmark models ranging from low-dimensional analytical models to high-dimensional differential equations. All of these models have previously been used as numerical examples for surrogate modeling or reliability analysis. None of these models has an exactly sparse representation. Note that we do not aim at benchmarking the ability of the methods to recover "true" sparse solutions, but instead their approximation capabilities on engineering models, which are typically not exactly sparse but *compressible*, meaning that the magnitude of their PC coefficients decays rapidly. Table 4.1 provides an overview of the benchmark models. For a more detailed presentation, we refer the interested reader to the respective publications (see column "Reference" of Table 4.1).

### 4.3.1 Basis adaptivity

We benchmark the sparse solvers LARS, OMP, $\text{SP}_{k=5}$, $\text{SP}_{\text{LOO}}$, BCS, and SPGL1 combined with the basis-adaptive schemes described in Section 4.2.3:

1. degree and q-norm (p&q) adaptivity (abbreviation: PQ)
2. forward neighbor basis adaptivity (FN)
3. anisotropic degree basis adaptivity (AD)

As a base case, we include a *static basis* following the rule $P \approx \frac{10}{3}N$ (abbreviation: ST), where we use hyperbolic truncation with $q = 0.5$ for high-dimensional models ($d \geq 20$).

[4] https://www.uqlab.com/sensitivity-high-dimension

**Table 4.1:** *Overview of the 11 computational models used in our benchmark. Finite element models are marked in italic font, all other models are analytical. The column "Reference" provides the literature in which the models and their input are described in detail. The column "ED sizes" contains the two sizes of experimental design (small and large) used in the basis adaptivity benchmark.*

| Model | Dimension | Distributions | Reference | ED sizes (small, large) |
|---|---|---|---|---|
| Ishigami function | 3 | uniform | Blatman and Sudret (2011) | 50, 150 |
| Undamped oscillator | 6 | Gaussian | Echard et al. (2013) | 60, 120 |
| Borehole function | 8 | Gaussian, lognormal, uniform | Morris et al. (1993) | 100, 250 |
| Damped oscillator | 8 | lognormal | Dubourg (2011) | 150, 350 |
| Wingweight function | 10 | uniform | Forrester et al. (2008) | 100, 250 |
| *Truss model* | 10 | lognormal, Gumbel | Blatman and Sudret (2011) | 100, 250 |
| Morris function | 20 | uniform | Blatman and Sudret (2010b) | 150, 350 |
| *Structural frame model* | 21 | lognormal, Gaussian; dependent input variables | Blatman and Sudret (2010) | 150, 350 |
| *2-dim diffusion model* | 53 | Gaussian | Konakli and Sudret (2016) | 100, 400 |
| *1-dim diffusion model* | 62 | Gaussian | Fajraoui et al. (2017) | 100, 400 |
| 100D function | 100 | uniform | UQLab example[4] | 400, 1200 |

The benchmark is performed on all 11 models presented in Table 4.1. The experimental design (ED) is created by Latin hypercube sampling (LHS) with optimized maximin distance. We investigate one "small" and one "large" ED size per model (see last column of Table 4.1), which correspond to the second-smallest and second-largest experimental design size, respectively, of an earlier benchmark (Lüthen et al., 2021) dedicated to investigating solvers and sampling schemes on a static basis. The small ED size is chosen to be at the lower end of the range of reasonable ED sizes. The large ED size represents the "highly informative" regime (in the engineering sense), which for costly engineering models is in the order of a few hundred model evaluations. Here we restrict ourselves to two ED sizes to control the complexity of the results; however, two ED sizes are needed since the earlier benchmark showed that the solver behavior is different in the two regimes. Since almost no engineering model is exactly sparse in the PCE basis, adaptivity can be expected to help identify relevant higher-order terms in all regimes of ED size – as soon as the ED is large enough to contain some information about the model. For each model and experimental design size, we repeat the analysis $R = 30$ times to account

for the stochasticity of the sampling method. Due to their excessive computational cost[5], we omit SPGL1 and anisotropic degree basis adaptivity from the benchmark for high-dimensional models ($d \geq 20$). More details on the settings for the basis adaptivity schemes (e.g., initial basis and investigated degree ranges) can be found in 4.A.

#### 4.3.1.1 Boxplots of results for the Ishigami function

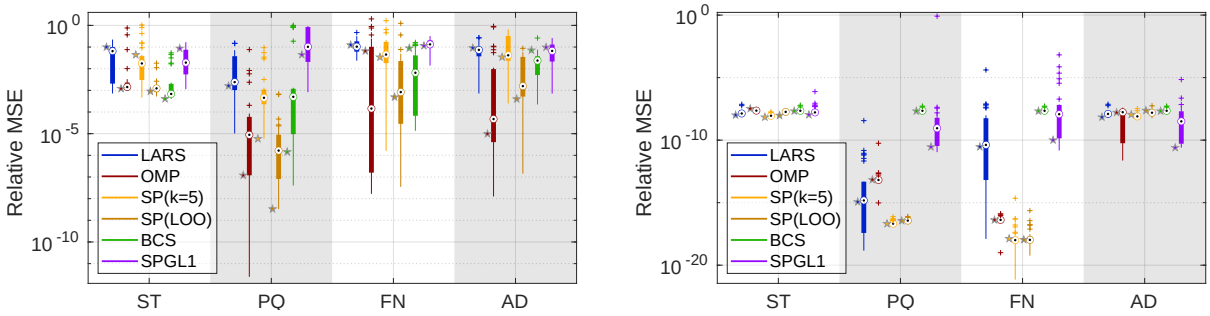
The results from this benchmark for the Ishigami function are displayed in Fig. 4.1 in the form of boxplots. Results for the remaining models are shown in 4.B.1. The boxplots visualize the results for all combinations on 30 independent ED realizations. The star-shaped markers denote for one (arbitrarily selected) ED realization the respective validation errors of each of the combinations, highlighting one set of data points which is also contained in the boxplots. We observe that the ranking based on statistics (e.g., median as denoted by the white circle) and the ranking based on the actually attained error on a specific ED can be quite different (see e.g. $\text{SP}_{k=5}$ & FN vs. BCS & FN in Fig. 4.1a). From the star-shaped markers, it is obvious that the solvers and basis-adaptive schemes do not exploit the available information in the same ways: while some combinations show their best performance on the selected ED, others perform average or worse on the same ED. We want to find the "best" method, i.e., a solver-basis adaptivity combination that consistently, on each different model and ED realization, achieves a close-to-optimal error. In other words, in this comparative study we are less interested in the absolute values of the error (since we assume that the tested methods have been validated before in the associated literature), but rather in the *relative performance* of the methods. The boxplots alone do not give the full picture, since they do not show which errors correspond to the same ED realization. Also, looking at the results for this and other benchmark models in 4.B.1, it is difficult to visually extract information about the overall performance of the methods.

#### 4.3.1.2 Bar plots – results aggregated over models and replications, separately per model dimensionality and ED size

Therefore, we propose an aggregated visualization of the results as displayed in Fig. 4.2 and described in the following. For every model and ED size, we determine on each unique ED realization which combination attained the smallest error $\epsilon^*$ (for example, for the ED realization visualized in Fig. 4.1a by the star-shaped markers, $\text{SP}_{\text{LOO}}$ & PQ attains rank 1). We also determine *on the same ED* for each combination whether it came within a factor of 2, 5, or 10 of $\epsilon^*$. For example, for the ED realization visualized in Fig. 4.1a by the star-shaped markers, the best error is attained by $\text{SP}_{\text{LOO}}$ & PQ. All other combinations achieve an error that is more than a magnitude larger, and would therefore not get a count for this ED realization.

We then aggregate the counts over all repetitions and all models in four different classes: low dimensionality ($d \leq 10$) and small experimental design, low dimensionality and large experi-

[5] Anisotropic degree basis adaptivity increases the degree of the basis uniformly by 1 in each dimension, which for high-dimensional models often results in infeasibly large bases. SPGL1, or the implementation we are using, takes considerably longer than the other solvers for an increasing number of basis elements $P$. Since our benchmark setup requires several thousand runs of the sparse solver for the case of high-dimensional models, the computational cost of the full benchmark is infeasible given our computational resources.




*(a) Ishigami function, $N = 50$* *(b) Ishigami function, $N = 150$*

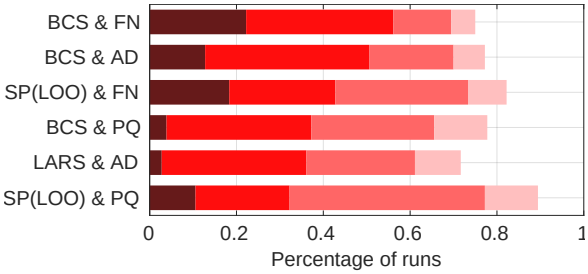
***Figure 4.1:*** *Comparison of all possible combinations of solvers and basis adaptivity schemes for the Ishigami function. Abbreviations of basis adaptivity schemes: ST – static basis, PQ – degree- and q-norm, FN – forward neighbor, AD – anisotropic degree. We show validation errors attained by all 24 combinations of methods on 30 realizations of experimental design. The boxplots visualize the attained errors across all 30 realizations. The star-shaped markers denote for one arbitrarily selected ED realization the errors that the different combinations attain (highlighting one set of data points which is also contained in the boxplots). Left: small ED; right: large ED.*

mental design, and their corresponding high dimensionality counterparts ($d \geq 20$). In other words, for each of the four classes we count how often each combination achieved the best error or came within a factor of 2, 5, or 10 of the best error, for each unique experimental design. These counts, scaled as percentages of all runs[6], are then visualized in the form of overlapping horizontal bars in Figs. 4.2a–4.2e as follows: The dark red bar shows the percentage of runs in which the particular combination of sparse solver and basis-adaptive scheme found the smallest relative MSE $\epsilon^*$. The other three bars in hues of red illustrate how often the respective combination was within a factor of 2, 5, or 10 of $\epsilon^*$ *on the same ED*.

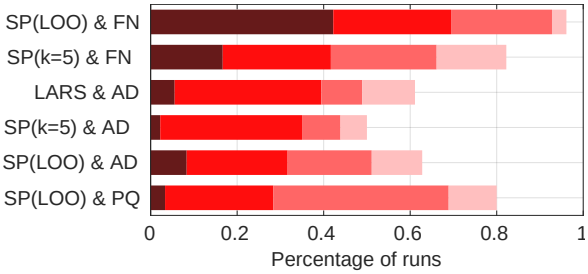
These bars are therefore a measure of the dispersion of the accuracy of each method in terms of distance to the best attainable accuracy. This measure is more interesting than the absolute scatter of a method, since a large variability does not matter as long as the method always provides a close-to-optimal solution; and small variability is of no value if the method is always orders of magnitude away from the best possible solution. We only show the six combinations of solver and basis adaptivity scheme whose relative MSE was most often within two times the best relative MSE (denoted by the bright red bar), and sort the combinations according to this criterion. (For the full list of combinations, see Figs. 4.B.3–4.B.6 in the appendix.)

Comparing the subplots 4.2a–4.2d, we observe that the results for small and large ED sizes and low- and high-dimensional models are indeed quite different, which justifies analyzing the results separately. As detailed in the previous paragraph, the plots show both which combinations of solver and basis adaptivity scheme attain the smallest relative MSE how often (length of the dark red bar), and how robust the combination is (lengths of the other three bars), i.e., how often the solution was within a small factor of the best solution. We observe that depending on the criterion considered to assess the combinations of solver and basis adaptivity scheme, different combinations turn out to perform best. In the following, we summarize the performance of solver-

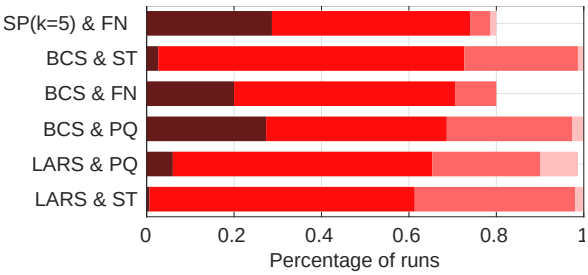
[6]There are $6 \times 30$ realizations for the two low-dimensional cases (Figs. 4.2a and 4.2b), and $5 \times 30$ realizations for the two high-dimensional cases (Figs. 4.2c and 4.2d). Fig. 4.2e aggregates over all 660 ED realizations.



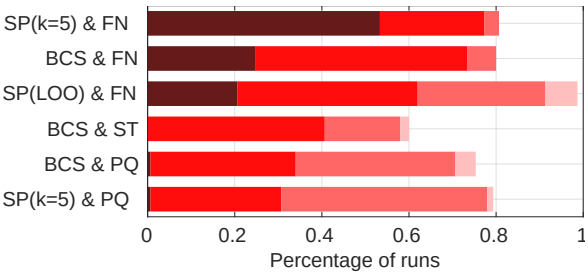

*(a) low-dimensional models, small ED ($6 \times 30$ runs)*



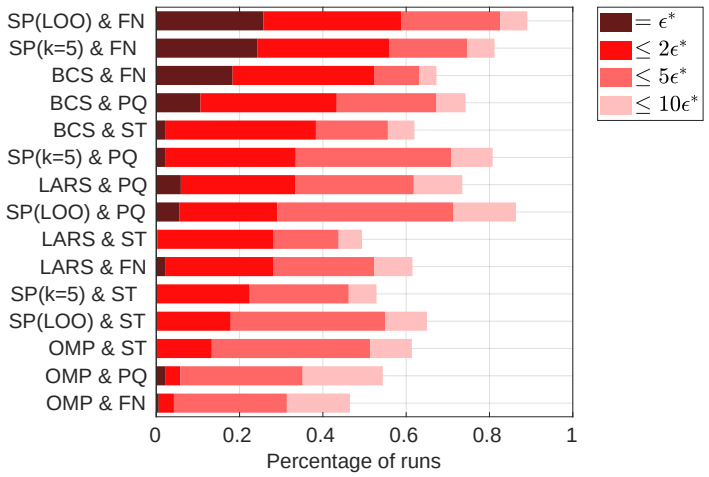

*(b) low-dimensional models, large ED ($6 \times 30$ runs)*




*(c) high-dimensional models, small ED ($5 \times 30$ runs)*




*(d) high-dimensional models, large ED ($5 \times 30$ runs)*




*(e) Aggregated results for all 11 models and both ED sizes ($11 \times 2 \times 30$ runs)*

***Figure 4.2:*** *Aggregated results of our comparative study. For each of the 11 models, 2 ED sizes, and 30 replications (i.e., ED realizations), we have run each of the combinations of sparse solver and basis adaptivity scheme and computed the corresponding relative mean-squared errors (MSE) (see Fig. 4.1 for results for the Ishigami function). We determine the relative performance of the method combinations separately on each ED realization: denoting the best attained relative MSE on a specific ED by $\epsilon^*$, we record which of the method combinations reached this error or came within a factor of $\{2, 5, 10\}$ of $\epsilon^*$ on the same ED.*
*We then aggregate the counts over all replications and models, separating by low/high dimensionality and small/large ED size (panels 4.2a–4.2d). This results in $6 \times 30$ runs for the low-dimensional cases, and $5 \times 30$ runs for the high-dimensional cases (per method combination). For readability, in panels 4.2a–4.2d only the six best combinations are shown (for the full results, see 4.B.2). The results are displayed in the form of overlapping bars, representing percentages. The dark red bar represents the percentage of runs where the method combination reached the best relative MSE on the specific ED realization. The other bars in shades of red denote the percentages of runs where the error of the particular method combination was within a factor of $\{2, 5, 10\}$ of the best error on the same ED realization. The method combinations are sorted by the length of the bright red bar (number of runs with an error $\leq 2\epsilon^*$).*
*In panel 4.2e, we display the relative performance of the method combinations aggregated over all models, ED sizes and replications (i.e., $11 \times 2 \times 30$ runs).*

basis adaptivity combinations with three numbers $X$-$Y$-$Z$, where $X$ denotes the percentage of runs where the respective combination turned out best (i.e., the length of the dark red bar), $Y$ denotes the percentage of runs that were within a factor of two of the smallest error on that ED (length of the bright red bar), and $Z$ denotes the percentage of runs that were within 1 order of magnitude of the smallest error (light red bar). The numbers are rounded to integer values. Italic numbers indicate that this is the best value achieved among all combinations. The enumeration tags refer to the panels in Figure 4.2.

(4.2a) *Low-dimensional models, small ED*: BCS together with forward neighbor basis adaptivity (*22*-*56*-75) achieves the smallest error more often than any other combination. However, this combination is not the most robust: in $100 - 75 = 25\%$ of runs, its error is more than one magnitude worse than the best error. $\mathrm{SP}_{\mathrm{LOO}}$ together with p&q adaptivity (11-32-*89*) is the most robust instead.

(4.2b) *Low-dimensional models, large ED*: by far the best combination in all three categories is $\mathrm{SP}_{\mathrm{LOO}}$ together with forward neighbor basis adaptivity (*42*-*69*-*96*).

(4.2c) *High-dimensional models, small ED*: there are two combinations that outperform the others: $\mathrm{SP}_{k=5}$ with forward neighbor basis adaptivity (*29*-*74*-80) and BCS with a static basis (3-73-*100*). A close third is BCS with p&q adaptivity (27-69-*100*).

(4.2d) *High-dimensional models, large ED*: the two best combinations are $\mathrm{SP}_{k=5}$ with forward neighbor basis adaptivity (*53*-*77*-81) and $\mathrm{SP}_{\mathrm{LOO}}$ with forward neighbor basis adaptivity (21-62-*99*).

We observe from this enumeration that only the solvers BCS, $\mathrm{SP}_{k=5}$ and $\mathrm{SP}_{\mathrm{LOO}}$ are found among the best combinations. Considering the best six combinations (based on the second number $Y$) as in Fig. 4.2, only LARS is joining the list. OMP does not perform well in any of the cases (see also Appendix, Figs. 4.B.3–4.B.6). This is likely due to its tendency to severely underestimate the generalization error, which might make the comparison between error estimates of different bases unreliable. Likewise, SPGL1 is never among the best six combinations for low-dimensional models.

Regarding basis adaptivity schemes, the static basis is not among the best six combinations (based on the second number $Y$) for low-dimensional models. However for high-dimensional models, especially for the small ED case, it is among the six best combinations several times and performs well.

Note that in the high-dimensional, small ED case (Figs. 4.2c and 4.B.4), the static basis shows the most robust behavior (in terms of the third number $Z$) while any solver with forward neighbor basis adaptivity is off by more than a magnitude in at least 20% of runs. This is due to its bad performance for the 100D function (see also Fig. 4.B.2u). The 100D function is also responsible for the peculiar results of the high-dimensional, large ED case (Figs. 4.2d and 4.B.6): only $\mathrm{SP}_{\mathrm{LOO}}$ together with forward neighbor basis adaptivity is able to find a very accurate solution, reaching a validation error of $10^{-10}$ which is several orders of magnitude smaller than what the other combinations reach. However, several other combinations are within one order of magnitude of the smallest error for all other high-dimensional models, reaching overall 80% for $Z$, as can be seen from Fig. 4.B.6.

#### 4.3.1.3 Bar plots – results aggregated over all models, ED sizes, and replications

To get a complete picture of solver and basis adaptivity scheme performance, we display in Fig. 4.2e the relative performance of all combinations of solvers and basis adaptivity schemes, averaged over all models and experimental designs regardless of their dimensionality and size. We exclude combinations involving SPGL1 and anisotropic degree basis adaptivity (AD), which were only tested for low-dimensional models. Therefore, Fig. 4.2e contains $3 \times 5 = 15$ combinations. We see that $\text{SP}_{\text{LOO}}$ together with forward neighbor basis adaptivity (*26-59-89*) performs best in all three categories. $\text{SP}_{k=5}$ together with forward neighbor basis adaptivity (24-56-81) is on the second place in terms of the first two criteria ($X$ and $Y$), while $\text{SP}_{\text{LOO}}$ together with p&q basis adaptivity (6-29-86) is the second-best solver in terms of achieving an error within one order of magnitude of the best error most often ($Z$). BCS together with any basis adaptivity scheme performs well, while all combinations involving OMP are on the bottom of the list. Combinations with a static basis are found more towards the end of the list, and those with forward neighbor basis adaptivity are found more towards the beginning of the list.

#### 4.3.1.4 Conclusion of the comparative study

From these plots, we see clearly that there is no single combination of sparse solver and basis adaptivity scheme that always outperforms all others. While $\text{SP}_{\text{LOO}}$ together with forward neighbor basis adaptivity shows superior performance when averaged over all models and ED sizes, we identify better-performing combinations when we differentiate by model dimension and ED size. In some cases, we are faced with a trade-off between accuracy and robustness: e.g., for low-dimensional models and small EDs, the choice of BCS & FN has a higher probability of yielding a near-optimal solution (within a factor of 2 to the best), while the choice of $\text{SP}_{\text{LOO}}$ & PQ has a higher probability of not being more than a magnitude off from the optimal one.

It seems to be easier to identify trends regarding solvers than regarding basis adaptivity schemes. For example, BCS performs well when information is scarce (small ED sizes or high-dimensional models) while $\text{SP}_{\text{LOO}}$ performs well for large ED sizes, as already observed in (Lüthen et al., 2021), and OMP generally underperforms. This might be because sparse regression solvers are based on quite different principles, from gradient descent over greedy stepwise regression to Bayesian reasoning, while the basis adaptivity schemes all work with variations of the same concept – namely, gradually increasing the degree of the basis. Basis-adaptive schemes generally outperform the static basis, most likely because they offer more basis elements to choose from. However, it seems to not matter as much how the additional basis elements are generated, since none of the three basis-adaptive methods always finds the best basis. Note that the selection of the bases is guided by the cross-validation error. If the cross-validation error does not correlate well with the real validation error, the scheme will select a suboptimal basis. OMP generally has an unreliable cross-validation error, which might explain its bad performance with basis-adaptive schemes.

### 4.3.2 Automatic selection of sparse solver and basis adaptivity scheme

As we saw in the previous section, there is no single best-performing combination of sparse solver and basis adaptivity scheme. In this section, we investigate the question: is there any criterion which can help us choose the best combination of solver and basis adaptivity scheme for a given problem? One such criterion could be a deterministic rule based on dimensionality of the model and ED size, relying on the results from the previous section. Another option could be to use cross-validation-based model selection criteria as described in Section 4.2.4 (see also Section 4.2.1). We call this process of automatically choosing a well-performing combination from a number of candidate combinations using a model selection criterion *automatic selection*.

Due to the performance of the methods in the benchmark in the previous sections, we restrict our investigation to the solvers $\mathrm{SP}_{k=5}$, $\mathrm{SP}_{\mathrm{LOO}}$ and BCS, and to the basis adaptivity schemes

- p&q adaptivity (PQ), forward neighbor adaptivity (FN), and anisotropic degree adaptivity (AD) for low-dimensional models
- static basis (static), p&q adaptivity (PQ), and forward neighbor adaptivity (FN) for high-dimensional models

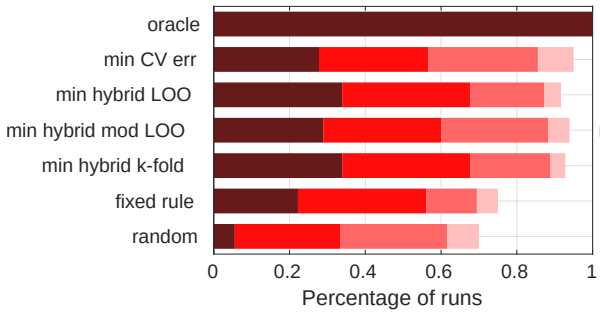
resulting in 9 possible solutions for each model, ED size and repetition.

We use the following model selection criteria (see also Section 4.2.4):

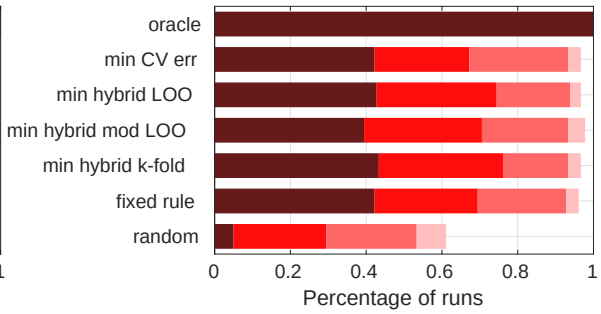
1. the oracle solution, i.e, the smallest relative MSE attained among *all* 24 or 15 combinations of methods (for low-dimensional or high-dimensional models, respectively) – i.e., among all methods tested in Section 4.3.1, not only among the 9 candidate solutions considered here – on each ED realization, as an ideal lower bound. Of course, this information is not available in practice.
2. the criterion used for basis and hyperparameter selection by the respective solver (see Section 4.2.1)
3. hybrid LOO, computed by OLS on the active basis functions only
4. hybrid modified LOO, computed by OLS on the active basis functions only, and using the correction factor from Eq. (4.6)
5. hybrid 10-fold cross-validation error, computed by OLS on the active basis functions only
6. A fixed rule dependent on dimensionality and ED size, according to the findings in Section 4.3.1, choosing the solver that achieved the smallest error most often (i.e., having the longest dark red bar, and coincidentally also the longest bright red bar):
   - low-dimensional models, small ED: BCS & FN
   - low-dimensional models, large ED: $\mathrm{SP}_{\mathrm{LOO}}$ & FN
   - high-dimensional models, small ED: $\mathrm{SP}_{k=5}$ & FN
   - high-dimensional models, large ED: $\mathrm{SP}_{k=5}$ & FN

   This implies that by design, the lengths of the bars for this selection criterion are identical with the results for the respective combination in Fig. 4.2.
7. A randomly picked combination of solver and basis adaptivity scheme from the 9 available options as upper bound: any sensible model selection criterion must perform better than this.

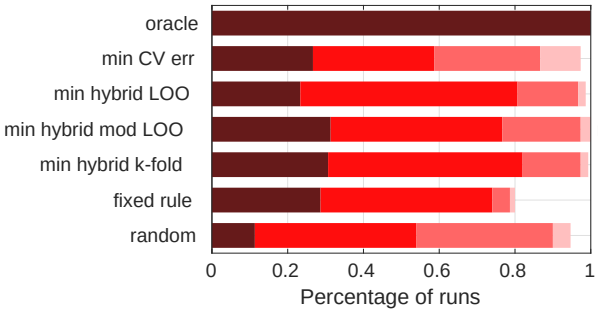
The results are presented in Figure 4.3. Note that we do not display which sparse solver and



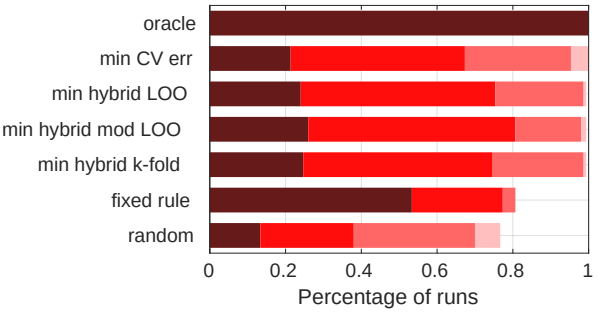

*(a) low-dimensional models, small ED*



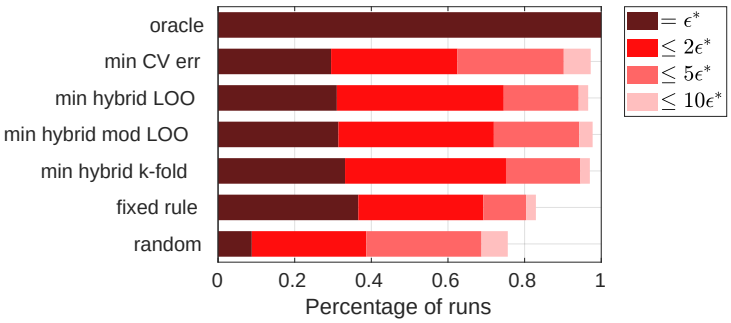

*(b) low-dimensional models, large ED*




*(c) high-dimensional models, small ED*




*(d) high-dimensional models, large ED*




*(e) Aggregated over all 11 models and both ED sizes*

***Figure 4.3:*** *Testing different automatic selection strategies. As before, the overlapping bar plot visualizes the percentage of runs where the respective strategy achieved the best error $\epsilon^*$ or was within a factor of $\{2, 5, 10\}$ of the best error $\epsilon^*$ among the seven selection strategies. The bar labels are explained in the enumeration of model selection criteria in the text.*

basis adaptivity scheme was chosen – we only show how close the chosen solution comes to the best possible solution.

- By construction, the oracle selection performs best in all cases. (We only include it into the plot to emphasize that all other criteria are measured against the best attained error on each ED.)
- The random selection is by far the worst selection criterion, which shows that automatic selection provides a statistically significant improvement over random selection.
- All three hybrid selection criteria are in general more robust than the fixed rule, attaining an error within 2 times of the best solution (bright red bar) similarly or more often than the fixed rule, and being within one order of magnitude of the best error (light red bar) even more than 90% of runs (low-dimensional models) or 99% of runs (high-dimensional models). Averaging over all runs (Fig. 4.3e), the hybrid criteria attain the best error almost as often as the fixed rule, which was chosen for this reason, i.e., because it attained the best error most often. This shows that automatic selection is able to return a solution with almost the same accuracy as the best-performing method, while being more robust than a fixed rule or a random choice of methods.
- The criterion used for basis and hyperparameter selection, i.e., k-fold CV for $\mathrm{SP}_{k=5}$ and BCS, and LOO for $\mathrm{SP}_{\mathrm{LOO}}$, performs slightly worse than the three hybrid selection criteria in all cases shown in Fig. 4.3. The slightly worse performance of this selection criterion might be explained by selection bias: since the criterion was already used twice for selection, it is likely that it underestimates the generalization error. Another reason might be that the criteria used by different solvers (OLS-based LOO vs. k-fold CV) might not be comparable among each other (inconsistency).

We conclude that automatic selection, i.e., using cross-validation to choose a PCE solution from a number of candidate solutions computed with different methods on the same experimental design, can achieve results that are as accurate as and more robust than results from a fixed rule for which combination of solver and basis adaptivity scheme to use, even if this rule is based on a thorough benchmark such as the one in Section 4.3.1. While automatic selection leads to increased computational cost due to the need of training the surrogate with different solvers and basis adaptivity schemes on the same experimental design, it is a case-independent strategy applicable to models of any dimensionality and experimental designs of any size, and it finally results in a PCE that is both accurate and more reliable.

## 4.4 Conclusion and discussion

Our goal was to provide guidance for the choice of sparse PCE methods in engineering applications, by answering the two questions: (1) Is there a significant difference between different combinations of sparse solvers and basis adaptivity strategies, and does the proper choice matter in actual applications? (2) Is there a smart strategy to automatically select a good combination?

To answer these questions, we performed an extensive comparative study investigating several approaches for computing sparse PCE with the goal of surrogate modeling, using the relative

mean squared error on a validation set as the main performance metric. In particular, we studied the performance of combinations of basis adaptivity schemes and sparse solvers in order to identify combinations that yield the smallest generalization error. Our investigations are based on 11 analytical and numerical benchmark models of varying input dimension and complexity, representative of a wide range of engineering problems. We considered the sparse solvers least angle regression (LARS), orthogonal matching pursuit (OMP), subspace pursuit based on $k$-fold cross-validation ($\mathrm{SP}_{k=5}$), subspace pursuit based on LOO ($\mathrm{SP}_{\mathrm{LOO}}$), Bayesian compressive sensing/FastLaplace (BCS), and spectral projected gradient-$\ell^1$ (SPGL1). The basis adaptivity schemes we compared were a fixed basis truncation scheme following the rule $N \approx \frac{10}{3}P$, degree- and q-norm adaptivity, forward neighbor basis adaptivity, and anisotropic degree basis adaptivity. We made a distinction between four cases, namely low- and high-dimensional models as well as small and large experimental design sizes.

The comparative study revealed that it is important to carefully select the strategy, since the difference in generalization error between different combinations of methods can be large, even more than an order of magnitude. No single solver or basis-adaptive scheme significantly outperformed the others. However, by dividing the analysis into classes (low-/high-dimensional models and small/large ED sizes), some significant patterns can be identified. The combinations that performed well always involved the solvers $\mathrm{SP}_{\mathrm{LOO}}$, $\mathrm{SP}_{k=5}$, or BCS, but never SPGL1 or OMP. For the basis-adaptive schemes, the picture is less clear, except that combinations involving forward neighbor basis adaptivity achieved the best accuracy most often, and that the static basis (i.e., no basis adaptivity) was in nearly all cases outperformed by basis-adaptive schemes. This might imply that basis-adaptive schemes provide the opportunity to improve the solution by offering more regressors to choose from, but that none of them consistently generates the most suitable basis. The choice of the final basis is guided by the cross-validation error, which is computed based on the solution returned by the respective sparse solver. A good correlation of cross-validation and validation error is crucial for the scheme to make good choices – which for example is not the case for the solver OMP. Overall, there is no combination of solver and basis-adaptive scheme that was consistently best across all models and experimental design sizes, although averaging over all four cases, $\mathrm{SP}_{\mathrm{LOO}}$ together with forward neighbor basis adaptivity outperformed all other combinations.

Since no clear best combination of solver and basis adaptivity scheme emerged, we introduced an automatic selection step. Based on a suitable error estimate, automatic selection chooses one PCE solution out of several candidate solutions computed by various combinations of solvers and basis adaptivity schemes. We found that automatic selection using any hybrid cross-validation error estimate performs better than the fixed rules obtained from the basis adaptivity benchmark: automatic selection attains the best possible relative MSE almost as often as the best solver-basis adaptivity combination, while being significantly more robust. An additional advantage of automatic selection is that it is independent of model dimension or size of the available experimental design, unlike the proposed fixed rules, which rely on the somewhat arbitrary, albeit simple, classification we applied (low/high dimension and small/large experimental design).

These findings demonstrate that when building a sparse PCE for an expensive black-box computational model, it is worth it to carefully select a sparse solver, and to apply a basis-adaptive

scheme, because the difference in relative MSE between different combinations of methods on the same experimental design can be larger than one order of magnitude. While we could identify a number of methods that generally perform well, and others that should be avoided, as we described above, a superior strategy is to compute several PCE solutions and perform a final model selection using one of the presented hybrid cross-validation error estimators.

Further research could investigate the use of true cross-validation for automatic selection instead of the hybrid estimates which we used here. Also, it might be possible to identify other problem characteristics besides model dimension and the size of the available experimental design to guide the choice of methods in the sparse PCE framework. A promising class of methods combines basis adaptivity with the sequential enrichment of the experimental design, adapted to the current candidate or active basis (Fajraoui et al., 2017; Diaz et al., 2018; Hampton and Doostan, 2018), which might be able to further improve on the results obtained here.

## Acknowledgements

We thank John Jakeman (Sandia National Laboratories), Negin Alemazkoor (University of Virginia), Hadi Meidani (University of Illinois at Urbana-Champaign), and Jerrad Hampton (University of Colorado Boulder) for generously providing their code and explanations. We thank John Jakeman for his useful hints to relevant literature on basis adaptivity.

## References

Abbiati, G., Marelli, S., Tsokanas, N., Sudret, B., and Stojadinovic, B. (2021). A global sensitivity analysis framework for hybrid simulation. *Mech. Syst. Signal Pr.*, 146(106997).

Adams, B. M., Bauman, L. E., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., Hough, P. D., Hu, K. T., Jakeman, J. D., Stephens, J. A., Swiler, L. P., Vigil, D. M., , and Wildey, T. M. (2014). *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual*. Sandia National Laboratories. Technical Report SAND2014-4633 (Updated November 2015 (Version 6.3)).

Alemazkoor, N. and Meidani, H. (2017). Divide and conquer: An incremental sparsity promoting compressive sampling approach for polynomial chaos expansions. *Comput. Methods Appl. Mech. Engrg.*, 318:937–956.

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2010). Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.*, 19(1):53–63.

Blatman, G. and Sudret, B. (2009). Anisotropic parcimonious polynomial chaos expansions based on the sparsity-of-effects principle. In Furuta, H., Frangopol, D., and Shinozuka, M., editors, *Proc. 10th Int. Conf. Struct. Safety and Reliability (ICOSSAR'2009), Osaka, Japan.*

Blatman, G. and Sudret, B. (2010a). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Prob. Eng. Mech.*, 25:183–197.

Blatman, G. and Sudret, B. (2010b). Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 95:1216–1229.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys*, 230:2345–2367.

Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81.

Candès, E. J. and Plan, Y. (2011). A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Inform. Theory*, 57(11):7235–7254.

Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Process. Mag.*, 25(2):21–30.

Chapelle, O., Vapnik, V., and Bengio, Y. (2002). Model selection for small sample regression. *Machine Learning*, 48(1):9–23.

Chatterjee, T., Chakraborty, S., and Chowdhury, R. (2019). A critical review of surrogate assisted robust design optimization. *Arch. Comput. Method. E.*, 26(1):245–274.

Diaz, P. (2018). DOPT_PCE. https://github.com/CU-UQ/DOPT_PCE. [Online; accessed 14-May-2020].

Diaz, P., Doostan, A., and Hampton, J. (2018). Sparse polynomial chaos expansions via compressed sensing and D-optimal design. *Comput. Methods Appl. Mech. Engrg.*, 336:640–666.

Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.*, 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.

Echard, B., Gayton, N., Lemaire, M., and Relun, N. (2013). A combined importance sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliab. Eng. Syst. Saf.*, 111:232–240.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Stat.*, 32:407–499.

Ernst, O., Mugler, A., Starkloff, H.-J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.*, 46(02):317–339.

Fajraoui, N., Marelli, S., and Sudret, B. (2017). Sequential design of experiment for sparse polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):1061–1085.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. Wiley.

Gerstner, T. and Griebel, M. (2003). Dimension-adaptive tensor-product quadrature. *Computing*, 71:65–87.

Guo, X., Dias, D., and Pan, Q. (2019). Probabilistic stability analysis of an embankment dam considering soil spatial variability. *Comput. Geotech.*, 113:103093.

Hampton, J. and Doostan, A. (2017). BASE_PC. https://github.com/CU-UQ/BASE_PC. [Online; accessed 14-May-2020].

Hampton, J. and Doostan, A. (2018). Basis adaptive sample efficient polynomial chaos (BASE-PC). *J. Comput. Phys.*, 371:20–49.

Hariri-Ardebili, M. and Sudret, B. (2020). Polynomial chaos expansion for uncertainty quantification of dam engineering problems. *Eng. Struct.*, 203(#109631).

Jakeman, J. D., Eldred, M. S., and Sargsyan, K. (2015). Enhancing $\ell_1$-minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.*, 289:18–34.

Konakli, K. and Sudret, B. (2016). Global sensitivity analysis using low-rank tensor approximations. *Reliab. Eng. Sys. Safety*, 156:64–83.

Le Gratiet, L., Marelli, S., and Sudret, B. (2017). *Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes*, chapter 38, pages 1289–1325. Springer International Publishing. Cham, Switzerland.

Loukrezis, D., Römer, U., and De Gersem, H. (2019). Assessing the performance of Leja and Clenshaw-Curtis collocation for computational electromagnetics with random input data. *Int. J. Uncertainty Quantification*, 9(1).

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif.*, 9(2):593–649.

Mai, C. V. and Sudret, B. (2015). Hierarchical adaptive polynomial chaos expansions. In Papadrakakis, M., Papadopoulos, V., and Stefanou, G., editors, *1st Int. Conf. on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP)*, Creta, Greece.

Marelli, S., Lüthen, N., and Sudret, B. (2021). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-104.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35:243–255.

Narayan, A., Gittelson, C., and Xiu, D. (2014). A stochastic collocation algorithm with multifidelity models. *SIAM J. Sci. Comput.*, 36(2):A495–A521.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proc. of 27th Asilomar Conf. on Signals, Systems and Computers*, pages 40–44. IEEE.

Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *Wires. Data Min. Knowl.*, 8(4):e1249.

Slot, R. M., Sorensen, J. D., Sudret, B., Svenningsen, L., and Thogersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Saf.*, 93:964–979.

Thapa, M., Mulani, S. B., and Walters, R. W. (2020). Adaptive weighted least-squares polynomial chaos expansion with basis adaptivity and sequential adaptive sampling. *Comput. Meth. Appl. Mech. Eng.*, 360:112759.

van den Berg, E. and Friedlander, M. P. (2008). Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912.

Van den Berg, E. and Friedlander, M. P. (2015). SPGL1 - A Matlab solver for sparse optimization. https://friedlander.io/software/spgl1/. [Online; accessed 28-August-2019].

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644.

# Appendix

## 4.A Details on the settings for the basis adaptivity schemes in the benchmark

In Table 4.A.1, we list the basis adaptivity settings for each of the 11 models in our benchmark.

- The rule for the static basis (ST) is to choose a total-degree basis with $p$ so that the number of basis functions $P$ is closest to $\frac{10}{3}N$, where $N$ is the number of ED points. Here, for low-dimensional models, the q-norm is chosen as $q = 1$, while for high-dimensional models, we use $q = 0.5$.
- For degree and q-norm basis adaptivity (PQ), we choose the ranges for degree and q-norm as large as possible while still keeping the number of basis functions computationally manageable (rule of thumb $\lesssim 10^4$).
- For forward neighbor degree adaptivity (FN), the degree of the initial basis is chosen so that the size of the basis is closest to $10N$ (as recommended in (Jakeman et al., 2015)), while we set the q-norm to the maximum of the q-norm-range for PQ basis adaptivity.
- Finally, for anisotropic degree basis adaptivity, which is only used for low-dimensional models, we use $q = 1$ and $p = \lceil \frac{p_{\max}}{2} \rceil$, where $p_{\max}$ is the maximum of the degree range for PQ basis adaptivity.

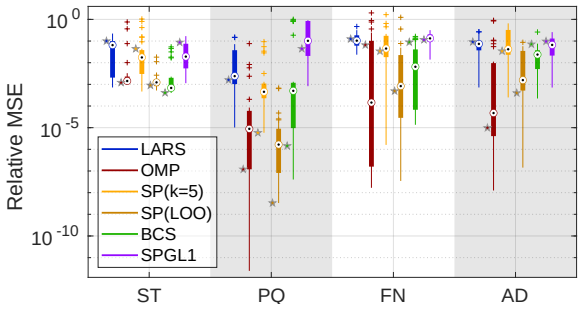
## 4.B Additional results

### 4.B.1 Basis adaptivity benchmark: raw data

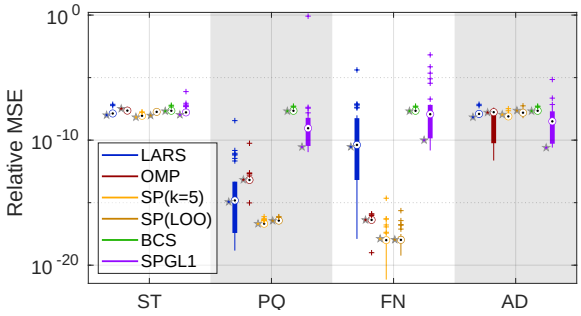
Figs. 4.B.1 and 4.B.2 show the raw data of the basis adaptivity benchmark: for each model and ED size, we display the boxplots of resulting relative MSE for each combination of sparse solver and basis-adaptive scheme. The star-shaped markers visualize the attained error for one specific ED realization. They illustrate that the resulting errors are not always well correlated: on the same ED, some methods achieve one of their smallest errors while others produce one of their largest. Therefore, we continue in Section 4.3.1 by investigating the relative performance of the different methods.

***Table 4.A.1:*** *Details on the initial bases and degree and q-norm ranges for the various basis adaptivity schemes*

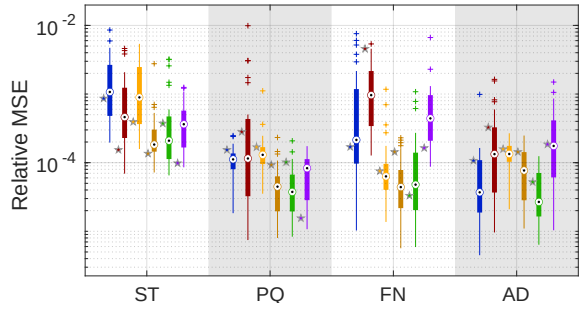
| Model | dim $d$ | static basis | PQ range | FN initial | AD initial |
|---|---|---|---|---|---|
| Ishigami function | 3 | $p = 8$ (small ED) / $p = 12$ (large), $q = 1$ | $p \in [1, \ldots, 25]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 12$ (small ED) / $p = 19$ (large) | $p = 13$ |
| Undamped oscillator | 6 | $p = 4$ / $p = 4$ , $q = 1$ | $p \in [1, \ldots, 10]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 5$ / $p = 6$ | $p = 5$ |
| Borehole function | 8 | $p = 4$ / $p = 4$ , $q = 1$ | $p \in [1, \ldots, 10]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 5$ / $p = 6$ | $p = 5$ |
| Damped oscillator | 8 | $p = 4$ / $p = 5$ , $q = 1$ | $p \in [1, \ldots, 7]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 5$ / $p = 6$ | $p = 4$ |
| Wingweight function | 10 | $p = 3$ / $p = 4$ , $q = 1$ | $p \in [1, \ldots, 7]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 4$ / $p = 5$ | $p = 4$ |
| *Truss model* | 10 | $p = 3$ / $p = 4$ , $q = 1$ | $p \in [1, \ldots, 6]$, $q \in [0.5, 0.6, \ldots, 1]$ | $p = 4$ / $p = 5$ | $p = 3$ |
| Morris function | 20 | $p = 6$ / $p = 8$ , $q = 0.5$ | $p \in [1, \ldots, 8]$, $q \in [0.4, 0.5, 0.6]$ | $p = 6$ / $p = 8$, $q = 0.6$ | – |
| *Structural frame model* | 21 | $p = 5$ / $p = 8$ , $q = 0.5$ | $p \in [1, \ldots, 8]$, $q \in [0.4, 0.5, 0.6]$ | $p = 6$ / $p = 8$, $q = 0.6$ | – |
| *2-dim diffusion model* | 53 | $p = 3$ / $p = 4$ , $q = 0.5$ | $p \in [1, \ldots, 6]$, $q \in [0.4, 0.5, 0.6]$ | $p = 4$ / $p = 5$, $q = 0.6$ | – |
| *1-dim diffusion model* | 62 | $p = 3$ / $p = 4$ , $q = 0.5$ | $p \in [1, \ldots, 5]$, $q \in [0.4, 0.5, 0.6]$ | $p = 3$ / $p = 4$, $q = 0.6$ | – |
| 100D function | 100 | $p = 3$ / $p = 4$ , $q = 0.5$ | $p \in [1, \ldots, 5]$, $q = 0.5$ | $p = 4$ / $p = 5$, $q = 0.5$ | – |



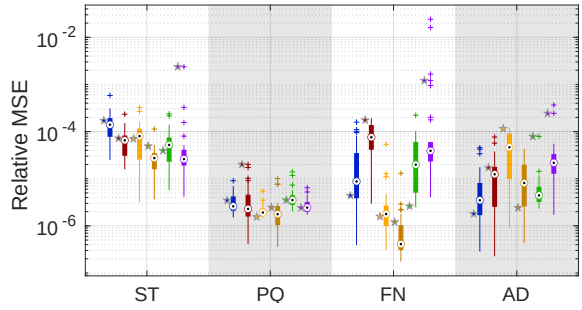

*(a) Ishigami function, $N = 50$*



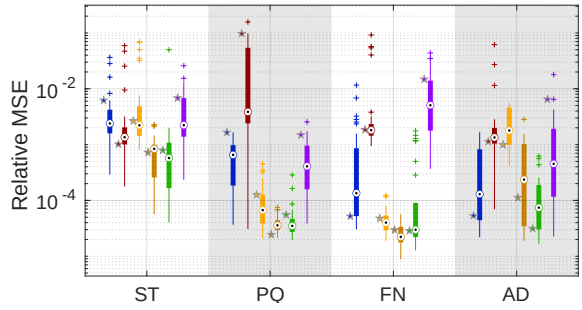

*(b) Ishigami function, $N = 150$*



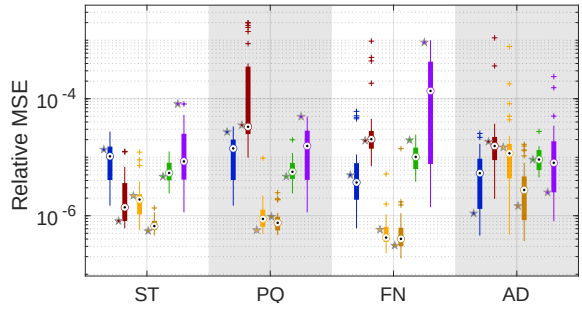

*(c) Undamped oscillator, $N = 60$*



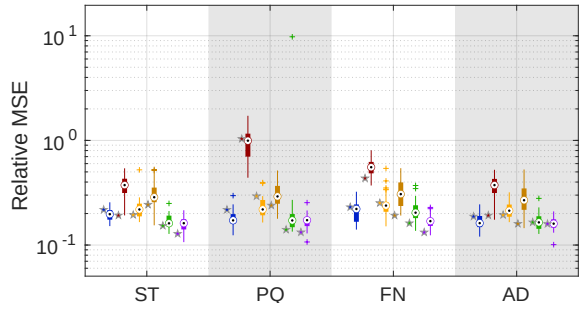

*(d) Undamped oscillator, $N = 120$*



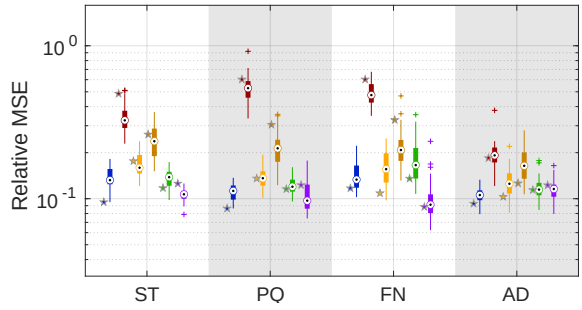

*(e) Borehole function, $N = 100$*




*(f) Borehole function, $N = 250$*

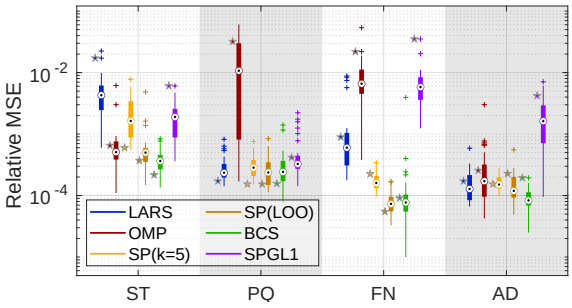



*(g) Damped oscillator, $N = 150$*

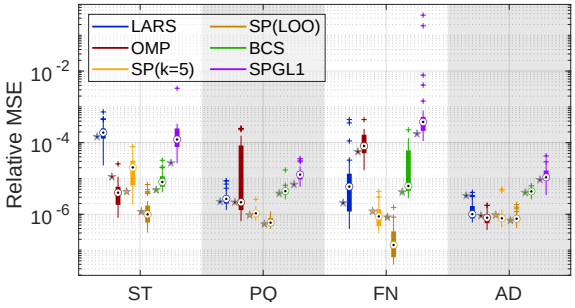



*(h) Damped oscillator, $N = 350$*

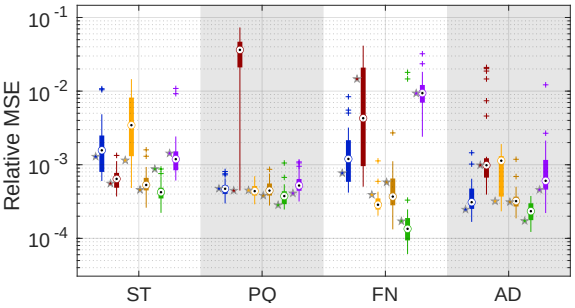
***Figure 4.B.1:*** *Comparison of different combinations of solvers and basis adaptivity schemes for* ***low-dimensional models****. We show validation errors attained by all 24 combinations of methods on 30 realizations of experimental design. The boxplots visualize the attained errors across all 30 realizations. The star-shaped markers denote the attained errors of all combinations for one selected ED realization. They highlight one set of data points which is also part of the larger set visualized by the boxplots. Left: small ED; right: large ED. Abbreviations of basis-adaptive schemes: ST: static basis; PQ: degree and q-norm; FN: forward neighbor; AD: anisotropic degree. Continued in next figure.*



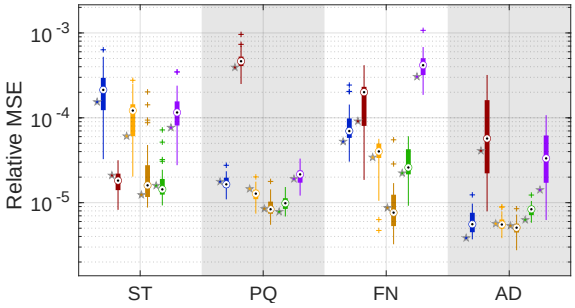

*(i)* Truss model, $N = 100$




*(j)* Truss model, $N = 250$

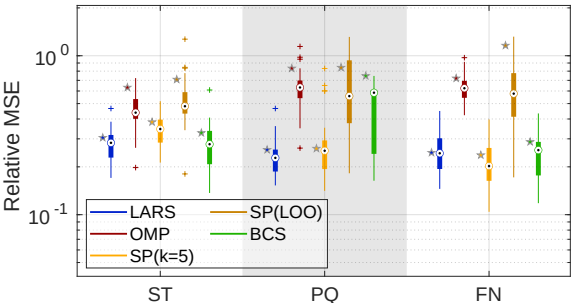



*(k)* Wingweight function, $N = 100$

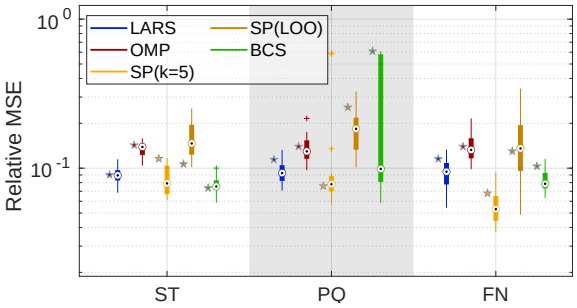



*(l)* Wingweight function, $N = 250$

***Figure 4.B.1:*** *Continued. Comparison of different combinations of solvers and basis adaptivity schemes for* ***low-dimensional models****. We show validation errors attained by all 24 combinations of methods on 30 realizations of experimental design. The boxplots visualize the attained errors across all 30 realizations. The star-shaped markers denote the attained errors of all combinations for one selected ED realization. They highlight one set of data points which is also part of the larger set visualized by the boxplots. Left: small ED; right: large ED. Abbreviations of basis-adaptive schemes: ST: static basis; PQ: degree and q-norm; FN: forward neighbor; AD: anisotropic degree.*

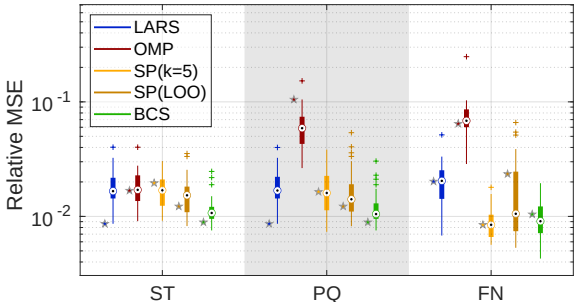



*(m)* Morris function, $N = 150$

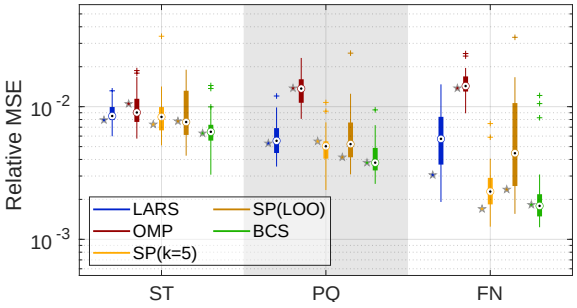



*(n)* Morris function, $N = 350$

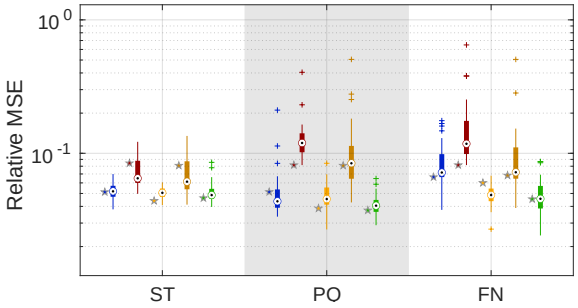
***Figure 4.B.2:*** *Comparison of different combinations of solvers and basis adaptivity schemes for* ***high-dimensional models****. We show validation errors attained by all 15 combinations of methods on 30 realizations of experimental design. The boxplots visualize the attained errors across all 30 realizations. The star-shaped markers denote the attained errors of all combinations for one selected ED realization. They highlight one set of data points which is also part of the larger set visualized by the boxplots. Left: small ED; right: large ED. Abbreviations of basis-adaptive schemes: ST: static basis; PQ: degree and q-norm; FN: forward neighbor. Continued in next figure.*



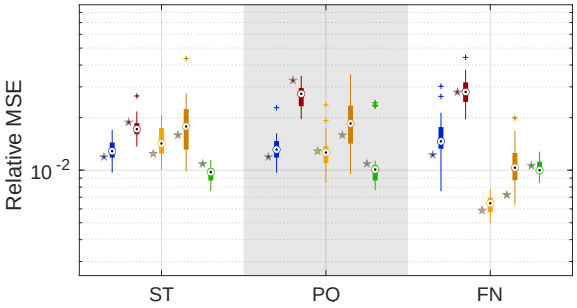

*(o) Structural frame, $N = 150$*



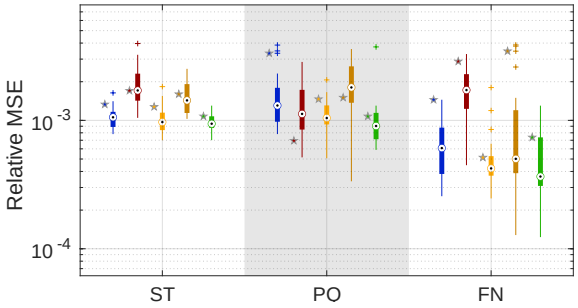

*(p) Structural frame, $N = 350$*



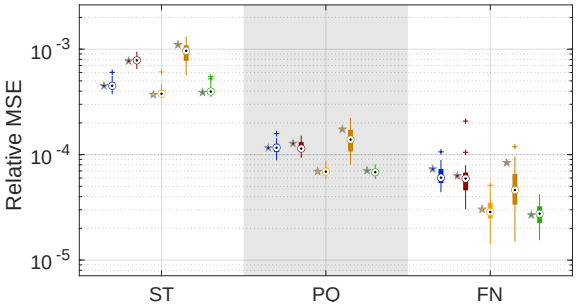

*(q) Diffusion 2D, $N = 100$*



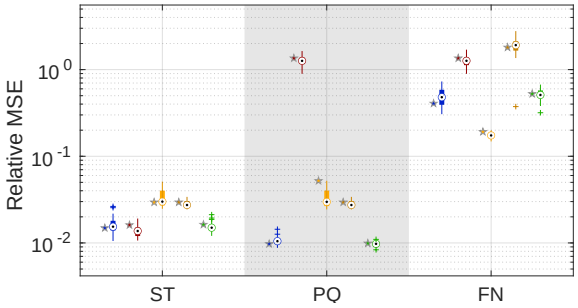

*(r) Diffusion 2D, $N = 400$*



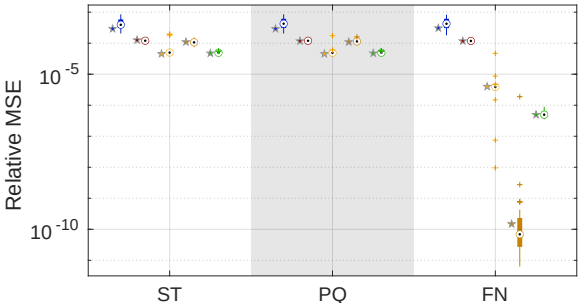

*(s) Diffusion 1D, $N = 100$*




*(t) Diffusion 1D, $N = 400$*




*(u) 100D function, $N = 400$*




*(v) 100D function, $N = 1200$*

***Figure 4.B.2:*** *Continued. Comparison of different combinations of solvers and basis adaptivity schemes for* ***high-dimensional models****. We show validation errors attained by all 15 combinations of methods on 30 realizations of experimental design. The boxplots visualize the attained errors across all 30 realizations. The star-shaped markers denote the attained errors of all combinations for one selected ED realization. They highlight one set of data points which is also part of the larger set visualized by the boxplots. Left: small ED; right: large ED. Abbreviations of basis-adaptive schemes: ST: static basis; PQ: degree and q-norm; FN: forward neighbor; AD: anisotropic degree.*

### 4.B.2 Bar plots for all combinations of methods

In Figs. 4.B.3-4.B.6, we display the aggregated results for all combinations of solvers and sampling schemes (while in Section 4.3.1, we only displayed the six best combinations). We sort the method combinations according to how often they achieved an error within two times the best relMSE (bright red bar). For a detailed description of how to read this plot, we refer to Section 4.3.1.

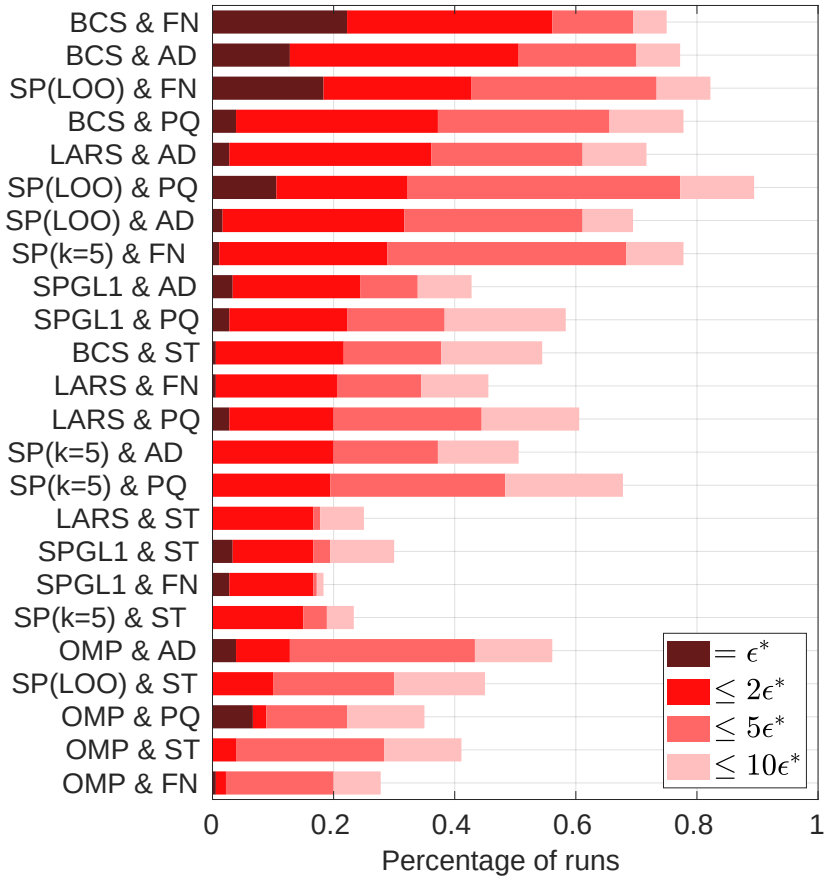



***Figure 4.B.3:*** ***Small ED sizes****. Aggregated results for* ***low-dim*** *models.*

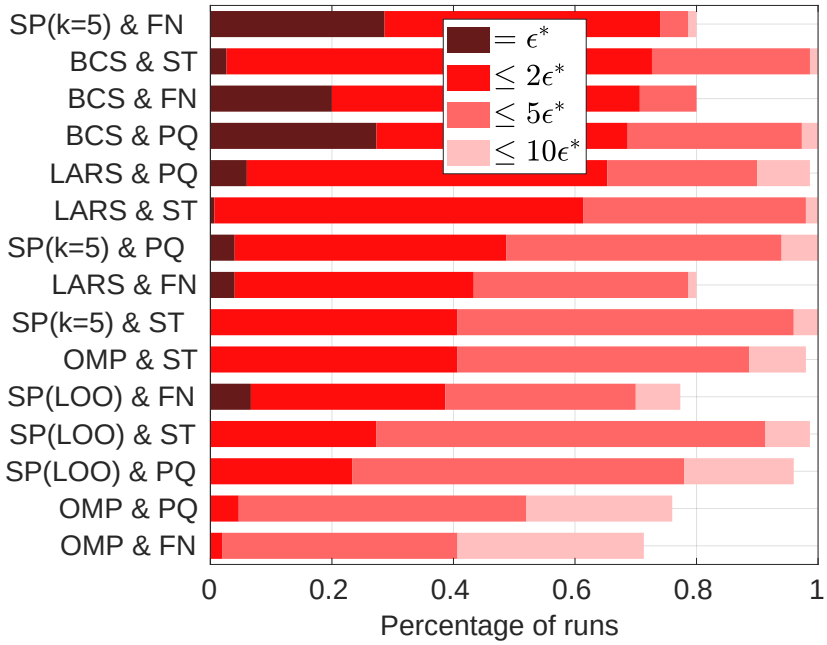



***Figure 4.B.4:*** ***Small ED sizes****. Aggregated results for* ***high-dim*** *models.*

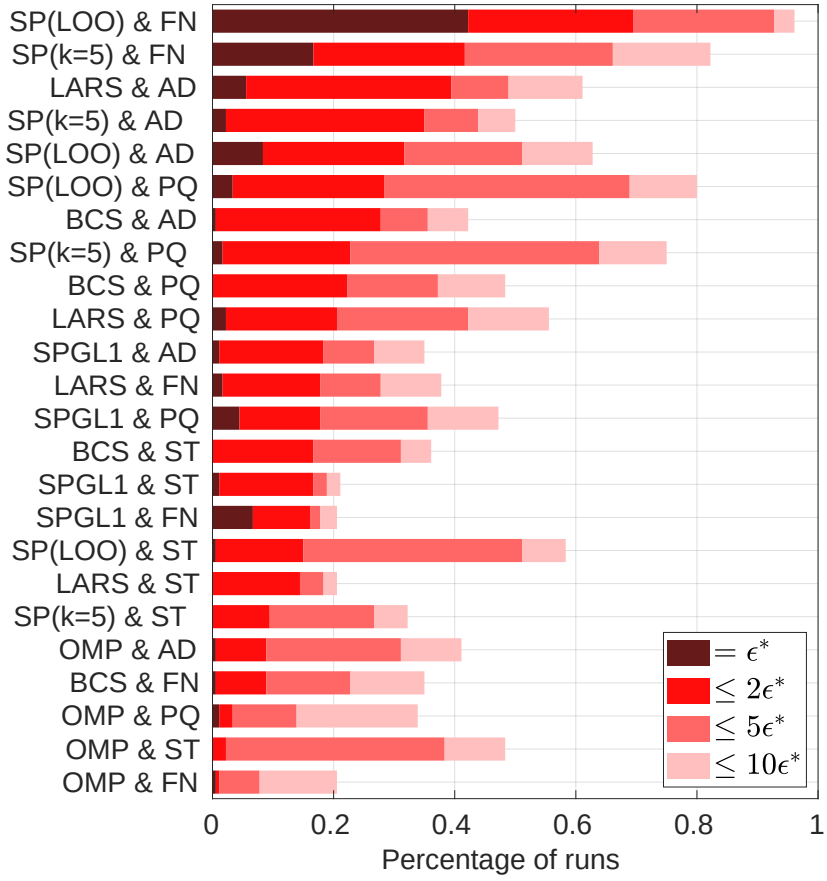



***Figure 4.B.5:*** ***Large ED sizes****. Aggregated results for* ***low-dim*** *models.*

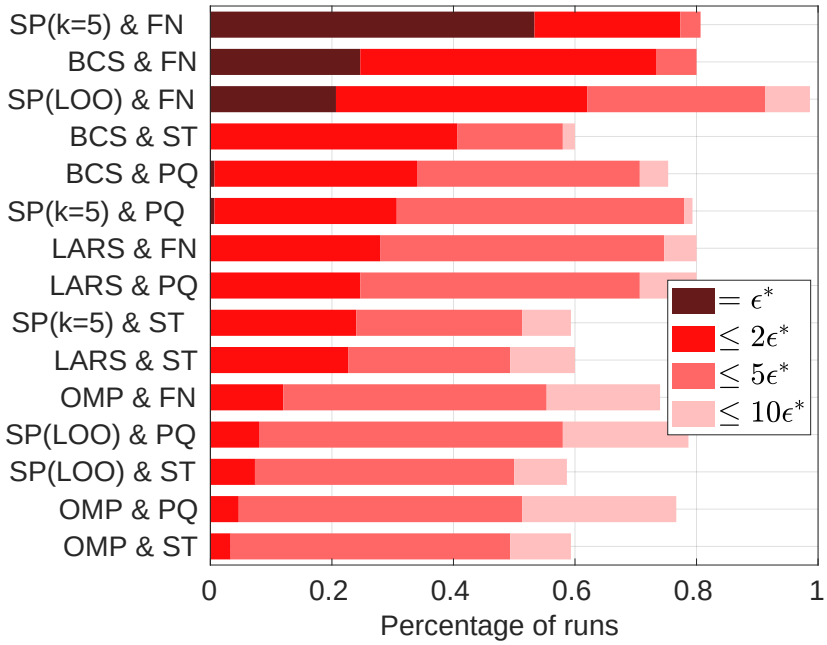



***Figure 4.B.6:*** ***Large ED sizes****. Aggregated results for* ***high-dim*** *models.*

CHAPTER 5

# Global sensitivity analysis using derivative-based sparse Poincaré chaos expansions

This chapter contains the preprint of

> Lüthen, N., Roustant, O., Gamboa, F., Iooss, B., Marelli, S., Sudret, B. (2022). Global sensitivity analysis using derivative-based sparse Poincaré chaos expansions. https://arxiv.org/abs/2107.00394.

submitted for publication in the *International Journal of Uncertainty Quantification* in March 2022.

**Author contributions.** **N. Lüthen:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization. **O. Roustant:** Conceptualization, Methodology, Formal Analysis, Writing - Original Draft, Supervision. **F. Gamboa:** Conceptualization, Methodology, Formal Analysis, Writing - Original Draft. **B. Iooss:** Conceptualization, Methodology, Formal Analysis, Writing - Original Draft. **S. Marelli:** Writing - Review & Editing, Supervision. **B. Sudret:** Conceptualization, Writing - Review & Editing, Supervision.

## Abstract

Variance-based global sensitivity analysis, in particular Sobol' analysis, is widely used for determining the importance of input variables to a computational model. Sobol' indices can be computed cheaply based on spectral methods like polynomial chaos expansions (PCE). Another choice are the recently developed Poincaré chaos expansions (PoinCE), whose orthonormal

tensor-product basis is generated from the eigenfunctions of one-dimensional Poincaré differential operators. In this paper, we show that the Poincaré basis is the unique orthonormal basis with the property that partial derivatives of the basis form again an orthogonal basis with respect to the same measure as the original basis. This special property makes PoinCE ideally suited for incorporating derivative information into the surrogate modelling process. Assuming that partial derivative evaluations of the computational model are available, we compute spectral expansions in terms of Poincaré basis functions or basis partial derivatives, respectively, by sparse regression. We show on two numerical examples that the derivative-based expansions provide accurate estimates for Sobol' indices, even outperforming PCE in terms of bias and variance. In addition, we derive an analytical expression based on the PoinCE coefficients for a second popular sensitivity index, the derivative-based sensitivity measure (DGSM), and explore its performance as upper bound to the corresponding total Sobol' indices.

## 5.1 Introduction

Computer models simulating physical phenomena and industrial systems are commonly used in engineering and safety studies, for prediction, validation or optimisation purposes. These numerical models often take as inputs a high number of physical parameters, whose values are variable or not perfectly known, creating the need for uncertainty quantification on model computations (Smith, 2014). Uncertainty quantification typically becomes more challenging the higher the input dimension is (curse of dimensionality). In this situation, global sensitivity analysis (GSA) is an invaluable tool that allows the analyst to rank the relative importance of each input of the model and to detect non-influential inputs (Borgonovo and Plischke, 2016; Razavi et al., 2021). Most often relying on a probabilistic modeling of the model input variables, GSA tries to explain model output uncertainties on the basis of model input uncertainties, accounting for the full range of variation of the variables.

A well-known and widely used GSA method is Sobol' analysis (Sobol', 1993), which relies on the functional ANOVA (analysis of variance) decomposition (Efron and Stein, 1981). For a square-integrable model and independent input variables, Sobol' analysis determines which part of the model output variance can be attributed to each input and to each interaction between inputs. The overall contribution of each input, including interactions with other inputs, is provided by the total Sobol' index (Homma and Saltelli, 1996). Sobol' indices can be estimated efficiently using various Monte Carlo-based techniques as well as metamodel-based techniques (Prieur and Tarantola, 2017). The latter save on expensive model evaluations by first performing a small number of model runs, which are used to compute an accurate approximation to the original model – the meta- or surrogate model – from which the Sobol' indices are finally computed (Fang et al., 2006; Le Gratiet et al., 2017).

One of the most popular and powerful metamodelling methods is the polynomial chaos expansion (PCE) (Xiu and Karniadakis, 2002). PCE represents the model in a specific basis consisting of polynomials that are orthonormal with respect to the input distribution. Orthogonal polynomial systems have been studied throughout the last century and they have many useful properties (see, e.g., Szegö (1939) and Simon (2010)). One particular strength of PCE is that once it is computed,

it easily gives all the variance-based quantities defined through the ANOVA decomposition, and in particular the Sobol' indices at all orders (Sudret, 2008). In practice, the expansion cannot use infinitely many terms and must be truncated. Among the many approaches available to compute the expansion coefficients, sparse regression techniques combined with adaptive basis selection appear to be especially promising (see Lüthen et al. (2021, 2022a) for an overview). Here, a small number of terms is selected which is able to best represent the computational model based on the available model evaluations.

In some practical situations, partial derivatives of the model output with respect to each input are easily accessible, for example by algorithmic differentiation of the numerical model in the reverse (adjoint) mode (Griewank and Walther, 2008). This technique allows for computing all partial derivatives of the model output at a cost independent of the number of input variables. Since PCEs are such a well-established metamodelling tool, there have been many efforts to leverage the additional information contained in model derivatives to improve the performance of PCE. The idea of including derivative information into sparse regression problems, often called *gradient-enhanced $\ell^1$-minimization*, is tested by Jakeman et al. (2015) for one numerical example with uniform inputs, and analyzed theoretically and numerically by Peng et al. (2016) for Hermite PCE. Both report favorable results. Roderick et al. (2010) and Li et al. (2011) apply polynomial regression (PCE) in the context of nuclear engineering. They include derivative information into the least-squares regression formulation and observe that most polynomial families are not orthogonal with respect to the $H^1$ inner product. This may deteriorate certain properties of the regression matrix. To alleviate this issue, Guo et al. (2018) develop a preconditioning procedure for gradient-enhanced sparse regression with certain polynomial families, with the goal of improving the orthogonality properties of the regression matrix. In all these approaches, the utilization of derivative information is not straightforward, but requires specific polynomial families and/or specialized sampling and preconditioning, because the partial derivatives of a PCE basis do in general not form an orthogonal system. Gejadze et al. (2019) have derived derivative-enhanced projection methods to compute the PCE coefficients but their method is restricted to Hermite polynomials and low polynomial degree.

On a different note, the availability of model derivatives has implications also for GSA. The so-called *Derivative-based Global Sensitivity Measures* (DGSM) are computed by integrating the squared partial derivatives of the model output over the domain of the inputs. These indices have been shown to be efficiently estimated by sampling techniques (as Monte Carlo or quasi-Monte Carlo) as well as from PCE (Sudret and Mai, 2015), and have been proven to be an excellent screening technique (i.e., detecting all the non-influential inputs among a large number), see e.g. the review in Kucherenko and Iooss (2017). Indeed, the interpretation of DGSM indices is straightforward due to their inequality relationship with Sobol' indices: multiplied with the associated Poincaré constant, DGSM indices provide an upper bound of the total Sobol' index (Lamboni et al., 2013), regardless of the input probability distribution.

Another way to utilize model derivatives, which solves the issues present for the polynomial chaos formulation, and naturally provides sharp lower bounds as well as upper bounds on total Sobol' indices, is to compute *Poincaré chaos expansions* (Roustant et al., 2020a), which we will abbreviate by *PoinCE* in the sequel. Similar to PCE, PoinCE is a spectral expansion in terms

of an orthonormal basis whose elements are eigenfunctions of the so-called *Poincaré differential operator*. The eigenfunctions are in general non-polynomial, except for the special case of the Gaussian distribution, where they coincide with the Hermite polynomials. The key property of PoinCE is that the partial derivatives of the basis form again an orthogonal basis with respect to the input distribution. This allows to conveniently expand the derivative of the computational model in terms of partial derivatives of the basis (PoinCE-der), which yields another estimator for partial variances and Sobol' indices. If the partial derivatives of the model have smaller variability than the model itself, the estimates based on the model derivatives might be more accurate. This makes PoinCE(-der) an efficient tool for screening (Roustant et al., 2020a).

Our present contribution to the field of generalized chaos expansions and GSA is two-fold. On the theoretical side, we provide a proof that the Poincaré basis is in fact characterized uniquely as the orthonormal basis which remains an orthogonal basis (w.r.t. the same probability measure) after differentiation. Furthermore, we show how PoinCE naturally generalizes an analytical formula for DGSM originally developed for Hermite PCE (Sudret and Mai, 2015), which implies that PoinCE simultaneously and efficiently provides lower and upper bounds to all partial variances. On the computational side, we improve on Roustant et al. (2020a), which introduced projection-based Poincaré chaos and demonstrated that small Sobol' indices were approximated particularly well by the derivative expansion. In this contribution, we compute PoinCE by sparse regression, thus generalizing the powerful and cost-effective sparse PCE methodology to non-polynomial functions. We explore the performance of PoinCE as an estimator for partial variances (upper and lower bounds) and compare it to standard PCE.

This paper is organized as follows. Section 5.2 revisits the mathematical foundations of PoinCE and presents several analytical results related to Sobol' indices and DGSM. Section 5.3 explains the computation of PoinCE basis functions, and the sparse regression methodology adapted from PCE to PoinCE. The methodology is applied in Section 5.4, where two example problems are investigated to demonstrate its performance for sensitivity analysis and screening. Finally, we summarize our conclusions in Section 5.5.

## 5.2 Mathematical background

### 5.2.1 Orthonormal bases in $L^2$

In this section, we recall some important facts about orthonormal bases in $L^2(E,\mu)$ where $E$ is a subset of $\mathbb{R}^d$ and $\mu$ is a probability measure on $E$. We first outline the general theory in Section 5.2.1.1. The particular cases of polynomial and Poincaré bases in several dimensions are developed in Sections 5.2.1.2 and 5.2.1.3.

#### 5.2.1.1 General theory

To begin with, recall that $L^2(E,\mu)$ endowed with the inner product

$$\langle f,g\rangle = \int_E f(x)g(x)\mu(dx), \quad \text{for } f,g \in L^2(E,\mu) \tag{5.1}$$

is a Hilbert space. Recall that a sequence of functions $(\Phi_\alpha)_{\alpha\in\mathcal{I}}$ ($\mathcal{I}\subset\mathbb{N}$) is an *orthonormal system* in $L^2(E,\mu)$ if it satisfies the two following assumptions:

1) For all $\alpha\neq\alpha'$, $\langle\Phi_\alpha,\Phi_{\alpha'}\rangle=0$, (orthogonality)
2) For all $\alpha$, $\langle\Phi_\alpha,\Phi_\alpha\rangle=1$ (unit norm).

An orthonormal system in $L^2(E,\mu)$ is called *complete* if the closure of the span generated by $(\Phi_\alpha)$ is $L^2(E,\mu)$. In this case, the system $(\Phi_\alpha)$ is called an *Hilbertian* or *orthonormal basis* of $L^2(E,\mu)$ and for any function $f\in L^2(E,\mu)$ the following expansion holds:

$$f=\sum_\alpha \langle\Phi_\alpha,f\rangle\,\Phi_\alpha,\quad (\mu \text{ almost surely}). \tag{5.2}$$

When used to represent random variables in terms of a basis of uncorrelated random variables, such an expansion is often called *chaos expansion* in the uncertainty quantification literature (Wiener, 1938; Ghanem and Spanos, 1991; Ernst et al., 2012).

An archetype example of chaos expansion is given by the so-called *Fourier expansion*. This corresponds to the case where the set $E=[0,1]$ is endowed with the Lebesgue measure and we have for $\alpha\in\mathbb{Z}$,

$$\Phi_\alpha(x)=\sqrt{2}\cos(2\pi\alpha x) \text{ if } \alpha<0,\ \Phi_0(x)=1, \text{ and } \Phi_\alpha(x)=\sqrt{2}\sin(2\pi\alpha x) \text{ if } \alpha>0.$$

In this frame, any square-integrable function $f$ may be expanded as

$$f(x)=a_0+\sqrt{2}\sum_{\alpha>0}\left(a_\alpha\cos(2\pi\alpha x)+b_\alpha\sin(2\pi\alpha x)\right).$$

Here, for all $\alpha\in\mathbb{Z}_*$,

$$a_0=\int_0^1 f(x)dx,\ \ a_\alpha=\sqrt{2}\int_0^1 f(x)\cos(2\pi\alpha)dx,\ b_\alpha=\sqrt{2}\int_0^1 f(x)\sin(2\pi\alpha)dx.$$

When the probability measure is a product measure $\mu=\mu_1\otimes\cdots\otimes\mu_d$ on a product space of intervals $E=E_1\times E_2\times\cdots\times E_d$, there is a canonical way to build a Hilbertian basis from a collection of univariate Hilbertian ones. Indeed, for $i=1,\ldots,d$ assume that $(\Phi_{\alpha_i}^{(i)})$ is a Hilbertian basis of $L^2(E_i,\mu_i)$. Then, setting $\boldsymbol{\alpha}:=(\alpha_1,\ldots,\alpha_d)$ and defining the tensor product functions $\Phi_{\boldsymbol{\alpha}}:=\prod_{i=1}^d\Phi_{\alpha_i}^{(i)}$, we obtain that $(\Phi_{\boldsymbol{\alpha}})$ is an orthonormal basis of $L^2(E,\mu)$.

In the following we describe two particular chaos types, namely the classical *polynomial chaos* and the recently developed *Poincaré chaos*, for a probability measure $\mu$ on $E\subset\mathbb{R}$.

#### 5.2.1.2 Polynomial chaos

A classical family of chaos expansions on an interval $E$ of $\mathbb{R}$ endowed with a probability measure are *polynomial chaos expansions* (PCE) given by orthonormal polynomial bases. A well-known example is the Hermite expansion for which the set $E$ is the whole line $\mathbb{R}$ endowed with the standard Gaussian distribution. In this example, for $\alpha\in\mathbb{N}$, $\Phi_\alpha=H_\alpha$ is the Hermite polynomial of degree $\alpha$. The first Hermite polynomials are

$$H_0(x)=1,\ H_1(x)=x,\ H_2(x)=\frac{x^2-1}{2},H_3(x)=\frac{x^3-3x}{6},\ H_4(x)=\frac{x^4-6x^2+3}{24}\quad(x\in\mathbb{R}).$$

In general, there exists an orthonormal polynomial basis for $L^2(\mu)$ whenever the moment problem for $\mu$ is determinate. This includes the uniform, Gaussian, Beta and Gamma distributions, as well as all distributions with compact support (Ernst et al., 2012).

#### 5.2.1.3 Poincaré chaos

The Poincaré basis is another example of an orthonormal basis of $L^2(\mu)$, consisting of functions that admit weak derivatives, i.e. that belong to $H^1(\mu) = \{f \in L^2(\mu) \text{ s.t. } f' \in L^2(\mu)\}$. Recall that $H^1(\mu)$, endowed with the norm $\|f\|^2_{H^1(\mu)} = \|f\|^2 + \|f'\|^2$, is a Hilbert space. This short summary is based on Roustant et al. (2017) in which more details can be found. We assume that:

**Assumption 2.** *The probability measure $\mu$ is supported on a bounded interval $(a, b)$ and admits a density of the form $\rho = e^{-V}$, where $V$ is continuous and piecewise $C^1$ on $[a, b]$ with respect to the Lebesgue measure.*

This assumption is sufficient to guarantee the existence of a Poincaré basis. On the topological side, it implies that the Hilbert space $L^2(\mu)$ (resp. $H^1(\mu)$) is equal to $L^2(a, b)$ (resp. $H^1(a, b)$), with an equivalent norm. Indeed, $\mu$ is a bounded perturbation of the uniform measure on $[a, b]$, meaning that the pdf $\rho$ is bounded from below and above by strictly positive constants (by continuity of $V$ on the compact support $[a, b]$).

**Theorem 2 (1D Poincaré basis).** *Under Assumption 2, there exists an orthonormal basis $(\varphi_\alpha)_{\alpha\geq 0}$ of $L^2(\mu)$ such that for all $f \in H^1(\mu)$ and for all integer $\alpha \geq 0$, we have:*

$$\langle f', \varphi'_\alpha \rangle = \lambda_\alpha \langle f, \varphi_\alpha \rangle , \tag{5.3}$$

*where $(\lambda_\alpha)_{\alpha\geq 0}$ is an increasing sequence that tends to infinity:*

$$0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_\alpha \underset{\alpha\to\infty}{\longrightarrow} +\infty.$$

*Here, the inner product $\langle \cdot, \cdot \rangle$ is the one on $L^2(\mu)$ as defined in (5.1). The basis functions $\varphi_\alpha$ are unique up to a sign change, and form the so-called Poincaré basis. Notice that $\varphi_0$ is a constant function equal to $\pm 1$; by convention, we choose $\varphi_0 = 1$.*
*Furthermore, the Poincaré basis functions are the eigenfunctions of the differential operator*

$$L(f) = f'' - V'f'$$

*i.e. satisfy $L(f) = -\lambda f$, subject to Neumann conditions $f'(a) = f'(b) = 0$. The $(\lambda_\alpha)_{\alpha\geq 0}$ are the corresponding eigenvalues.*
*Finally, $\alpha^{-2}\lambda_\alpha \to \pi^2$ when $\alpha$ tends to infinity, and for all $\alpha \in \mathbb{N}^\star$, the eigenfunction $\varphi_\alpha$ has exactly $\alpha$ zeros in $(a, b)$.*

*Proof.* The main part of the Theorem can be found in Roustant et al. (2017) or Bakry et al. (2014). The two last assertions come by rewriting the differential equation $f'' - V'f' = -\lambda f$ in the Sturm-Liouville form

$$-(pf')' + qf = \lambda w f \tag{5.4}$$

with $p = w = \rho$ and $q = 0$. Then, by the Sturm-Liouville theory (see e.g. Zettl (2010, Theorem 4.3.1, (1), (6) and (7))), we have that $\alpha^{-2}\lambda_\alpha \to \pi^2$ when $\alpha$ tends to infinity, and for all $\alpha \in \mathbb{N}^\star$, the eigenfunction $\varphi_\alpha$ has exactly $\alpha$ zeros in $(a, b)$. □

The Poincaré basis shares some similarity with both the polynomial chaos and the Fourier basis in terms of oscillations: by Theorem 2, the higher the order of the eigenvalue, the more oscillating the corresponding eigenfunction.

For some specific cases, the Poincaré basis is known analytically. For instance, for the uniform distribution, the Poincaré basis is a kind of Fourier basis (see e.g. Roustant et al. (2020a, §4)). Otherwise it has to be computed numerically, e.g., by a finite element technique (see Section 5.3.1).

Note that Assumption 2 is a convenient sufficient condition which guarantees the existence of a Poincaré basis. It is satisfied for a large range of truncated parametric probability distributions. The set of probability distributions for which the Poincaré chaos exists is larger, but not well known. For instance, the Poincaré chaos is defined for the Gaussian distribution, and then coincides with polynomial chaos, corresponding to Hermite polynomials. This is the only case where Poincaré chaos and polynomial chaos coincide (Bakry et al., 2014, §2.7). On the other hand, Poincaré chaos is not defined for the Laplace distribution, since the eigenvalues of the associated operator do not form a countable set (Bakry et al., 2014, §4.4.1).

The Poincaré basis is useful for sensitivity analysis. First, it is linked to the Poincaré inequality

$$\mathrm{Var}_\mu(f) \leq C_P(\mu) \int f'^2 d\mu, \tag{5.5}$$

which holds for all functions $f \in H^1(\mu)$ under the assumptions on $\mu$. Indeed, the smallest constant $C_P(\mu)$ such that (5.5) is satisfied is equal to $C_P(\mu) = 1/\lambda_1$, and choosing $f = \varphi_1$ corresponds to the equality case (Roustant et al., 2017). Roughly speaking, the Poincaré basis function associated to the first non-zero eigenvalue is the function with the largest possible variance for a given amount of integrated squared derivative (in the $L^2$ sense). A second appealing property for the analysis of variance is that the derivatives of the Poincaré basis remain orthogonal functions:

**Proposition 1.** *Under Assumption 2, the sequence* $\left(\frac{1}{\sqrt{\lambda_\alpha}}\varphi'_\alpha\right)_{\alpha\geq 1}$ *is an orthonormal basis of* $L^2(\mu)$.

*Proof.* The orthonormality of the sequence is a consequence of (5.3) by choosing $f = \varphi_\beta$, with $\beta \in \mathbb{N}^*$. It remains to show that the system is dense in $L^2(\mu)$, or equivalently, that its orthogonal is null. Let thus $f \in L^2(\mu)$ such that

$$\langle f, \varphi'_\alpha \rangle = 0, \quad \text{for all } \alpha \geq 1.$$

As explained when stating Assumption 2, $L^2(\mu)$ (resp. $H^1(\mu)$) is equal to $L^2(a, b)$ (resp. $H^1(a, b)$), with an equivalent norm. Now, there exists $g \in H^1(\mu)$ such that $f = g'$. Indeed, let us define $g$ by $g(x) = g(a) + \int_a^x f(t)dt$. As $f \in L^2(\mu) = L^2(a, b)$, then $g$ belongs to $H^1(a, b) = H^1(\mu)$, and $g' = f$. Then we have

$$\langle g', \varphi'_\alpha \rangle = 0, \quad \text{for all } \alpha \geq 1.$$

By (5.3), we obtain $\langle g, \varphi_\alpha \rangle = 0$ for all $\alpha \geq 1$ (as $\lambda_\alpha > 0$ for $\alpha \geq 1$). As the functions $\varphi_\alpha$ form an orthonormal basis of $L^2(\mu)$ with $\varphi_0 = 1$, this implies that $g$ is a constant function, and finally $f = 0$. The proof is completed. □

In fact, the property in Proposition 1, i.e., that the derivatives of the Poincaré basis form again an orthogonal basis in $L^2(\mu)$, uniquely characterizes the Poincaré basis:

**Proposition 2.** *Under Assumption 2, Poincaré bases are the only orthonormal bases $(\varphi_\alpha)$ of $L^2(\mu)$ in $H^1(\mu)$ such that $(\varphi'_\alpha)$ is an orthogonal basis of $L^2(\mu)$.*

This result seems difficult to find in the literature. German-speaking readers can find a similar proposition in Mikolas (1955), stated in the frame of Sturm-Liouville theory for twice-differentiable functions satisfying boundary conditions. See also Kwon and Lee (2003) for a similar result under the assumption that all functions involved in the Sturm-Liouville problem Eq. (5.4) are of class $C^\infty$. We provide below a proof based on Hilbertian arguments.

As a corollary, if there exists a basis different from the Poincaré basis for which derivatives form an orthogonal system, then that system is not dense in $L^2(\mu)$. As an example, for the uniform probability measure on $[0, 2\pi]$, consider the usual Fourier basis formed by $\{\cos(nx), \sin(nx) : n \geq 0\}$ (up to multiplicative constants). Taking derivatives results in the same set of functions (up to multiplicative constants) – except for the constant function $\cos(0x) = 1$. Thus, the derivatives form an orthogonal system which covers the orthogonal of constant functions in $L^2(\mu)$, which is a strict subspace of $L^2(\mu)$. Meanwhile, the Poincaré basis for this probability measure is formed by functions proportional to $\cos\left(\frac{n}{2}x\right)$ for $n \geq 0$. Proposition 2 guarantees that all functions of $L^2(\mu)$, including the constant functions, are spanned by the derivatives. Indeed, this is explained intuitively by the presence of half-frequencies: when $n$ is odd, the functions $\sin\left(\frac{n}{2}x\right)$ are not orthogonal to 1.

*Proof of Proposition 2.* The fact that a Poincaré basis remains an orthogonal basis by derivation has been proved in Proposition 1. Conversely, let $(\varphi_\alpha)_{\alpha \geq 0}$ be a system of $H^1(\mu)$, with $\varphi_0 = 1$, such that $(\varphi_\alpha)$ is an orthonormal basis of $L^2(\mu)$ and $(\varphi'_\alpha)_{\alpha \geq 1}$ is an orthogonal basis of $L^2(\mu)$. Let us first prove that $(\varphi_\alpha)$ is an orthogonal basis of $H^1(\mu)$. The orthogonality is a direct consequence of the definition of the inner product of $H^1(\mu)$:

$$\langle \varphi_\alpha, \varphi_\beta \rangle_{H^1(\mu)} = \langle \varphi_\alpha, \varphi_\beta \rangle_{L^2(\mu)} + \left\langle \varphi'_\alpha, \varphi'_\beta \right\rangle_{L^2(\mu)} = (1 + \|\varphi'_\alpha\|^2_{L^2(\mu)})\delta_{\alpha,\beta}.$$

Let us prove that $(\varphi_\alpha)$ is dense in $H^1(\mu)$. As explained when stating Assumption 2, $L^2(\mu)$ (resp. $H^1(\mu)$) is equal to $L^2(a,b)$ (resp. $H^1(a,b)$), with an equivalent norm. Hence, it is equivalent to prove that $(\varphi_\alpha)$ is dense in $H^1(a,b)$. Now, let $f$ be in $H^1(a,b)$. As $(\varphi'_\alpha)$ is dense in $L^2(a,b)$ (equivalently in $L^2(\mu)$), then $f'$ expands as $f' = \sum_{\alpha \in \mathbb{N}} c_\alpha \varphi'_\alpha$. In $H^1(a,b)$ each function is equal to the primitive function of its derivative, hence we have:

$$\begin{aligned} \left| f(t) - f(a) - \sum_{\alpha=1}^{N} c_\alpha(\varphi_\alpha(t) - \varphi_\alpha(a)) \right| &= \left| \int_a^t \left( f'(x) - \sum_{\alpha=1}^{N} c_\alpha \varphi'_\alpha(x) \right) dx \right| \\ &\leq (b-a) \left\| f' - \sum_{\alpha=1}^{N} c_\alpha \varphi'_\alpha \right\|_{L^2(a,b)} \end{aligned}$$

where the inequality comes from the Cauchy-Schwarz inequality. We deduce that $\|f - f(a) - \sum_{\alpha=1}^{N} c_\alpha(\varphi_\alpha - \varphi_\alpha(a))\|_{L^2(a,b)} \to 0$ when $N$ tends to infinity. Together with $f' = \sum_{\alpha\in\mathbb{N}} c_\alpha \phi'_\alpha$, this implies that $\|f - f(a) - \sum_{\alpha=1}^{N} c_\alpha(\varphi_\alpha - \varphi_\alpha(a))\|_{H^1(a,b)} \to 0$. As $\varphi_0 = 1$, this proves that $(\varphi_\alpha)$ is dense in $H^1(a,b)$, which was to be proved.

Now, let us fix $\alpha \geq 0$. Consider the linear form $L_\alpha$ defined on $H^1(\mu)$ by $L_\alpha(f) = \langle f', \varphi'_\alpha\rangle_{L^2(\mu)}$. The Cauchy-Schwarz inequality gives $|L_\alpha(f)| \leq \|f'\|_{L^2(\mu)}\|\varphi'_\alpha\|_{L^2(\mu)} \leq \|f\|_{H^1(\mu)}\|\varphi'_\alpha\|_{L^2(\mu)}$. This proves that $L_\alpha$ is continuous. Hence, by the Riesz representation theorem, there exists a unique $\zeta_\alpha \in H^1(\mu)$ such that for all $f \in H^1(\mu)$, $L_\alpha(f) = \langle f, \zeta_\alpha\rangle_{H^1(\mu)}$, i.e. $\langle f', \varphi'_\alpha\rangle_{L^2(\mu)} = \langle f, \zeta_\alpha\rangle_{H^1(\mu)}$. Choosing $f = \varphi_\beta$ with $\beta \neq \alpha$, we obtain by orthogonality of $(\varphi'_\alpha)$ that for all $\beta \neq \alpha$, $\langle \varphi_\beta, \zeta_\alpha\rangle_{H^1(\mu)} = 0$. As $(\varphi_\beta)_{\beta\geq 0}$ is an orthogonal basis of $H^1(\mu)$, this implies that $\zeta_\alpha$ is collinear to $\varphi_\alpha$, i.e., there exists $\tilde{\lambda}_\alpha \in \mathbb{R}$ such that $\zeta_\alpha = \tilde{\lambda}_\alpha\varphi_\alpha$. Thus, for all $f \in H^1(\mu)$, we have $\langle f', \varphi'_\alpha\rangle_{L^2(\mu)} = \tilde{\lambda}_\alpha \langle f, \varphi_\alpha\rangle_{H^1(\mu)}$. Choosing $f = \varphi_\alpha$, we get $\tilde{\lambda}_\alpha = \frac{\|\varphi'_\alpha\|^2_{L^2(\mu)}}{1+\|\varphi'_\alpha\|^2_{L^2(\mu)}}$, which belongs to $[0,1)$. Finally, we obtain that $\langle f', \varphi'_\alpha\rangle_{L^2(\mu)} = \lambda_\alpha \langle f, \varphi_\alpha\rangle_{L^2(\mu)}$, where $\lambda_\alpha = \frac{\tilde{\lambda}_\alpha}{1-\tilde{\lambda}_\alpha}$ is a non-negative real number. As it is true for all $f$ in $H^1(\mu)$ and all $\alpha \in \mathbb{N}$, this implies, by uniqueness of the Poincaré basis (under Assumption 2), that $(\varphi_\alpha)_{\alpha\geq 0}$ is a Poincaré basis. □

Turning to higher dimensions, we assume that for all $i = 1, \ldots, d$, the probability measure $\mu_i$ satisfies Assumption 2, and we denote by $(\varphi_{i,\alpha_i})_{\alpha_i\geq 0}$ the sequence of 1-dimensional Poincaré basis functions, and by $(\lambda_{i,\alpha_i})_{\alpha_i\geq 0}$ the sequence of associated eigenvalues. The Poincaré chaos basis is then defined by the tensor product $\Phi_{\boldsymbol{\alpha}} = \varphi_{1,\alpha_1} \otimes \cdots \otimes \varphi_{d,\alpha_d}$. Using the properties of $L^2$ bases, (5.3) thus implies that for all $f \in H^1(\mu)$, for all $i = 1, \ldots, d$:

$$\left\langle \frac{\partial f}{\partial x_i}, \frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_i} \right\rangle = \lambda_{i,\alpha_i} \langle f, \Phi_{\boldsymbol{\alpha}}\rangle . \tag{5.6}$$

Similarly, applying Proposition 1, we get:

**Proposition 3.** *Under Assumption 2, for all $i = 1, \ldots, d$, the sequence* $\left(\frac{1}{\sqrt{\lambda_{i,\alpha_i}}}\frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_i}\right)_{\boldsymbol{\alpha},\alpha_i\geq 1}$ *is an orthonormal basis of $L^2(\mu)$.*

### 5.2.2 Variance-based indices, derivative-based indices

We first recall the definition of variance-based sensitivity indices, which quantify the importance of each input variable in terms of function response variability.

Let $f$ be a real-valued function defined on $E = E_1 \times \cdots \times E_d \subseteq \mathbb{R}^d$. The uncertainty of the inputs is represented by a random vector $\boldsymbol{X} = (X_1, \ldots, X_d)^T$ with probability measure $\mu$ on $E$. We further assume that the $X_i$'s are independent and that $f(\boldsymbol{X})$ belongs to $L^2(E,\mu)$. Denoting by $\mu_i$ the marginal distribution of $X_i$ on $E_i$ $(i = 1, \ldots, d)$, we then have $\mu = \mu_1 \otimes \cdots \otimes \mu_d$. In this framework, $f(\boldsymbol{X})$ can be decomposed uniquely as a sum of terms of increasing complexity

$$f(\boldsymbol{X}) = f_0 + \sum_{1\leq i\leq d} f_i(X_i) + \sum_{1\leq i<j\leq d} f_{i,j}(X_i, X_j) + \cdots + f_{1,\ldots,d}(X_1, \ldots, X_d) \tag{5.7}$$

under centering conditions $\mathbb{E}[f_I(X_I)] = 0$ and non-overlapping conditions $\mathbb{E}[f_I(X_I)|X_J] = 0$, for all sets $I \subseteq \{1, \ldots, d\}$ and all strict subsets $J$ of $I$. We have used the set notation $X_I$ to represent

the subvector of $\boldsymbol{X}$ obtained by selecting the coordinates belonging to $I$. These conditions imply that all the terms of (5.7) are orthogonal, leading to the variance decomposition

$$\mathrm{Var} f(\boldsymbol{X}) = \sum_{1\le i\le d} \mathrm{Var} f_i(X_i) + \sum_{1\le i<j\le d} \mathrm{Var} f_{i,j}(X_i, X_j) + \cdots + \mathrm{Var} f_{1,\dots,d}(X_1, \dots, X_d) \tag{5.8}$$

Due to this property, the functional decomposition (5.7) is often called ANOVA (ANalysis Of VAriance) decomposition. Originating from Hoeffding (1948), it was revisited by Efron and Stein (1981), Antoniadis (1984), and Sobol' (1993). For a given set $I \subseteq \{1, \dots, d\}$, we call the corresponding term of (5.8) *partial variance* (denoted $D_I$), and call its normalized version *Sobol' index* (denoted $S_I$):

$$D_I = \mathrm{Var}(f_I(X_I)), \qquad S_I = \frac{D_I}{D},$$

where $D = \mathrm{Var} f(\boldsymbol{X})$ is the overall variance (*total variance*). In particular, for $i \in \{1, \dots, d\}$, the first-order Sobol' index $S_i$ corresponds to the proportion of variance of $f(\boldsymbol{X})$ explained by $X_i$ only. In order to include also the interactions of $X_i$ with the other variables, the total partial variance and the total Sobol' index are defined by

$$D_i^{\mathrm{tot}} = \sum_{I \supseteq \{i\}} \mathrm{Var}(f_I(X_I)), \qquad S_i^{\mathrm{tot}} = \frac{D_i^{\mathrm{tot}}}{D}.$$

Note that practitioners also call the (total) partial variances unnormalized (total) Sobol' indices. In the sequel, we will use these two words interchangeably.
The total Sobol' index can be used for screening. Indeed, under mild conditions, if $S_i^{\mathrm{tot}} = 0$ then the function $f$ does not depend on $x_i$ over $E$ (in the pointwise sense).

When the derivatives are available, a global sensitivity index can be obtained by integration. The so-called *derivative-based sensitivity measure* (DGSM) index of $f$ with respect to $X_i$ (Sobol and Gresham, 1995; Kucherenko et al., 2009) is defined by

$$\nu_i = \mathbb{E}\left[\left(\frac{\partial f}{\partial x_i}(\boldsymbol{X})\right)^2\right] = \int_{\mathbb{R}^d} \left(\frac{\partial f}{\partial x_i}(\boldsymbol{x})\right)^2 d\mu(\boldsymbol{x}) = \left\|\frac{\partial f}{\partial x_i}\right\|^2. \tag{5.9}$$

Contrarily to variance-based indices, DGSM are not associated to a variance decomposition. Nevertheless, they can be used for screening. Indeed, under mild conditions, $\nu_i = 0$ implies that $f$ does not depend on $x_i$ over $E$.

### 5.2.3 Chaos expansion serving sensitivity analysis

One main advantage of using an orthonormal basis for sensitivity analysis is that, once the expansion has been obtained, the variance-based indices can be computed in a straightforward way as a sum of squared coefficients (Sudret, 2006, 2008). More precisely, let $f$ be in $L^2(\mu)$, and let $(\Phi_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^d}$ be a multivariate orthonormal basis obtained by tensorization as described in Section 5.2.1. The expansion of $f$ in this basis is given by

$$f = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d} c_{\boldsymbol{\alpha}} \Phi_{\boldsymbol{\alpha}}. \tag{5.10}$$

By using the orthonormality we obtain the expression of the total variance

$$D = \sum_{\boldsymbol{\alpha} \neq \mathbf{0}} c_{\boldsymbol{\alpha}}^2. \tag{5.11}$$

The expression of the total Sobol' index $S_i^{\text{tot}}$ is obtained by only considering the terms of the decomposition (5.10) that contain the variable $x_i$, i.e. such that $\alpha_i \geq 1$. Hence, we have $S_i^{\text{tot}} = \frac{D_i^{\text{tot}}}{D}$ with $D_i^{\text{tot}}$ the total partial variance

$$D_i^{\text{tot}} = \sum_{\boldsymbol{\alpha}, \alpha_i \geq 1} c_{\boldsymbol{\alpha}}^2. \tag{5.12}$$

The first-order Sobol' index $S_i^1$ relies on the terms that include $x_i$ only, i.e., $S_i^1 = \frac{D_i^1}{D}$ with

$$D_i^1 = \sum_{\substack{\boldsymbol{\alpha}, \alpha_i \geq 1, \\ \alpha_j = 0 \text{ for } j \neq i}} c_{\boldsymbol{\alpha}}^2. \tag{5.13}$$

Let us now consider the case where the gradient of $f$ is available. The Poincaré basis is particularly suited to this situation. Indeed, we can derive in a straightforward way expressions of both variance-based and derivative-based indices, involving the derivatives of $f$. Due to orthonormality, the coefficients of the basis expansion in (5.10) are given by the projection of $f$ onto the associated basis element:

$$c_{\boldsymbol{\alpha}} = \langle f, \Phi_{\boldsymbol{\alpha}} \rangle . \tag{5.14}$$

From now on, let $(\Phi_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}}$ denote the Poincaré basis. Combining (5.6) and (5.14), and assuming that $\alpha_1 \geq 1$, $c_{\boldsymbol{\alpha}}$ can be written using the partial derivatives w.r.t variable $X_1$ (Roustant et al., 2020a):

$$c_{\boldsymbol{\alpha}} = \langle f, \Phi_{\boldsymbol{\alpha}} \rangle = \frac{1}{\lambda_{1,\alpha_1}} \left\langle \frac{\partial f}{\partial x_1}, \frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_1} \right\rangle = \frac{1}{\lambda_{1,\alpha_1}} \left\langle \frac{\partial f}{\partial x_1}, \frac{\partial \varphi_{1,\alpha_1}}{\partial x_1} \otimes \varphi_{2,\alpha_2} \otimes \cdots \otimes \varphi_{d,\alpha_d} \right\rangle \tag{5.15}$$

and equivalently using partial derivatives w.r.t variable $X_i$ if $\alpha_i \geq 1$. Thus, (5.11), (5.12) and (5.13) can also be computed using the various partial derivatives of $f$. Whereas the theoretical expressions are equal, their estimators have different properties. For example, if the integral is evaluated by Monte Carlo simulation, the expression whose integrand has smaller variance will be more accurate. We describe in Section 5.3 the computation of the expansion coefficients by regression, and we empirically compare the two estimation procedures in Section 5.4.

Furthermore, DGSM can be computed directly from the Poincaré expansion. More precisely, we have the following proposition.

**Proposition 4 (DGSM formula for Poincaré chaos).** *Let $f \in H^1(\mu)$. Let $f = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \Phi_{\boldsymbol{\alpha}}$ be the expansion of $f$ in the Poincaré chaos basis, with $c_{\boldsymbol{\alpha}} = \langle f, \Phi_{\boldsymbol{\alpha}} \rangle$. Then the DGSM index of $f$ with respect to $X_i$ is equal to:*

$$\nu_i = \sum_{\boldsymbol{\alpha},\, \alpha_i \geq 1} \lambda_{i,\alpha_i} \left(c_{\boldsymbol{\alpha}}\right)^2 . \tag{5.16}$$

*Proof.* Write $f = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \Phi_{\boldsymbol{\alpha}}$. Then by Proposition 3, we get

$$\frac{\partial f}{\partial x_i} = \sum_{\boldsymbol{\alpha},\, \alpha_i \geq 1} c_{\boldsymbol{\alpha}} \frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_i},$$

where we can constrain the sum to multi-indices $\boldsymbol{\alpha}$ such that $\alpha_i \geq 1$, since $\varphi_{i,\alpha_i} = 1$ for $\alpha_i = 0$. Now, using again the orthogonality of Poincaré basis derivatives (Proposition 3), it follows that

$$\nu_i = \left\| \frac{\partial f}{\partial x_i} \right\|^2 = \sum_{\boldsymbol{\alpha},\, \alpha_i \geq 1} (c_{\boldsymbol{\alpha}})^2 \left\| \frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_i} \right\|^2 = \sum_{\boldsymbol{\alpha},\, \alpha_i \geq 1} \lambda_{i,\alpha_i} (c_{\boldsymbol{\alpha}})^2 .$$

□

Formula (5.16) extends a previous result given by Sudret and Mai (2015) when all the $\mu_i$ are standard Gaussian. Indeed, in that case, Poincaré chaos coincides with polynomial chaos, and $\lambda_{i,\alpha_i} = \alpha_i$.

Using the expressions provided in (5.12) and (5.16) and an inequality derived by Sobol' and Kucherenko (2009) and Lamboni et al. (2013), we obtain lower and upper bounds to total partial variances as follows:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}, \alpha_i \geq 1} (c_{\boldsymbol{\alpha}})^2 \;\leq\; D_i^{\text{tot}} \;\leq\; C_P(\mu_i)\nu_i \;=\; \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d,\, \alpha_i \geq 1} \frac{\lambda_{i,\alpha_i}}{\lambda_{i,1}} (c_{\boldsymbol{\alpha}})^2 . \tag{5.17}$$

The lower bound is an obvious consequence of the truncation. The upper bound holds only for the full infinite expansion and is otherwise underestimated. Comparing the form of the right-hand side of Eq. (5.17) with the total Sobol' formula Eq. (5.12) gives insight into how tight this upper bound is: equality is attained only if the Poincaré chaos expansion does not contain terms of higher degree than 1 for $X_i$ (then, $\frac{\lambda_{i,\alpha_i}}{\lambda_{i,1}} = 1$). Else, depending on the decay behavior of $c_{\boldsymbol{\alpha}}$ the gap can be significant, since the eigenvalues are diverging to infinity (see Theorem 2).

## 5.3 Computation of sparse Poincaré expansions

Let $f \in H^1(\mu, E)$ be a computational model defined on the input space $E \subset \mathbb{R}^d$, with independent input random variables and with the input probability measure $\mu$ admitting a probability density function $\rho$ fulfilling Assumption 2 for each of the marginals. In the remainder of this paper, we assume that $\rho$ is known. We also assume that we are provided with an i.i.d. sample from the input distribution and with the corresponding model evaluations and model gradient values at each of the points.

With *Poincaré expansion* (PoinCE) we denote the expansion of the computational model onto the Poincaré basis

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \Phi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{5.18}$$

and with *Poincaré derivative expansion in direction $i$* (PoinCE-der-$i$) the expression

$$\frac{\partial f}{\partial x_i}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}, \alpha_i \geq 1} c_{\boldsymbol{\alpha}}^{\partial, i} \frac{\partial \Phi_{\boldsymbol{\alpha}}}{\partial x_i}(\boldsymbol{x}), \tag{5.19}$$

or the equivalent expansion using normalized basis derivatives that have unit norm in $L^2(\mu)$. Note that (5.19) is the partial derivative of (5.18) w.r.t. variable $X_i$. Because the zeroth order basis function of a Poincaré basis is the constant function, basis terms for which $\alpha_i = 0$ have

zero partial derivative w.r.t. $X_i$ and are not included in (5.19). While in theory by Equation (5.15), the two expressions (5.18) and (5.19) provide identical coefficients for corresponding basis elements, i.e., $c_{\boldsymbol{\alpha}} = c_{\boldsymbol{\alpha}}^{\partial,i}$ for $\boldsymbol{\alpha} \in \{\boldsymbol{\alpha}' \in \mathcal{A} : \alpha'_i \geq 1\}$, in practice they will not coincide when estimated from a data set of finite size. This will be investigated in Section 5.4 for a number of numerical examples.

In this section, we describe how such expansions are computed in practice: this concerns the computation of the Poincaré basis functions, the choice of truncation, the location of the sampled points, and the method for computing the coefficients. The implementation relies on and integrates into the UQLab framework (Marelli and Sudret, 2014).

### 5.3.1 Implementation of Poincaré basis functions

As described in Section 5.2.1.3, Poincaré basis functions are tensor products of univariate Poincaré basis functions. Each 1D basis consists of the eigenfunctions of the Poincaré differential operator associated with the respective marginal distribution (Theorem 2).

A Poincaré basis is guaranteed to exist for marginal distributions fulfilling Assumption 2 and for the Gaussian distribution. Other distributions have to be transformed or truncated to allow for a Poincaré basis. Since an isoprobabilistic transformation to standard variables can be highly nonlinear (Torre et al., 2019; Oladyshkin and Nowak, 2012), we opt for truncation: if the distribution is not Gaussian and has (one- or two-sided) unbounded support, we truncate it to its $10^{-6}$- and $(1 - 10^{-6})$-quantiles, respectively.[1]

We consider here only standard parametric families of probability densities (bounded and unbounded), although a Poincaré basis can be computed for any input distribution which after truncation fulfills Assumption 2. In particular, without any changes to the methodology PoinCE could be used in a data-driven framework (Torre et al., 2019) by computing the Poincaré basis for a dimensionwise kernel density estimate of the input distribution (assuming independence) given the available data.

As can be seen from applying the change-of-variables formula for a linear transformation to (5.3), the eigenvalues of the Poincaré differential operator scale with the inverse of the squared support interval length. To avoid numerical difficulties, we therefore linearly transform (i.e., shift and rescale) parametric families to standard parameters using

- their bounds in the case of uniform, beta, triangular;
- their location and scale parameter in the case of Gaussian, Gumbel, Gumbel-min, Laplace, logistic;
- their (inverse) scale parameter in the case of exponential, gamma, Weibull, lognormal.

[1] One might argue that this can distort the results obtained with PoinCE, especially in the tails. It is true that this truncation introduces a small error. However, as all such methods, PoinCE by design approximates accurately mainly the bulk, not the tails (for this, specialized techniques like subset simulation shall be used). Furthermore, in practical applications it is a modelling choice how to represent the input distribution. Choosing an unbounded parametric distribution is common, but not necessarily the most sensible choice, since for virtually every quantity in the real world there is an upper bound that cannot be exceeded.

In the current implementation, distributions not belonging to this group of families are not being rescaled.

For standard uniform ($\mathcal{U}([-0.5, 0.5])$ and standard Gaussian ($\mathcal{N}(0, 1)$) marginals, the Poincaré basis can be analytically computed and is given by the Fourier (cosine) basis and the Hermite polynomial basis, respectively (Roustant et al., 2020a). Therefore, in the special case of uniform or Gaussian marginals, we always (after rescaling) use the analytical solution.

For all other marginals, the Poincaré basis is computed numerically using linear finite elements. We use a fine uniform grid within the bounds and piecewise linear functions with local support, commonly called 'hat' functions. Using the weak formulation of the eigenvalue problem of the Poincaré differential operator given in (5.3), we arrive at the shifted generalized eigenvalue problem

$$\boldsymbol{K}\boldsymbol{a}^{(n)} = (\lambda_n + 1)\boldsymbol{M}\boldsymbol{a}^{(n)} \tag{5.20}$$

as described in Roustant et al. (2017, section 4.3), where the eigenvector $\boldsymbol{a}^{(n)}$ denotes the vector of coefficients used to express eigenfunction $\varphi_n$ in terms of 'hat' functions. Here $\boldsymbol{M}$ is the mass matrix, and $\boldsymbol{K}$ is the sum of mass- and stiffness matrix. After solving this problem using Matlab's builtin function `eigs`, we interpolate the discrete eigenvectors with piecewise cubic splines, prescribing zero derivatives at the interval boundaries. Then, the basis derivatives are computed using centered finite differences. While more sophisticated techniques (e.g., Hermitian $C^1$ elements, or Haar wavelets (Bujurke et al., 2008)) could of course be used to improve this numerical computation procedure, it is accurate enough for our purposes of demonstrating the usefulness of PoinCE. Eigenfunctions and eigenfunction derivatives are scaled to have unit norm with respect to the measure $\mu_i$.

### 5.3.2 Choice of the basis truncation

In practice, the series in (5.18) and (5.19) cannot include an infinite number of terms, but must be truncated to a finite expansion. We denote by $\mathcal{A} \subset \mathbb{N}^d$ the subset of multi-indices that are included in the expansion. For PCE, $\mathcal{A}$ is typically chosen to include terms up to a certain degree $p$, resulting in the so-called *total degree basis*

$$\mathcal{A}^p = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \sum_{i=1}^{d} |\alpha_i| \leq p\}. \tag{5.21}$$

To further restrict the number of terms used in the expansion, another common truncation method is *hyperbolic truncation* (Blatman and Sudret, 2011)

$$\mathcal{A}^{p,q} = \{\boldsymbol{\alpha} \in \mathbb{N}^d : \|\boldsymbol{\alpha}\|_q \leq p\}. \tag{5.22}$$

with the $\ell^q$-(quasi-)norm $\|\boldsymbol{\alpha}\|_q = \left(\sum_{i=1}^{d} \alpha_i^q\right)^{\frac{1}{q}}$ for $q \in (0, 1]$.

Since the Poincaré basis is in general not polynomial, the concept of polynomial degree cannot be used to characterize the basis functions. Instead, we use the natural order of the basis functions corresponding to the increasing sequence of Poincaré eigenvalues, which also corresponds to an increasing number of oscillations (Theorem 2; recall that the $n$th eigenfunction has $n$ zeros).

Therefore, we use the PCE terminology "degree" also for PoinCE. In particular, a degree of $\alpha_i = 0$ denotes the constant basis function $\varphi_{i,0}(x_i) = 1$ associated to the eigenvalue $\lambda_{i,0} = 0$.

Often, in practice it is not known which degree is needed for a given problem. While in theory the expansion is more accurate the larger the total degree is, in practice accuracy is limited by the number of available sample points, since the quality of the regression solution (see Section 5.3.3) depends on the ratio of sample points to basis elements. In that case, a successful strategy consists of applying *degree adaptivity*, i.e., choosing the best degree for the expansion by cross-validation (Blatman and Sudret, 2011; Lüthen et al., 2022a). This procedure is computationally inexpensive, since it only requires a new surrogate model fit for each new total degree, but no additional model evaluations. We apply leave-one-out (LOO) cross-validation together with a modification factor introduced by Chapelle et al. (2002) (Blatman and Sudret, 2011).

Both hyperbolic truncation and degree adaptivity contribute to the sparsity of the resulting expansion by identifying a suitable subset of basis functions necessary for a good approximation. Sparsity is a successful concept in regression-based PCE (Lüthen et al., 2021). Denote by $P = |\mathcal{A}|$ the number of basis elements in the truncated expansion. $\mathcal{A}$, also called *candidate basis*, contains the basis elements available for approximation. We describe below how sparse regression further selects only a subset of $\mathcal{A}$ to be *active*, i.e., have a nonzero coefficient. The final expansion might (and indeed often will) have less than $P$ active terms.

### 5.3.3 Computation of the coefficients by sparse regression

For computing the coefficients of an orthogonal expansion as in (5.18) and (5.19), there exist two main approaches. One is *projection*: the model $f$ is projected onto the basis functions, see (5.2). The resulting integral may be evaluated by Monte Carlo (MC) simulation, as done by Roustant et al. (2020a) for Poincaré chaos, or by (sparse) quadrature methods (Le Maître et al., 2002; Matthies and Keese, 2005; Constantine et al., 2012). However, note that in general MC converges slowly, while quadrature (even when sparse) is affected by the curse of dimensionality.

The second approach is *regression*, introduced for PCE by Blatman and Sudret (2008). Here, after choosing an *experimental design* $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\}$ of input points, (5.18) is discretized as

$$\boldsymbol{y} \approx \boldsymbol{\Psi}\boldsymbol{c} \tag{5.23}$$

where $\boldsymbol{y} = (f(\boldsymbol{x}^{(1)}), \dots, f(\boldsymbol{x}^{(N)}))^T$ is the vector of model evaluations, $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$ is the regression matrix with entries $\Psi_{kj} = \Phi_j(\boldsymbol{x}^{(k)})$ where $j$ refers to an enumeration of the multivariate basis $(\Phi_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathcal{A}}$, and $\boldsymbol{c}$ is the vector of expansion coefficients. The discretization of (5.19) is analogous, with a vector

$$\boldsymbol{y}_{\partial,i} = \left( \frac{\partial f}{\partial x_i}(\boldsymbol{x}^{(1)}), \dots, \frac{\partial f}{\partial x_i}(\boldsymbol{x}^{(N)}) \right)^T \tag{5.24}$$

containing model partial derivatives and a regression matrix $\boldsymbol{\Psi}_{\partial,i}$ with entries

$$\Psi_{kj}^{\partial,i} = \frac{1}{\sqrt{\lambda_{i,\boldsymbol{\alpha}_j(i)}}} \frac{\partial \Phi_j}{\partial x_i}(\boldsymbol{x}^{(k)}), \tag{5.25}$$

where $\boldsymbol{\alpha}_j(i)$ denotes the $i$th component of the $j$th basis element characterized by the multi-index $\boldsymbol{\alpha}_j$ (see also Proposition 3).

The regression problem can be solved by ordinary least squares as

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 , \tag{5.26}$$

provided that enough model evaluations are available – at least $N \geq P$, or better $N \geq kP$ with $k = 2, 3$ to avoid overfitting. Due to the rapid growth of the total-degree basis with increasing dimension and degree, this requirement on model evaluations is often too restrictive for real-world problems.

To avoid this problem, sparse regression can be used, which regularizes the problem by encouraging solutions with few nonzero coefficients (Candès and Wakin, 2008; Kougioumtzoglou et al., 2020). An example is $\ell^1$-minimization:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}\boldsymbol{c} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{c}\|_1 . \tag{5.27}$$

The $\ell^1$-norm penalizes the coefficient vector so that sparse solutions are preferred. The sparse regression formulation allows for accurate solutions even in the case $N < P$. There exist many sparse regression methods utilizing different formulations of the sparse regression problem, see e.g. Lüthen et al. (2021) for an overview of available sparse regression solvers in the context of PCE. In this work, we use the sparse solver Hybrid Least Angle Regression (Hybrid-LARS) (Blatman and Sudret, 2011; Marelli et al., 2021a).

A result by Candès and Plan (2011) on sparse recovery emphasizes the importance of *isotropy* of the row distribution of the regression matrix, i.e., the requirement that for a row $\boldsymbol{a} = (\Phi_{\boldsymbol{\alpha}_1}(\boldsymbol{x}), \ldots, \Phi_{\boldsymbol{\alpha}_P}(\boldsymbol{x}))$ of the regression matrix $\boldsymbol{\Psi}$ it holds that $\mathbb{E}\left[\boldsymbol{a}^T\boldsymbol{a}\right] = I_P$, where $I_P$ is the identity matrix of size $P$, and the expectation is with respect to the distribution of the experimental design points. If the experimental design points are chosen to follow the input distribution, the distributions of regression matrix rows for Poincaré as well as for normalized Poincaré derivative expansions are isotropic by construction due to orthonormality of the bases w.r.t. the input distribution. To improve the space-filling property of the experimental design, we use Latin Hypercube Sampling (LHS) (McKay et al., 1979) with maximin distance optimization.

### 5.3.4 Coefficients and Sobol' indices for Poincaré derivative expansions

Let $\hat{\boldsymbol{c}}^{\partial,i}$ be the solution to the sparse regression problem corresponding to the $i$-th Poincaré derivative expansion (PoinCE-der-$i$) (5.19) with regression matrix (5.25) and data vector (5.24).[2] By construction, this expansion only provides coefficients corresponding to the basis elements from the set $\mathcal{A}^i := \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i \geq 1\}$, since the partial derivatives w.r.t. $X_i$ of the basis elements $\{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i = 0\}$ are zero and therefore no coefficient value can be computed for those elements. Theoretically, for $\boldsymbol{\alpha} \in \mathcal{A}^i$ the coefficient $c_{\boldsymbol{\alpha}}^{\partial,i}$ from (5.19) is equal to the PoinCE solution $c_{\boldsymbol{\alpha}}$ from (5.18), however when estimated from a data set of finite size they will in general not coincide.

The coefficients from the set $\mathcal{A}^i$ are sufficient for computing partial variances for variable $i$ as in (5.12) and (5.13), but not enough for computing the total variance (5.11), which requires

[2] Note that in practice, we normalize and rescale the regression matrix as described in Section 5.3.1 to improve the estimation of the coefficients.

all coefficients $c_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}$, and which is needed for normalizing the partial variances to Sobol' indices.

To compute the total variance from PoinCE-der expansions, we therefore aggregate the coefficients of all $d$ PoinCE-der-$i$ expansions into one vector $\hat{\boldsymbol{c}}^{\partial,\text{avg}}$ as follows:

$$\hat{c}_{\boldsymbol{\alpha}}^{\partial,\text{avg}} = \frac{1}{\#\{i : \alpha_i \geq 1\}} \sum_{i:\alpha_i \geq 1} \hat{c}_{\boldsymbol{\alpha}}^{\partial,i} \quad \text{for each } \boldsymbol{\alpha} \in \mathcal{A} \setminus \mathbf{0}, \tag{5.28}$$

i.e., every PoinCE-der-$i$ expansion which computed a coefficient value for the basis element with index $\boldsymbol{\alpha}$ contributes equally to the averaged value. It follows that in theory, the averaged coefficient $\hat{c}_{\boldsymbol{\alpha}}^{\partial,\text{avg}}$ is equal to the PoinCE solution $c_{\boldsymbol{\alpha}}$ from (5.18), too. It can therefore be used to estimate the total variance according to (5.11).

The averaging procedure yields PoinCE-der estimates for all coefficients except for the coefficient $c_{\mathbf{0}}$ corresponding to the constant term $\psi_{\mathbf{0}}$. Let $\hat{\boldsymbol{c}}_{\boldsymbol{\alpha}}^{\partial,\text{avg}} = (\hat{c}_{\boldsymbol{\alpha}}^{\partial,\text{avg}})_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}}$ be in the form of a column vector in $\mathbb{R}^{(P-1)\times 1}$. In order to use the averaged PoinCE-der expansion also as a surrogate model, we estimate the remaining coefficient $\hat{c}_{\mathbf{0}}^{\partial,\text{avg}}$ corresponding to the constant term by ordinary least-squares on the residual $\boldsymbol{y}_{\text{res}}$:

$$\boldsymbol{y}_{\text{res}} = \boldsymbol{y} - \boldsymbol{\Psi} \begin{pmatrix} 0 \\ \hat{\boldsymbol{c}}_{\boldsymbol{\alpha}}^{\partial,\text{avg}} \end{pmatrix},$$

$$\hat{c}_{\mathbf{0}}^{\partial,\text{avg}} = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{y}_{\text{res}}^{(k)}$$

Note that the described construction uses model evaluations and partial derivatives separately. An obvious question is whether one could use these simultaneously to compute an estimate for the coefficients. Although tempting, the simple stacking of regression matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}_{\partial,i}$ into a big regression matrix (as done by Peng et al. (2016) for Hermite PCE) is not satisfactory, since the increasing norm of the basis partial derivatives (see Proposition 3) introduces an undesired weighting into the problem. The simultaneous use of evaluation and derivative data is a topic of further research.

## 5.4 Numerical results

We investigate the performance of PoinCE (both based on model evaluations and on derivatives) on two numerical examples. The focus of our study is on Sobol' sensitivity analysis, but we also investigate DGSM-based upper bounds to partial variances, validation error (relative mean-squared error) and sparsity. Our implementation is based on UQLab (Marelli and Sudret, 2014) and integrates into its PCE module (Marelli et al., 2021a).

We use the following estimation techniques to compute the Sobol' indices of the models:

- PoinCE-LARS / PoinCE-der-LARS: Poincaré expansion and Poincaré derivative expansion computed by LARS as proposed in Section 5.3.3
- PoinCE-MC / PoinCE-der-MC: As a baseline, we compare to MC-based computation using the Poincaré basis/the Poincaré partial derivative basis as in Roustant et al. (2020a)

- PCE-LARS: As a second baseline, we compare to PCE computed by LARS (with generalized polynomial chaos adapted to the respective input) (Blatman and Sudret, 2011; Marelli et al., 2021a). Sparse PCE is a state-of-the-art method for computing Sobol' indices for real-world models (Le Gratiet et al., 2017).

We do not compare to any sample-based estimates of Sobol' indices, since it is known that ANOVA-based estimation outperforms sample-based estimation. For example, Sudret (2008) and Crestaux et al. (2009) have shown that polynomial chaos-based estimators of Sobol' indices are much more efficient than Monte Carlo or quasi-Monte Carlo-based estimators (for smooth models and dimensions up to 20). Recently, Becker (2020) has shown that certain sample-based approaches can be more efficient than metamodel-based ones for screening with total Sobol' indices. However, the screening performance metrics of Becker (2020) are only based on input ranking. In contrary, our practical purpose is to perform a so-called quantitative screening which aims at providing a correct screening and a good estimation of Sobol' indices.

We do not include a comparison to gradient-enhanced PCE (Peng et al., 2016; Guo et al., 2018) because so far these methods are developed only for Gaussian, uniform and Beta input and are not immediately usable for other input distributions. Furthermore, the code of the relatively involved sampling- and preconditioning approach is not readily available. The development and comparison of gradient-enhanced PoinCE to gradient-enhanced PCE is a topic of future research.

Partial variances are normalized to Sobol' indices using the total variance. For PCE-LARS and PoinCE-LARS, the total variance is computed from the expansion coefficients as in (5.11). For PoinCE-MC and PoinCE-der-MC, we use the sample variance as done by Roustant et al. (2020a). For PoinCE-der-LARS, the total variance is obtained by the procedure detailed in Section 5.3.4.

The DGSM-based upper bound to the total partial variances is computed from (5.17) using the coefficients of the PoinCE derivative expansions as described in Section 5.3.4. Note that the inequalities in (5.17) are analytical bounds that do not necessarily hold for the estimated quantities.

For uniform and Gaussian input variables, the analytical expression for the Poincaré basis functions is used, while for all others, the basis functions are computed numerically using a resolution of $10^3$ points for the uniform grid within the given bounds (see Section 5.3.1).[3]

### 5.4.1 Dyke cost model

Our first application is a simplified analytical model computing the cost associated to a dyke that is to be constructed along a stretch of river to prevent flooding (Iooss and Lemaître, 2015; Roustant et al., 2020a). Its output is the cost in million euros given by

$$Y = \mathbb{1}_{S>0} + \left[0.2 + 0.8\left(1 - \exp^{-\frac{1000}{S^4}}\right)\right] \cdot \mathbb{1}_{S\leq 0} + \frac{1}{20}\left(8 \cdot \mathbb{1}_{H_d\leq 8} + H_d \cdot \mathbb{1}_{H_d>8}\right) \tag{5.29}$$

where $S$ is the maximal annual overflow and $H_d$ is the dyke height. Here, the first term represents the cost of the consequences of a flooding event, the second describes the maintenance costs,

[3]For the flood model, the change in the resulting Sobol' indices when instead using a grid with $10^2$ or $10^4$ points is in the order of $10^{-4}$ or $10^{-6}$, respectively.

and the third is associated to the construction cost. $S$ is computed from the river characteristics detailed in Table 5.1 via the 1D Saint-Venant equations under several simplifying assumptions as follows:

$$S = \left( \frac{Q}{BK_s \sqrt{\frac{Z_m - Z_v}{L}}} \right)^{0.6} + Z_v - H_d - C_b \tag{5.30}$$

The model Eq. (5.29) is continuous and piecewise $C^1$, and therefore in $H^1$. It has 8 input variables, of which $Q, K_s, Z_v$ and $H_d$ are important, and $C_b, Z_m, L$ and $B$ are unimportant (see also the last two columns of Table 5.1).

***Table 5.1:*** *Input variables to the dyke cost model and reference values of first-order and total Sobol' indices. The reference values were obtained to good precision by Monte-Carlo-based Sobol' index estimation using a large sample (Roustant et al., 2020a).*

| Input | Function | Unit | Distribution | $S_i$ | $S_i^{\text{tot}}$ |
|---|---|---|---|---|---|
| $Q$ | Maximal annual flowrate | $\text{m}^3/\text{s}$ | Gumbel $\mathcal{G}(1013, 558)$ truncated to $[500, 3000]$ | 0.358 | 0.483 |
| $K_s$ | Strickler coefficient | – | Gaussian $\mathcal{N}(30, 8^2)$ truncated to $[15, +\infty]$ | 0.156 | 0.252 |
| $Z_v$ | River downstream level | m | Triangular $\mathcal{T}(49, 51)$ | 0.167 | 0.223 |
| $Z_m$ | River upstream level | m | Triangular $\mathcal{T}(54, 56)$ | 0.003 | 0.008 |
| $H_d$ | Dyke height | m | Uniform $\mathcal{U}([7, 9])$ | 0.119 | 0.177 |
| $C_b$ | Bank level | m | Triangular $\mathcal{T}(55, 56)$ | 0.029 | 0.040 |
| $L$ | Length of river stretch | m | Triangular $\mathcal{T}(4990, 5010)$ | 0.000 | 0.000 |
| $B$ | River width | m | Triangular $\mathcal{T}(295, 305)$ | 0.000 | 0.000 |

The dyke cost model has been used by Roustant et al. (2020a) to demonstrate the performance of projection-based PoinCE. We compare the new regression-based methods PoinCE-LARS and PoinCE-der-LARS with the projection-based counterparts PoinCE-MC and PoinCE-der-MC, and additionally with the standard PCE method PCE-LARS. The projection-based estimates use a basis of total degree 2, while the regression-based estimates use degree adaptivity with a degree of up to 5 (remember that for PoinCE, the degree corresponds to the ordering of the eigenfunctions by the magnitude of the eigenvalues). The experimental design is sampled by LHS with maximin distance optimization. Gradients are computed here by finite differences. For each size of the experimental design, we perform 50 independent repetitions. We display the resulting estimates in the form of boxplots. We show results only for three input variables: the most important variable $Q$, the low-importance variable $C_b$, and the unimportant variable $B$. The results for the remaining input variables can be found in Section 5.A.

#### 5.4.1.1 Comparison of MC-based and regression-based computation of PoinCE(der)

First we investigate the two different ways to compute PoinCE: projection-based as in Roustant et al. (2020a) versus sparse regression-based as described in Section 5.3. Figure 5.1 and 5.2 show estimates for first-order and total Sobol' indices. We observe that in all cases the regression-based estimates have a smaller variance than the corresponding projection-based estimates.

Also, the median of the regression-based estimates is closer to the true Sobol' index value than the median of the projection-based estimates. Note that while the regression-based estimates use a degree-adaptive basis of $p \leq 5$, the projection-based estimates use a fixed degree of only $p = 2$. While this choice introduces a certain bias to the projection-based estimates, a larger value for $p$ leads to unfeasibly large variance for those estimates. This is because the coefficients of higher-order terms cannot be estimated precisely with few experimental design points, which makes the overall estimate less precise.

We also observe that regression-based estimates are often clustered around the true Sobol' index already for very small experimental design sizes. This might be due to the generally smaller variance of sparse-regression-based coefficient estimates compared to MC-based estimates, as well as due to the choice of normalization factor for Sobol' indices (coefficient-based variance Eq. (5.12) vs. sample variance as described in the beginnning of Section 5.4). Since regression generally leads to more precise estimates than projection, in the remainder of this paper we focus on regression-based PoinCE estimates.

Furthermore, as already observed by Roustant et al. (2020a), PoinCE-der estimates for Sobol' indices have a smaller variance than PoinCE estimates. In the case of projection-based estimates, this is the case if the derivative has a smaller variance than the original model. In the case of regression, the explanation might be that PoinCE-$i$-der has to compute less coefficients than PoinCE for the same number of experimental design points ($\{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i > 0\}$ vs. $\mathcal{A}$), which can result in a more precise estimate of the true coefficient values.

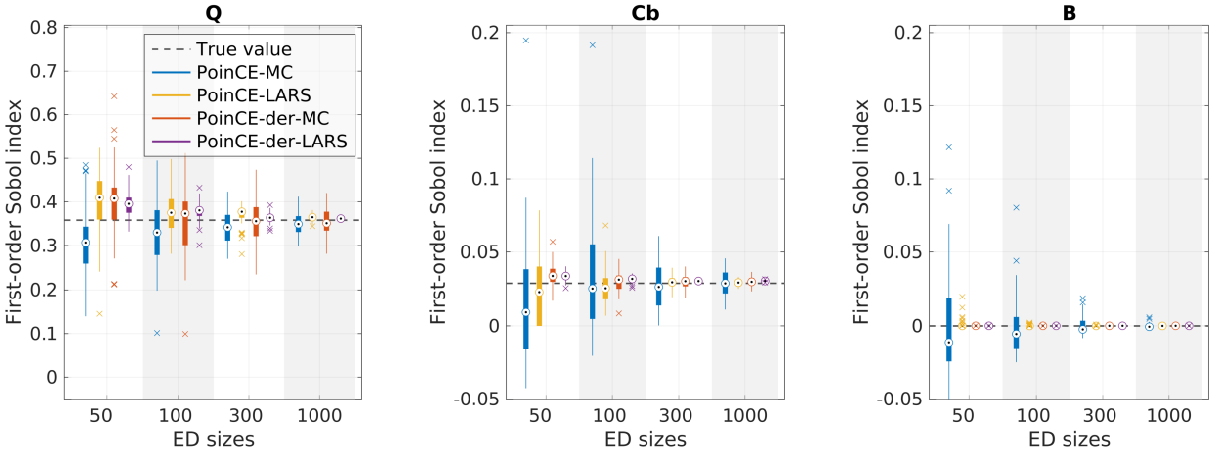



***Figure 5.1:*** *Comparison of PoinCE estimates of first-order Sobol' indices for the dyke cost model. Degree $p = 2$ for the MC-based estimates and $p \leq 5$ (degree-adaptive) for the regression-based estimates. Results for the remaining variables are displayed in Figure 5.A.1 in the appendix.*

#### 5.4.1.2 Comparison of regression-based PoinCE-der with PCE and the DGSM-based upper bound

Next, we investigate the performance of regression-based PoinCE compared to state-of-the-art PCE, and the usefulness of the DGSM-based upper bound to partial variances derived in (5.17). The corresponding results, unnormalized[4] estimates for first-order and total Sobol' indices, are

[4]We show unnormalized indices because the DGSM-based upper bound is not normalized.

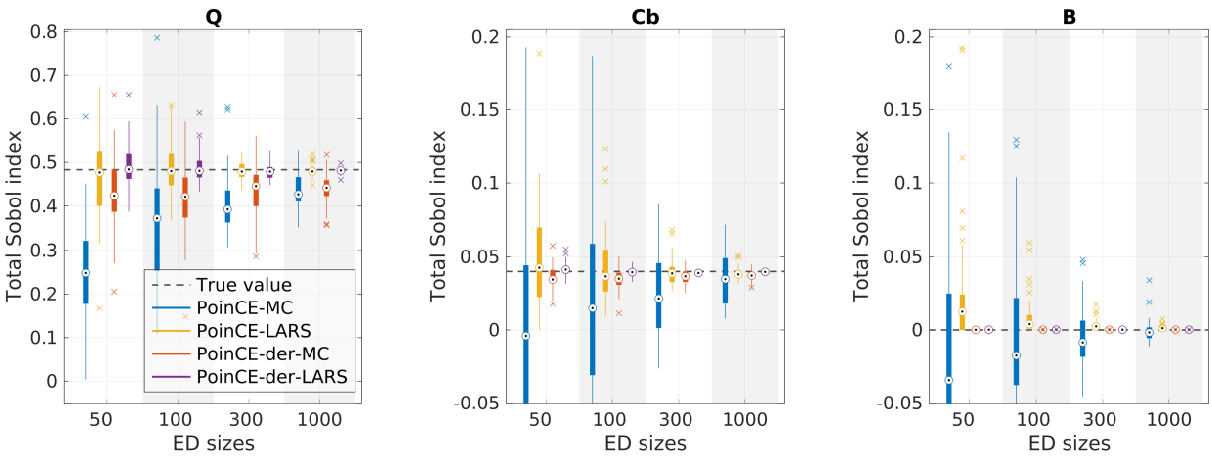



***Figure 5.2:*** *Comparison of PoinCE estimates of total Sobol' indices for the dyke cost model. Degree $p = 2$ for the MC-based estimates and $p \leq 5$ (degree-adaptive) for the regression-based estimates. Results for the remaining variables are displayed in Figure 5.A.2 in the appendix.*

displayed in Figure 5.3 and 5.4. Because PoinCE-der achieves more accurate estimates than PoinCE, we compute the DGSM-based upper bound using the PoinCE-der-$i$ coefficients. For total Sobol' indices, we also include a precise Monte Carlo estimate for the DGSM-based upper bound (using $10^7$ derivative samples) computed from (5.9) and the second inequality of (5.17).

We make the following observations: the PCE-LARS estimates are generally very similar to the PoinCE-LARS estimates, but the latter often have a slightly larger range. The similarity might be because both rely on model evaluations only. However, the respective basis functions have a very different shape (for inputs that do not follow a Gaussian distribution). In particular, the PoinCE basis functions by construction obey Neumann boundary conditions, i.e., have zero derivative on the boundary.

As observed before for normalized indices, PoinCE-der performs better than PoinCE: the median is closer to the true value, and the range is smaller. This effect is especially pronounced for low-importance variables. In 8 dimensions, a PoinCE-der expansion of degree 5 has 495 terms while the total-degree basis of PCE and PoinCE has 1287 terms. This means that a PoinCE-der expansion has to estimate less than half of the coefficients. PoinCE-der generally gives a tighter "lower bound" than PCE (but note that the estimates are not guaranteed to be a lower bound).

By construction (5.17), the DGSM-based upper bound estimate is larger than or equal to the corresponding total Sobol' index estimate. However, it would be an upper bound to the true Sobol' index value only if the full infinite expansion was used. This is visible in Figure 5.A.4 and 5.4: for some inputs, the upper bound estimate almost coincides with the Sobol' index estimate, and is smaller than the true Sobol' index.

For some inputs, such as $K_s$ and especially $H_d$ (see Figure 5.A.4), the DGSM-based upper bound is not tight. An explanation for the large gap between the Sobol' index and the upper bound for $H_d$, using (5.17), might be that $H_d$ is responsible for a kink in the model (through the last term of (5.29)), while the Poincaré basis is differentiable in $H_d$ (cosine functions). Therefore, its expansion needs high-order terms. From (5.17) it follows that even if the PoinCE-der found the true coefficients, the estimated upper bound is not tight due to the quickly-growing eigenvalues.

Finally, we see that in many cases the gap between the estimated upper bound estimate and its true value is larger than the gap between Sobol' index estimate and its true value. The reason might be that some higher-order terms are still missing from the considered expansion. Due to the eigenvalue factor involved in the estimate of the upper bound (5.17), this has a larger influence on the upper bound than on the Sobol' index estimate.

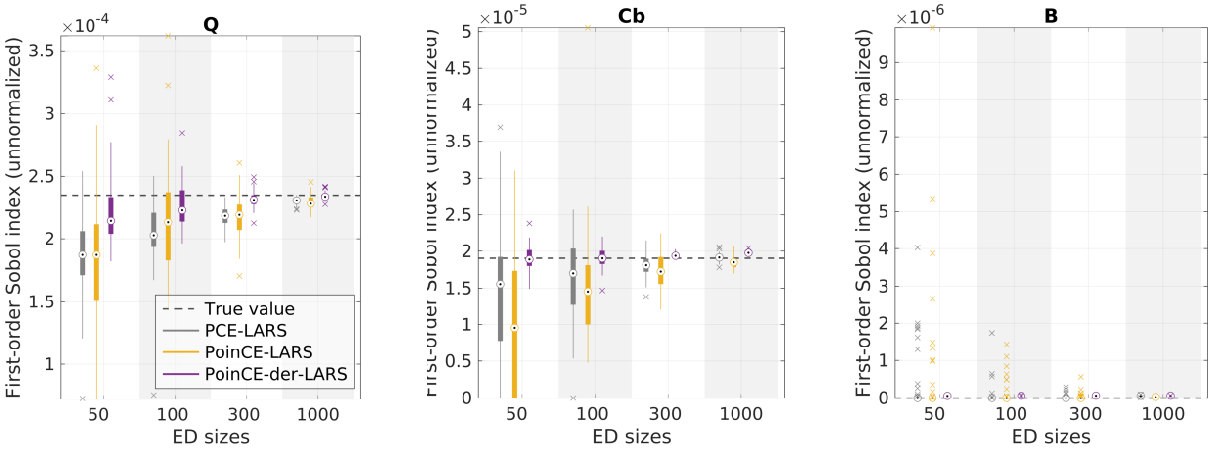



***Figure 5.3:*** *Estimates of unnormalized first-order Sobol' indices for the dyke cost model ($p \leq 5$). Boxplots: in grey the PCE-based estimates. The dashed line ("True value") denotes a high-precision estimate for the unnormalized first-order Sobol' index. Results for the remaining input variables can be found in Figure 5.A.3 in Section 5.A.*

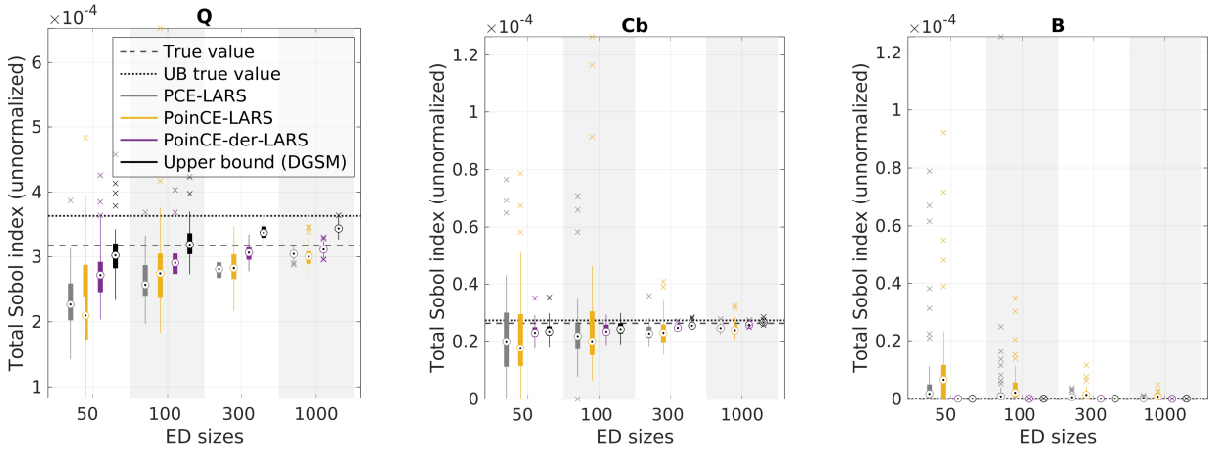



***Figure 5.4:*** *Estimates of unnormalized total Sobol' indices for the dyke cost model ($p \leq 5$). Boxplots: in grey the PCE-based estimates and in black the DGSM-based upper bound from (5.17). Lines: the dashed line ("True value") denotes a high-precision estimate for the unnormalized total Sobol' index, while the dotted line ("UB true value") is a MC-based high-precision estimate for the DGSM-based upper bound. Results for the remaining input variables can be found in Figure 5.A.4 in Section 5.A.*

#### 5.4.1.3 Comparison of total variance, relative mean-squared error, and sparsity

To estimate Sobol' indices precisely, it is crucial to have a good estimate for the total variance. For PCE-LARS, PoinCE-LARS, and PoinCE-der-LARS, this value can directly be computed

from the expansion coefficients, while for PoinCE computed by projection, we are using the empirical variance, as detailed in the beginning of Section 5.4. In Figure 5.5a we display the scatter of the variance estimates (50 replications). The empirical estimate has the largest variation, while PoinCE-der-LARS has the smallest. PCE-LARS and PoinCE-LARS underestimate the total variance more than PoinCE-der-LARS. This is likely one reason for the good performance of PoinCE-der for the estimation of Sobol' indices: a more accurate total variance leads to more accurate Sobol' indices.

Interestingly, while PoinCE performs well for the estimation of Sobol' indices, this is not true for the generalization error, given by the relative mean-squared error

$$\text{RelMSE} = \frac{\mathbb{E}_X\left[(f(X) - f^{\text{surr}}(X))^2\right]}{\text{Var}_X\left[f(X)\right]} \tag{5.31}$$

with the surrogate model $f^{\text{surr}}$. The RelMSE is computed by Monte Carlo integration on a validation set of size $10^6$ sampled from the input distribution $\mu$. In Figure 5.5b we display boxplots of estimates for the generalization error on a validation set of size $10^6$ (mean-squared error normalized by the variance of the validation set). PoinCE-der attains a smaller relative MSE than PoinCE. PCE shows faster convergence behavior than both, and attains a smaller relative MSE than PoinCE. PoinCE-der performs better than PCE for the two small experimental design sizes, which shows that the information brought by derivatives might be especially useful when data is scarce.

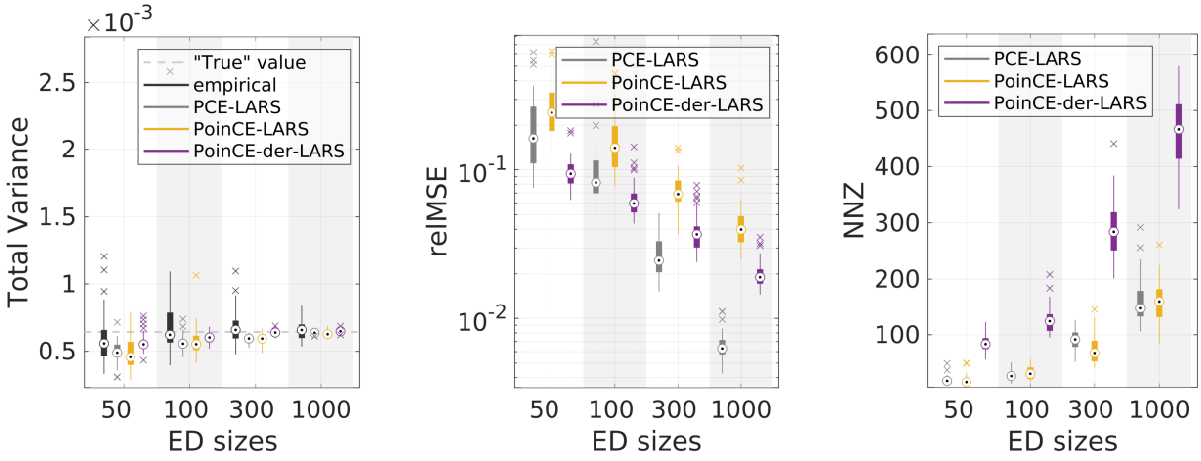
Finally, we display the number of nonzero coefficients of each expansion in Figure 5.5c. PCE and PoinCE have a similar number of active coefficients, while PoinCE-der has considerably more active coefficients. This corresponds to a better validation error only for the smallest experimental design size (Figure 5.5b). A possible explanation is the following: using the derivative information to compute PoinCE-der, the same amount of data is used for a (truncated) expansion containing less terms than the PoinCE expansion has (since some terms, which do not contain the variable for which the partial derivative is taken, drop out). In this way, more coefficients can be computed.

### 5.4.2 Mascaret data set

Our second application focuses on a phenomenological and industrial simulation model, called Mascaret (Goutal et al., 2012), based on a 1D solver of the Saint Venant equations and aiming at computing water height for river flood events. The studied case, taken from Petit et al. (2016) and also studied in Roustant et al. (2017), is the French Vienne river in permanent regime whose uncertain input data concern flowrate, several physical parameters and geometrical data (transverse river profiles). 37 independent inputs have then been considered as random variables:

- 12 Strickler coefficients of the main channel $K_{s,c}^i$, uniform in $[20, 40]$;
- 12 Strickler coefficients of the flood plain $K_{s,p}^i$, uniform in $[10, 30]$;
- 12 slope perturbations $dZ^i$, standard Gaussian with bounds $[-3, 3]$;
- 1 discharge value $Q$, Gaussian with zero mean and standard deviation 50, bounds $[-150, 50]$.

The derivatives of the model output with respect to these 37 inputs have been efficiently (with a cost independent of the number of inputs) computed by using the adjoint model of Mascaret




*(a) Estimates of the variance*

*(b) Relative mean-squared error*

*(c) Number of nonzero coefficients*

***Figure 5.5:*** *Dyke cost model: Comparison of PCE and PoinCE(-der) with respect to the following metrics: estimation of total variance, relative mean-squared error, and number of nonzero coefficients in the respective expansions.*

(Demangeon et al., 2015). This adjoint model has been obtained by automatic differentiation (Griewank and Walther, 2008) using the automatic differentiation software Tapenade (Hascoët and Pascual, 2013). A large-size Monte Carlo sample ($n = 20\,000$) is available from the study of Petit et al. (2016). This data set contains all the values of the 37 inputs, the water height as output and the 37 partial derivatives of the output (one derivative with respect to each input). Note that this sample, which has a very large size, has been obtained during a research work for a demonstrative purpose. In industrial practice, the aim is to use the minimal possible sample size: it is expected to use methods able to deal with sample sizes of the order of one hundred.

Previous studies on this data set (Petit et al., 2016; Roustant et al., 2017) have identified 32 of the 37 inputs as noninfluential. In our study, we display results for the 5 remaining inputs ($K^{11}_{s,c}$, $K^{12}_{s,c}$, $dZ^{11}$, $dZ^{12}$, and $Q$) and for one of the noninfluential inputs ($K^{1}_{s,c}$). We choose a basis with hyperbolic truncation using $q = 0.5$, and degree adaptivity $p = 1, 2, \ldots, 8$. We analyze several experimental design sizes ranging from 30 to 300. For each experimental design size, we run 30 replications, sampling the design randomly without replacement from the given full data set. "True" values for Sobol' indices and total variance are computed from a PCE using all 20 000 points.

#### 5.4.2.1 Comparison of regression-based PoinCE(der) with PCE and the DGSM-based upper bound

Estimates of first-order and total Sobol' indices are displayed in Figure 5.6 and 5.7. We display results for regression-based PCE, PoinCE, and PoinCE-der. In addition, we display the upper bound computed as in (5.17), computed based on PoinCE-der coefficients and normalized by the PoinCE-der total variance. We observe that for the non-influential variable $K^{1}_{s,c}$ (and indeed all other 31 non-influential variables), derivative-based PoinCE correctly identify a total and first-order Sobol' index of 0. Overall, PoinCE and PCE show very similar results, with PoinCE

having slightly larger variance in a few cases. For some variables such as $dZ^{11}$ and $Q$, the DGSM-based estimate of the upper bound almost coincides with the PoinCE-der estimate. Overall, we observe that PoinCE-der estimates have smaller variance than PCE and PoinCE for the important variables $K_{s,c}^{11}, dZ^{11}$, and $Q$, even already for 30 experimental design points. For low-importance variables such as $K_{s,c}^{12}$ and $dZ^{12}$, PoinCE-der correctly identifies a value away from zero already for the smallest experimental design, while half of the PCE and PoinCE estimates are zero. For small experimental design sizes, the PoinCE-der estimates also have a smaller bias than the PCE and PoinCE estimates. Sometimes the PoinCE-der estimates seem to systematically over- or underestimate the true Sobol' index by a small amount. However, note that the "true" value was computed by a PCE (based on all 20 000 points), and might therefore itself be slightly inaccurate.

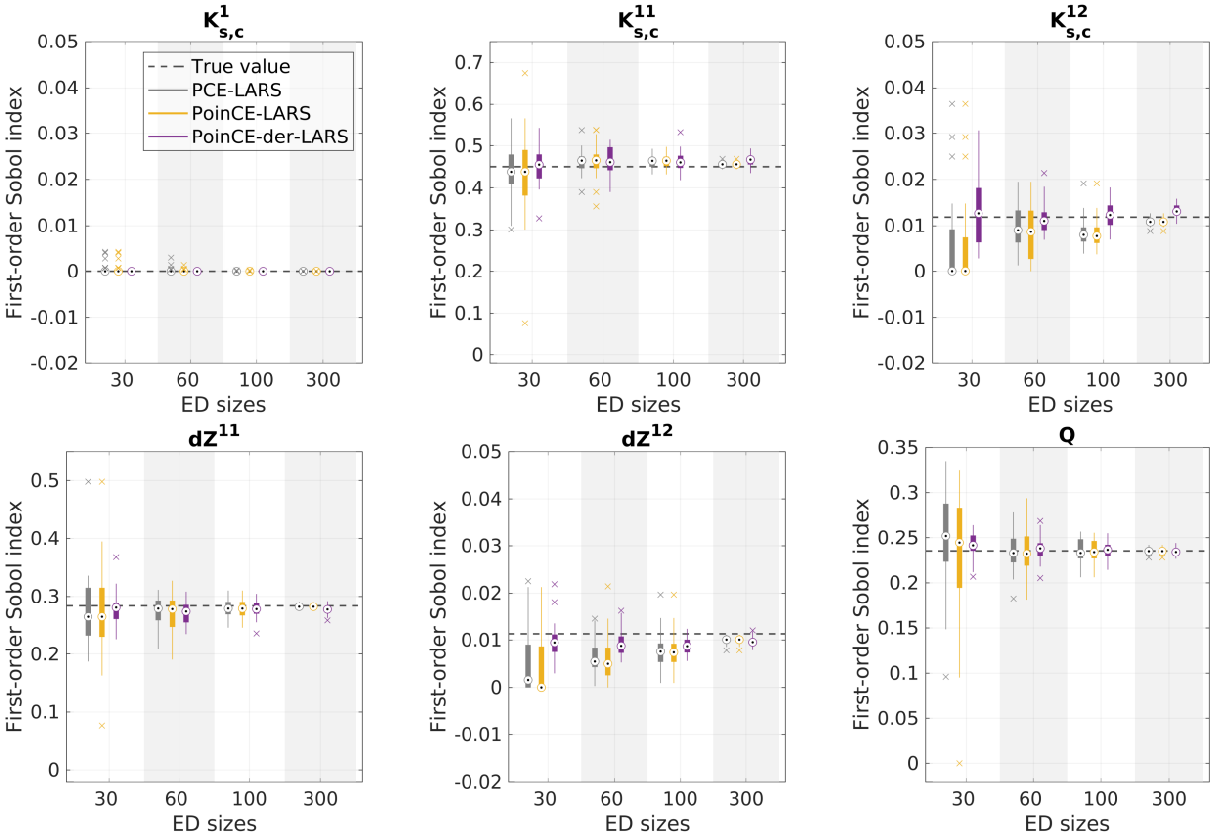



***Figure 5.6:*** *First-order Sobol' indices for the Mascaret data set (30 replications). "True" values computed from a PCE using all* 20 000 *points.*

### 5.4.2.2 Comparison of total variance and relative mean-squared error

In Figure 5.8a we display various estimates for the total variance. We observe again that PoinCE-der yields an estimate with smaller variance and less bias than PoinCE and PCE. PoinCE and PCE both have smaller variance than the empirical estimate, but generally underestimate the total variance.

While PoinCE-der estimates Sobol' indices and total variance well, we observe in Figure 5.8b showing the relative MSE that PCE and PoinCE are performing better as global surrogate

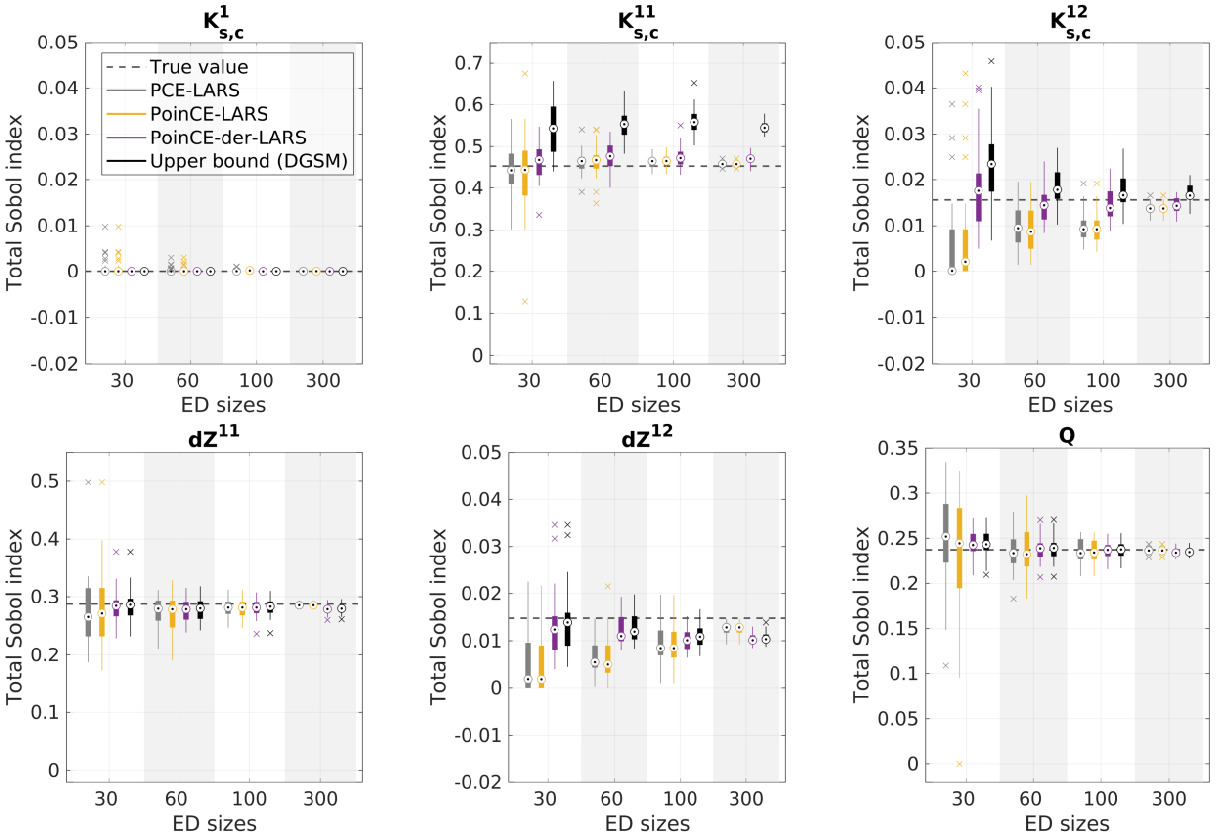



***Figure 5.7:*** *Total Sobol' indices for the Mascaret data set (30 replications). "True" values computed from a PCE using all* 20 000 *points.*

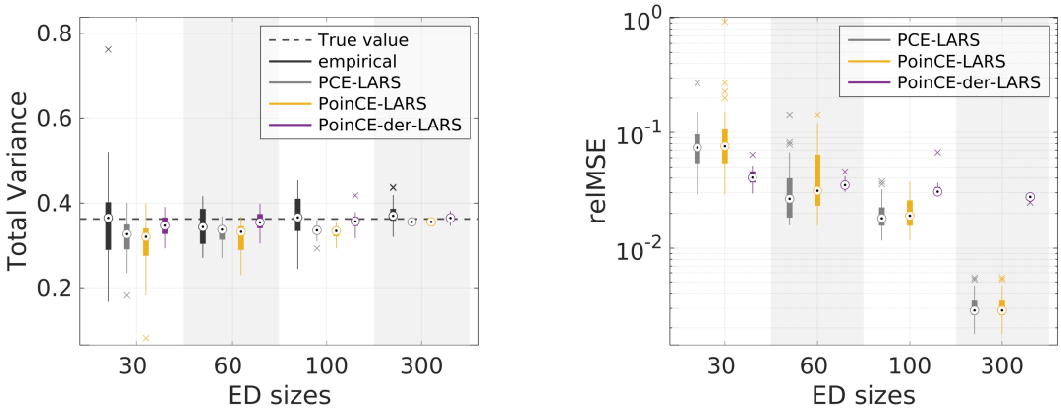
models: their model approximation error is for large experimental designs almost an order of magnitude better than for PoinCE-der.

## 5.5 Conclusion

In this paper we studied PoinCE, an expansion in terms of the Poincaré basis, which is an orthonormal basis of $L^2(\mu)$ with the unique property that all its partial derivatives form again an orthogonal basis for the same space. We provided a proof of this property as well as a few analytical results as direct consequences. In particular, we showed how upper and lower bounds for partial variances can be obtained analytically from PoinCE coefficients.

We described the computation of PoinCE and Poincaré derivative expansions by sparse regression and applied the method to two numerical examples. We found that while PoinCE does not outperform PCE in terms of validation error, it can be advantageous for estimating Sobol' indices in the low-data regime. PoinCE is therefore a valuable tool if model derivatives are cheaply available (e.g., by automatic differentiation or as a by-product of the simulation). Taking partial derivatives reduces the size of the truncated basis especially for high-dimensional, low-order total-degree bases, which gives an advantage to derivative-based PoinCE over expansions relying on model evaluations.

Future work on the topic of PoinCE will investigate the simultaneous use of model evaluations




*(a) Estimates of the variance* *(b) Relative mean-squared error*

***Figure 5.8:*** *Mascaret data set. Comparison of PCE and PoinCE(-der) with respect to the following metrics: estimation of total variance, and relative mean-squared error. "True" value of total variance computed from a PCE using all* 20 000 *points.*

and derivatives for the computation of the coefficients, and compare to the related topic of gradient-enhanced PCE.

## Funding

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. Part of this research was conducted within the frame of the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Etienne, University of Grenoble, University of Nice, University of Toulouse) around advanced methods for computer experiments. Support from the ANR-3IA Artificial and Natural Intelligence Toulouse Institute is gratefully acknowledged.

## Declaration of competing interests

No competing interests.

## References

Antoniadis, A. (1984). Analysis of variance on function spaces. *Statistics*, 15(1):59–71.

Bakry, D., Gentil, I., and Ledoux, M. (2014). *Analysis and geometry of Markov diffusion operators*, volume 348 of *Fundamental Principles of Mathematical Sciences*. Springer, Cham.

Becker, W. (2020). Metafunctions for benchmarking in sensitivity analysis. *Reliab. Eng. Sys. Safety*, 204:107189.

Blatman, G. and Sudret, B. (2008). Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *C. R. Mécanique*, 336(6):518–523.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys*, 230:2345–2367.

Borgonovo, E. and Plischke, E. (2016). Sensitivity analysis: A review of recent advances. *Eur. J. Oper. Res.*, 248(3):869–887.

Bujurke, N. M., Salimath, C. S., and Shiralashetti, S. C. (2008). Computation of eigenvalues and solutions of regular Sturm–Liouville problems using Haar wavelets. *J. Comput. Appl. Math.*, 219(1):90–101.

Candès, E. J. and Plan, Y. (2011). A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Inform. Theory*, 57(11):7235–7254.

Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Process. Mag.*, 25(2):21–30.

Chapelle, O., Vapnik, V., and Bengio, Y. (2002). Model selection for small sample regression. *Machine Learning*, 48(1):9–23.

Constantine, P. G., Eldred, M. S., and Phipps, E. T. (2012). Sparse pseudospectral approximation method. *Comput. Methods Appl. Mech. Engrg.*, 229:1–12.

Crestaux, T., Le Maître, O., and Martinez, J.-M. (2009). Polynomial chaos expansion for sensitivity analysis. *Reliab. Eng. Sys. Safety*, 94(7):1161–1172.

Demangeon, F., Goeury, C., Zaoui, F., Goutal, N., Pascual, V., and Hascoët, L. (2015). Algorithmic differentiation applied to the optimal calibration of a shallow water model. *La Houille Blanche - Revue internationale de l'eau*, 102(4):57–65.

Efron, B. and Stein, C. (1981). The Jacknife estimate of variance. *Ann. Stat.*, 9(3):586–596.

Ernst, O., Mugler, A., Starkloff, H.-J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.*, 46(02):317–339.

Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Computer Science and Data Analysis Series. Chapman & Hall/CRC, Boca Raton, Florida, USA.

Gejadze, I., Malaterre, P.-O., and Shutyaev, V. (2019). On the use of derivatives in the polynomial chaos based global sensitivity and uncertainty analysis applied to the distributed parameter models. *J. Comput. Phys.*, 381:218–245.

Ghanem, R. G. and Spanos, P. (1991). *Stochastic finite elements – A spectral approach*. Springer Verlag, New York. (Reedited by Dover Publications, Mineola, 2003).

Goutal, N., Lacombe, J.-M., Zaoui, F., and El-Kadi-Abderrezak, K. (2012). MASCARET: a 1-D open-source software for flow hydrodynamic and water quality in open channel networks. In Murillo Muñoz, R., editor, *River Flow 2012: Proceedings of the International Conference on Fluvial Hydraulics*, volume 2, pages 1169–1174, San José, Costa Rica. CRC Press.

Griewank, A. and Walther, A. (2008). *Evaluating derivatives: Principles and techniques of automatic differentiation*. SIAM Philadelphia.

Guo, L., Narayan, A., and Zhou, T. (2018). A gradient enhanced $\ell^1$-minimization for sparse approximation of polynomial chaos expansions. *J. Comput. Phys.*, 367:49–64.

Hascoët, L. and Pascual, V. (2013). The Tapenade automatic differentiation tool: Principles, model and specification. *ACM T. Math. Software*, 39(3):1–43.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Ann. Math. Stat.*, 19:293–325.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of non linear models. *Reliab. Eng. Sys. Safety*, 52:1–17.

Iooss, B. and Lemaître, P. (2015). *Uncertainty Management in Simulation-Optimization of Complex Systems*, volume 59 of *Operations Research/Computer Science Interfaces Series (ORCS)*, chapter A review on global sensitivity analysis methods. Springer.

Jakeman, J. D., Eldred, M. S., and Sargsyan, K. (2015). Enhancing $\ell_1$-minimization estimates of polynomial chaos expansions using basis selection. *J. Comput. Phys.*, 289:18–34.

Kougioumtzoglou, I. A., Petromichelakis, I., and Psaros, A. F. (2020). Sparse representations and compressive sampling approaches in engineering mechanics: A review of theoretical concepts and diverse applications. *Prob. Eng. Mech.*, 61:103082.

Kucherenko, S. and Iooss, B. (2017). Derivative-based global sensitivity measures. In *Handbook of Uncertainty Quantification*, pages 1241–1263. Springer International Publishing.

Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C., and Shah, N. (2009). Monte Carlo evaluation of derivative-based global sensitivity measures. *Reliab. Eng. Syst. Safety*, 94:1135–1148.

Kwon, K. H. and Lee, D. W. (2003). Orthogonal functions satisfying a second-order differential equation. *J. Comput. Appl. Math.*, 153(1-2):283–293.

Lamboni, M., Iooss, B., Popelin, A.-L., and Gamboa, F. (2013). Derivative-based global sensitivity measures: general links with Sobol' indices and numerical tests. *Math. Comput. Simul.*, 87:44–54.

Le Gratiet, L., Marelli, S., and Sudret, B. (2017). *Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes*, pages 1289–1325. Springer.

Le Maître, O., Reagan, M., Najm, H., and Ghanem, R. (2002). A stochastic projection method for fluid flow - II. random process. *J. Comp. Phys.*, 181:9–44.

Li, Y., Anitescu, M., Roderick, O., and Hickernell, F. (2011). Orthogonal bases for polynomial regression with derivative information in uncertainty quantification. *Int. J. Uncertainty Quantification*, 1:297–320.

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif.*, 9(2):593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification*, 12(3):49–74.

Marelli, S., Lüthen, N., and Sudret, B. (2021). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-104.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Matthies, H. G. and Keese, A. (2005). Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194:1295–1331.

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*, 2:239–245.

Mikolas, M. (1955). Über gewisse Eigenschaften orthogonaler Systeme der Klasse $L^2$ und die Eigenfunktionen Sturm-Liouvillescher Differentialgleichungen. *Acta Math. Acad. Sci. H.*, 6:147–190.

Oladyshkin, S. and Nowak, W. (2012). Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliab. Eng. Syst. Saf.*, 106:179–190.

Peng, J., Hampton, J., and Doostan, A. (2016). On polynomial chaos expansion via gradient-enhanced $\ell_1$-minimization. *J. Comput. Phys.*, 310:440–458.

Petit, S., Zaoui, F., Popelin, A.-L., Goeury, C., and Goutal, N. (2016). Couplage entre indices à base de dérivées et mode adjoint pour l'analyse de sensibilité globale. Application sur le code Mascaret. Technical report. Preprint.

Prieur, C. and Tarantola, S. (2017). Variance-based sensitivity analysis: Theory and estimation algorithms. In *Springer Handbook on Uncertainty Quantification*, pages 1217–1239. Springer.

Razavi, S., Jakeman, A., Saltelli, A., Prieur, C., Iooss, B., Borgonovo, E., Plischke, E., Lo Piano, S., Iwanaga, T., Becker, W., Tarantola, S., Guillaume, J., Jakeman, J., Gupta, H., Melillo, N., Rabiti, G., Chabridon, V., Duan, Q., Sun, X., Smith, S., Sheikholeslami, R., Hosseini, N., Asadzadeh, M., Puy, A., Kucherenko, S., and Maier, H. (2021). The future of sensitivity analysis: An essential discipline for systems modelling and policy making. *Environ. Modell. Softw.*, 137(104954).

Roderick, O., Anitescu, M., and Fischer, P. (2010). Polynomial regression approaches using derivative information for uncertainty quantification. *Nucl. Sci. Eng.*, 164:122–139.

Roustant, O., Barthe, F., and Iooss, B. (2017). Poincaré inequalities on intervals – application to sensitivity analysis. *Electron. J. Stat.*, 11(2):3081–3119.

Roustant, O., Gamboa, F., and Iooss, B. (2020). Parseval inequalities and lower bounds for variance-based sensitivity indices. *Electron. J. Statist.*, 14(1):386–412.

Simon, B. (2010). *Szegő's theorem and its descendants: spectral theory for L2 perturbations of orthogonal polynomials.* Princeton University Press.

Smith, R. C. (2014). *Uncertainty Quantification: Theory, Implementation and Applications.* SIAM Computational Science and Engineering.

Sobol, I. and Gresham, A. (1995). On an alternative global sensitivity estimator. In *Proceedings of SAMO 1995*, pages 40–42, Belgirate.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Modeling & Comp. Exp.*, 1(4):407–414.

Sobol', I. M. and Kucherenko, S. (2009). Derivative based global sensitivity measures and their link with global sensitivity indices. *Math. Comput. Simul.*, 79(10):3009–3017.

Sudret, B. (2006). Global sensitivity analysis using polynomial chaos expansions. In Spanos, P. and Deodatis, G., editors, *Proc. 5th Int. Conf. on Comp. Stoch. Mech (CSM5), Rhodos, Greece.*

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Saf.*, 93:964–979.

Sudret, B. and Mai, C. V. (2015). Computing derivative-based global sensitivity measures using polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 134:241–250.

Szegö, G. (1939). *Orthogonal polynomials*, volume 23. American Mathematical Society.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *J. Comput. Phys*, 388:601–623.

Wiener, N. (1938). The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936.

Xiu, D. and Karniadakis, G. E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644.

Zettl, A. (2010). *Sturm-Liouville theory.* Number 121. American Mathematical Society.

# Appendix

## 5.A Additional results

In Figure 5.A.1 to 5.A.4, we show additional results for the dyke cost model, namely Sobol' index estimates (normalized and unnormalized) for the remaining five input variables. For the corresponding discussion, see Section 5.4.1.

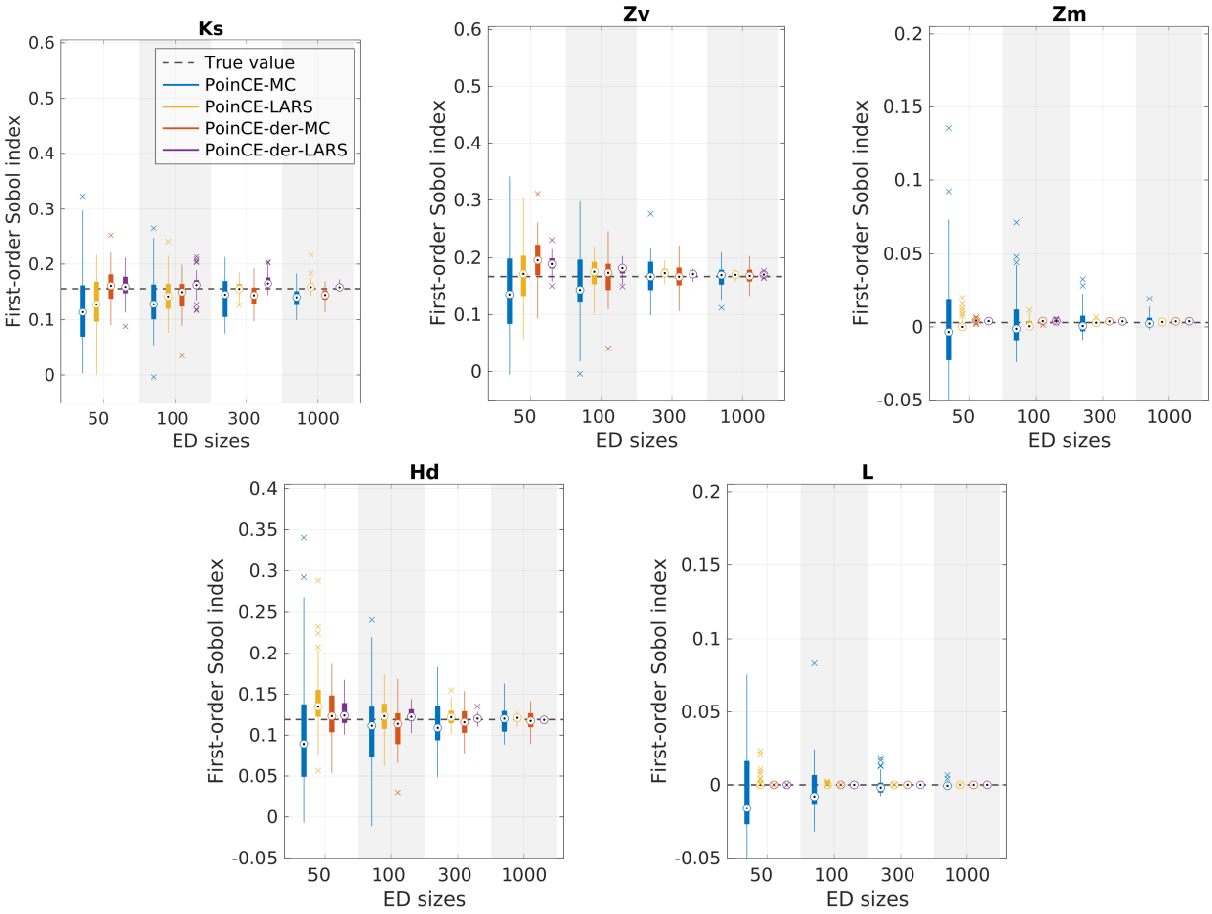



***Figure 5.A.1:*** *Comparison of PoinCE estimates of first-order Sobol' indices for the dyke cost model. Degree $p = 2$ for the MC-based estimates and $p \leq 5$ (degree-adaptive) for the regression-based estimates. See also Figure 5.1 in the main part of the paper.*

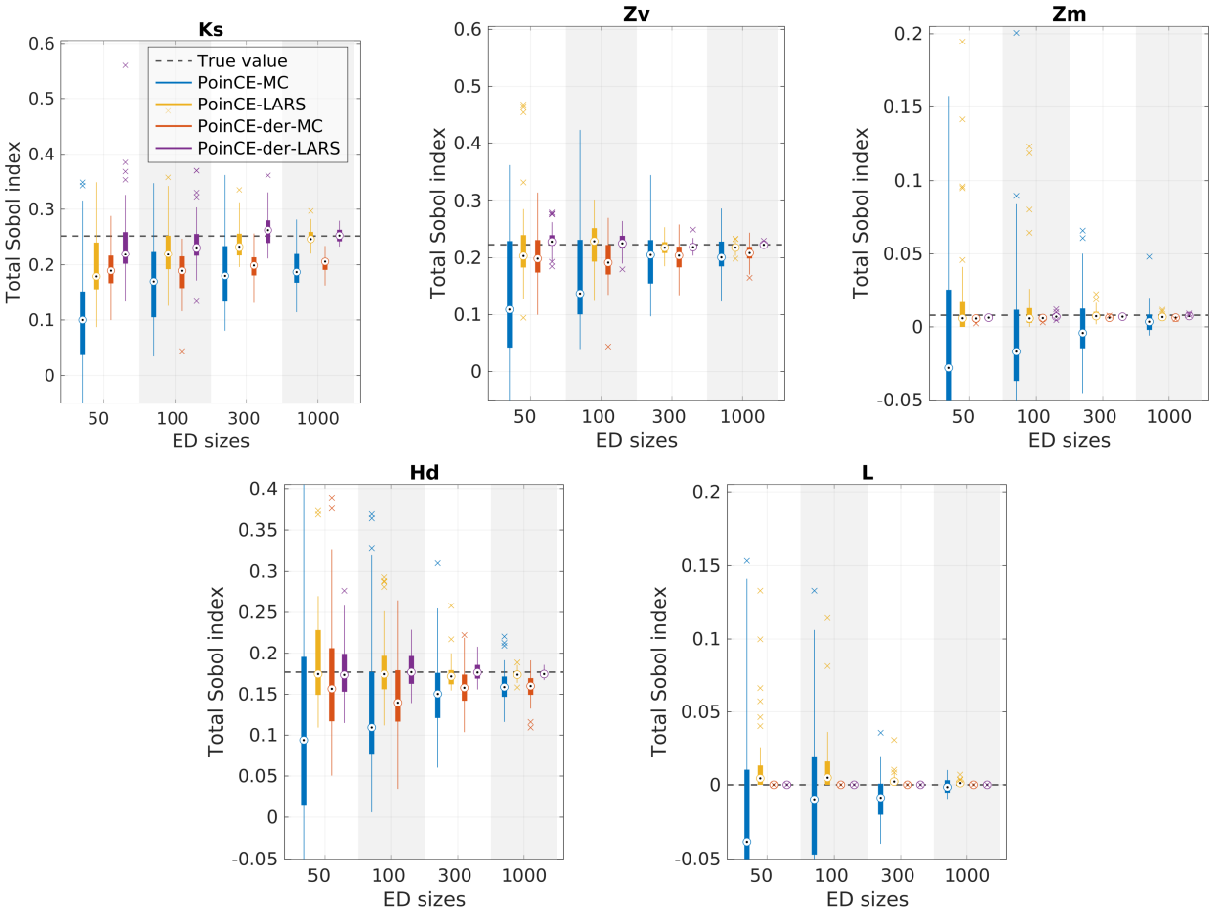



***Figure 5.A.2:*** *Comparison of PoinCE estimates of total Sobol' indices for the dyke cost model. Degree $p = 2$ for the MC-based estimates and $p \leq 5$ (degree-adaptive) for the regression-based estimates. See also Figure 5.2 in the main part of the paper.*

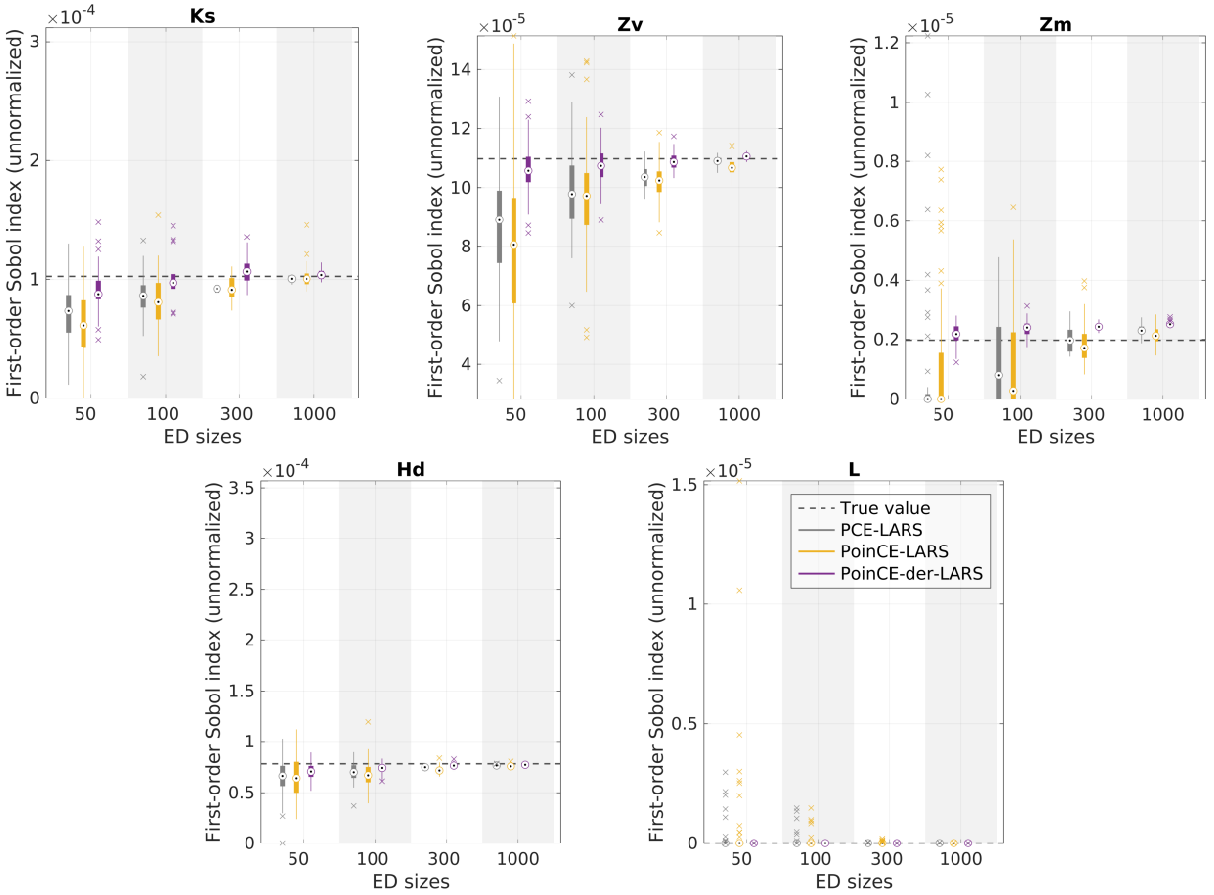



***Figure 5.A.3:*** *Estimates of unnormalized first-order Sobol' indices for the dyke cost model ($p \leq 5$). Boxplots: in grey the PCE-based estimates. The dashed line ("True value") denotes a high-precision estimate for the unnormalized first-order Sobol' index. See also Figure 5.3.*

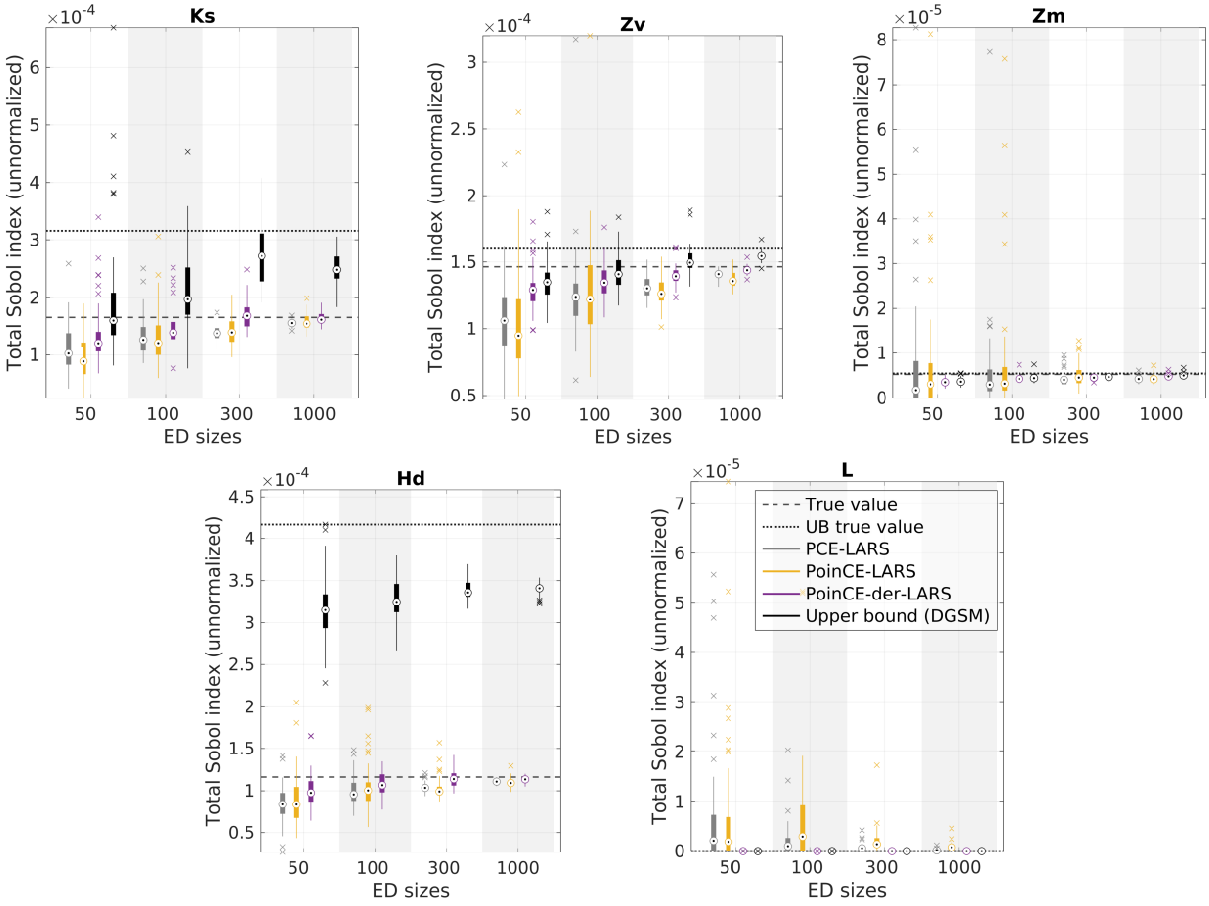



***Figure 5.A.4:*** *Estimates of unnormalized total Sobol' indices for the dyke cost model ($p \leq 5$). Boxplots: in grey the PCE-based estimates and in black the DGSM-based upper bound from (5.17). Lines: the dashed line ("True value") denotes a high-precision estimate for the unnormalized total Sobol' index, while the dotted line ("UB true value") is a MC-based high-precision estimate for the DGSM-based upper bound. See also Figure 5.4.*

CHAPTER 6

# A spectral surrogate model for stochastic simulators computed from trajectory samples

This chapter contains the preprint of

> Lüthen, N., Marelli, S., Sudret, B. (2022). A spectral surrogate model for stochastic simulators computed from trajectory samples. https://arxiv.org/abs/2207.05653

submitted for publication in the journal *Computer Methods in Applied Mechanics and Engineering* in July 2022.

**Author contributions.** **N. Lüthen:** Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Visualization. **S. Marelli:** Conceptualization, Methodology, Writing - Review & Editing, Supervision. **B. Sudret:** Conceptualization, Writing - Review & Editing, Supervision, Funding acquisition.

## Abstract

Stochastic simulators are non-deterministic computer models which provide a different response each time they are run, even when the input parameters are held at fixed values. They arise when additional sources of uncertainty are affecting the computer model, which are not explicitly modeled as input parameters. The uncertainty analysis of stochastic simulators requires their repeated evaluation for different values of the input variables, as well as for different realizations

of the underlying latent stochasticity. The computational cost of such analyses can be considerable, which motivates the construction of surrogate models that can approximate the original model and its stochastic response, but can be evaluated at much lower cost.

We propose a surrogate model for stochastic simulators based on spectral expansions. Considering a certain class of stochastic simulators that can be repeatedly evaluated for the same underlying random event, we view the simulator as a random field indexed by the input parameter space. For a fixed realization of the latent stochasticity, the response of the simulator is a deterministic function, called trajectory. Based on samples from several such trajectories, we approximate the latter by sparse polynomial chaos expansion and compute analytically an extended Karhunen-Loève expansion (KLE) to reduce its dimensionality. The uncorrelated but dependent random variables of the KLE are modeled by advanced statistical techniques such as parametric inference, vine copula modeling, and kernel density estimation. The resulting surrogate model approximates the marginals and the covariance function, and allows to obtain new realizations at low computational cost. We observe that in our numerical examples, the first mode of the KLE is by far the most important, and investigate this phenomenon and its implications.

## 6.1 Introduction

Nowadays, computer simulations are an essential ingredient of the research and development workflow in virtually all fields of science and engineering. Typically, not all parameters and conditions needed for the simulations are known exactly, and this uncertainty affects the output of the simulations. This is the main focus of the field of uncertainty quantification (Smith, 2014).

Most computer simulations can be classified as *deterministic simulators*: repeatedly evaluating the model $\mathcal{M}$ for the same set of input parameters $\boldsymbol{x}$ always yields the same deterministic response $y = \mathcal{M}(\boldsymbol{x}) \in \mathbb{R}$.[1] To perform uncertainty quantification, the uncertainty on the input (parameters and conditions) is usually represented probabilistically, and we follow this approach in this paper. Propagating the input uncertainty through the deterministic simulator, the overall response of the simulation becomes a random quantity.

However, not all computer simulations can be classified as deterministic simulators. Some models contain intrinsic stochasticity that cannot be modeled as input parameter, e.g., epidemiological models where each transmission or recovery is a random event, governed by the respective rate of occurrence. Other models depend on uncontrollable environmental variables such as wind fields or earthquakes, for which it can be infeasible or undesirable to explicitly model their uncertainty. In these cases, it is more convenient to use the notion of a *stochastic simulator*: only some of the uncertainty is explicitly modeled as random input variables, and there is some residual randomness affecting the computational model that causes the model response $\mathcal{M}(\boldsymbol{x})$ for a fixed set of input parameters $\boldsymbol{x}$ to still be a random variable: $Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x})$. In other words, evaluating the computer model several times with the same input parameters $\boldsymbol{x}$ will result in

[1] We consider here only real-valued simulators. The extension to low-dimensional vector-valued simulators is straightforward. For the extension to high-dimensional vector-valued or function-valued simulators, see e.g. Nagel et al. (2020); Perrin et al. (2021).

different realizations $y$ of the random variable $Y_{\boldsymbol{x}}$. Of course, since there is no true randomness in a computer, every computer simulation can be made deterministic by fixing the random seed. However, the seed is in general not a useful parametrization of uncertainty.

Uncertainty quantification methods typically require many runs of the computational model, which can become costly or even infeasible for expensive engineering simulators. To save computational resources, the model is often replaced with a cheaper *surrogate model* (or *metamodel*), which provides a reasonably good approximation to the original model. The surrogate model is computed from a small number of model evaluations and can subsequently be evaluated many times with negligible computational cost. Surrogate models often treat the model as a *black box*, i.e., they do not use any specific knowledge about the model and rely only on the available input-output data samples (and sometimes on the characteristics of the input parameter space). Popular surrogate models for deterministic simulators include polynomial chaos expansions (Ghanem and Spanos, 1991; Xiu and Karniadakis, 2002), Kriging (Sacks et al., 1989; Rasmussen and Williams, 2006), radial basis functions (Buhmann, 2000), and support vector regression (Vapnik, 1995; Smola and Schölkopf, 2004).

Since the response of stochastic simulators is a random variable for every set of input parameters, even more runs might be required to analyze their uncertainty, making surrogate models all the more relevant in this case. Research on surrogating stochastic simulators is comparatively recent. Most available methods focus on the marginal response distribution $\mathbb{P}(Y|\boldsymbol{X} = \boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{D}$ and emulate the conditional density itself or certain statistics of it. Early contributions aimed at characterizing the variation of the first two moments of the output response over the input domain using joint Gaussian process models (Iooss and Ribatet, 2009; Marrel et al., 2012). Another class of methods aims at directly modeling the variation of the marginal output probability density function (pdf) of the random variable $Y_{\boldsymbol{x}}$ over the input domain. Assuming that the true marginal response pdf at a number of input locations is known, Moutoussamy et al. (2015) represent the marginal pdf of a new input point as a linear combination of training examples (i.e., kernel regression) or of specifically constructed basis functions. However, the true marginal pdf is rarely known or its generation might require a lot of samples. For a finite number of stochastic simulator evaluations over the input domain (with or without replications), Zhu and Sudret (2020, 2021b) model the variation of the marginal output pdf over the input domain using the so-called *generalized lambda model*, a parametric distribution family that is able to approximate many classical families. In fact, stochastic simulators are akin to real-world scientific experiments, which are usually stochastic due to unavoidable measurement error and environmental noise. Therefore, standard statistical methods like quantile regression (Torossian et al., 2020) and kernel conditional density estimation (Hall et al., 2004) can also be used to emulate the marginal distribution of the response of a stochastic simulator. Furthermore, Zhu and Sudret (2022) developed an approach that emulates the stochastic simulator response in distribution, inspired by the weak PCE methodology based on maximum likelihood estimation (Xiu, 2010).

A related method from machine learning are Bayesian neural networks, whose weights are modeled as independent Gaussian random variables (MacKay, 1992; Goan and Fookes, 2020). Bayesian methods such as Markov Chain Monte Carlo or variational inference are used to de-

termine the parameters of the weight densities from the given data. Furthermore, generative models like variational autoencoders (Kingma and Welling, 2014) and generative adversarial networks (Goodfellow et al., 2014) can be seen as surrogate models in distribution, learning a conditional target density from data.

All the methods cited above aim at emulating only the univariate probability density functions of the response random variables of the stochastic simulator. However, they do not take into account the correlation and higher-order information between the stochastic simulator responses at different points in the input domain. This close relation between the responses at different input locations can be best illustrated by fixing the stochasticity of the simulator (e.g., by fixing the random seed)[2]: in this case, the stochastic simulator response over the input domain becomes a deterministic function, which we call a *trajectory*. In other words, the stochastic simulator can be seen as a random field, i.e., as a collection of random functions.

Surrogating a stochastic simulator based on few model evaluations becomes therefore the task of inferring a random field from discrete samples (often called "limited data"). Popular methods for modeling random fields include orthogonal series expansions, such as spectral representation (Shinozuka and Deodatis, 1991; Grigoriu, 1993) or Karhunen-Loève expansion (KLE) (Loève, 1978; Karhunen, 1946; Zhang and Ellingwood, 1994; Ramsay and Silverman, 2005; Grigoriu, 2006), and translation processes, which are mappings of Gaussian processes (Yamazaki and Shinozuka, 1988; Grigoriu, 1998; Sakamoto and Ghanem, 2002; Shields et al., 2011). To our knowledge, the only publication in the specific context of stochastic simulators which takes the random field point of view and aims at emulating trajectories (including the higher-order relations between responses at different input locations) is by Azzi et al. (2019), who construct a metamodel using Karhunen-Loève expansion together with the deterministic methods PCE and Kriging.

The goal of our paper is to develop a surrogate model that is able to emulate the trajectories of a stochastic simulator, and allows insight into the dependence between the simulator responses at different input locations. Our method of choice in this paper is Karhunen-Loève expansion, one of the most popular methods for random field inference from limited data. The main challenges in constructing a trajectory-based surrogate for a stochastic simulator (a *stochastic emulator*) are explained in more detail in the following:

1. Accuracy and efficiency: the surrogate should be accurate while needing as few model evaluations as possible.
2. Continuous surrogate from discrete data: the surrogate should emulate the response over the whole (continuous) input domain, while the available data consists of trajectories sampled at a few points throughout the input domain (i.e., discrete samples).
3. The stochastic simulator is in general a non-Gaussian random field. This introduces additional complexity into the Karhunen-Loève model.

We are addressing each of these challenges by introducing a novel approach that combines several state-of-the-art methodologies. We use Karhunen-Loève expansion in conjunction with sparse PCE (Blatman and Sudret, 2011; Lüthen et al., 2021), which is a powerful and sample-efficient

[2]Note that this does not require this randomness to be modeled. In practice, fixing the seed might not be possible for all computational models, as it depends on their implementation.

surrogate modeling method for deterministic simulators, to address Challenge 1. This circumvents the otherwise high computational cost of solving the integral eigenvalue problem of KLE (Schwab and Todor, 2006; Betz et al., 2014) by reducing the integral eigenvalue problem to finite-dimensional discrete principal component analysis (PCA) in the truncated space of PCE coefficients. The joint distribution of the resulting sample of dependent random KLE coefficients (Challenge 3) is identified using statistical inference within the marginal-copula framework (Torre et al., 2019b). The procedure results in an analytical formula for the stochastic emulator that can be used for computing marginals and correlations, as well as for generating new trajectories that resemble trajectories of the original stochastic simulator.

In our approach, the extension from discrete data to the continuous model (Challenge 2) is achieved by approximating the sampled trajectories by sparse regression-based PCE. A similar approach has been used by Navarro Jimenez et al. (2017) in the context of stochastic differential equations with the goal of sensitivity analysis, using non-intrusive pseudospectral projection to compute the PCE coefficients. The representation by sparse PCE can be seen as a variant of *orthogonal series expansion (OSE)* (Zhang and Ellingwood, 1994), which expands a second-order random process in terms of an orthogonal basis of the associated Hilbert space.

Note that when random fields are approximated based on a set of samples, it is most often assumed that the latter are collected on a discrete mesh in the index set, whereas this is not a requirement for our method. In such mesh-based approximations to random fields, PCE is often used for modeling the random variables arising in dimension-reduced expansions (Desceliers et al., 2006; Doostan et al., 2007; Das et al., 2009; Raisee et al., 2015; Abraham et al., 2018; Dai et al., 2019). This is distinct from our approach, as we use PCE to approximate the trajectories in the input space. Our approach yields an emulator for the whole input space (including unseen locations), while existing approaches are mostly focused on building an emulator on the discrete mesh where the samples were collected.

KLE represents a random field using an optimal orthogonal basis of the index space, resulting in an expansion in terms of deterministic functions weighted by random coefficients. These random coefficients are by construction uncorrelated, but unless the random field is a Gaussian random field, they are in general statistically dependent. Inferring the joint distribution of dependent random variables from samples is challenging but necessary for approximating a general non-Gaussian random field by KLE. To address this challenge of inference, several approaches have been proposed. Grigoriu (2010) suggests two methods to infer the joint distribution of the random coefficients of a series expansion model, of which one amounts to kernel density estimation, and the other to the fitting of a discrete joint distribution. Poirion and Zentner (2013, 2014) use KLE for modeling seismic ground motion time series, and model the random KLE coefficients by 1D sample CDFs assuming at most pairwise dependence (Poirion and Zentner, 2013), or by kernel density estimation (Poirion and Zentner, 2014). In the present paper, we investigate the use of kernel density estimation and inference of parametric joint distributions based on marginals and vine copulas.

This paper is organized as follows: in Section 6.2 we recall the relevant theory and definitions. In Section 6.3 we present our new stochastic emulator. The proposed method is then applied in Section 6.4, where we assess its performance on several examples of varying complexity. Here we

observe that the KLE is often significantly dominated by its first mode, a phenomenon that we investigate in Section 6.5. Finally, we draw conclusions and give an outlook on possible further developments in Section 6.6.

## 6.2 Theoretical foundation

We provide a brief summary of the relevant theory and concepts needed to construct our proposed stochastic emulator for stochastic simulators: random fields, polynomial chaos expansions, Karhunen-Loève expansion, and inference of joint probability distributions.

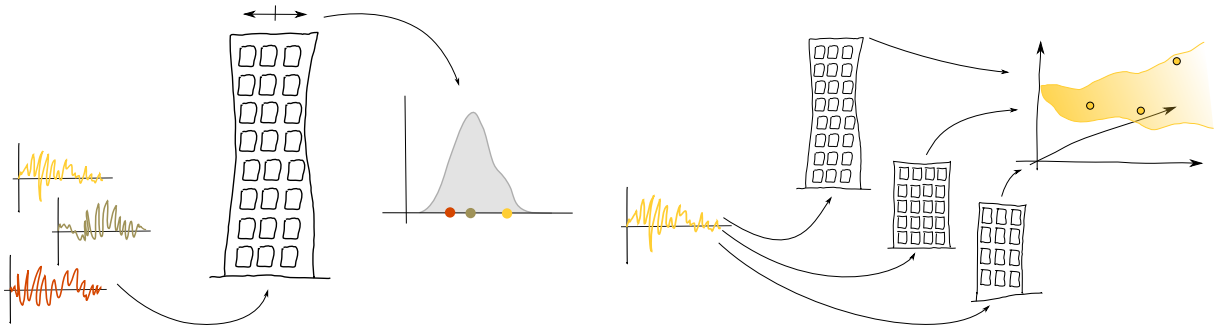
### 6.2.1 Stochastic simulators as random fields

Let $\boldsymbol{X}$ be a random vector with values in $\mathcal{D} \subset \mathbb{R}^d$, with finite variance and joint probability density function (pdf) $f_{\boldsymbol{X}}$. Denote by $\omega \subset \Omega$ an abstract random event in a probability space $(\Omega, \mathcal{F}, P)$. A *stochastic simulator* is a mapping

$$\mathcal{M} : \mathcal{D} \times \Omega \to \mathbb{R}, \tag{6.1}$$

$$(\boldsymbol{x}, \omega) \mapsto \mathcal{M}(\boldsymbol{x}, \omega). \tag{6.2}$$

Fixing $\boldsymbol{x} \in \mathcal{D}$, the quantity $Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x}, \cdot) : \Omega \to \mathbb{R}$ is a random variable. Fixing $\omega \in \Omega$, $\mathcal{M}(\cdot, \omega) : \mathcal{D} \to \mathbb{R}$ is a function in the input parameters, which we call *trajectory* or *realization* of the stochastic simulator (see also Figure 6.1). We assume that $Y_{\boldsymbol{x}}$ has finite variance for all $\boldsymbol{x}$, and that $\mathcal{M}(\cdot, \omega) \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ for all $\omega \in \Omega$.



*(a)* Random variable $Y_{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x}, \cdot)$ *(b)* Trajectory $\mathcal{M}(\cdot, \omega) : \mathcal{D} \to \mathbb{R}$

***Figure 6.1:*** *Visual representation of a stochastic simulator when either the input parameters $\boldsymbol{x}$ or the random event $\omega$ are held fixed, resulting in a random variable (left) or a deterministic function (right). The computational model is a high-rise building parametrized by several properties $\boldsymbol{x}$ (visualized in the sketch by the shape of the building) subject to random earthquake events $\omega$ (visualized by 1D time series in different colors), whose output $\mathcal{M}(\boldsymbol{x}, \omega)$ is a real number (e.g., the maximal displacement at the top floor).*

These definitions imply that a stochastic simulator $\mathcal{M}$ can be seen as a *random field* (also: *stochastic process* or *random process*) $\{Y_{\boldsymbol{x}}\}_{\boldsymbol{x} \in \mathcal{D}}$ with *index set* $\mathcal{D}$, i.e., as a family of random variables $\{Y_{\boldsymbol{x}}\}$ indexed by $\boldsymbol{x} \in \mathcal{D}$. In the following, we provide a brief reminder of a few random field basics. For more details, see e.g. Grigoriu (2002).

To fully characterize a general random field, one needs to specify the collection of all its finite-dimensional distributions

$$F_{Y_{\boldsymbol{x}_1},\dots,Y_{\boldsymbol{x}_n}}(y_1,\dots,y_n) = \mathbb{P}\left(Y_{\boldsymbol{x}_1} \leq y_1 \wedge \dots \wedge Y_{\boldsymbol{x}_n} \leq y_n\right) \tag{6.3}$$

for *all* $n \geq 1$ and *any* $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{D}$. Extending the concept of moments of random variables to random fields, the deterministic *mean function* of the random field is given by $\mu(\boldsymbol{x}) = \mathbb{E}\left[Y_{\boldsymbol{x}}\right]$. If $\mu(\boldsymbol{x}) = 0$, the random field is called *centered*. The *(auto-)covariance function* is defined by

$$c(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(Y_{\boldsymbol{x}} - \mu(\boldsymbol{x}))(Y_{\boldsymbol{x}'} - \mu(\boldsymbol{x}'))\right]. \tag{6.4}$$

In general, a random field is not uniquely defined by its mean and covariance function. The only exception is the family of *Gaussian processes*, for which all finite-dimensional joint distributions are multivariate Gaussian distributions. For Gaussian processes, conditional distributions are again multivariate Gaussians, which lies at the foundation of the popular surrogate modeling technique Kriging/Gaussian process modeling. While Gaussian random fields are computationally convenient, random fields encountered in real-world problems (and in particular, stochastic simulators) are often non-Gaussian. One obvious argument is that Gaussian variables are unbounded while physical quantities are almost always bounded (Grigoriu, 2002).

A special feature of a stochastic simulator $\mathcal{M}$, as opposed to classical random fields, is that its index set is not an interval or a hypercube, but a general domain $\mathcal{D} \in \mathbb{R}^d$ with weight function $f_{\boldsymbol{X}}$. We will use this property to build an accurate surrogate model for $\mathcal{M}$ respecting the probability density $f_{\boldsymbol{X}}$ of the input space.

### 6.2.2 Polynomial chaos expansion

Polynomial chaos expansion (PCE) is a technique for modeling random variables using a basis of polynomials that are orthonormal w.r.t. a given probability density function (Ghanem and Spanos, 1991; Xiu and Karniadakis, 2002). In our algorithm (Section 6.3), we will use PCE to approximate trajectories $\mathcal{M}(\cdot, \omega)$ of the stochastic simulator, which can be seen as random variables $\mathcal{M}(\boldsymbol{X}, \omega)$ with their randomness induced by the uncertainty in the input $\boldsymbol{X}$.

Consider a random vector $\boldsymbol{X}$ with values in $\mathcal{D} \subset \mathbb{R}^d$ and independent components. Let $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^d f_{X_i}(x_i)$ be its probability density function (pdf) and assume that $\boldsymbol{X}$ has finite variance. Let $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ be the space of real-valued function that are square-integrable under $f_{\boldsymbol{X}}$, i.e., $L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = \{g : \mathcal{D} \to \mathbb{R} \mid \mathrm{Var}_{\boldsymbol{X}}\left[g(\boldsymbol{X})\right] < +\infty\}$. Under certain assumptions on the random vector $\boldsymbol{X}$ (Xiu and Karniadakis, 2002; Ernst et al., 2012), there exists an orthonormal polynomial basis $\{\psi_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^d\}$ of $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$, where each element is the product of univariate polynomials characterized by the multi-index $\boldsymbol{\alpha}$.

Let $\mathcal{M} \in L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ be a (computational) model. Its output $Y = \mathcal{M}(\boldsymbol{X})$ is a random variable, which can be represented in terms of the orthonormal polynomial basis as $\mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ with $a_{\boldsymbol{\alpha}} = \mathbb{E}_{\boldsymbol{X}}\left[\mathcal{M}(\boldsymbol{X})\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\right]$. This representation is called *polynomial chaos expansion*. In practice, a truncated expansion is computed,

$$\mathcal{M}(\boldsymbol{X}) \approx \mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{6.5}$$

where $\mathcal{A} \subset \mathbb{N}^d$ is a finite subset of the full basis. The accuracy of a truncated PCE depends on three ingredients: the choice of $\mathcal{A}$, the method used for computing the coefficients $\boldsymbol{a} = (a_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathcal{A}}$, and the choice of points $\mathcal{X} \subset \mathcal{D}$ used in the coefficient computation method. An extensive overview of the state-of-the-art methods to determine these is given in Lüthen et al. (2021, 2022a).

### 6.2.3 Karhunen-Loève expansion

Karhunen-Loève expansion (KLE) is a well-established spectral expansion technique through which a random field is represented in terms of an optimal orthogonal basis for the index space, weighted by random coefficients (Karhunen, 1946; Loève, 1978). KLE transforms the random field, which is an uncountably infinite but correlated family of random variables $\{\mathcal{M}_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{D}}$, into a countably infinite but uncorrelated family of different random variables $\{\xi_i\}_{i=1,2,\ldots}$. Furthermore, the random variables $\xi_i$ are typically of decreasing importance. KLE is therefore well suited and often used for discretization and modeling efforts for random fields.

To make these notions more precise, let $\{\mathcal{M}_{\boldsymbol{x}}(\omega)\}_{\boldsymbol{x}\in\mathcal{D}}$ be a random field. Denote by $\mu(\boldsymbol{x}) = \mathbb{E}[\mathcal{M}_{\boldsymbol{x}}]$ its mean function, and by $c(\boldsymbol{x},\boldsymbol{x}') = \mathrm{Cov}[\mathcal{M}_{\boldsymbol{x}}, \mathcal{M}_{\boldsymbol{x}'}]$ its covariance function. Let $\mathcal{D}$ be closed and bounded. Let $c$ be continuous on $\mathcal{D}\times\mathcal{D}$ and assume that $\mathcal{M}_{\boldsymbol{x}}$ has finite variance for all $\boldsymbol{x}\in\mathcal{D}$. Then the *Karhunen-Loève expansion* of the random field $\mathcal{M}_{\boldsymbol{x}}$ is given by

$$\mathcal{M}_{\boldsymbol{x}}(\omega) = \mu(\boldsymbol{x}) + \sum_{k=1}^{\infty} \sqrt{\lambda_k}\xi_k(\omega)\phi_k(\boldsymbol{x}) \tag{6.6}$$

where $(\phi_k)_{k=1,2,\ldots}$ is an orthonormal basis of $L^2(\mathcal{D})$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ is a non-increasing sequence of non-negative real numbers, and $\{\xi_k\}_{k=1,2,\ldots}$ is a countable family of zero mean, unit variance, uncorrelated random variables.

Here, $(\lambda_k, \phi_k)$ are solutions to the integral eigenvalue problem

$$\int_{\mathcal{D}} c(\boldsymbol{x},\boldsymbol{x}')\phi_k(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}' = \lambda_k\phi_k(\boldsymbol{x}), \tag{6.7}$$

and $\xi_k$ is the result of the projection of $\mathcal{M}$ onto the spatial basis

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}}\int_{\mathcal{D}} \mathcal{M}(\boldsymbol{x},\omega)\phi_k(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{6.8}$$

From Eq. (6.6) and the properties of $\phi_k$ and $\xi_k$ it follows immediately that the covariance function can be expressed as

$$c(\boldsymbol{x},\boldsymbol{x}') = \sum_{k=1}^{\infty} \lambda_k\phi_k(\boldsymbol{x})\phi_k(\boldsymbol{x}') \tag{6.9}$$

(Mercer's theorem). Note that the KLE random variables $\{\xi_k\}$ (herein *KL-RV*) do not enter this expression.

KLE is especially well-suited to Gaussian random fields, since in this case the random variables $\xi_k$ are standard Gaussian and independent. However, Eq. (6.6) holds for all random fields fulfilling the assumptions, not only for Gaussian random fields. The non-Gaussianity is modeled by the (possibly complex) joint distribution $f_{\boldsymbol{\xi}}$ of the KL-RV.

Eqs. (6.6) to (6.8) are formulated in terms of $L^2(\mathcal{D})$, but they can be generalized: let $\boldsymbol{X}$ be a random variable with values in $\mathcal{D} \subset \mathbb{R}^d$, density $f_{\boldsymbol{X}}$, and finite variance. Then KLE can be generalized to the space $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ instead of $L^2(\mathcal{D})$. In that case, the index set $\mathcal{D}$ does not have to be bounded, since the volume of $\mathcal{D}$ under measure $f_{\boldsymbol{X}}\mathrm{d}\boldsymbol{x}$ is finite. This is called *extended KLE* (Iemma et al., 2006). This property is crucial for our proposed stochastic emulator, which we will introduce in Section 6.3.

In practice, the infinite expansion in Eq. (6.6) must be truncated. From the orthonormality of $\{\phi_k\}$ it follows from Eq. (6.9) that the variance of the random field is equal to $\sum_{k=1}^{\infty} \lambda_k$. The sequence $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ is non-increasing, and typically (depending on the correlation length of the random field) this sequence decays rather quickly to zero. Loosely speaking, the higher the correlation between different locations in the index set, the fewer spatial basis functions are needed to approximate the trajectories, and the faster the decay of the eigenvalues. Knowing this, the KLE can be truncated at the $M$-th term with $M$ chosen so that the fraction of *explained variance* is sufficiently large:

$$\frac{\sum_{k=1}^{M} \lambda_k}{\sum_{k=1}^{\infty} \lambda_k} > 1 - \epsilon \tag{6.10}$$

for a small threshold parameter $\epsilon > 0$ (e.g., $\epsilon = 0.001$).

KLE is closely related to function principal component analysis (fPCA) (Besse and Ramsay, 1986; Ramsay and Silverman, 2005). To compute a solution to the integral eigenvalue problem in Eq. (6.7), there are several possibilities (Ramsay and Silverman, 2005, Section 8.4): the integrals can be approximated numerically; the eigenproblem can be discretized on a number of representative grid points in $\mathcal{D}$ (this is the approach chosen by the majority of modelers, including Azzi et al. (2019)); or the eigenproblem can be written in terms of a suitable (truncated) spatial basis, which transforms the problem into a (finite-dimensional) discrete eigenvalue problem. The third approach is related to orthogonal series expansion (OSE) (Zhang and Ellingwood, 1994). It is used by Poirion and Zentner (2014), who derive the explicit discrete problem for a basis consisting of interpolation functions, building on results by Besse and Ramsay (1986) and Besse (1991). We use this approach together with the orthogonal basis provided by polynomial chaos expansion (Section 6.3). Detailed calculations are provided in Section 6.A.

### 6.2.4 Inference of the joint distribution of the Karhunen-Loève random variables

Characterizing the dependent (but uncorrelated) Karhunen-Loève random variables (KL-RV) $\xi_k, k = 1, \ldots, M$ correctly is important for the accurate modeling of a general non-Gaussian stochastic process (Grigoriu, 2010). However, inferring the joint distribution of a random vector is a challenging task. The main challenge is the scarcity of data: the higher the dimensionality, the more samples are needed to be able to correctly infer the dependence structure of the data. We need to construct a suitable parametric or non-parametric model to accurately describe the joint distribution. In the following, we introduce the marginal-copula framework, which is a powerful tool to represent and infer complex dependence structures between random variables (Nelsen, 2006; Torre et al., 2019).

Let $\boldsymbol{Z}$ be any $M$-dimensional random vector with multivariate cumulative distribution function (CDF) $F_{\boldsymbol{Z}}$ and marginal distributions $F_{Z_i}$. The so-called *Sklar's theorem* states that $F_{\boldsymbol{Z}}$ can be written as

$$F_{\boldsymbol{Z}}(z_1, \dots, z_M) = C\left(F_{Z_1}(z_1), \dots, F_{Z_M}(z_M)\right), \tag{6.11}$$

where the function $C : [0,1]^d \to \mathbb{R}$ is called *copula* (Sklar, 1959; Nelsen, 2006). $C$ is a CDF with uniform marginals, which defines the dependence structure of the random vector $\boldsymbol{Z}$. $C$ is unique if all marginals $F_{Z_i}$ are continuous, and it holds that

$$C(u_1, \dots, u_M) = F_{\boldsymbol{Z}}\left(F_{Z_1}^{-1}(u_1), \dots, F_{Z_M}^{-1}(u_M)\right). \tag{6.12}$$

Let an i.i.d. sample $\mathcal{Z} = \{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(N)}\}$ of the random vector $\boldsymbol{Z}$ be given. The goal is to infer the joint distribution $F_{\boldsymbol{Z}}$ from this sample. For this, the copula representation of Eq. (6.11) is convenient, since it allows inferring the marginals and the dependence structure of the data separately, as briefly explained in the following.

To infer the marginal distributions, we consider two options. The first is parametric inference, where we choose from a set of parametric probability distributions with zero mean and unit standard deviation (see Table 6.1) the distribution with the smallest Akaike information criterion (AIC). If a distribution family has more than two parameters, its remaining parameters are chosen by maximum likelihood. We utilize the uncertainty quantification software UQLab (Marelli and Sudret, 2014; Torre et al., 2021) with a modification prescribing the desired moments.

***Table 6.1:*** *Considered marginal families with zero mean and unit standard deviation. The last column lists the remaining degrees of freedom $k$ after fixing the first two moments. The Akaike information criterion is then given as $AIC = 2k - 2\log\mathcal{L}$, where $\mathcal{L}$ is the likelihood.*

| Type | Parameter | $k$ |
|---|---|---|
| Uniform $\mathcal{U}([a,b])$ | $a = -\sqrt{3}, b = \sqrt{3}$ | 0 |
| Gaussian $\mathcal{N}(\mu,\sigma)$ | $\mu = 0, \sigma = 1$ | 0 |
| Gumbel (for maxima) $\mathcal{G}(\mu,\beta)$ | $\mu \approx -0.4501, \beta \approx 0.7797$ | 0 |
| Gumbel (for minima) $\mathcal{G}_{\min}(\mu,\beta)$ | $\mu \approx 0.4501, \beta \approx 0.7797$ | 0 |
| Logistic $P(\mu,s)$ | $\mu = 0, s \approx 0.5513$ | 0 |
| Laplace $\mathcal{L}(\mu,b)$ | $\mu = 1, b = \frac{1}{\sqrt{2}}$ | 0 |
| Beta $\mathcal{B}(a,b,r,s)$ | $a, b$ chosen according to data bounds<br>$r = \frac{a(ab+1)}{b-a}, s = \frac{b(ab+1)}{a-b}$ | 2 |

A second popular method to represent marginal behavior non-parametrically is *kernel density estimation* (KDE) (Wand and Jones, 1995; Simonoff, 1996), which has also been proposed for estimating the distribution of KL-RV (Grigoriu, 2010; Poirion and Zentner, 2014). Here the distribution is modeled as a Gaussian mixture, where the Gaussian density functions are centered in the data points and share the same standard deviation, called *bandwidth* in the case of 1D KDE. We adopt a bandwidth estimation method optimal for data with Gaussian distribution (Bowman and Azzalini, 1997).

To characterize the dependence structure, we use a copula. While any multivariate CDF with uniform marginals $\mathcal{U}([0,1])$ constitutes a copula, there are a number of well-known parametric families (see, e.g., Nelsen (2006); Joe (2014)). Besides the independence copula and the families

derived from multivariate elliptical distributions, most of these parametric families are pair copulas, i.e., they couple only two variables. Constructing meaningful parametric copulas for more than two variables (other than elliptical copulas) is in general difficult (Nelsen, 2006).

A solution is to decompose the $M$-variate copula into a product of conditional pair copulas, which is known as vine copula construction (Bedford and Cooke, 2002). This is always possible as a consequence of the chain rule of probability. In general, a vine copula is the product of $\frac{M(M-1)}{2}$ pair copulas.[3] The factorization into pair copulas is not unique but depends on the ordering and grouping of variables. Two classes of vine copulas, differing in the order in which the variables are grouped into pairs, are the drawable vine (D-vine) (Kurowicka and Cooke, 2005) and the canonical vine (C-vine) (Aas et al., 2009). For a more detailed description of the vine copula construction, we refer to Aas et al. (2009) and Torre et al. (2019b).

To infer a copula from data, we first map the multivariate data to $[0,1]^d$ by applying element-wise the inferred marginal CDFs (see Eq. (6.11)). Then we infer the dependence structure by using Kendall's tau to determine the groupings of variables as well as their order in the vine copula (Aas et al., 2009; Torre et al., 2019b). For each pair copula, the parameters are identified by maximum likelihood. Finally, the best-fitting copula is chosen using AIC. This approach is implemented in the statistical inference module of UQLab (Torre et al., 2021). The list of available copula families can be found in Lataniotis et al. (2021, Section 1.4).

## 6.3 Surrogating a stochastic simulator from a set of samples

We are now ready to describe the construction of our spectral surrogate model for a stochastic simulator. Assume that discrete samples of the stochastic simulator $\mathcal{M}$ are available in the following form:

$$\mathcal{T}_r = \left\{ \left( \boldsymbol{x}^{(r,i)}, \mathcal{M}(\boldsymbol{x}^{(r,i)}, \omega^{(r)}) \right) : i = 1, \dots, N_r \right\}, \quad r = 1, \dots, R \tag{6.13}$$

i.e., in the form of discrete evaluations of the stochastic simulator on $R$ trajectories, where for every $r$, $\{\boldsymbol{x}^{(r,i)} : i = 1, \dots, N_r\}$ is an i.i.d. sample from the input distribution $f_{\boldsymbol{X}}$, the so-called *experimental design*. In particular, for different trajectories the samples can be taken at different locations, i.e., for $r_1 \neq r_2$ we can have $\boldsymbol{x}^{(r_1,i)} \neq \boldsymbol{x}^{(r_2,i)}$ and in principle even different numbers of samples $N_{r_1} \neq N_{r_2}$. However, here we assume for notational simplicity that $N_r = N$ for all $r$.

Our proposed method consists of the following steps (see also Figure 6.1):

1. **Approximate each discrete trajectory** $\mathcal{T}_r$ by a sparse PCE $\mathcal{M}_r^{\mathrm{PCE}}$ in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$:

$$\mathcal{M}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{(r)}} a_{\boldsymbol{\alpha}}^{(r)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{6.14}$$

   with $\mathcal{A}^{(r)}$ the set of regressors with nonzero associated coefficient $a_{\boldsymbol{\alpha}}^{(r)}$. We use a total-degree basis with degree- and $q$-norm adaptivity to determine the truncation set $\mathcal{A}^{(r)}$ (Blatman and Sudret, 2011; Lüthen et al., 2022a) and apply the least-angle regression

[3] There are $M-1$ unconditional pair copulas; $M-2$ pair copulas conditioned on 1 other variable; $M-3$ conditioned on 2 other variables; and so on, until there is 1 pair copula conditioned on all except 2 variables.

solver to compute the coefficients (sparse PCE) (Blatman and Sudret, 2011; Lüthen et al., 2021).

2. **Determine a set $\mathcal{A}$ of regressors** that jointly represents all trajectories well:
   - Identify the union $\mathcal{A} = \bigcup_{r=1}^{R} \mathcal{A}^{(r)}$ of all chosen regressors.
   - To keep the size of the basis manageable, discard the regressors with the smallest sum of squares of coefficients over all trajectories ($\sum_{r=1}^{R} \left(a_{\boldsymbol{\alpha}}^{(r)}\right)^2$) until $P = |\mathcal{A}| \leq \frac{N}{2}$ regressors or less are left in $\mathcal{A}$.
   - To avoid discontinuous behavior resulting from sparse selection, recompute the coefficients of every trajectory by ordinary least squares (OLS), using the chosen set of regressors $\mathcal{A}$.

   This results in $R$ PCE trajectories, where each trajectory uses the same set of $P$ PCE basis functions.
3. **Center the PCE trajectories** by subtracting the sample mean

$$\hat{\mu}^{\mathrm{PCE}}(\boldsymbol{x}) = \frac{1}{R}\sum_{r=1}^{R} \mathcal{M}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\frac{1}{R}\sum_{r=1}^{R} a_{\boldsymbol{\alpha}}^{(r)}\right) \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{6.15}$$

   which is itself a PCE. We denote by $\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \mathcal{M}_r^{\mathrm{PCE}}(\boldsymbol{x}) - \hat{\mu}^{\mathrm{PCE}}(\boldsymbol{x})$ the centered PCE trajectories. Extract the coefficients $\tilde{a}_{\boldsymbol{\alpha}}^{(r)}$ of the centered trajectories and store them in a $P \times R$ matrix $\tilde{\boldsymbol{a}}$.
4. **Apply extended KLE** to the set of PCE trajectories. The sample covariance function has the form

$$\hat{c}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{R-1}\sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x})\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}'). \tag{6.16}$$

   Computing the eigenfunctions $\phi(\boldsymbol{x})$ of the associated integral eigenvalue problem in Eq. (6.7) is equivalent to computing a PCA on the PCE coefficients, i.e., equivalent to solving the following $P$-dimensional eigenproblem for $\tilde{\boldsymbol{a}}$:

$$\boldsymbol{\Sigma}\boldsymbol{b} = \lambda \boldsymbol{b}, \tag{6.17}$$

   where $\boldsymbol{\Sigma} = \frac{1}{R-1}\tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^T$ (see Section 6.A for the derivation of this equivalence). The eigenvectors $\boldsymbol{b}$ contain the coefficients of the eigenfunctions represented in the PCE basis: $\phi(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} b_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$.
5. **Identify the truncation order** $K \ll P$ for the KLE based on a given threshold for the explained variance. We use a threshold of 99.9% (see Eq. (6.10)).
6. **Compute the realizations** of the KL-RV $\xi_i$ from the sample trajectories by projecting onto the eigenfunctions. Due to the orthonormality of the PCE basis, this can be done analytically (see Section 6.A.2). Denote the realizations by $\boldsymbol{\xi}^{(r)} \in \mathbb{R}^K$.
7. **Infer the joint distribution** $f_{\boldsymbol{\xi}}$ of random KL coefficients from the data set $\{\boldsymbol{\xi}^{(r)}\}_{r=1,\dots,R}$. We will test four methods consisting of the techniques described in Section 6.2.4:
   (a) Option 1: assume standard Gaussian marginals, which implies independence;
   (b) Option 2: parametric inference of the marginals (with moment constraints) and of the copula;
   (c) Option 3: 1D kernel density estimation of each marginal, assuming independence;
   (d) Option 4: 1D kernel density estimation of each marginal and parametric inference of the copula.

The resulting stochastic model for the random field $\mathcal{M}$ is

$$\hat{\mathcal{M}}(\boldsymbol{x}, \cdot) = \hat{\mu}(\boldsymbol{x}) + \sum_{k=1}^{K} \sqrt{\lambda_k}\, Z_k(\cdot) \underbrace{\left( \sum_{\boldsymbol{\alpha} \in \mathcal{A}} b_{\boldsymbol{\alpha}}^{(k)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \right)}_{=\phi_k(\boldsymbol{x})} \tag{6.18}$$

where $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_K)$ is a random vector distributed according to the inferred joint distribution $f_{\boldsymbol{\xi}}$.

The full procedure is visualized in Figure 6.1.

Using the stochastic emulator constructed in Eq. (6.18), we can easily compute the following quantities.

- The mean function $\hat{\mu}$ is given by the sample mean of the approximated trajectories (PCE trajectories), see Eq. (6.15).
- The covariance function $\hat{c}(\cdot, \cdot)$ can be computed from the KLE eigenfunctions using the truncated version of Eq. (6.9). Note that this relation does not involve the KL-RV.
- New trajectories (i.e., realizations of the random field) can be generated by drawing new samples of the KL-RV $\xi_k$, and evaluating Eq. (6.6).
- A histogram of the marginal pdf $f_{\mathcal{M}_{\boldsymbol{x}'}}$ of the random field at any input space location $\boldsymbol{x}'$ can be created by generating many new trajectories and evaluating them at $\boldsymbol{x}'$.

**Remark 1 (Another stochastic emulator).** *A simple stochastic emulator able to model marginal distributions $f_{\mathcal{M}_{\boldsymbol{x}'}}$ can be constructed by evaluating all PCE trajectories from Step 2 above at the new location $\boldsymbol{x}'$ and computing a kernel density estimate on the resulting set of predictions. This method will be used as a comparison method for marginal estimation in Section 6.4. However, unlike our stochastic emulator in Eq. (6.18), this simple emulator is not able to resample trajectories.*

**Remark 2 (Alternatives to PCE).** *We choose PCE to approximate the sampled trajectories because it is a powerful method for deterministic surrogate modeling. However, the choice of PCE in the above method is not crucial: without any changes to the methodology, PCE could be replaced by any other spectral expansion onto an orthonormal basis of $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$, e.g., a Poincaré basis (Lüthen et al., 2022c) or a spline basis (Rahman, 2020). From the orthonormality of the basis it follows that functional PCA in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ becomes traditional (unweighted) PCA in the coefficient space (see Section 6.A), which avoids the expensive numerical solution of the integral eigenvalue problem in $d$ dimensions, and instead solves an inexpensive discrete eigenvalue problem.*

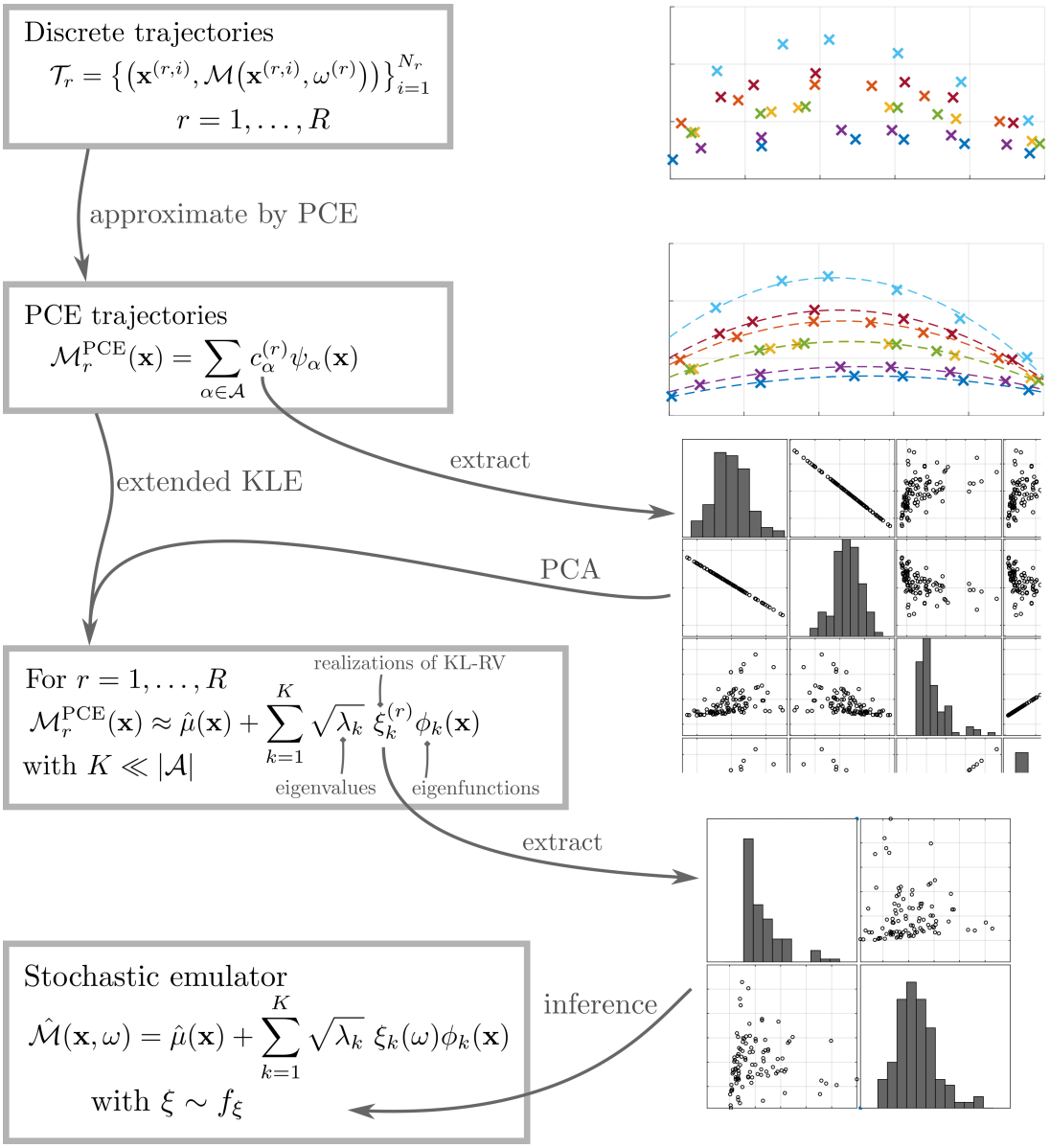



***Figure 6.1:*** *Sketch of our stochastic emulator, starting with stochastic simulator samples (discrete trajectories) at the top and resulting in the stochastic emulator at the bottom, which is a KLE that includes a probabilistic model of the KL-RV. The sketch is purely for illustration and does not display real data. Note that there are two equivalent ways to arrive at the third box: through extended KLE and through PCA on the coefficients.*

## 6.4 Numerical experiments

To analyse the performance of our stochastic emulator, we apply it to three models of increasing complexity: the three-dimensional Ishigami function with two random parameters (Section 6.4.1), the borehole model with five hidden (latent) variables (Section 6.4.2), and finally the Heston stochastic volatility model, a system of two stochastic ODEs with six inputs that has already been used by Zhu and Sudret (2021c) as a stochastic emulator benchmark model (Section 6.4.3).

We first investigate the pointwise approximation capabilities of our emulator by plotting the stochastic simulator and emulator responses at selected points throughout the input domain. Then, we investigate the convergence behavior of our stochastic emulator using the following global error measures:

- The global convergence of the marginal distributions is assessed using the *averaged normalized Wasserstein distance*. The *Wasserstein distance of order two* between two random variables $Y_1$, $Y_2$ with quantile functions (inverse CDF) $Q_1$, $Q_2$ is defined by (Villani, 2009)

$$d_{\mathrm{WS}}(Y_1, Y_2) = \|Q_1 - Q_2\|_2 = \sqrt{\int_0^1 (Q_1(u) - Q_2(u))^2 \, \mathrm{d}u}. \tag{6.19}$$

  To measure the global quality of marginal approximation, we consider the quantity

$$\epsilon_{\mathrm{marg}} = \mathbb{E}_{\boldsymbol{X}} \left[ \frac{d_{\mathrm{WS}}(\mathcal{M}(\boldsymbol{X}, \cdot), \hat{\mathcal{M}}(\boldsymbol{X}, \cdot))}{\sigma(\mathcal{M}(\boldsymbol{X}, \cdot))} \right], \tag{6.20}$$

  computed by Monte-Carlo integration on a validation set with $N_{\mathrm{val}} = 1,000$ points and $R_{\mathrm{val}} = 10,000$ replications (Zhu and Sudret, 2021b).
- The global error between the true covariance function $c$ and the emulated one $\hat{c}$ is computed by

$$\epsilon_{\mathrm{cov}} = \|c - \hat{c}\|_{L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) \times L^2_{f_{\boldsymbol{X}}}(\mathcal{D})} \approx \frac{1}{N_{\mathrm{val}}} \left\| C - \hat{C} \right\|_F \tag{6.21}$$

  where $C$ and $\hat{C}$ denote the true and emulated covariance matrices for a validation sample $\{\boldsymbol{x}^{(i)} : i = 1, \ldots, N_{\mathrm{val}}\}$, and $\|\cdot\|_F$ is the Frobenius norm.

### 6.4.1 Stochastic Ishigami function

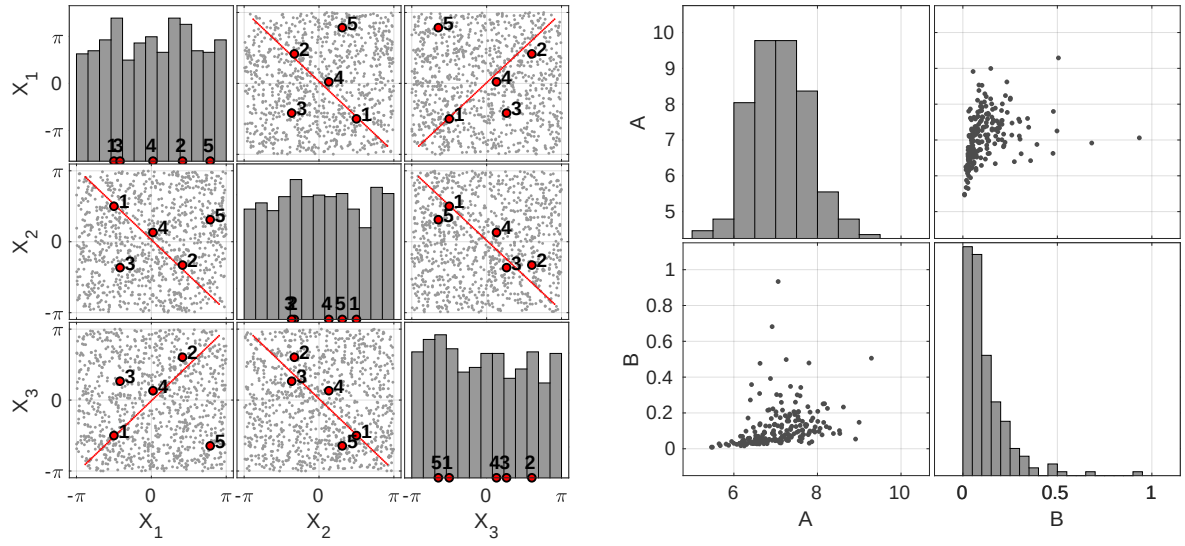
#### 6.4.1.1 Problem statement

The Ishigami function is a a well-known benchmark function for deterministic surrogate models. It is highly non-linear and has significant interaction terms. It becomes a stochastic simulator by treating its parameters $a$ and $b$, which are usually fixed at $a = 0.7$ and $b = 0.1$, as additional random variables:

$$f(\boldsymbol{X}; A, B) = \sin(X_1) + A \sin(X_2)^2 + B X_3^4 \sin(X_1). \tag{6.22}$$

$A$ and $B$ have here the role of so-called *hidden* or *latent* random variables. In other words, we assume that they cannot be observed, and that therefore their values cannot be utilized in

the surrogate modeling process. They introduce stochasticity into the otherwise deterministic Ishigami model. Here we model $A$ and $B$ as lognormal random variables with mean 7 and standard deviation 0.7, and mean 0.1 and standard deviation 0.1, respectively. We assume that both variables are coupled with a Clayton pair copula with parameter 1.5. The non-hidden (explicit) input variables are as usual $\boldsymbol{X} = (X_1, X_2, X_3)$, which are independent and uniformly distributed in $[-\pi, \pi]$. A Sobol' analysis of $f(\boldsymbol{X}; A, B)$ in Eq. (6.22) reveals that the main effect of the group of explicit input parameters ($\mathcal{M}_1$ in Eq. (6.28)) is approx. 75%, while the interaction effect between the explicit and the latent group is approx. 25%, and the main effect of the group of latent variables is negligible. Samples of the input space and the latent space are displayed in Figure 6.1.




***(a)*** *Input space with validation points used for visualization*

***(b)*** *Latent space*

***Figure 6.1:*** *Samples of the input space (left) and the latent space (right). The red line in Figure 6.1a is the trajectory along which the simulator/emulator response is plotted in Figure 6.7. The red dots annotated by small numbers denote the five points that are used for visualization in the following. Figure 6.1b shows a sample of the latent space ($A, B$ in Eq. (6.22)).*

We use different experimental design sizes $N \in \{50, 100, 150\}$ and a maximum degree of $p = 14$ for the PCE trajectories (with degree-adaptivity (Blatman and Sudret, 2011)). This results in a relative mean-squared error in the order of $10^{-3}/10^{-5}/10^{-10}$, respectively. We also test different numbers of trajectories $R \in \{10, 30, 100, 300\}$, and use a different experimental design for each trajectory. For each combination of experimental design size and number of trajectories, we conduct 50 independent repetitions. All resulting stochastic emulators are evaluated on the same validation set, consisting of $R_{\text{val}} = 10,000$ trajectories of the true stochastic simulator, each evaluated on a set of $N_{\text{val}} = 1,000$ points in the input space.

#### 6.4.1.2 Analysis of the KL-RV samples

To illustrate the type of result obtained with our proposed stochastic emulator described in Section 6.3, we now present scatterplots showing realizations of the following random quantities: 1) the KL-RV (compressed representation of PCE coefficients) resulting from step 7; 2) the PCE coefficients resulting from transforming the KL-RV samples to the PCE coefficient space. Detailed results for the prediction $\mathcal{M}(\boldsymbol{x}', \cdot)$ at a new location $\boldsymbol{x}'$ for a number of new trajectories are presented in Section 6.4.1.3 below. The results shown here are based on parametric inference of marginals and copula (Option (b) of Step 7).

The truncation of the KLE (Step 5 of our algorithm) typically results in two modes with eigenvalues $\lambda_1 \in [3, 5]$ and $\lambda_2 \approx 0.1$ depending on the size and realization of the experimental design. We display a specific example in Figure 6.2, which is computed from 100 trajectories with 150 samples each, and has eigenvalues $\lambda_1 = 4.46$ and $\lambda_2 = 0.10$. The figure shows resampled values for the KL-RV: in black, samples computed from validation trajectories by projecting first onto the truncated PCE space and then onto the eigenfunctions; in red, new samples drawn from the input object inferred in Step 7 of our algorithm (Section 6.3). We see that their inferred joint distribution (Beta and Gumbel marginals, with a Clayton copula) visually matches the validation data well.

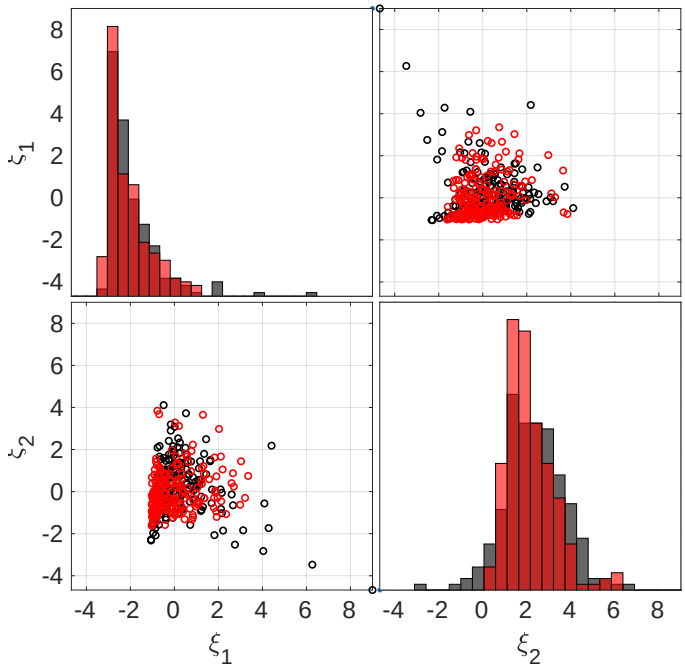



***Figure 6.2:*** *KL-RV coefficient samples computed from validation trajectories (black) and new samples from the stochastic emulator (red). This is data from one experiment with $N = 150$ and $R = 100$, max degree $p = 14$. Number of validation trajectories and resampled PCE coefficients: 200 each. Number of KL modes: $M = 2$. Inferred distribution of KL-RV: Beta and Gumbel, with Clayton copula (parameter 0.32). The corresponding eigenvalues are $\lambda_1 = 4.46$ and $\lambda_2 = 0.10$.*

The KL-RV are the compressed representation of the random PCE coefficients (which in turn encode trajectories). Mapping the realizations of the KL-RV back to the PCE coefficient space, we obtain the samples displayed in Figure 6.3 for $N = 150$ and $R = 100$. Validation samples from

the original stochastic simulator (generated by regressing them onto the truncated PCE basis) are displayed in black, while 200 resampled PCE coefficient vectors generated from the stochastic emulator are shown in red. We only show the 5 coefficients with maximal mean absolute value. We see that the validation samples have a slightly larger spread than the emulator samples, but that overall the behavior is matched well. Some parameters have linear functional dependence, e.g., $a_1, a_8$ and $a_{17}$, which is perfectly reproduced by the emulator. These parameters correspond to the basis functions $\boldsymbol{\alpha}_1 = (0,0,0)$ (constant term), $\boldsymbol{\alpha}_8 = (0,4,0)$ and $\boldsymbol{\alpha}_{17} = (0,6,0)$ and are needed to emulate the second term of the stochastic Ishigami model in Eq. (6.22). There are no interactions with the other terms, therefore a different value of $A$ just proportionally changes the relative weighting of these terms. A similar explanation holds for $a_2$ and $a_6$ with $\boldsymbol{\alpha}_2 = (1,0,0)$ and $\boldsymbol{\alpha}_6 = (1,0,2)$, which are involved in emulating the first and the third term of Eq. (6.22) and change proportionally with $B$.

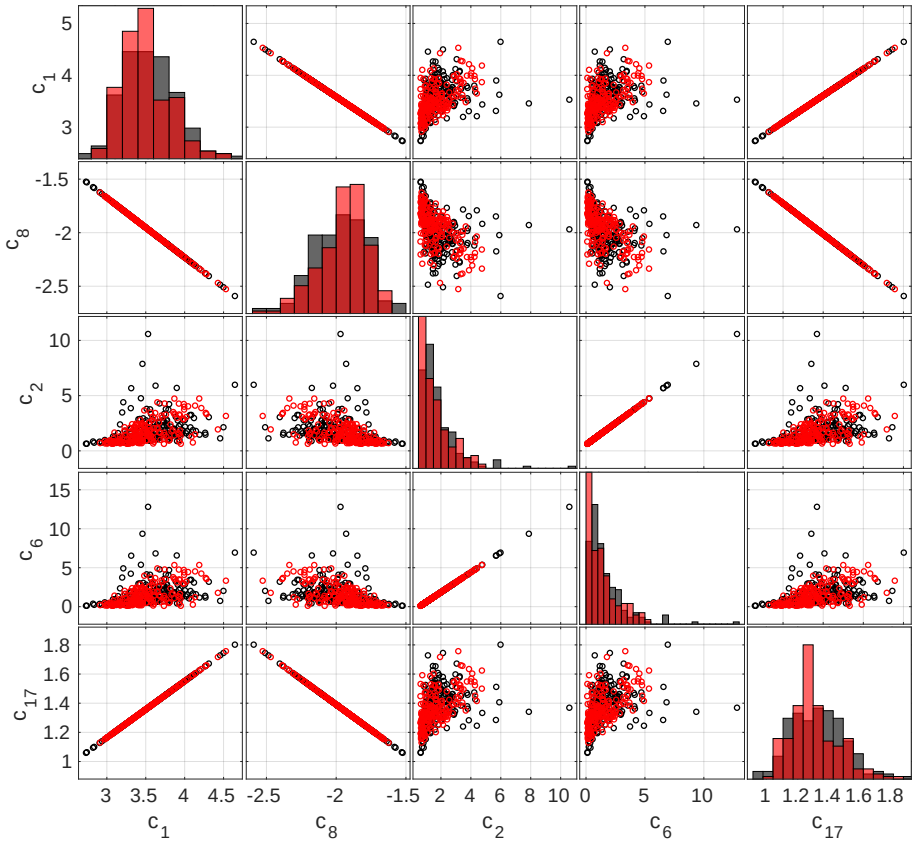



***Figure 6.3:*** *PCE coefficient samples computed from validation trajectories (black) and new samples from the stochastic emulator (red). This is data from one experiment with $N = 120$ and $R = 100$, max degree $p = 14$. Number of validation trajectories and resampled PCE coefficients: $R_{\text{val}} = 200$ each. The PCE coefficients are sorted by mean magnitude, and we only display the largest 5 out of total 75 nonzero coefficients.*

#### 6.4.1.3 Marginal performance on selected validation points

We now investigate the performance of the stochastic emulator with parametric inference of KL-RV marginals and copulas (choice 7b) on a selection of out-of-sample validation points, i.e., points that were not used for training.

Figure 6.4 shows the histograms and pairwise scatterplots of samples from the output random variables $Y_i = \mathcal{M}(\boldsymbol{x}^{(i)}, \cdot)$ and $\hat{Y}_i = \hat{\mathcal{M}}(\boldsymbol{x}^{(i)}, \cdot)$ of the stochastic simulator and emulator, respectively. The five selected validation locations $\{\boldsymbol{x}^{(i)}\}_{i=1}^5$ in the input space are visualized in Figure 6.1a by red dots. Each black (resp. red) point in Figure 6.4 is a new trajectory of the stochastic simulator (resp. emulator) evaluated at the five given points. Both samples have the same size (200 new trajectories). Overall, the model behavior is captured well, but the stochastic simulator has a slightly larger spread (see e.g. $Y_2$ vs. $Y_3$).

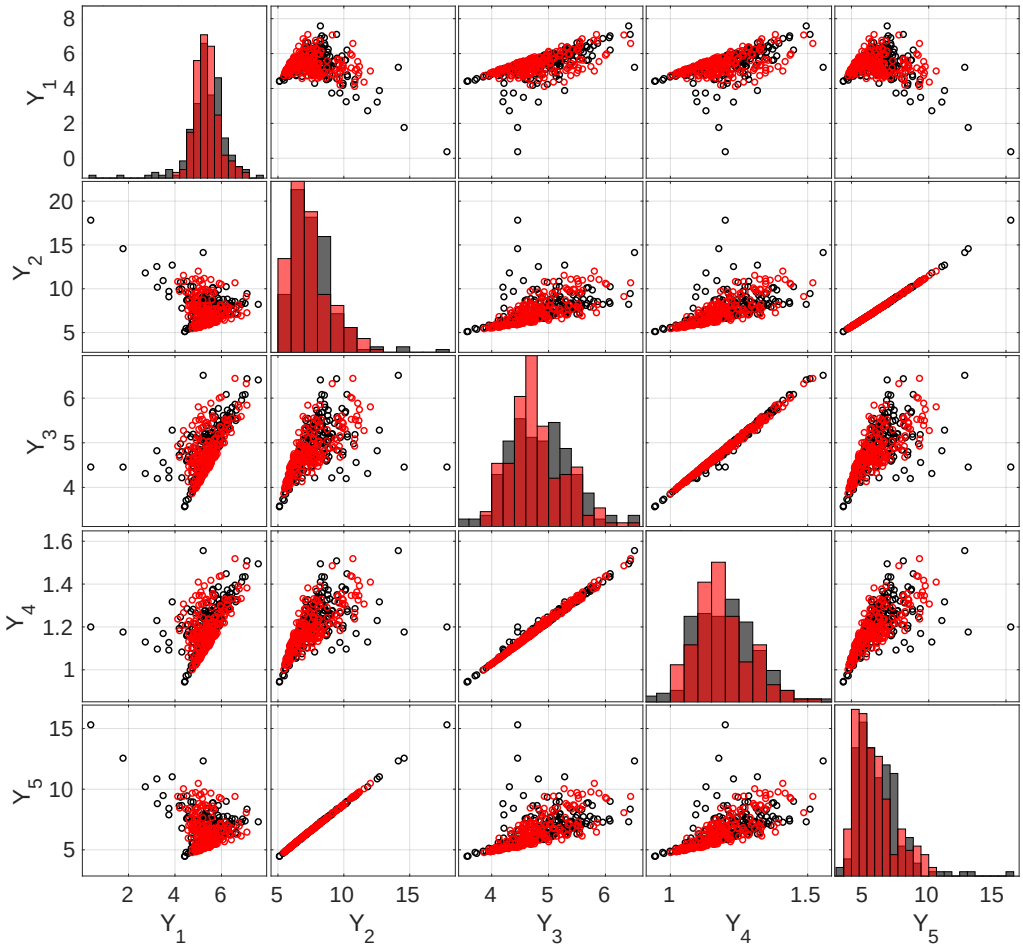



***Figure 6.4:*** *Scatterplot of output $Y_i = \mathcal{M}(\boldsymbol{x}^{(i)}, \cdot)$ of the stochastic simulator (black) and of output $\hat{Y}_i = \hat{\mathcal{M}}(\boldsymbol{x}^{(i)}, \cdot)$ of the parametric stochastic emulator (red, created from training set with $N = 150$, $R = 100$) for five validation points sampled from the input space. The location of these five points is illustrated in Figure 6.1a.*

From the data in the off-diagonal scatter plots in Figure 6.4, we can compute the sample covariance matrix. However, we can also compute the covariance analytically from the KLE eigenfunctions, using Eq. (6.9). In Figure 6.5, we use the five illustrative points shown in Figure 6.1a to compare this covariance estimate to a validation covariance matrix computed empirically from $10,000$ trajectories of the stochastic simulator. Qualitatively, the covariance is reproduced well, although the KLE-based covariance is slightly smaller in magnitude than the empirical covariance.

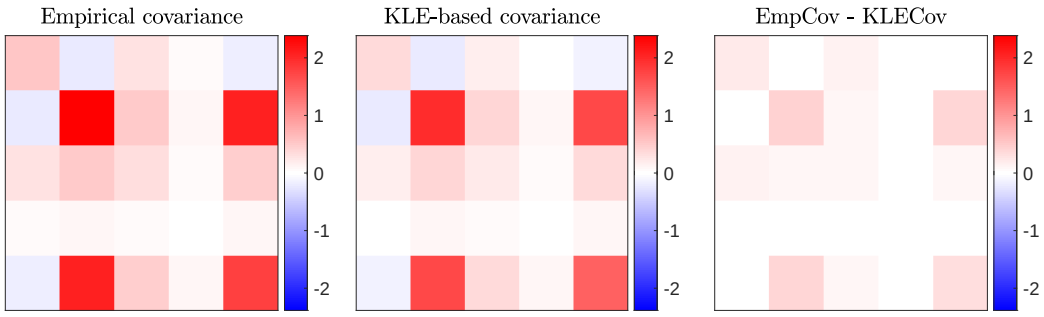
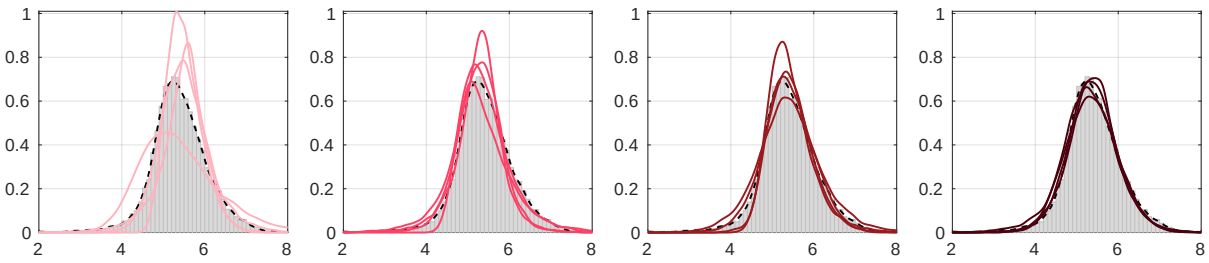



***Figure 6.5:*** *Covariance approximation for the Ishigami model for the $N_{\text{val}} = 5$ validation locations in the input space illustrated in Figure 6.1a. Computation based on $N = 150, R = 100$, max degree $p = 14$. KLE-based covariance: computed from eigenfunctions as in Eq. (6.9). Empirical covariance: based on the validation set comprising $10,000$ trajectories of the stochastic simulator.*

In Figure 6.6, we visualize the marginal distribution $f_{Y_1}$ of $Y_1 = \mathcal{M}(\boldsymbol{x}^{(1)}, \cdot)$ at one validation point (the point marked with "1" in Figure 6.1a) for an increasing number of trajectories in the training set, and 4 independent repetitions of each experiment. The estimates for the marginal distribution $f_{Y_1}$ are computed by KDE from $10,000$ samples from the constructed stochastic emulator, while the histogram and the dashed curve represents a validation set of $10,000$ samples of the original stochastic simulator. As expected, we observe that with an increasing number $R$ of trajectories, the shape of the predicted marginal becomes closer to the kernel density estimate of the validation set and shows less variation.

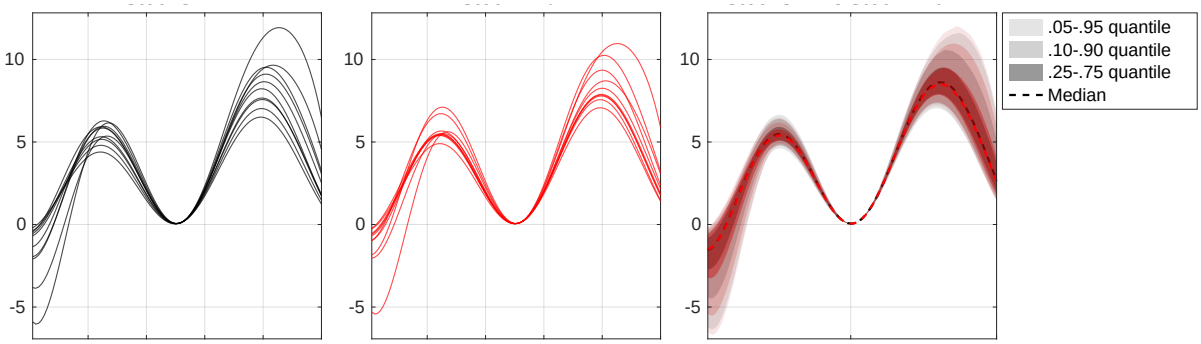


***(a)*** $R = 10$ ***(b)*** $R = 30$ ***(c)*** $R = 100$ ***(d)*** $R = 300$

***Figure 6.6:*** *Prediction of $Y$-marginal at validation point $\boldsymbol{x}_{\text{val}}^{(1)} = (-\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$ for the Ishigami model and $R = \{10, 30, 100, 300\}$ trajectories with the parametric stochastic emulator for 4 independent repetitions. Visualization of predicted marginals by KDE using $10,000$ samples. Number of experimental design points $N = 150$, max degree $p = 14$. The approximation error is in the order $\mathcal{O}(10^{-10})$.*

Finally, to assess visually how well the resampled trajectories match the behavior of the original stochastic model, we plot in Figure 6.7 a 1D slice of 10 new trajectories generated by the stochastic simulator (left) and the stochastic emulator (middle). The slice through the input space is shown in Figure 6.1a by a red line. On the right, data for the same slice is shown, but this time we show quantiles aggregated over $10,000$ new trajectories each. The trajectory slices look qualitatively similar, although there is a lot of variability between individual realizations. From the aggregated data on the left, we see that the bulk of the distribution (10%-90% quantile) is predicted quite accurately. Interestingly, in Figure 6.7c it seems that the trajectories of the

stochastic emulator (red) have a larger spread than the ones of the simulator (black), contrary to the results earlier in this section, which always showed the simulator having a larger spread than the emulator. This illustrates the difficulty of inferring global behavior from local observations. Theoretically, the emulator should have a smaller variance than the simulator, because terms are missing from Eq. (6.9) due to truncation.




***(a)*** *Stochastic simulator*

***(b)*** *Stochastic emulator*

***(c)*** *Comparison of quantiles for both models*

***Figure 6.7:*** *Visualization of stochastic simulator/emulator response $Y$ when following a 1D slice through the input space, which is illustrated with a red line in Figure 6.1a. $N = 150, R = 100$. The left and middle plots show 10 trajectories each. The right plot aggregates the values from $10,000$ trajectories to show quantile information.*

#### 6.4.1.4 Convergence with the number of trajectories

To assess the global performance of our proposed method, we now construct stochastic emulators for all combinations of input space experimental design sizes $N \in \{50, 100, 150\}$ and numbers of trajectories in the range $R \in \{10, 30, 100, 300\}$. We then evaluate each of the resulting stochastic emulators $R_{\text{val}} = 10,000$ times at $N_{\text{val}} = 1,000$ validation points in the input space (out-of-sample, i.e. not used for training) and compute the errors as described in Section 6.4. Each combination is independently repeated 50 times to account for the statistical uncertainty of the sampling of both experimental design and trajectories, which allows us to display results in the form of Tukey boxplots.

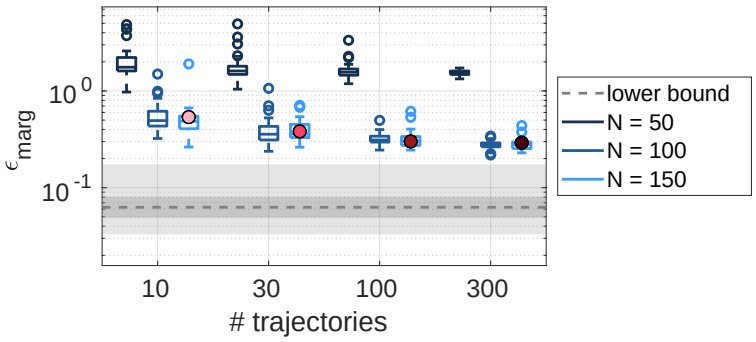
In Figure 6.8a, we display the global convergence of marginal predictions for the parametric stochastic emulator in terms of $\epsilon_{\text{marg}}$ defined in Eq. (6.20). Each boxplot represents one experiment (i.e., a specific number of experimental design points and number of trajectories), repeated independently 50 times. The value of the averaged and normalized Wasserstein distance $\epsilon_{\text{marg}}$ is, by itself, difficult to interpret. To aid the interpretation and give an idea of the quality of the approximation, we add two auxiliary quantities to the plot:

- The averaged and normalized Wasserstein distance is computed based on samples. As a lower bound, we independently sample $100 \times 2$ validation sets (each consisting of $R_{\text{val}} = 10,000$ trajectories evaluated at $N_{\text{val}} = 1,000$ points in the input domain; each pair of validation sets shares the same points in the input domain). We then compute the error in Eq. (6.20) for each of the 100 pairs. The median and quantiles (0.25–0.75 and 0.05–0.95)

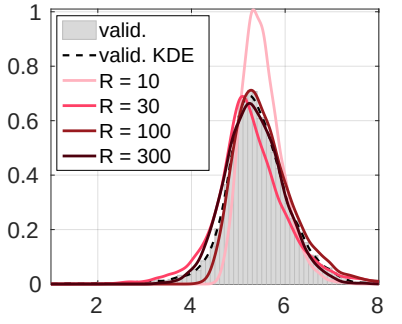
of this value are displayed in Figure 6.8a in gray, indicating the best possible error that can be achieved due to the natural variability of the sample estimates.

- A priori, it is unclear which value of the (averaged) normalized Wasserstein distance corresponds to predicted marginals that are visually close to the true marginals. To have some concrete examples on what a specific value of the normalized Wasserstein distance means, we consider the marginals predicted at one chosen validation point, shown in Figure 6.8b. We add the corresponding value of the normalized Wasserstein distance between simulator and emulator prediction as a small colored circle to the plot in Figure 6.8a.

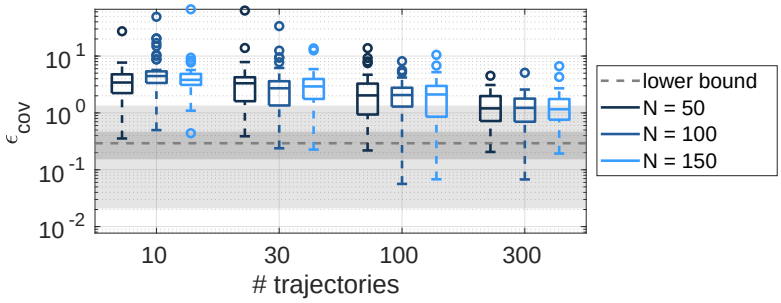
We observe that the quality of the marginal estimates improves as we increase the size of the input parameter sample, which is expected since the PCE approximation of the trajectories becomes better with increasing experimental design size. $N = 50$ points are clearly too few to achieve a good estimate, whereas $N = 100$ and $N = 150$ show convergence with the number of trajectories, indicating that the error from statistical inference is the dominating one. While the convergence of the marginals with the number of trajectories looks slow, the improvement of the marginal shapes is actually significant (compare the values with Figure 6.8b).




*(a) Convergence of $\epsilon_{marg}$*




*(b) Marginal at $\boldsymbol{x}_{val}^{(1)} = (-\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$*




*(c) Convergence of $\epsilon_{cov}$*

***Figure 6.8:*** *Convergence of $\epsilon_{marg}$ and $\epsilon_{cov}$ (Eqs. (6.20) and (6.21)) for increasing number of available trajectories and parameter locations. Results for the stochastic emulator with parametric inference (choice 7b of our algorithm in Section 6.3) and 50 replications. The errors are computed based on a validation set of size $N_{val} = 1,000$, $R_{val} = 10,000$. The gray areas and the dashed line represent quantiles and the median of a lower bound estimate for the respective error measure computed from $100 \times 2$ independent MC samples of size $R_{val} = 10,000$ generated by the true stochastic simulator. The colored points in Figure 6.8a correspond to the results for a single replication and validation point as shown in Figure 6.8b and help assess the meaning of the numerical error measures.*

In addition to marginal predictions, our stochastic emulator can also emulate the covariance function, using Eq. (6.9). Since this equation relies only on the KL eigenfunctions, not on the KL-RV, the choice of inference method in Step 7 of our algorithm in Section 6.3 does not affect these results. In Figure 6.8c we display the convergence of $\epsilon_{\text{cov}}$ from Eq. (6.21). We observe that the error decreases with increasing numbers of trajectories. For the largest numbers of trajectories and experimental design points, the error is already in the range of the rough lower bound on achievable accuracy (obtained as described above by empirical sampling of the true model). Interestingly, unlike the marginal error in Figure 6.8a, an increasing number $N$ of input parameter samples does not lead to a smaller covariance error. This indicates that the covariance estimate is less sensitive to the quality of the trajectory approximation, while the inference of the distribution of the KL-RV is more sensitive to it.

So far, we showed results for the stochastic emulator with parametric inference only (Option 7b). Now, in Figure 6.9 we compare the four inference options described in Step 7 of the algorithm in Section 6.3 with the results of a fifth method described in Remark 1, which we call here PCE-KDE. We use the experimental design size $N = 100$, which yields PCE approximations with relative validation error of $10^{-5}$. Due to this close fit, the PCE-KDE estimate can be considered as near-optimal estimate given the available data.

The error $\epsilon_{\text{marg}}$ is again computed on a validation set consisting of $R_{\text{val}} = 10,000$ trajectories evaluated at $N_{\text{val}} = 1,000$ points in the input space.

We observe that for 20 trajectories, the five methods show almost identical performance. This suggests that 20 trajectories are not yet enough to infer a distribution that is able to generalize to unseen data, so that any marginal distribution with mean zero and unit standard deviation provides a reasonable approximation. Comparing with the results for PCE-KDE, we see that our emulator is similarly accurate in prediction at an unseen point as a kernel density estimate using the training set of highly accurate PCE trajectories. This suggests that we do not lose much accuracy by applying our KLE approach on top of the PCE approximation, which can be seen as a form of dimension reduction in the stochastic space.

For the larger number of trajectories, $R = 100$ and $R = 500$, we do observe a difference between the performance of the different marginal inference methods: standard Gaussians perform worst, while kernel density estimation without copula performs best of all the inference methods considered. Kernel density estimation with independence assumption performs almost on par with the PCE-KDE estimate. This suggests that the KL-RV are close to independent in this case, and that fitting a vine copula (using the available pair copulas) does not improve the overall inference, at least in the considered cases of limited data. It also demonstrates that the true KL-RV distribution is not well approximated by independent standard Gaussians nor by other currently available parametric families, but that it can be approximated well by the more flexible kernel density estimation. Parametric inference, offering a variety of standard marginal shapes, performs slightly better than Gaussian random variables.

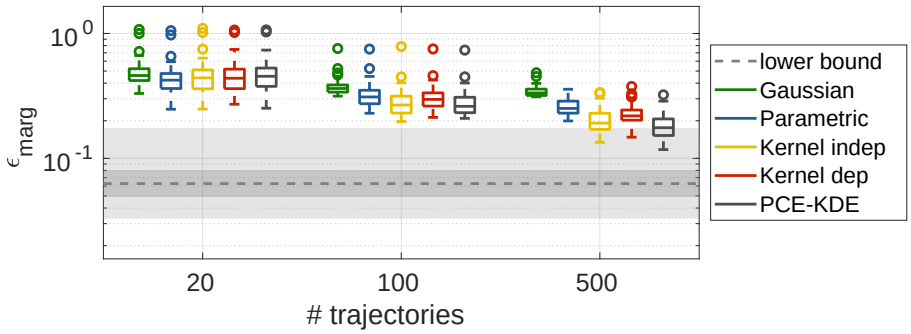



***Figure 6.9:*** *Convergence of average normalized Wasserstein distance. Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with the method PCE-KDE described in Remark 1. $N = 100$ and $p = 14$. Errors are computed based on $N_{\text{val}} = 1,000$, $R_{\text{val}} = 10,000$, for 50 replications. The gray areas and dashed line have the same meaning as in Figure 6.8.*

### 6.4.2 Borehole function with latent variables

As a second example, we consider the well-known borehole function, which computes the water flow between two aquifers that are connected by a borehole (Harper and Gupta, 1983). It depends on eight parameters and is defined by

$$B(R_w, H_u, K_w, R, T_u, T_l, H_l, L) = \frac{2\pi T_u (H_u - H_l)}{\ln(R/R_w)\left(1 + \frac{2LT_u}{\ln(R/R_w)R_w^2 K_w} + \frac{T_u}{T_l}\right)}. \tag{6.23}$$

Its input random variables and their distributions are provided in Table 6.1.

We consider five parameters ($\boldsymbol{\Xi} = (R, T_u, T_l, H_l, L)$) of the borehole function to be latent, resulting in the three-dimensional stochastic simulator $\tilde{B}(r_w, h_u, k_w) = B(r_w, h_u, k_w; \boldsymbol{\Xi})$.

For the three-dimensional input space, we use $N = \{20, 30, 60\}$ input samples and a maximal PCE degree of $p = 6$. The accuracy of the borehole approximation in terms of relative mean-squared validation error is in the order of $10^{-3}/10^{-7}/10^{-10}$ for the different experimental design sizes. The number of trajectories is in the range $R = \{10, 30, 100, 300\}$.

This results in typically $M = 2$ KL modes for an explained variance threshold of 99.9%. The eigenvalues of the KLE are approximately $\lambda_1 \approx 170$ and $\lambda_2 \approx 0.5$. The first mode alone covers more than 99.5% of the total variance, even though five independent parameters are used as latent variables. Two of these have a significant total Sobol' index, and the sum of the first-order indices of the latent group is 19%. We will investigate this phenomenon in more detail in Section 6.5 below.

As before, we analyze the global convergence of the marginal and covariance approximation for increasing numbers of input samples and trajectories, and we compare different methods for inferring the distribution of the KL-RV as described in Step 7 of our algorithm (Section 6.3). For the detailed explanation of the error measures, the setup of the convergence study, and the interpretation of the plots, see Sections 6.4 and 6.4.1.

In Figure 6.10a, we see that our method converges in both global error metrics ($\epsilon_{\text{marg}}$ and $\epsilon_{\text{cov}}$) towards the rough empirical lower bound indicated by the gray area and dashed line. For $\epsilon_{\text{marg}}$,

***Table 6.1:*** *Borehole function: Input random variables and their distributions. For the borehole stochastic simulator with hidden variables, five of the eight variables (marked by italic letters) are considered latent.*

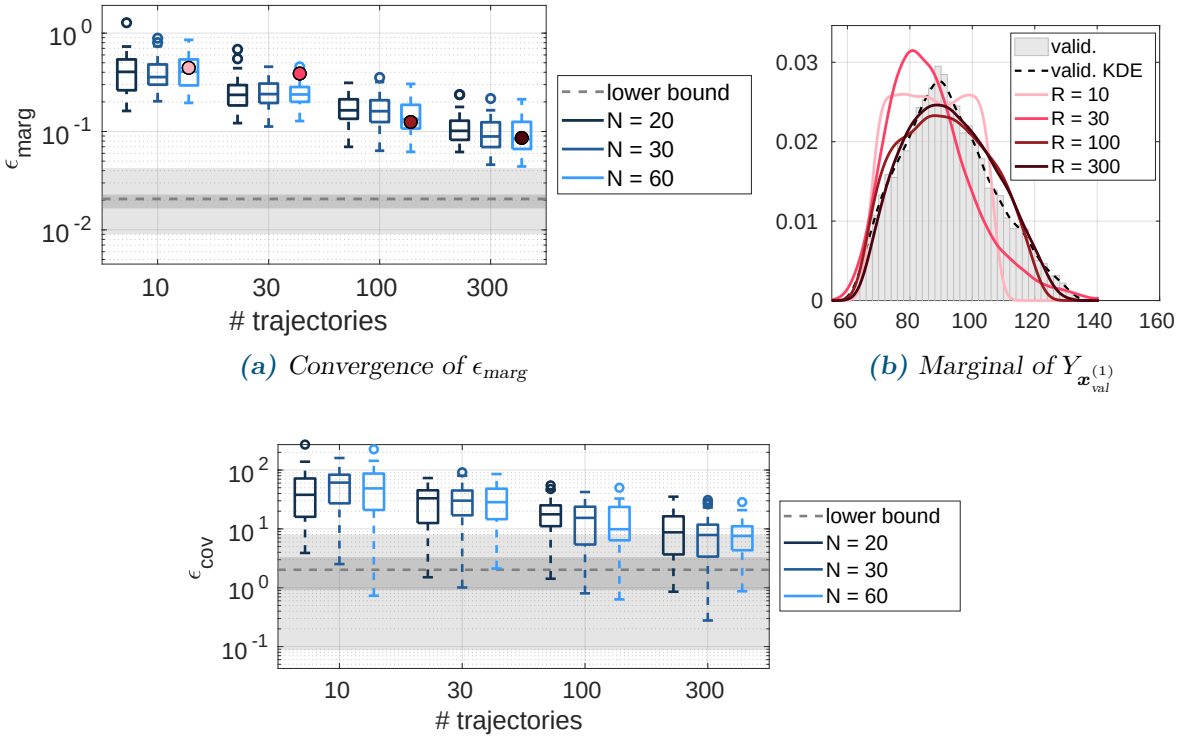
| Variable | Distribution | Description | Total Sobol' index |
|---|---|---|---|
| $R_w$ | $\mathcal{N}(0.10, 0.0161812)$ | borehole radius | 6.94e-01 |
| $H_u$ | $\mathcal{U}([990, 1110])$ | potentiometric head of upper aquifer | 1.06e-01 |
| $K_w$ | $\mathcal{U}([9855, 12045])$ | borehole hydraulic conductivity | 2.51e-02 |
| $R$ | Lognormal$([7.71, 1.0056])$ | *radius of influence* | *2.77e-06* |
| $T_u$ | $\mathcal{U}([63070, 115600])$ | *transmissivity of upper aquifer* | *2.10e-08* |
| $T_l$ | $\mathcal{U}([63.1, 116])$ | *transmissivity of lower aquifer* | *8.23e-06* |
| $H_l$ | $\mathcal{U}([700, 820])$ | *potentiometric head of lower aquifer* | *1.06e-01* |
| $L$ | $\mathcal{U}([1120, 1680])$ | *borehole length* | *1.03e-01* |

the difference between the results for the three experimental design sizes $N = 20, 30$, and 60 is small. This indicates that at least for this model, a validation mean-squared error smaller than $\mathcal{O}(10^{-3})$ does not lead to significantly more accurate results, and that below this accuracy the error is dominated by the inference error.

The convergence of the emulated covariance function is displayed in Figure 6.10c. As expected, the global error become smaller with an increasing number of trajectories, and approaches the lower bound representing the variability due to the error being computed from samples. Again, the difference in the results for the three experimental design sizes $N = 20, 30, 60$ is small. Since the first mode accounts for more than 99.5% of the explained variance, the first KLE eigenfunction has the dominating influence on the covariance estimation (Eq. (6.9)). The results indicate that the first eigenfunction and its eigenvalue are estimated accurately already for the smallest experimental design sizes.

The comparison of the different inference methods for the distribution of the KL-RV (Step 7 of the algorithm) is displayed in Figure 6.11. Similarly as for the Ishigami function, we observe that for a small number of trajectories ($R = 10$ and 30) the four inference methods and PCE-KDE show almost the same performance. Modeling the KL-RV with standard Gaussian distributions seems to offer a slight advantage (resulting in a slightly smaller median error and smaller variability) for small numbers of trajectories, probably because they make the strongest assumptions on the distribution shape, which is advantageous for generalizability in the case of small data. As the number of trajectories grows, a similar pattern as in Section 6.4.1 emerges: standard Gaussian inference shows the worst performance, followed by parametric inference. Inference with kernel density estimation (dependent and independent) shows the best performance, on par with the PCE-KDE estimate, which (due to the high accuracy of the PCE approximations for $N = 60$) represents the near-optimal estimate given the available training data.

Interestingly, there is no significant difference between the performance of KDE with and without the independence assumption. Here the magnitude of the eigenvalues might offer an explanation: with more than two orders of magnitude difference between $\lambda_1$ and $\lambda_2$, the dependence between the two random variables $\xi_1$ and $\xi_2$ does not influence the resulting predictions as much as the correct identification of the marginal shape of the first KL-RV $\xi_1$.




*(a)* Convergence of $\epsilon_{marg}$

*(b)* Marginal of $Y_{\boldsymbol{x}_{val}^{(1)}}$

*(c)* Convergence of $\epsilon_{cov}$

***Figure 6.10:*** *Convergence of the $\epsilon_{marg}$ and $\epsilon_{cov}$ (Eqs. (6.20) and (6.21)) for an increasing number of available trajectories and parameter locations. Results for the stochastic emulator with parametric inference (Option 7b) and 50 replications. The errors are computed based on a validation set of size $N_{val} = 1,000$, $R_{val} = 10,000$. The gray areas and the dashed line represent quantiles and the median of a lower bound estimate for the respective error measure computed from $100 \times 2$ independent MC samples of size $R_{val} = 10,000$ generated by the true stochastic simulator. The colored points in Figure 6.10a correspond to the results for a single replication and validation point as shown in Figure 6.10b and help assess the meaning of the numerical error measures.*

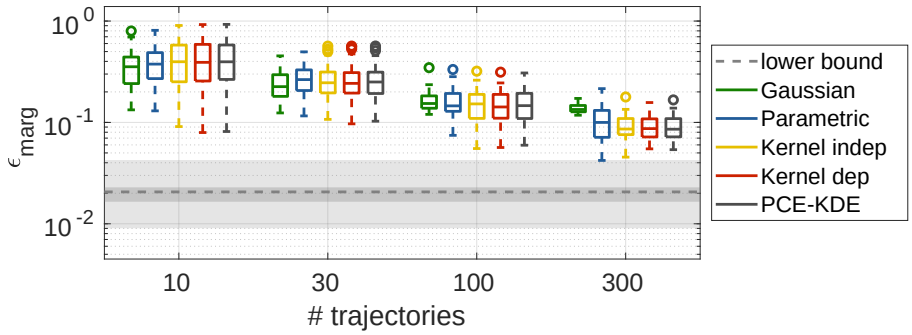



***Figure 6.11:*** *Convergence of $\epsilon_{marg}$. Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with PCE-KDE described in Remark 1. $N = 60$ and $p = 6$. Errors are computed based on $N_{val} = 1,000$, $R_{val} = 10,000$, and 50 replications. The gray areas and dashed line have the same meaning as in Figure 6.10.*

### 6.4.3 Heston stochastic volatility model for a stock price

As a third example, we consider the Heston stochastic volatility model, which describes a stock price $Y_t$ (Heston, 1993) with its volatility $\nu_t$ modeled as stochastic process:

$$\mathrm{d}U_t = \mu U_t \mathrm{d}t + \sqrt{\nu_t} U_t \mathrm{d}W_t^{(1)}, \tag{6.24}$$

$$\mathrm{d}\nu_t = \kappa(\theta - \nu_t)\mathrm{d}t + \sigma\sqrt{\nu_t}\mathrm{d}W_t^{(2)} \tag{6.25}$$

with two Wiener processes $W_t^{(1)}$ and $W_t^{(2)}$ with correlation coefficient $\rho$. This model has six uniformly distributed parameters $\boldsymbol{X} = (\mu, \kappa, \theta, \sigma, \rho, \nu_0)$ detailed in Table 6.2, the bounds of which are calibrated from real data as described in Zhu and Sudret (2021c). The quantity of interest is

$$Y_{\boldsymbol{x}} = U_1(\boldsymbol{X} = \boldsymbol{x}), \tag{6.26}$$

i.e., the stock price after 1 year. As proposed by Zhu and Sudret (2021c), we set $U_0 = 1$ and use the Euler-Maruyama method to integrate the system of stochastic differential equations (SDEs) and replace $\nu_t$ by $\max(\nu_t, 0)$ to avoid negative values of $\nu_t$. This model is a stochastic simulator due to the stochasticity induced by the two Wiener processes $W_t^{(1)}$ and $W_t^{(2)}$ driving the SDEs. A trajectory in the parameter space $\mathcal{D}$ is obtained by fixing the realizations of these processes and evaluating Eqs. (6.24) and (6.25) for $\boldsymbol{x} \in \mathcal{D}$.

For the six-dimensional input space, we use $N = \{50, 100, 150\}$ input samples and a maximal PCE degree of $p = 7$. The accuracy of the approximation in terms of relative mean-squared validation error is ca. $\mathcal{O}(0.03)/\mathcal{O}(0.02)/\mathcal{O}(0.006)$ for the different experimental design sizes. This means that the Heston model is not particularly well approximated by PCE, even for rather large experimental designs. We use a number of trajectories in the range $R = \{10, 30, 100, 300\}$.

This results in typically $M = 4$ to 6 KL modes for an explained variance threshold of 99.9%. The first eigenvalue is $\lambda_1 \approx 0.05$ and usually covers more than 97% of the variance.

***Table 6.2:*** *Parameters and their distributions for the Heston SDE model.*

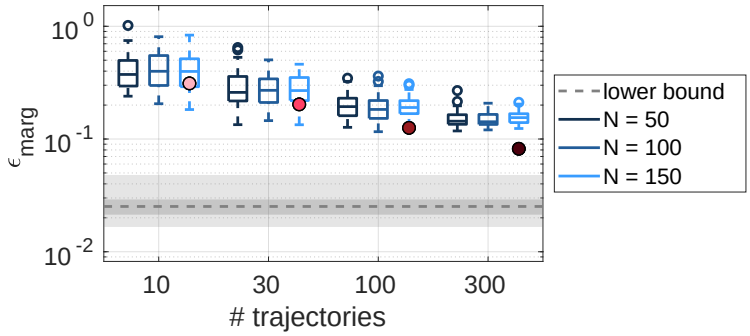
| Variable | Distribution | Description |
|---|---|---|
| $\mu$ | $\mathcal{U}([0, 0.1])$ | Expected return rate |
| $\kappa$ | $\mathcal{U}([0.3, 2])$ | Mean reversion speed of the volatility |
| $\theta$ | $\mathcal{U}([0.02, 0.07])$ | Long term mean of the volatility |
| $\sigma$ | $\mathcal{U}([0.2, 0.4])$ | Volatility of the volatility |
| $\rho$ | $\mathcal{U}([-1, -0.5])$ | Correlation coefficient between $\mathrm{d}W_t^{(1)}$ and $\mathrm{d}W_t^{(2)}$ |
| $\nu_0$ | $\mathcal{U}([0.02, 0.07])$ | Initial volatility |

Again, we analyze the global convergence of the marginal and covariance approximation in the same way as in the preceding sections. The marginal approximations of the parametric stochastic emulator converge with increasing experimental design size and number of trajectories, but slowly, as displayed in Figure 6.12a. There is no significant difference between the three experimental design sizes $N = 50, 100, 150$. This indicates that the improvement due to a better PCE approximation for an increasing number of experimental design points is overshadowed by the inaccuracy due to the inference of the KL-RV. This, in turn, could be because the PCE approximation is not yet sufficiently accurate (note that the relative validation error is in the

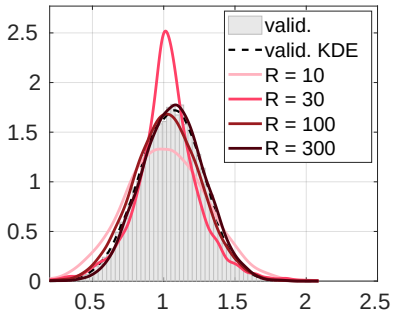
order of $10^{-2}$ for all ED sizes.) Even for the largest number of trajectories, the averaged and normalized Wasserstein distance between the responses of the true model and the emulator is still much larger than the variability resulting from sampling the true model, which is illustrated by the gray areas and the dashed line in Figure 6.12a (quantiles and median, respectively). Comparing the boxplots to the colored points corresponding to the marginal estimates illustrated in Figure 6.12b, we observe that the marginal shape of the stochastic response for one validation point $\boldsymbol{x}_{\text{val}}^{(1)}$ is already captured quite well for 100-300 trajectories (with the value for $R = 300$ being a bit of an outlier).

The convergence of the covariance function is shown in Figure 6.12c. As expected, the covariance estimate becomes better with increasing number of trajectories, even approaching the lower bound obtained by resampling the original model. However, we observe again that an increasing number of experimental design points does not influence the estimate much, which indicates that the covariance estimate is quite robust against the trajectory approximation quality. Since the first mode is also dominant for this example (accounting for ca. 97% of the explained variance), it indicates that the first eigenfunction is well estimated already for small experimental design sizes.

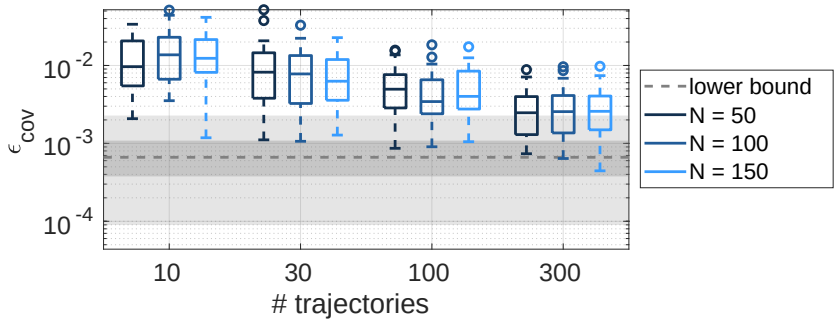
Figure 6.13 shows the comparison between the different methods for KL-RV inference (Step 7 of the algorithm in Section 6.3) as described in Section 6.4.1.4. All four methods perform comparably. The independent standard Gaussian approximation performs slightly better than the other methods in the case of few trajectories and slightly worse for the case of many trajectories, which is consistent with the previous cases. Again, KDE with and without dependence shows almost identical performance, indicating that the either the true copula is the independence copula, or that the existing parametric copulas are not suitable for capturing the dependence structure. While the inference methods show a similar performance to PCE-KDE for smaller numbers of trajectories, PCE-KDE finds a better marginal approximation when $R = 300$ trajectories are available. This indicates that some information is lost in the KLE procedure of Section 6.3.




*(a) Convergence of $\epsilon_{marg}$*




*(b) Marginal of $Y_{\boldsymbol{x}_{val}^{(1)}}$*




*(c) Convergence of $\epsilon_{cov}$*

***Figure 6.12:*** *Convergence of $\epsilon_{marg}$ and $\epsilon_{cov}$ (Eqs. (6.20) and (6.21)) for an increasing number of available trajectories and parameter locations. Results for the stochastic emulator with parametric inference (choice 7b) and 50 replications. The errors are computed based on a validation set of size $N_{val} = 1,000$, $R_{val} = 10,000$. The gray areas and the dashed line represent quantiles and the median of a lower bound estimate for the respective error measure computed from $100 \times 2$ independent MC samples of size $R_{val} = 10,000$ generated by the true stochastic simulator. The colored points in Figure 6.12a correspond to the results for a single replication and validation point as shown in Figure 6.12b and help assess the meaning of the numerical error measures.*

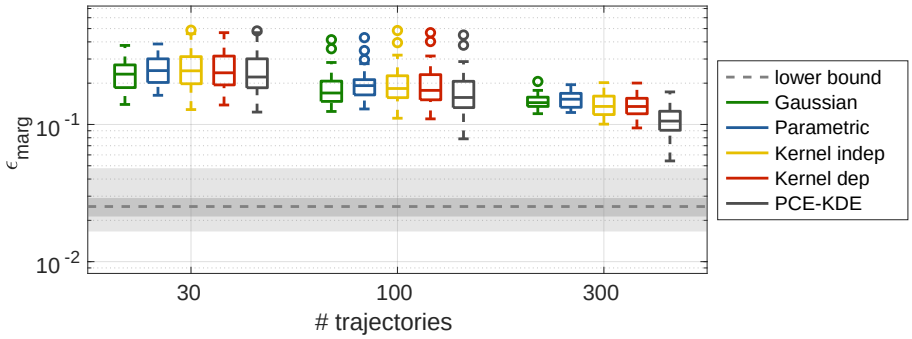



***Figure 6.13:*** *Convergence of marginals (Wasserstein distance). Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with PCE-KDE described in Remark 1. $N = 150$ and $p = 5$. Errors are computed based on $N_{val} = 1,000$, $R_{val} = 10,000$, and 50 replications. The gray areas and dashed line have the same meaning as in Figure 6.12.*

## 6.5 Considerations on the number of modes

In this section we investigate how many modes $K$ we can expect in the stochastic emulator of Eq. (6.18). We consider here a certain class of stochastic simulators that arise from a deterministic model $\boldsymbol{z} \mapsto \mathcal{M}(\boldsymbol{z})$ by considering some of its variables as *hidden* (or *latent*). In other words, the stochastic simulator is $\mathcal{M}(\boldsymbol{X}, \boldsymbol{\Xi})$, where $\boldsymbol{X}$ are the explicit input variables, and $\boldsymbol{\Xi}$ the latent variables inducing the stochasticity (the random events, see Section 6.2.1). Assume that all components of $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{\Xi})$ are independent, and denote by $f_k$ the marginal distribution of $Z_k$.

Assume further that the deterministic simulator $\mathcal{M}$ has finite variance under $f_{\boldsymbol{Z}}$. Then it can be decomposed into the Hoeffding-Sobol' decomposition (a.k.a. ANOVA decomposition, analysis of variance) (Hoeffding, 1948; Sobol and Gresham, 1995) as

$$\mathcal{M}(\boldsymbol{Z}) = m_0 + \sum_{1 \le i \le d} m_i(Z_i) + \sum_{1 \le i < j \le d} m_{i,j}(Z_i, Z_j) + \cdots + m_{1,\dots,d}(Z_1, \dots, Z_d) \tag{6.27}$$

where the terms satisfy $\int m_I(\boldsymbol{Z}_I) f_k(z_k) \mathrm{d}z_k = 0$ for all $k \in I \subset \{1, \dots, d\}$. $m_0$ is the mean of $\mathcal{M}(\boldsymbol{Z})$. The univariate terms $m_i$ are called *main effects*, and the remaining summands are *interaction terms* of increasing order.

Now we group the summands of Eq. (6.27) according to whether they involve only input variables, only latent variables, or some variables from both groups. This results in the following decomposition:

$$\mathcal{M}(\boldsymbol{X}, \boldsymbol{\Xi}) = m_0 + \mathcal{M}_1(\boldsymbol{X}) + \mathcal{M}_2(\boldsymbol{\Xi}) + \mathcal{M}_{12}(\boldsymbol{X}, \boldsymbol{\Xi}) \tag{6.28}$$

where, e.g., $\mathcal{M}_1(\boldsymbol{x}) = \sum_{I : \boldsymbol{Z}_I \subset \boldsymbol{X}} m_I(\boldsymbol{z}_I)$. The last summand of Eq. (6.28) contains all interaction terms from Eq. (6.27) that involve at least one input and at least one latent variable.

It is a rather common phenomenon in uncertainty quantification that engineering models actually have near-zero interaction terms. In that case, the model is essentially additive with respect to the two groups of variables $\boldsymbol{X}$ and $\boldsymbol{\Xi}$:

$$\mathcal{M}(\boldsymbol{X}, \boldsymbol{\Xi}) \approx m_0 + m_1(\boldsymbol{X}) + m_2(\boldsymbol{\Xi}). \tag{6.29}$$

This means that any realization $\boldsymbol{\xi}$ of the unknown latent variables $\boldsymbol{\Xi}$ results in a constant shift of $\mathcal{M}(\cdot, \boldsymbol{\xi})$ regardless of the value of the input parameters $\boldsymbol{x}$, a behavior that can be accurately modeled by a single KL mode: if equality holds in Eq. (6.29), the mean function is $\mu(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\Xi}}[\mathcal{M}(\boldsymbol{x}, \boldsymbol{\Xi})] = m_0 + m_1(\boldsymbol{X})$, the covariance function is $c(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Var}[m_2(\boldsymbol{\Xi})]$, and the only nonzero eigenvalue of Eq. (6.7) is $\lambda_1 = \mathrm{Var}[m_2(\boldsymbol{\Xi})]$ with eigenfunction $\phi_1(\boldsymbol{x}) = 1$.

We have observed this in the numerical examples in Section 6.4, e.g., for the case of the borehole model with hidden variables. The deterministic borehole model defined in Eq. (6.23) has relatively low interaction: despite its nonlinearity, under the input uncertainties in Table 6.1 its first order Sobol' indices (Sobol', 1993) sum up to $\sum_{i=1}^{8} S_i^1 \approx 96.7\%$. The interaction effect between the explicit and the latent group is around 2%. Therefore, only one mode is sufficient to model the stochastic simulator that results from treating several of the model's variables as latent.

## 6.6 Discussion and conclusions

We presented a spectral surrogate model for stochastic simulators, a special class of computational models whose response for a given input is a random variable. Our method relies on several advanced techniques for modeling uncertainties, such as polynomial chaos expansion (PCE), Karhunen-Loève expansion (KLE), and statistical inference of multivariate distributions. The resulting surrogate model is not only able to emulate the marginal distributions and the covariance structure, but it can also generate new trajectories.

The form of our surrogate model provides insight into the model structure. The number of expansion modes indicates how high-dimensional the underlying stochasticity is. The eigenfunctions of the KLE, which are polynomials, give information about how the input parameters influence the stochastic simulator output. Even though our numerical examples were chosen to represent a range of cases of increasing complexity, we found that one mode was usually sufficient to explain more than 95% of the variance of the stochastic simulator. We were able to explain this phenomenon for the common case of stochastic simulators that arise from finite-dimensional deterministic models with independent inputs and finite variance by treating some of their input variables as latent. Considering the Hoeffding-Sobol' decomposition of the underlying deterministic simulator, we found that if the interaction terms between the explicit and latent variables are negligible, one single KL mode will be sufficient to emulate the behavior of the stochastic simulator. Indeed, by experience, this is a common occurrence in applications of uncertainty quantification. Interactions are rarely dominant in engineering problems, and so one KL mode might suffice in many cases.

From our numerical experiments, we found that the Gaussian (or more generally, parametric) approximation of marginals of the KL-random variables (KL-RV) can be preferable if the number of available trajectories is very small. When more trajectories are available, the better choice is kernel density estimation. Since the first mode was dominant in our numerical examples, the characterization of the dependence between KL-RV turned out not to be crucial, at least for the class of applications considered.

Our numerical tests reveal that the emulator is able to capture the true model behavior reasonably well, as long as enough input samples and trajectories are used. Due to the sequential nature of our approach, it is important to use enough points in the experimental design: if the PCE approximation is not accurate enough, also the inferred distribution for the KL-RV will be inaccurate. Interestingly, we observed in all three examples that the covariance estimate was not heavily influenced by using a larger experimental design, even though the latter typically results in more accurate PCE approximations of the trajectories. This indicates that the number of trajectories is more important for the covariance approximation than the quality of the PCE approximation. Also, it seems that (since the first mode was dominant for all investigated models) the first KLE eigenfunction can be identified accurately already from a small experimental design.

Note that our surrogate relies on the assumption that the trajectories are well approximated by sparse PCE, an assumption not fulfilled by some stochastic simulators, e.g., ones with discontinuous or non-differentiable trajectories in the input parameter space. This could be circumvented

by using another basis specially adapted to the purpose, such as wavelets. Furthermore, by construction, our emulator is only suitable for stochastic simulators whose response is correlated throughout the input domain. If there is little to no correlation between the responses at different points in the input space, KLE (which is ultimately a dimension reduction technique), would need infeasibly many modes to converge.

Our methodology can be extended in several ways. The computation of the sparse PCE approximation of the trajectories could be done jointly for all trajectories, instead of fitting each trajectory separately and later discarding regressors. While in our study the dependence between KL-RV was not crucial for the accuracy of the emulator, an improved estimation of the dependence structure would be desirable if for future models more modes turn out to be important. Furthermore, an interesting question is under which circumstances one mode is enough to represent the underlying stochasticity of stochastic simulators, and how the methodology can be adapted to take advantage of this phenomenon. The general idea of representing trajectories by their coefficients, and after dimension reduction modeling their joint distribution, can also be applied outside spectral expansions, e.g. for Bayesian neural networks, at the cost of losing some of the analytical properties following from orthogonality. Finally, in the spirit of *common random numbers* (Pearce et al., 2022), the applicability of our stochastic emulator for purposes such as optimization should be explored.

## Acknowledgments

This paper is 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

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Ins. Math. Eco.*, 44(2):182–198.

Abraham, S., Tsirikoglou, P., Miranda, J., Lacor, C., Contino, F., and Ghorbaniasl, G. (2018). Spectral representation of stochastic field data using sparse polynomial chaos expansions. *J. Comput. Phys.*, 367:109–120.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions – application to computational electromagnetic dosimetry. *Int. J. Uncertainty Quantification*, 9(4):351–363.

Bedford, T. and Cooke, R. M. (2002). Vines – a new graphical model for dependent random variables. *Ann. Stat.*, 30(4):1031–1068.

Besse, P. (1991). Approximation spline de l'analyse en composantes principales d'une variable aléatoire hilbertienne. *Annales de la Faculté des sciences de Toulouse: Mathématiques*, 12(3):329–349.

Besse, P. and Ramsay, J. (1986). Principal components analysis of sampled functions. *Psychometrika*, 51(2):285–311.

Betz, W., Papaioannou, I., and Straub, D. (2014). Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion. *Comput. Methods Appl. Mech. Engrg.*, 271:109–129.

Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys*, 230:2345–2367.

Bowman, A. W. and Azzalini, A. (1997). *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, volume 18. OUP Oxford.

Buhmann, M. D. (2000). Radial basis functions. *Acta Numer.*, 9:1–38.

Dai, H., Zheng, Z., and Ma, H. (2019). An explicit method for simulating non-Gaussian and non-stationary stochastic processes by Karhunen–Loève and polynomial chaos expansion. *Mech. Syst. Signal Pr.*, 115:1–13.

Das, S., Ghanem, R., and Finette, S. (2009). Polynomial chaos representation of spatio-temporal random fields from experimental measurements. *J. Comput. Phys.*, 228(23):8726–8751.

Desceliers, C., Ghanem, R., and Soize, C. (2006). Maximum likelihood estimation of stochastic chaos representations from experimental data. *Int. J. Numer. Meth. Engng.*, 66:978–1001.

Doostan, A., Ghanem, R., and Red-Horse, J. (2007). Stochastic model reduction for chaos representations. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40):3951–3966.

Ernst, O., Mugler, A., Starkloff, H.-J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.*, 46(02):317–339.

Ghanem, R. G. and Spanos, P. (1991). *Stochastic finite elements – A spectral approach*. Springer Verlag, New York. (Reedited by Dover Publications, Mineola, 2003).

Goan, E. and Fookes, C. (2020). Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87. Springer.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Adv. Neur. In.*, 27:1–9.

Grigoriu, M. (1993). Simulation of stationary process via a sampling theorem. *J. Sound Vib.*, 166(2):301–313.

Grigoriu, M. (1998). Simulation of stationary non-Gaussian translation processes. *J. Eng. Mech.*, 124(2):121–126.

Grigoriu, M. (2002). *Stochastic Calculus: Applications in Science and Engineering*. Springer Science+Business Media.

Grigoriu, M. (2006). Evaluation of Karhunen-Loève, spectral and sampling representations for stochastic processes. *J. Eng. Mech.*, 132(2):179–189.

Grigoriu, M. (2010). Probabilistic models for stochastic elliptic partial differential equations. *J. Comput. Phys.*, 229(22):8406–8429.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *J. Am. Stat. Assoc.*, 99(468):1015–1026.

Harper, W. V. and Gupta, S. K. (1983). Sensitivity/uncertainty analysis of a borehole scenario comparing Latin hypercube sampling and deterministic sensitivity approaches. Technical Report No. BMI/ONWI-516, Battelle Memorial Inst. - Office of Nuclear Waste Isolation, Columbus, OH (USA).

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.*, 6(2):327–343.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Ann. Math. Stat.*, 19:293–325.

Iemma, U., Diez, M., and Morino, L. (2006). An extended Karhunen-Loève decomposition for modal identification of inhomogeneous structures. *J. Vib. Acoust.*, 128(3):357–365.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliab. Eng. Syst. Saf.*, 94:1194–1204.

Joe, H. (2014). *Dependence Modeling with Copulas*. Chapman and Hall/CRC, 1 edition.

Karhunen, K. (1946). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fenn. A1*, A1(37).

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Conference proceedings: papers accepted to the International Conference on Learning Representations (ICLR) 2014*. Amsterdam Machine Learning lab (IVI, FNWI).

Kurowicka, D. and Cooke, R. (2005). Distribution-free continuous Bayesian belief nets. In *Modern Statistical and Mathematical Methods in Reliability*, pages 309–322. World Scientific Publishing.

Lataniotis, C., Torre, E., Marelli, S., and Sudret, B. (2021). UQLab user manual – The Input module. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-102.

Loève, M. (1978). *Probability theory – Graduate texts in mathematics*, volume 2. Springer Verlag, New-York, 4th edition.

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif.*, 9(2):593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022a). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification*, 12(3):49–74.

Lüthen, N., Roustant, O., Gamboa, F., Iooss, B., Marelli, S., and Sudret, B. (2022b). Global sensitivity analysis using derivative-based sparse Poincaré chaos expansions. *Submitted*.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pages 2554–2563.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput.*, 22:833–847.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Math. Model. Num.*, 48:116–155.

Nagel, J., Rieckermann, J., and Sudret, B. (2020). Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: application to urban drainage simulation. *Reliab. Eng. Sys. Safety*, 195(#106737).

Navarro Jimenez, M., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):378–402.

Nelsen, R. B. (2006). *An introduction to copulas*, volume 139 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 2nd edition.

Pearce, M., Poloczek, M., and Branke, J. (2022). Bayesian optimization allowing for common random numbers. *Oper. Res.*, 0(0):1–16.

Perrin, T., Roustant, O., Rohmer, J., Alata, O., Naulin, J., Idier, D., Pedreros, R., Moncoulon, D., and Tinard, P. (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliab. Eng. Syst. Saf.*, 211:107522.

Poirion, F. and Zentner, I. (2013). Non-Gaussian non-stationary models for natural hazard modeling. *Appl. Math. Model.*, 37(8):5938–5950.

Poirion, F. and Zentner, I. (2014). Stochastic model construction of observed random phenomena. *Prob. Eng. Mech.*, 36:63–71.

Rahman, S. (2020). A spline chaos expansion. *SIAM/ASA J. Uncertain. Quantif.*, 8(1):27–57.

Raisee, M., Kumar, D., and Lacor, C. (2015). A non-intrusive model reduction approach for polynomial chaos expansion using proper orthogonal decomposition. *Int. J. Num. Meth. Eng.*, 103:293–312.

Ramsay, J. and Silverman, B. (2005). *Functional data analysis*. Springer Series in Statistics. Springer Science and Business Media, 2nd edition.

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.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Stat. Sci.*, 4:409–435.

Sakamoto, S. and Ghanem, R. (2002). Simulation of multi-dimensional non-Gaussian non-stationary random fields. *Prob. Eng. Mech.*, 17(2):167–176.

Schwab, C. and Todor, R. (2006). Karhunen–Loève approximation of random fields by generalized fast multipole methods. *J. Comput. Phys.*, 217(1):100–122.

Shields, M., Deodatis, G., and Bocchini, P. (2011). A simple and efficient methodology to approximate a general non-Gaussian stationary stochastic process by a translation process. *Prob. Eng. Mech.*, 26(4):511–519.

Shinozuka, M. and Deodatis, G. (1991). Simulation of stochastic processes by spectral representation. *Appl. Mech. Rev.*, 3(4).

Simonoff, J. (1996). *Smoothing methods in statistics*. Springer Series in Statistics. Springer-Verlag New York.

Sklar, A. (1959). Fonctions de répartition à $n$ dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris 8*, 8(1):11.

Smith, R. C. (2014). *Uncertainty Quantification: Theory, Implementation and Applications*. SIAM Computational Science and Engineering.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Stat. Comput.*, 14:199–222.

Sobol, I. and Gresham, A. (1995). On an alternative global sensitivity estimator. In *Proceedings of SAMO 1995*, pages 40–42, Belgirate.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Modeling & Comp. Exp.*, 1(4):407–414.

Torossian, L., Picheny, V., Faivre, R., and Garivier, A. (2020). A review on quantile regression for stochastic computer experiments. *Reliab. Eng. Syst. Saf.*, 201:106858.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019a). Data-driven polynomial chaos expansion for machine learning regression. *J. Comput. Phys*, 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. *Prob. Eng. Mech.*, 55:1–16.

Torre, E., Marelli, S., and Sudret, B. (2021). UQLab user manual – Statistical inference. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-114.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Villani, C. (2009). *Optimal transport: old and new*, volume 338. Springer.

Wand, M. and Jones, M. C. (1995). *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, Boca Raton, London New York.

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 J. Sci. Comput.*, 24(2):619–644.

Yamazaki, F. and Shinozuka, M. (1988). Digital generation of non-Gaussian stochastic fields. *J. Eng. Mech.*, 114(7):1183–1197.

Zhang, J. and Ellingwood, B. (1994). Orthogonal series expansions of random fields in reliability analysis. *J. Eng. Mech.*, 120(12):2660–2677.

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *Int. J. Uncertainty Quantification*, 10(3):249–275.

Zhu, X. and Sudret, B. (2021a). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA J. Unc. Quant.*, 9(4):1345–1380.

Zhu, X. and Sudret, B. (2021b). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliab. Eng. Sys. Safety*, 214:107815.

Zhu, X. and Sudret, B. (2022). Stochastic polynomial chaos expansions to emulate stochastic simulators. *Int. J. Uncertainty Quantification*. Accepted.

# Appendix

## 6.A Analytical derivations for extended KLE on PCE trajectories

The following is a detailed exposition of the analytical computations for extended KLE using the PCE basis in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$. A less detailed derivation for $L^2(\mathcal{D})$ can be found in Ramsay and Silverman (2005, Section 8.4.2).

We show in Section 6.A.1 that if the trajectories $\boldsymbol{x} \mapsto \mathcal{M}(\boldsymbol{x}, \omega)$ are represented by PCE, and extended KLE is applied, then the sample covariance function is a polynomial, the integral eigenvalue problem reduces to PCA in the expansion coefficients, and the eigenfunctions are polynomials. In Section 6.A.2, we show that the realizations of the random KLE coefficients can be determined analytically.

Let for $r = 1, \ldots, R$

$$\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{6.30}$$

be the centered PCE trajectory computed from discrete evaluations of the trajectory $\mathcal{T}_r$ (Eq. (6.13)). The sample covariance function is a polynomial given by

$$\hat{c}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{R-1} \sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}'). \tag{6.31}$$

### 6.A.1 Analytical solution of the extended KLE eigenvalue problem

Computing an extended KLE in the function space $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ corresponds to solving the following eigenproblem:

$$\langle \hat{c}(\boldsymbol{x}, \cdot), \phi_i(\cdot) \rangle_{L^2_{f_{\boldsymbol{X}}}(\mathcal{D})} = \int_{\mathcal{D}} \hat{c}(\boldsymbol{x}, \boldsymbol{x}') \phi_i(\boldsymbol{x}') f_{\boldsymbol{X}}(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' = \lambda_i \phi_i(\boldsymbol{x}). \tag{6.32}$$

Since $\hat{c}$ is polynomial, also the eigenfunctions will be polynomials and can be represented in terms of the PCE basis as follows:

$$\phi_i(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} b_{\boldsymbol{\alpha}}^{(i)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}). \tag{6.33}$$

Solving Eq. (6.32) reduces to finding $(\lambda_i, (b_{\boldsymbol{\alpha}}^{(i)})_{\boldsymbol{\alpha}\in\mathcal{A}})$ for $i = 1, \ldots, M$.

Dropping the $i$-subscript of the eigenfunction for convenience, we compute

$$\begin{aligned}
\int_{\mathcal{D}} \hat{c}(\boldsymbol{x}, \boldsymbol{x}')\phi(\boldsymbol{x}')f_{\boldsymbol{X}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' &= \int_{\mathcal{D}} \frac{1}{R-1}\sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x})\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}')\phi(\boldsymbol{x}')f_{\boldsymbol{X}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \\
&= \frac{1}{R-1}\sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \int_{\mathcal{D}} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}')\phi(\boldsymbol{x}')f_{\boldsymbol{X}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \\
&= \frac{1}{R-1}\sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \int_{\mathcal{D}} \left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}')\right)\left(\sum_{\boldsymbol{\beta}\in\mathcal{A}} b_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(\boldsymbol{x}')\right) f_{\boldsymbol{X}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \\
&= \frac{1}{R-1}\sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r b_{\boldsymbol{\alpha}}\right) \qquad \text{(orthonormality of PCE basis)} \\
&= \frac{1}{R-1}\sum_{r=1}^{R} \left(\sum_{\boldsymbol{\beta}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\beta}}^r \psi_{\boldsymbol{\beta}}(\boldsymbol{x})\right)\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r b_{\boldsymbol{\alpha}}\right) \\
&= \sum_{\boldsymbol{\beta}\in\mathcal{A}} \left(\frac{1}{R-1}\sum_{r=1}^{R} \tilde{a}_{\boldsymbol{\beta}}^r \left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r b_{\boldsymbol{\alpha}}\right)\right) \psi_{\boldsymbol{\beta}}(\boldsymbol{x}).
\end{aligned}$$

The eigenvalue problem reduces to

$$\sum_{\boldsymbol{\beta}\in\mathcal{A}} \left(\frac{1}{R-1}\sum_{r=1}^{R} \tilde{a}_{\boldsymbol{\beta}}^r \underbrace{\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r b_{\boldsymbol{\alpha}}\right)}_{=(\tilde{\boldsymbol{a}}^r)^T\boldsymbol{b}\,\in\,\mathbb{R}}\right) \psi_{\boldsymbol{\beta}}(x) \overset{!}{=} \lambda \sum_{\boldsymbol{\beta}\in\mathcal{A}} b_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(x).$$

Because the PCE basis functions $\psi_{\boldsymbol{\beta}}$ are of different orders, we can rewrite this into matrix form:

$$\frac{1}{R-1}\sum_{r=1}^{R} \begin{pmatrix} \tilde{a}_{\boldsymbol{\beta}_1}^r (\tilde{\boldsymbol{a}}^r)^T \\ \tilde{a}_{\boldsymbol{\beta}_2}^r (\tilde{\boldsymbol{a}}^r)^T \\ \vdots \\ \tilde{a}_{\boldsymbol{\beta}_P}^r (\tilde{\boldsymbol{a}}^r)^T \end{pmatrix} \boldsymbol{b} = \left(\frac{1}{R-1}\sum_{r=1}^{R} \tilde{\boldsymbol{a}}^r\ (\tilde{\boldsymbol{a}}^r)^T\right)\boldsymbol{b} = \left(\frac{1}{R-1}\ \tilde{\boldsymbol{a}}\ \tilde{\boldsymbol{a}}^T\right)\boldsymbol{b} \overset{!}{=} \lambda\boldsymbol{b}, \tag{6.34}$$

where $\tilde{\boldsymbol{a}}$ is the $P \times R$-matrix of active and centered PCE trajectory coefficients. Defining the matrix $\boldsymbol{\Sigma} = \frac{1}{R-1}\ \tilde{\boldsymbol{a}}\ \tilde{\boldsymbol{a}}^T$, which is the empirical covariance matrix of the centered PCE coefficients, we see that Eq. (6.34) is nothing else than principal component analysis (PCA) on the coefficients.

Note that it was necessary to apply KLE in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ to arrive at this equation, since the PCE basis is orthonormal in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ but in general not in $L^2(\mathcal{D})$.

The solution vectors $\boldsymbol{b}^{(i)}$ to the eigenvalue problem $\boldsymbol{\Sigma}\boldsymbol{b}^{(i)} = \lambda_i\boldsymbol{b}^{(i)}$ are the PCE coefficients of the KLE eigenfunctions: $\phi_i(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} b_{\boldsymbol{\alpha}}^{(i)}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$. Since the PCE basis is orthonormal, and assuming that the eigenvectors $\boldsymbol{b}^{(i)}$ are normalized to unit norm, it follows that the eigenfunctions $\{\phi_i\}$ are orthonormal.

### 6.A.2 Analytical computation of the realizations of the KL-RV

Let $\lambda_i$ be an eigenvalue and

$$\phi_i(x) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} b_{\boldsymbol{\alpha}}^{(i)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{6.35}$$

the associated eigenfunction expressed in the PCE basis. The projection of the PCE trajectories onto the KLE basis is given by

$$\begin{aligned} \xi_i^r &= \frac{1}{\sqrt{\lambda_i}} \int_{\mathcal{D}} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \phi_i(x) f_{\boldsymbol{X}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{\sqrt{\lambda_i}} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}}^r b_{\boldsymbol{\alpha}}^{(i)} = \frac{1}{\sqrt{\lambda_i}} (\tilde{\boldsymbol{a}}^r)^T \boldsymbol{b}^{(i)} \in \mathbb{R}. \end{aligned}$$

Let $\tilde{\boldsymbol{a}} \in \mathbb{R}^{P \times R}$ the matrix of coefficients of centered PCE trajectories and $\boldsymbol{b} \in \mathbb{R}^{P \times M}$ the matrix of PCE coefficients of the KLE functions. Then we can compute the matrix $\boldsymbol{\Xi} \in \mathbb{R}^{M \times R}$ of KLE coefficient realizations by

$$\boldsymbol{\Xi} = \left( \mathrm{diag}\left( \frac{1}{\sqrt{\lambda_1}}, \ldots, \frac{1}{\sqrt{\lambda_M}} \right) \boldsymbol{b} \right)^T \tilde{\boldsymbol{a}}. \tag{6.36}$$

CHAPTER 7

# Discussion and conclusions

We live in exciting times. Computers can execute billions of calculation steps per second and thus dramatically extend our human thinking capabilities (with respect to anything that can be expressed by numbers, at least). When sitting at a laptop or desktop computer, or even holding a smartphone, this power is literally at our fingertips. The challenge is now to do something meaningful with this power: to contribute to the better understanding or even to the solution of real problems, while at the same time doing so efficiently with regard to time and energy. There is no doubt that carefully prepared computer simulations can help address this challenge, advance science and guide decisions – as long as the ever-present uncertainties of our real world are not neglected in the process.

Dealing with these uncertainties is precisely the task of the field of uncertainty quantification (UQ), which provides methods for investigating the influence of stochasticity on the model in question. In order to be useful, UQ methods must be efficient, well-tested, and accessible to practitioners, i.e., creators of computational models that might themselves not be experts in UQ.

This thesis made contributions towards these goals. Its overarching topic, surrogate modeling, is a general methodology for improving the efficiency of UQ techniques by approximating the original simulator with a less expensive replacement model, based on well-chosen model evaluations and regularity assumptions. In particular, we investigated the popular sparse polynomial chaos expansions (PCE) method. To make the large literature on sparse PCE more accessible, we performed an extensive literature survey and benchmark of various methods, which resulted in problem-dependent recommendations. We also investigated two new surrogate models. The first, Poincaré chaos expansions, is a spectral expansion technique in the spirit of sparse PCE, but using a non-polynomial basis that is especially well suited for sensitivity analysis when partial derivatives of the model are available. The second method is a surrogate for stochastic simulators, i.e., computational models with latent stochasticity. Taking the random function view, we use trajectory samples to approximate the spatial as well as the stochastic behavior of the model.

We now discuss the conclusions and outlooks associated to each of these topics. These discussions are not meant to merely repeat the conclusions of the respective papers, but to summarize the results and to point out connections with related work from a more global perspective. In particular, we will discuss the nature and role of sparse expansions, and share considerations regarding benchmark design.

## 7.1 Sparse polynomial chaos expansions

The literature on sparse regression-based PCE, a spectral surrogate modeling method making use of orthonormal polynomials and compressed sensing ideas, has grown tremendously in the recent years. To provide an overview of the various available approaches and to benchmark their performance with respect to the global mean-square error between surrogate and true model, we conducted an extensive literature review and a thorough benchmark, resulting in the two publications presented in Chapters 3 and 4.

### 7.1.1 Conclusions

**Literature review and benchmark.** In Chapter 3, we identified a framework for the sparse PCE computation procedure consisting of the following components (see Figure 3.1 for an illustration): construction of the initial basis, construction of the initial experimental design, solution of the sparse regression problem (i.e., computation of the coefficients), evaluation of an error measure or model selection criterion, adaptation of the basis, and sequential enrichment of the experimental design. A large part of the literature on sparse PCE targets one or several of these components. Furthermore, there are a number of methods outside this framework that are meant to improve the sparse PCE computation procedure, e.g., by pre-conditioning the regression problem, or by identifying a more suitable coordinate system in the input domain.

Because the number of components and associated proposed methods is large, and the number of possible combinations even larger, we had to restrict our benchmark to a few components and to a selection of most promising methods. In Chapter 3, we investigated the impact of the initial experimental design and of the sparse regression solver on the global approximation quality of the final result. For the experimental design, we investigated the schemes MC, LHS, coherence-optimal sampling, D-optimal sampling, and near-optimal sampling. For the solvers, we considered least-angle regression (LARS), orthogonal matching pursuit (OMP), subspace pursuit (SP), Bayesian compressive sensing (BCS), and spectral projected gradient-$\ell^1$ (SPGL1). We tested 11 benchmark models of various dimensionality and complexity repeatedly for several sizes of the experimental design, while using a fixed basis.

We found that the performance of the methods varied considerably across the models and over the repetitions. To aggregate the results from the various models, experimental design sizes, and repetitions, we divided the experiments into four classes according to two criteria: small/large size of experimental design, and low/high dimensionality of the model. For each combination of sparse regression solver and sampling scheme, we then counted the fraction of experiments that resulted in an optimal or close-to-optimal error, and visualized this ranking.

Based on this ranking, we were able to identify solvers and sampling schemes that consistently showed better or worse performance. Our conclusions are explained in detail in Section 3.4. To highlight a few results: we found that the solvers OMP, LARS, and SPGL1 performed comparatively poorly, whereas BCS and SP based on leave-one-out cross-validation performed rather well. Regarding the sampling schemes, LHS was the best choice for high-dimensional models. For low-dimensional models, coherence-optimal sampling performed well for large design sizes, and the near-optimal scheme performed best for problems with a rather small basis.

**Basis adaptivity and automatic selection.** We continued the benchmark in Chapter 4, where we investigated the component of basis adaptivity. This introduces an iterative aspect into the PCE computation procedure: after computing an expansion, new regressors are chosen and added to the basis. Then, the expansion is recomputed and the procedure is repeated until convergence. We investigated three methods from the literature, namely degree- and q-norm, forward neighbor, and anisotropic degree basis adaptivity. The latter two choose the new regressors based on the previously computed solution. We used LHS for the experimental design, and paired the basis-adaptive schemes with the solvers LARS, OMP, SP, SPGL1, and BCS. For the evaluation of the results, we conducted the same aggregation as previously described, and focused on the relative performance, measured in the percentage of runs in which the combination attained an error within a certain factor of the best possible error on the same experimental design achieved by any other combination.

We found that the three basis-adaptive methods we tested generally produced a more accurate solution than a static (i.e., fixed) basis. This was especially pronounced for low-dimensional models, where in a large number of cases, basis adaptivity achieved an error that was more than one order of magnitude smaller than the one of a static basis. In contrast, in the case of high-dimensional models with a small experimental design (i.e., the case of very scarce data), the static basis produced more robust solutions than the adaptive basis schemes.

Among the tested solvers, SP and BCS performed well, whereas SPGL1 and OMP did not. This is consistent with the performance of these solvers in Chapter 3. Aggregated over all experiments, forward-neighbor basis adaptivity together with leave-one-out-based SP showed good performance most often. However, even this combination was far from the smallest possible error in a rather large percentage of cases.

Therefore, we implemented the automatic selection of one out of several sparse solutions generated by different methods based on cross-validation, which was able to achieve close-to-optimal errors, and outperformed any single recommendation of solver and basis adaptivity scheme. This result is another example for the good performance of cross-validation for model selection, which has already been utilized successfully e.g. for determining important regressors in sparse PCE (Liu et al., 2020b) and for ensemble modeling (Viana et al., 2009; Parisi et al., 2021).

**The nature of sparse expansions.** The basis adaptivity scheme "forward neighbor" provides insight into the sparsity pattern of sparse PCE. This scheme views the set of regressors as a directed graph; see Section 4.2.3.2 for a detailed explanation. This scheme would not be successful if the target sparse expansion contained isolated regressors whose backward neighbors

all have negligible coefficients. This suggests that expansions of real-world models possess a specific sparsity pattern: building on the exposition of forward neighbor basis adaptivity by Jakeman et al. (2015) who state that "the magnitude of the ancestors of a PCE coefficient is a reasonable indicator of the size of the child coefficient", we hypothesize that sparse expansions are actually dense with respect to the important input variables.[1] The expansions are sparse only within a larger basis of candidate regressors, such as a total-order basis: unimportant input variables and low interaction between variables cause a substantial part of this candidate basis to have negligible coefficients.

In the classification of models for global sensitivity analysis by Kucherenko et al. (2011), this corresponds to function type A, i.e., functions of small effective dimension, of which they say that "Type A functions are probably the most common type of functions encountered in practice".

Following this reasoning, the type of basis is not necessarily essential for achieving a sparse expansion: any tensor-product basis that includes the constant regressor in each univariate basis would similarly allow sparse expansions. Of course, the type of basis has an influence on compressibility, i.e., on the decay in coefficient magnitude. We will talk more about this topic in Section 7.2 in the context of Poincaré chaos expansions.

Furthermore, this has implications for the design of benchmark cases to test sparse PCE algorithms. It is not uncommon for methodological papers to use artificial test cases, such as the "manufactured sparse PCE" model constructed as the weighted sum of a few randomly selected polynomials, for a first test of the algorithm in question. However, for the result to be indicative of the real-world performance of the tested method, such models should follow the sparsity pattern described above.

**The role of sparsity.** We want to discuss the role of sparsity in sparse PCE using the example of BCS. In our benchmark, this solver performed especially well for high-dimensional models or for small experimental designs, in other words: whenever information is scarce. However, for low-dimensional models and a large design, it did not perform well. BCS uses a hierarchical Bayesian formulation of the regression problem, in which the parameters of the prior distribution of the coefficients are again random variables, resulting in a sparsity-encouraging effective prior for the coefficients. Empirically, we often found that BCS generates sparser solutions than other solvers, which fits the conclusions of Babacan et al. (2010) comparing BCS with another Bayesian method.

However, as opposed to compressed sensing, in UQ we are usually not interested in the sparsest possible solution. Instead, sparsity is a device to achieve reasonably accurate approximations under limited data. Sparsity avoids overfitting by only including the most relevant regressors into the model, thus acting as a regularizer. Viewed from another angle, sparsity is a tool for computational efficiency, allowing the solution of underdetermined systems of linear equations. However, real computational models are almost never sparse in the PCE basis. Whenever there is enough information to make it feasible (such as in the case of low-dimensional models with a

[1] To give an example: if in a three-dimensional model the regressor with multi-index $(3, 7, 0)$ has a non-negligible coefficient, then typically the same is true for all the smaller ones: $\{(i, j, 0) : i \leq 3, j \leq 7\}$. In the literature on least-squares approximation, such a set is also called *downward closed*.

large experimental design), a denser model will in general give a better approximation than a sparse one, which might explain the worse relative performance of BCS in this case.

**Benchmark design.** In an active methodological research field such as surrogate modeling, and in particular sparse PCE, it is important to perform comparative studies in order to identify the most promising methods for further research, to provide an incentive to study the reasons for better or worse performance of certain methods, and to create recommendations for practitioners looking for advice which method to use. Regarding the implementation of the benchmark, we found that it is important to clearly define the scope of the study, to include a sufficiently large number of test models, to use repetitions for the (stochastic) sampling of the experimental design, and to appropriately aggregate the results in order to see the big picture. For the latter, we developed a novel way to accumulate and visualize the relative performance of different methods using stacked bar plots. We concluded that the most relevant information to display is the fraction of experiments in which the respective method achieves a global validation error that is within a certain factor of the error of the best-performing method.

**Literature reviews.** Literature reviews can help make an established, specialized field of research more accessible to new researchers looking for the main directions and open questions. They can also help discover connections between topics and retrace the evolution of certain ideas. We hope that with our two review and benchmark papers, we were able to provide such guidance and overview for sparse regression-based PCE at least for the current point in time.

### 7.1.2 Outlook

The literature review and benchmarks could be extended in several ways. In our benchmarks, we investigated the effect of the initial experimental design, the sparse solver, and the basis adaptivity scheme. Further work could target the other components of the sparse PCE computation procedure, e.g., sequential enrichment of the experimental design, pre-conditioning, or the identification of potentially more suitable coordinates for the expansion.

Furthermore, for the components we tested, we were not able to consider all available methods. For example, in the solver benchmark we did not include the class of heuristic stepwise regression algorithms, which usually show superior performance in the numerical tests of the respective authors. Among the sampling schemes, techniques such as Leja sequences and Christoffel sparse approximation appear promising. At the same time, new methods are continuously being developed. Preferably, the benchmark should be continued on the same (or even on an extended) suite of models to allow for the fair comparison of methods.

Going one step further, the ideal solution would be a free public benchmarking service similar to the "RPrepo" platform, which Rozsas and Slobbe (2019) developed for black-box reliability algorithms. This service collects benchmark problems of various complexity and provides a server that allows the remote testing of new algorithms under exactly equal conditions.

Similar to a benchmark, a literature review becomes outdated as new methods continue to be published. A suitable format for keeping it up to date would be a live online document that is

regularly updated to collect the most relevant literature.

We have mentioned the difficulty of selecting a representative set of benchmark models, and the associated difficulty of inferring recommendations on which methods to use depending on the model characteristics and the available experimental design. For his benchmark of Sobol' sensitivity analysis methods, Becker (2020) has found an elegant solution called *metafunctions*: test cases are randomly constructed from a set of univariate basis functions, using specific distributions for the number of dimensions, the coefficient values, and the number of interaction terms. The resulting test functions are representative of real-world models with respect to the Pareto principle and the sparsity-of-effects principle (see also Section 2.5.3). While artificial, this principled way of constructing test cases allows detailed insight into the interplay between model properties and method performance.

Among the methods we tested, the solver BCS appeared to be especially promising. Apart from its good performance in the small-data regime, it is based on Bayesian inference, which has two beneficial consequences that are worth studying further: first, through the specification of the prior distributions it is possible to encode prior information about the magnitude of coefficients. This has already been utilized by Shao et al. (2017) in a stepwise regression algorithm. Second, the posterior distribution of the coefficients contains information about the confidence in the resulting estimates given the data, which can be used for active learning (Zhou et al., 2019; Pan et al., 2020).

Turning to model selection, with our automatic selection scheme choosing between several expansions generated by various sparse PCE algorithms, we have provided more evidence that cross-validation performs well for this purpose. While there is the danger of overfitting if used excessively (Cawley and Talbot, 2010), model selection by cross-validation, and more generally ensemble learning, are powerful concepts with great potential for improving surrogate modeling.

Among the sampling methods, coherence-optimal sampling showed good performance for low-dimensional models. However, so far this sampling scheme has only been analyzed and implemented for the uniform and the Gaussian distribution (Hampton and Doostan, 2015b). It is not yet known how to efficiently construct such a sample for other types of distributions or for dependent inputs.

More generally, many of the available sampling schemes are based either on optimizing the space-filling quality of the resulting sample (e.g., maximin LHS), or on optimizing a property of the regression matrix (e.g., D-optimal, coherence-optimal, or near-optimal sampling). While the authors proposing the latter schemes for PCE (Diaz et al., 2018; Hampton and Doostan, 2015b; Alemazkoor and Meidani, 2018a) demonstrate superior performance for these algorithms, Weise et al. (2022) find in their recent benchmark that for the considered test functions, LHS (with some adaptations improving the space-filling property) outperforms various optimal sampling schemes, including coherence- and D-optimal sampling. Therefore, it would be of special interest to further investigate in a principled way the impact of matrix properties on the approximation quality of the resulting PCE. Answering this question is particularly challenging because the result also crucially depends on the benchmark model and its input distribution.

Finally, we have listed several insights into the nature of sparse expansions. This information might be utilized to develop new algorithms.

## 7.2 Poincaré chaos expansions

In Chapter 5, we investigated Poincaré chaos expansions (PoinCE), a new type of spectral expansion especially suited for global sensitivity analysis. Similar to sparse PCE, we construct a tensor-product basis and compute the coefficients by sparse regression. The Poincaré basis is in general not polynomial (with the exception of the basis associated to the Gaussian distribution, which coincides with the Hermite polynomials). Poincaré functions are similar to polynomials in that they always include the constant function, and that the $n$-th order function has $n$ roots. They arise as eigenfunctions to a differential eigenvalue problem that was originally derived for computing Poincaré constants for univariate measures with bounded support.

### 7.2.1 Conclusions

While the Poincaré basis functions associated to the Gaussian distribution are the well-known Hermite polynomials, the basis associated to uniform input consists of cosine functions. Hence, PoinCE constitutes a connecting link between the two well-known spectral methods PCE and Fourier expansion with the potential of generalizing results for each of them.

Poincaré basis functions are orthonormal and have the special property that the partial derivatives of the basis form again an orthonormal basis with respect to the same measure. In Chapter 5, we were able to show that this is in fact a characterizing property, i.e., every basis with this property must be a Poincaré basis.

As a consequence, we derived an analytical formula for the computation of DGSM, which extends a formula by Sudret and Mai (2015) to the non-Gaussian case and provides insight into the inequality bound between total Sobol' indices and DGSM: namely, this bound is tight if the expansion does not include relevant terms of order larger than one in the respective variable. Otherwise, the gap can be significant, which we also observed in the numerical examples.

The characterizing property of orthogonal derivatives also provides a convenient way to utilize model derivative evaluations for the computation of expansion coefficients. We implemented an algorithm that combines the partial derivative evaluations with respect to the different input variables to compute accurate estimates of Sobol' indices. We found that this procedure generally resulted in more accurate and less biased estimates than the ones computed by PCE or by Poincaré chaos expansions (PoinCE) based on model evaluations alone.

When taking partial derivatives of a multivariate expansion with respect to a certain input variable, all regressors that do not involve this variable will vanish. This reduces the number of unknown coefficients particularly for high-dimensional problems and rather low-dimensional expansions, and might be one possible explanation for the good performance of derivative-based PoinCE.

We also compared the performance of PCE, PoinCE, and derivative-based PoinCE as a global surrogate model, based on the $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$-error. Surprisingly, we found that here PCE outperforms the other two methods, particularly for larger experimental design sizes. This difference in performance between sensitivity analysis and surrogate modeling requires more investigation.

Furthermore, we found that the two expansions PCE and PoinCE have almost the same number of nonzero coefficients for a given experimental design size. This supports the hypothesis mentioned in Section 7.1 that expansions are sparse because of unimportant input variables and low interaction order, not because of true sparsity in the basis. However, given that the same level of sparsity in PCE and PoinCE leads to a different quality of global approximation, we speculate that polynomial functions might be better suited than Poincaré chaos expansions to approximate the model of interest with a low-degree expansion. In other words, it seems that the model is less compressible in the Poincaré basis than in the PCE basis.

### 7.2.2 Outlook

While PoinCE is unlikely to replace PCE in the near future, it is an interesting example of a non-polynomial orthonormal basis that is relatively easy to construct. The direct comparison allows us to obtain insights also about PCE, e.g., regarding compressibility.

An open question about PoinCE is the performance gap between sensitivity analysis, where it performs well, and surrogate modeling, where it is outperformed by PCE. This should be investigated on a larger number of benchmark models. Answering this question might provide further insight into what is important for accurately estimating Sobol' sensitivity indices.

Furthermore, in our computations we have used either model evaluations or model partial derivatives to compute the PoinCE coefficients. We are working on combining the information of both sources in the spirit of derivative-enhanced PCE (Jakeman et al., 2015; Peng et al., 2016).

In general, the existence of the Poincaré basis with its favorable orthogonality property raises the question whether we could construct other non-polynomial orthonormal bases with beneficial properties for certain applications.

## 7.3 Spectral surrogate for stochastic simulators

In Chapter 6 we proposed a surrogate model for stochastic simulators based on the random function view of stochastic simulators. Fixing the seed, the response of a stochastic simulator becomes a deterministic function, which we surrogate using sparse PCE. Then, we perform Karhunen-Loève expansion (KLE) on the resulting trajectories and characterize the stochastic behavior by statistical inference of the random coefficients of the KLE (KL-RV).

### 7.3.1 Conclusions

The method approximates the full stochastic simulator including mean function, covariance function, marginal distributions, and trajectories. It does not assume that the stochastic simulator is a Gaussian random field. Our numerical examples show that non-Gaussian distributions for the KL-RV outperform Gaussian ones.

KLE requires the solution of an integral eigenvalue equation involving the covariance function, which is generally expensive to solve. In our case, several factors contribute to circumventing

this problem: we perform extended KLE in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ and utilize an orthonormal polynomial basis to approximate the trajectories. Here, sparsity is key for obtaining accurate approximations at low cost. The covariance function is computed from the sample trajectories. All of this results in the integral eigenvalue equation becoming a cheap-to-solve finite-dimensional matrix eigenvalue problem.

In our numerical examples, we observe that $M = 1$ term in the KL expansion is often sufficient to emulate a large part of the variance. We explain this phenomenon using the Hoeffding-Sobol' decomposition of the model considering both its explicit and latent parameters – assuming it exists – and find that one mode is sufficient if there are no interaction terms between the explicit and the latent group.

In general, by identifying the number of modes through the explained variance, our method can discover the effective dimensionality of the latent stochastic space.

### 7.3.2 Outlook

The random function view of stochastic simulators shares similarities with the "common random numbers" approach used in optimization (Pearce et al., 2022). This connection should be explored further; possibly, our stochastic emulator could be applied for optimization. Furthermore, the precise connection of our approach to kernel PCA (Sarma et al., 2008) should be investigated.

Navarro Jimenez et al. (2017) and Zhu and Sudret (2021c) have discussed Sobol' sensitivity analysis for stochastic simulators. This methodology can easily be applied to our simulator, and can even be computed analytically from the PCE trajectories.

While KLE works especially well for Gaussian random fields, our approach is not restricted to the latter, because our methodology characterizes the dependent distribution of the random coefficients of KLE. We use the copula framework to infer the dependence, which could benefit from the introduction of additional copula families that better describe the type of dependence found in KL-RV of stochastic simulators.

Finally, the observation that only one mode covers a large amount of the explained variance is worth exploring further. This phenomenon should be tested and analyzed for more complex and realistic engineering stochastic simulators.

## 7.4 Final conclusion

We have investigated several types of sparse spectral surrogate models in this thesis, with the goals of testing their performance in practice, giving recommendations, and understanding their behavior in more detail. Spectral methods, which utilize an orthonormal basis of $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$ for approximation, are global methods adapted to the distribution of the input random vector. The orthogonality of the basis is an important property for computational reasons (see e.g. Eq. (2.12) and the concept of isotropy in Section 2.5.2).

We focused on sparse expansions computed by sparse regression. Here, sparsity is a tool for computational efficiency, allowing the accurate calculation of coefficients even for rather small experimental design sizes, which is the data regime typically encountered in UQ. With our literature review and benchmarks of sparse polynomial chaos expansions, and the two methodological contributions of sparse Poincaré chaos expansions for global sensitivity analysis and our new trajectory-based surrogate for stochastic simulators, we have addressed the above-mentioned goals of performance evaluation, recommendation, and improved understanding, as discussed in the preceding sections.

Furthermore, to disseminate the results of our research, we have already made the most promising sparse regression solvers from our benchmark available in the general-purpose UQ software UQLab (Marelli et al., 2021a). We are planning to add further functionality related to basis adaptivity and automatic selection for PCE, as well as a module for the treatment of stochastic simulators.

Finally, we want to comment on the interdisciplinary nature of the field of uncertainty quantification, specifically on its location at the intersection of engineering and applied mathematics. During our research, we have seen various examples for exciting, rigorous results stemming from thorough mathematical investigation. To name a few examples, there is coherence-optimal sampling (Candès and Plan, 2011; Hampton and Doostan, 2015b), the inequality relation between Sobol' indices and DGSM (Lamboni et al., 2013), and the surprising fact that there is no polynomial basis associated to the lognormal distribution (Ernst et al., 2012). Rigorous mathematical analysis has been essential in obtaining these results.

On the other hand, the theoretical performance and proven validity of methods is not always indicative of their performance in practice. For example, the bound on the number $N$ of samples in compressive sensing (Theorem 1 in Section 2.5.2) involves a constant whose value is not known, making the bound useful mainly for large problem sizes $P$. For the case of rather small degree, this theorem does not necessarily guarantee that $N < P$ is sufficient for sparse recovery.

Furthermore, engineering research can be ahead of the associated mathematical results by several decades. For example, Xiu and Karniadakis (2002) proposed to compute PCE associated to distributions from the Wiener-Askey scheme ten years before Ernst et al. (2012) provided the formal proof of convergence. Also, polynomial approximation by least-squares regression (Isukapalli, 1999; Berveiller et al., 2006) has been used long before in the recent years Cohen and Migliorati (2017) and colleagues were able to prove rigorous bounds for its performance.

These small examples illustrate that applied mathematics and engineering research should go hand in hand to further advance the field of uncertainty quantification, finally resulting in methods that are able to efficiently obtain reliable and meaningful solutions to real-world problems from computer simulations.

# Bibliography

(2020). *MATLAB version 9.8.0 (r2020a)*. Natick, Massachusetts.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inform. Fusion*, 76:243–297.

Abraham, S., Tsirikoglou, P., Miranda, J., Lacor, C., Contino, F., and Ghorbaniasl, G. (2018). Spectral representation of stochastic field data using sparse polynomial chaos expansions. *Journal of Computational Physics*, 367:109–120.

Adams, B. M., Bauman, L. E., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., Hough, P. D., Hu, K. T., Jakeman, J. D., Stephens, J. A., Swiler, L. P., Vigil, D. M., , and Wildey, T. M. (2014). *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual*. Sandia National Laboratories. Technical Report SAND2014-4633 (Updated November 2015 (Version 6.3)).

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Alemazkoor, N. and Meidani, H. (2017). Divide and conquer: An incremental sparsity promoting compressive sampling approach for polynomial chaos expansions. *Computer Methods in Applied Mechanics and Engineering*, 318:937–956.

Alemazkoor, N. and Meidani, H. (2018a). A near-optimal sampling strategy for sparse recovery of polynomial chaos expansions. *Journal of Computational Physics*, 371:137–151.

Alemazkoor, N. and Meidani, H. (2018b). A preconditioning approach for improved estimation of sparse polynomial chaos expansions. *Computer Methods in Applied Mechanics and Engineering*, 342:474–489.

Andrianov, G., Burriel, S., Cambier, S., Dutfoy, A., Dutka-Malen, I., de Rocquigny, E., Sudret, B., Benjamin, P., Lebrun, R., Mangeant, F., and Pendola, M. (2007). Open TURNS, an open source initiative to Treat Uncertainties, Risks'N Statistics in a structured industrial approach. In *Proc. ESREL'2007 Safety and Reliability Conference, Stavenger, Norway*.

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Oper. Res.*, 58:371–382.

Arjoune, Y., Kaabouch, N., El Ghazi, H., and Tamtaoui, A. (2017). Compressive sensing: Performance comparison of sparse recovery algorithms. In *2017 IEEE CCWC*, pages 1–7. IEEE.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Stat. Surv.*, 4:40–79.

Arnst, M., Ghanem, R., and Soize, C. (2010). Identification of Bayesian posteriors for coefficients of chaos expansions. *Journal of Computational Physics*, 229(9):3134–3154.

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 Resour. Res.*, 51(8):5957–5973.

Atkinson, K. E. (1996). *The numerical solution of integral equations of the second kind*. Cambridge University Press.

Azzi, S., Huang, Y., Sudret, B., and Wiart, J. (2019). Surrogate modeling of stochastic functions – application to computational electromagnetic dosimetry. *Int. J. Uncertainty Quantification*, 9(4):351–363.

Azzi, S., Sudret, B., and Wiart, J. (2020). Sensitivity analysis for stochastic simulators using differential entropy. *Int. J. Uncertainty Quantification*, 10(1):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.

Babuška, I., Nobile, F., and Tempone, R. (2007). A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Num. Anal.*, 45(3):1005–1034.

Bao, Y., Beck, J. L., and Li, H. (2011). Compressive sampling for accelerometer signals in structural health monitoring. *Struct. Health Monit.*, 10(3):235–246.

Becker, W. (2020). Metafunctions for benchmarking in sensitivity analysis. *Reliability Engineering & System Safety*, 204:107189.

Bénesse, C., Gamboa, F., Loubes, J., and Boissin, T. (2022). Fairness seen as global sensitivity analysis. *Mach. Learn.*, pages 1–28.

Benner, P., Gugercin, S., and Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531.

Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite elements: a non intrusive approach by regression. *Eur. J. Comput. Mech.*, 15(1–3):81–92.

Besse, P. and Ramsay, J. (1986). Principal components analysis of sampled functions. *Psychometrika*, 51(2):285–311.

Betz, W., Papaioannou, I., and Straub, D. (2014). Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 271:109–129.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *J. Comput. Graph. Stat.*, 27(4):808–821.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer-Verlag New York Inc.

Blatman, G. and Sudret, B. (2008). Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *C. R. Mécanique*, 336(6):518–523.

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.

Bochner, S. (1929). Über Sturm-Liouvillesche Polynomsysteme. *Math. Z.*, 29(1):730–736.

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. and Plischke, E. (2016). Sensitivity analysis: A review of recent advances. *Eur. J. Oper. Res.*, 248(3):869–887.

Bourinet, J.-M., Deheeger, F., and Lemaire, M. (2011). Assessing small failure probabilities by combined subset simulation and support vector machines. *Struct. Saf.*, 33(6):343–353.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Belmont, Wadsworth.

Britton, T. (2010). Stochastic epidemic models: a survey. *Math. Biosci.*, 225(1):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. *Qual. Reliab. Eng. Int.*, 32(6):2067–2080.

Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81.

Buhmann, M. D. (2000). Radial basis functions. *Acta Numer.*, 9:1–38.

Candès, E., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Inform. Theory*, 52(2):489–509.

Candès, E. J. and Plan, Y. (2011). A probabilistic and RIPless theory of compressed sensing. *IEEE Transactions on Inform. Theory*, 57(11):7235–7254.

Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Processing Mag.*, 25(2):21–30.

Candès, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted $\ell_1$ minimization. *J. Fourier Anal. Appl.*, 14(5-6):877–905.

Cawley, G. C. and Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107.

Chatterjee, T., Chakraborty, S., and Chowdhury, R. (2019). A critical review of surrogate assisted robust design optimization. *Archives of Computational Methods in Engineering*, 26(1):245–274.

Cheng, B. and Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Stat. Sci.*, 9(1):2–30.

Cheng, K. and Lu, Z. (2020). Structural reliability analysis based on ensemble learning of surrogate models. *Struct. Saf.*, 83:101905.

Cheng, K., Lu, Z., Ling, C., and Zhou, S. (2020). Surrogate-assisted global sensitivity analysis: an overview. *Structural and Multidisciplinary Optimization*, 61(3):1187–1213.

Chevreuil, M., Lebrun, R., Nouy, A., and Rai, P. (2015). A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA J. Uncertain. Quantif.*, 3(1):897–921.

Cohen, A. and Migliorati, G. (2017). Optimal weighted least-squares methods. *SMAI J. Comp. Math.*, 3:181–203.

Conrad, P. R. and Marzouk, Y. M. (2013). Adaptive Smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing*, 35(6):A2643–A2670.

Constantine, P. G., Eldred, M. S., and Phipps, E. T. (2012). Sparse pseudospectral approximation method. *Computer Methods in Applied Mechanics and Engineering*, 229:1–12.

Da Veiga, S. (2015). Global sensitivity analysis with dependence measures. *J. Stat. Comput. Sim.*, 85(7):1283–1305.

Da Veiga, S. (2021). Kernel-based ANOVA decomposition and Shapley effects – Application to global sensitivity analysis. *arXiv preprint arXiv:2101.05487*.

Da Veiga, S., Bénard, C., Chazalviel, D., El Bachir, A., and Sinoquet, D. (2022). Lagun: An open source platform for data exploration and optimization. In *SIAM Conf. on Uncertainty Quantification, Atlanta (USA), April 12-15*. (Talk by D. Sinoquet).

Da Veiga, S., Gamboa, F., Iooss, B., and Prieur, C. (2021). *Basics and Trends in Sensitivity Analysis: Theory and Practice in R*. SIAM.

Da Veiga, S., Wahl, F., and Gamboa, F. (2009). Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463.

Dai, H., Zheng, Z., and Ma, H. (2019). An explicit method for simulating non-Gaussian and non-stationary stochastic processes by Karhunen–Loève and polynomial chaos expansion. *Mechanical Systems and Signal Processing*, 115:1–13.

Dai, W. and Milenkovic, O. (2009). Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Inform. Theory*, 55(5):2230–2249.

Das, S., Ghanem, R., and Finette, S. (2009). Polynomial chaos representation of spatio-temporal random fields from experimental measurements. *Journal of Computational Physics*, 228(23):8726–8751.

Das, S., Ghanem, R., and Spall, J. C. (2008). Asymptotic sampling distribution for polynomial chaos representation from data: A maximum entropy and Fisher information approach. *SIAM Journal on Scientific Computing*, 30(5):2207–2234.

Deodatis, G. and Micaletti, R. (2001). Simulation of highly skewed non-Gaussian stochastic processes. *Journal of Engineering Mechanics (ASCE)*, 127(12):1284–1295.

Deodatis, G. and Shinozuka, M. (1991). Simulation of stochastic processes by spectral representation. *Applied Mech. Rev. (ASME)*, 44(4):191–204.

Der Kiureghian, A. and Ke, J. (1988). The stochastic finite element method in structural reliability. *Probabilistic Engineering Mechanics*, 3(2):83–91.

Desceliers, C., Ghanem, R., and Soize, C. (2006). Maximum likelihood estimation of stochastic chaos representations from experimental data. *International Journal for Numerical Methods in Engineering*, 66:978–1001.

Diaz, P., Doostan, A., and Hampton, J. (2018). Sparse polynomial chaos expansions via compressed sensing and D-optimal design. *Computer Methods in Applied Mechanics and Engineering*, 336:640–666.

Dimitrov, N., Kelly, M. C., Vignaroli, A., and Berg, J. (2018). From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. *Wind Energy Sci.*, 3(2):767–790.

Dimitrov, N. K., Marelli, S., and Schär, S. (2022). Novel surrogate modelling approaches for wind turbine reliability assessment. Technical Report D4.1, HIPERWIND project, ETH Zurich, DTU.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Inform. Theory*, 52(4):1289–1306.

Donoho, D. L. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell^1$ minimization. *P. Natl. Acad. Sci. USA*, 100(5):2197–2202.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Inform. Theory*, 52(1):6–18.

Doostan, A., Ghanem, R., and Red-Horse, J. (2007). Stochastic model reduction for chaos representations. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40):3951–3966.

Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034.

Eckert, C., Beer, M., and Spanos, P. D. (2020). A polynomial chaos method for arbitrary random inputs using B-splines. *Probabilistic Engineering Mechanics*, 60:103051.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Stat.*, 32:407–499.

Efron, B. and Stein, C. (1981). The Jacknife estimate of variance. *Ann. Stat.*, 9(3):586–596.

Eldred, M., Webster, C., and Constantine, P. (2008). Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos. In *49th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, page 1892.

Ernst, O., Mugler, A., Starkloff, H.-J., and Ullmann, E. (2012). On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. Numer. Anal.*, 46(02):317–339.

Faes, M. G. R., Daub, M., Marelli, S., Patelli, E., and Beer, M. (2021). Engineering analysis with probability boxes: A review on computational methods. *Struct. Saf.*, 93:102092.

Fajraoui, N., Marelli, S., and Sudret, B. (2017). Sequential design of experiment for sparse polynomial chaos expansions. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):1061–1085.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*, volume 66 of *Monographs on Statistics and Applied Probability*. Chapman and Hall.

Fel, T., Cadène, R., Chalvidal, M., Cord, M., Vigouroux, D., and Serre, T. (2021). Look at the variance! Efficient black-box explanations with Sobol-based sensitivity analysis. *Adv. Neur. In.*, 34.

Forrester, A. I. J. and Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.*, 45(1-3):50–79.

Freimer, M., Kollia, G., Mudholkar, G. S., and Lin, C. T. (1988). A study of the generalized Tukey lambda family. *Comm. Stat. Theor. Meth.*, 17:3547–3567.

Gamboa, F., Janon, A., Klein, T., and Lagnoux, A. (2014). Sensitivity analysis for multidimensional and functional outputs. *Electron. J. Stat.*, 8(1):575–603.

Gautschi, W. (1982). On generating orthogonal polynomials. *SIAM J. Sci. Stat. Comp.*, 3(3):289–317.

Gautschi, W. (2004). *Orthogonal polynomials: computation and approximation*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, UK.

Gerstner, T. and Griebel, M. (2003). Dimension-adaptive tensor-product quadrature. *Computing*, 71:65–87.

Ghanem, R. and Doostan, A. (2006). On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data. *Journal of Computational Physics*, 217(1):63–81.

Ghanem, R. G. and Spanos, P. (1991). *Stochastic finite elements – A spectral approach*. Springer Verlag, New York. (Reedited by Dover Publications, Mineola, 2003).

Giovanis, D. G. and Shields, M. D. (2018). Uncertainty quantification for complex systems with very high dimensional response using Grassmann manifold variations. *Journal of Computational Physics*, 364:393–415.

Goan, E. and Fookes, C. (2020). Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87. Springer.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Adv. Neur. In.*, 27:1–9.

Grigoriu, M. (1984). Crossings of non-Gaussian translation processes. *Journal of Engineering Mechanics (ASCE)*, 110(4):610–620.

Grigoriu, M. (1998). Simulation of stationary non-Gaussian translation processes. *Journal of Engineering Mechanics (ASCE)*, 124(2):121–126.

Grigoriu, M. (2002). *Stochastic Calculus: Applications in Science and Engineering*. Springer Science+Business Media.

Grigoriu, M. (2006). Evaluation of Karhunen-Loève, spectral and sampling representations for stochastic processes. *Journal of Engineering Mechanics (ASCE)*, 132(2):179–189.

Grigoriu, M. (2010). Probabilistic models for stochastic elliptic partial differential equations. *Journal of Computational Physics*, 229(22):8406–8429.

Gutiérrez, R., Ruiz, J. C., and Valderrama, M. J. (1992). On the numerical expansion of a second order stochastic process. *Appl. Stoch. Model. D. A.*, 8(2):67–77.

Hajela, P. and Berke, L. (1992). Neural networks in structural analysis and design: an overview. *Comput. Syst. Eng.*, 3(1-4):525–538.

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *J. Am. Stat. Assoc.*, 99(468):1015–1026.

Hampton, J. and Doostan, A. (2015a). Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. *Computer Methods in Applied Mechanics and Engineering*, 290:73–97.

Hampton, J. and Doostan, A. (2015b). Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. *Journal of Computational Physics*, 280:363–386.

Hampton, J. and Doostan, A. (2018). Basis adaptive sample efficient polynomial chaos (BASE-PC). *Journal of Computational Physics*, 371:20–49.

Hariri-Ardebili, M. and Sudret, B. (2020). Polynomial chaos expansion for uncertainty quantification of dam engineering problems. *Eng. Struct.*, 203(#109631).

Hart, J. L., Alexanderian, A., and Gremaud, P. A. (2017). Efficient computation of Sobol' indices for stochastic models. *SIAM Journal on Scientific Computing*, 39(4):A1514–A1530.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer, New York.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Ann. Math. Stat.*, 19:293–325.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of non linear models. *Reliability Engineering & System Safety*, 52:1–17.

Hosder, S., Walters, R., and Balch, M. (2007). Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In *48th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, page 1939.

Iemma, U., Diez, M., and Morino, L. (2006). An extended Karhunen-Loève decomposition for modal identification of inhomogeneous structures. *J. Vib. Acoust.*, 128(3):357–365.

Iooss, B., Da Veiga, S., Janon, A., Pujol, G., et al. (2021). Sensitivity: Global sensitivity analysis of model outputs. R package, CRAN.

Iooss, B. and Lemaître, P. (2015). *Uncertainty Management in Simulation-Optimization of Complex Systems*, volume 59 of *Operations Research/Computer Science Interfaces Series (ORCS)*, chapter A review on global sensitivity analysis methods. Springer.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94:1194–1204.

Isukapalli, S. S. (1999). *Uncertainty analysis of transport-transformation models*. PhD thesis, The State University of New Jersey.

Jakeman, J. D. (2022). Pyapprox: Approximation and probabilistic analysis of data. In *SIAM Conf. on Uncertainty Quantification, Atlanta (USA), April 12-15*. (Talk by J. D. Jakeman).

Jakeman, J. D., Eldred, M. S., and Sargsyan, K. (2015). Enhancing $\ell_1$-minimization estimates of polynomial chaos expansions using basis selection. *Journal of Computational Physics*, 289:18–34.

Jakeman, J. D., Franzelin, F., Narayan, A., Eldred, M., and Pflüger, D. (2019). Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351:643–666.

Jakeman, J. D., Narayan, A., and Zhou, T. (2017). A generalized sampling and preconditioning scheme for sparse approximation of polynomial chaos expansions. *SIAM Journal on Scientific Computing*, 39(3):A1114–A1144.

Janon, A., Klein, T., Lagnoux, A., Nodet, M., and Prieur, C. (2014). Asymptotic normality and efficiency of two Sobol index estimators. *ESAIM: Probab. Stat.*, 18:342–364.

Jansen, M. J. W. (1999). Analysis of variance designs for model output. *Comput. Phys. Commun.*, 117(1-2):35–43.

Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356.

Jolliffe, I. T. (2002). *Principal component analysis*. Springer Series in Statistics. Springer Verlag (Second Edition), New York, 2 edition.

Karhunen, K. (1946). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fenn. A1*, A1(37).

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nat. Rev. Phys.*, 3(6):422–440.

Kersaudy, P., Sudret, B., Varsier, N., Picon, O., and Wiart, J. (2015). A new surrogate modeling technique combining Kriging and polynomial chaos expansions – Application to uncertainty analysis in computational dosimetry. *Journal of Computational Physics*, 286:103–117.

Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *Ann. Math. Stat.*, 30(2):271–294.

Kim, H. and Shields, M. D. (2015). Modeling strongly non-Gaussian non-stationary stochastic processes using the iterative translation approximation method and Karhunen–Loève expansion. *Comput. Struct.*, 161:31–42.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Conference proceedings: papers accepted to the International Conference on Learning Representations (ICLR) 2014*. Amsterdam Machine Learning lab (IVI, FNWI).

Konakli, K. and Sudret, B. (2016). Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions. *Journal of Computational Physics*, 321:1144–1169.

Kontolati, K., Loukrezis, D., dos Santos, K. R. M., Giovanis, D. G., and Shields, M. D. (2022a). Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models. *Int. J. Uncertainty Quantification*, 12(4).

Kontolati, K., Loukrezis, D., Giovanis, D. G., Vandanapu, L., and Shields, M. D. (2022b). A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems. *Journal of Computational Physics*, page 111313.

Kosambi, D. D. (1943). Statistics in function space. *J. Indian Math. Soc.*, 7:76–88.

Kougioumtzoglou, I. A., Petromichelakis, I., and Psaros, A. F. (2020). Sparse representations and compressive sampling approaches in engineering mechanics: A review of theoretical concepts and diverse applications. *Probabilistic Engineering Mechanics*, 61:103082.

Krige, D. G. (1951). A statistical approach to some mine valuation and allied problems on the Witwatersrand. Master's thesis, University of the Witwatersrand, South Africa.

Kucherenko, S., Feil, B., Shah, N., and Mauntz, W. (2011). The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety*, 96(4):440–449.

Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C., and Shah, N. (2009). Monte Carlo evaluation of derivative-based global sensitivity measures. *Reliab. Eng. Syst. Safety*, 94:1135–1148.

Lamboni, M., Iooss, B., Popelin, A.-L., and Gamboa, F. (2013). Derivative-based global sensitivity measures: general links with Sobol' indices and numerical tests. *Math. Comput. Simul.*, 87:44–54.

Lamboni, M., Monod, H., and Makowski, D. (2011). Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety*, 96(4):450–459.

Lataniotis, C., Marelli, S., and Sudret, B. (2020). Extending classical surrogate modelling to high dimensions through supervised dimensionality reduction: a data-driven approach. *Int. J. Uncertainty Quantification*, 10(1):55–82.

Lataniotis, C., Marelli, S., and Sudret, B. (2021). Uncertainty quantification in the cloud with UQCloud. In *Proc. 4th Int. Conf. Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP), Athens (Greece), June 27-30.* (Talk given by S. Marelli).

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.

Le Gratiet, L., Marelli, S., and Sudret, B. (2017). *Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes*, chapter 38, pages 1289–1325. Springer International Publishing. Cham, Switzerland.

Le Maître, O. P., Najm, H. N., Ghanem, R. G., and Knio, R. G. (2004). Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *Journal of Computational Physics*, 197(2):502–531.

Le Maître, O. P., Reagan, M., Najm, H. N., Ghanem, R. G., and Knio, O. M. (2002). A stochastic projection method for fluid flow – II. Random process. *Journal of Computational Physics*, 181:9–44.

Lebrun, R. and Dutfoy, A. (2009). Do Rosenblatt and Nataf isoprobabilistic transformations really differ? *Probabilistic Engineering Mechanics*, 24(4):577–584.

Lei, H., Li, J., Gao, P., Stinis, P., and Baker, N. A. (2019). A data-driven framework for sparsity-enhanced surrogates with arbitrary mutually dependent randomness. *Computer Methods in Applied Mechanics and Engineering*, 350:199–227.

Li, C. C. and Der Kiureghian, A. (1993). Optimal discretization of random fields. *Journal of Engineering Mechanics (ASCE)*, 119(6):1136–1154.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. (2020a). Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Adv. Neur. In.*, 33:7498–7512.

Liu, Z., Lesselier, D., Sudret, B., and Wiart, J. (2020b). Surrogate modeling based on resampled polynomial chaos expansions. *Reliability Engineering & System Safety*, 202:107008.

Loève, M. (1978). *Probability theory – Graduate texts in mathematics*, volume 2. Springer Verlag, New-York, 4th edition.

Loukrezis, D., Römer, U., and De Gersem, H. (2019). Assessing the performance of Leja and Clenshaw-Curtis collocation for computational electromagnetics with random input data. *Int. J. Uncertainty Quantification*, 9(1).

Lüthen, N., Marelli, S., and Sudret, B. (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Uncertain. Quantif.*, 9(2):593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022a). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *Int. J. Uncertainty Quantification*, 12(3):49–74.

Lüthen, N., Marelli, S., and Sudret, B. (2022b). A spectral surrogate model for stochastic simulators computed from trajectory samples. *Comput. Meth. Appl. Mech. Eng.* Accepted.

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*.

Ma, X. and Zabaras, N. (2011). Kernel principal component analysis for stochastic input model generation. *Journal of Computational Physics*, 230(19):7311–7331.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472.

Mara, T. and Tarantola, S. (2012). Variance-based sensitivity indices for models with dependent inputs. *Reliability Engineering & System Safety*, 107:125–131.

Mara, T., Tarantola, S., and Annoni, P. (2015). Non-parametric methods for global sensitivity analysis of model output with dependent inputs. *Environ. Modell. Softw.*, 72:173–183.

Marelli, S., Lüthen, N., and Sudret, B. (2021a). UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-104.

Marelli, S. and Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. 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. *Struct. Saf.*, 75:67–74.

Marelli, S., Wagner, P.-R., Lataniotis, C., and Sudret, B. (2021b). Stochastic spectral embedding. *Int. J. Uncertainty Quantification*, 11(2):25–47.

Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput.*, 22:833–847.

Mathelin, L. and Gallivan, K. (2012). A compressed sensing approach for partial differential equations with random input data. *Commun. Comput. Phys.*, 12(4):919–954.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

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*, 2:239–245.

Migliorati, G., Nobile, F., Von Schwerin, E., and Tempone, R. (2014). Analysis of discrete $L^2$ projection on polynomial spaces with random evaluations. *Found. Comp. Math.*, 14(3):419–456.

Montgomery, D. C. (2004). *Design and analysis of experiments*. John Wiley and Sons, New York.

Morris, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174.

Moustapha, M. and Sudret, B. (2019). A two-stage surrogate modelling approach for the approximation of functions with non-smooth outputs. In Papadrakakis, M., Papadopoulos, V., and Stefanou, G., editors, *Proc. 3rd Int. Conf. Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP), Crete Island (Greece), June 24-26*.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: Math. Model. Num.*, 48:116–155.

Nagel, J., Rieckermann, J., and Sudret, B. (2020). Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: application to urban drainage simulation. *Reliability Engineering & System Safety*, 195(#106737).

Narayan, A., Jakeman, J., and Zhou, T. (2017). A Christoffel function weighted least squares algorithm for collocation approximations. *Math. Comput.*, 86(306):1913–1947.

Narayan, A. and Jakeman, J. D. (2014). Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36(6):A2952–A2983.

Narayan, A. and Zhou, T. (2015). Stochastic collocation on unstructured multivariate meshes. *Commun. Comput. Phys.*, 18(1):1–36.

Navarro Jimenez, M., Le Maître, O. P., and Knio, O. M. (2017). Nonintrusive polynomial chaos expansions for sensitivity analysis in stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):378–402.

Nouy, A. (2010). Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Arch. Comput. Meth. Eng.*, 17:403–434.

Oladyshkin, S. and Nowak, W. (2012). Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190.

Olivier, A., Giovanis, D. G., Aakash, B., Chauhan, M., Vandanapu, L., and Shields, M. D. (2020). UQpy: A general purpose Python package and development environment for uncertainty quantification. *Journal of Computational Science*, 47:101204.

Owen, A. B. (2014). Sobol' indices and Shapley value. *SIAM/ASA J. Uncertain. Quantif.*, 2(1):245–251.

Owen, A. B. and Prieur, C. (2017). On Shapley value for measuring importance of dependent inputs. *SIAM/ASA J. Uncertain. Quantif.*, 5(1):986–1002.

Pan, Q., Qu, X., Liu, L., and Dias, D. (2020). A sequential sparse polynomial chaos expansion using Bayesian regression for geotechnical reliability estimations. *Int. J. Numer. Anal. Met.*, 44(6):874–889.

Papaioannou, I., Ehre, M., and Straub, D. (2019). PLS-based adaptation for efficient PCE representation in high dimensions. *Journal of Computational Physics*, 387:186–204.

Parisi, P., Lataniotis, C., Marelli, S., and Sudret, B. (2021). Combining surrogate models by ensemble learning techniques for uncertainty quantification. In *Proc. 4th Int. Conf. Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP), Athens (Greece), June 27-30.* (Talk given by P. Parisi).

Patelli, E., Broggi, M., de Angelis, M., and Beer, M. (2014). OpenCossan: an efficient open tool for dealing with epistemic and aleatory uncertainties. In *Vulnerability, Uncertainty, and*

*Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom).*

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proc. of 27th Asilomar Conf. on Signals, Systems and Computers*, pages 40–44. IEEE.

Pearce, M., Poloczek, M., and Branke, J. (2022). Bayesian optimization allowing for common random numbers. *Oper. Res.*, 0(0):1–16.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Peng, J., Hampton, J., and Doostan, A. (2016). On polynomial chaos expansion via gradient-enhanced $\ell_1$-minimization. *Journal of Computational Physics*, 310:440–458.

Perrin, G., Soize, C., Duhamel, D., and Funfschilling, C. (2012). Identification of polynomial chaos representations in high dimension from a set of realizations. *SIAM Journal on Scientific Computing*, 34:2917–2945.

Perrin, T., Roustant, O., Rohmer, J., Alata, O., Naulin, J., Idier, D., Pedreros, R., Moncoulon, D., and Tinard, P. (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliability Engineering & System Safety*, 211:107522.

Phoon, K., Huang, H., and Quek, S. (2005). Simulation of strongly non Gaussian processes using Karhunen-Loève expansion. *Probabilistic Engineering Mechanics*, 20(2):188–198.

Phoon, K., Huang, S., and Quek, S. (2002). Simulation of second-order processes using Karhunen-Loève expansion. *Comput. Struct.*, 80(12):1049–1060.

Pillaud-Vivien, L., Bach, F., Lelièvre, T., Rudi, A., and Stoltz, G. (2020). Statistical estimation of the Poincaré constant and application to sampling multimodal distributions. In *International Conference on Artificial Intelligence and Statistics*, pages 2753–2763. PMLR.

Plumlee, M. and Tuo, R. (2014). Building accurate emulators for stochastic simulations via quantile Kriging. *Technometrics*, 56(4):466–473.

Poirion, F. and Puig, B. (2010). Simulation of non-Gaussian multivariate stationary processes. *Int. J. Nonlin. Mech.*, 45(5):587–597.

Poirion, F. and Zentner, I. (2013). Non-Gaussian non-stationary models for natural hazard modeling. *Appl. Math. Model.*, 37(8):5938–5950.

Poirion, F. and Zentner, I. (2014). Stochastic model construction of observed random phenomena. *Probabilistic Engineering Mechanics*, 36:63–71.

Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., and Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *J. Commun. Netw.*, 15(5):443–456.

Rahman, S. (2020). A spline chaos expansion. *SIAM/ASA J. Uncertain. Quantif.*, 8(1):27–57.

Raisee, M., Kumar, D., and Lacor, C. (2015). A non-intrusive model reduction approach for polynomial chaos expansion using proper orthogonal decomposition. *Int. J. Num. Meth. Eng.*, 103:293–312.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Ramsay, J. and Silverman, B. (2005). *Functional data analysis*. Springer Series in Statistics. Springer Science and Business Media, 2nd edition.

Rao, R. B., Fung, G., and Rosales, R. (2008). On the dangers of cross-validation. An experimental evaluation. In *Proc. 2008 SIAM Int. Conf. Data Mining*, pages 588–596. SIAM.

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.

Razavi, S., Tolson, B. A., and Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resour. Res.*, 48(7).

Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R. (2008). Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA Journal*, 46(11):2814–2822.

Rohmer, J., Roustant, O., Lecacheux, S., and Manceau, J.-C. (2022). Revealing the interlevel dependence structure of categorical inputs in numerical environmental simulations with kernel model selection. *Environ. Modell. Softw.*, 151:105380.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Ann. Math. Stat.*, 23:470–472.

Roustant, O., Barthe, F., and Iooss, B. (2017). Poincaré inequalities on intervals – application to sensitivity analysis. *Electron. J. Stat.*, 11(2):3081–3119.

Roustant, O., Gamboa, F., and Iooss, B. (2020a). Parseval inequalities and lower bounds for variance-based sensitivity indices. *Electron. J. Statist.*, 14(1):386–412.

Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. (2020b). Group kernels for Gaussian process metamodels with categorical inputs. *SIAM/ASA J. Uncertain. Quantif.*, 8(2):775–806.

Rozsas, A. and Slobbe, A. (2019). Repository and Black-box Reliability Challenge 2019. https://gitlab.com/rozsasarpi/rprepo/ (Accessed: 2022-07-17).

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Stat. Sci.*, 4:409–435.

Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *Wires. Data Min. Knowl.*, 8(4):e1249.

Sakamoto, S. and Ghanem, R. (2002). Simulation of multi-dimensional non-Gaussian non-stationary random fields. *Probabilistic Engineering Mechanics*, 17(2):167–176.

Saltelli, A., Aleksankina, K., Becker, W., Fennell, P., Ferretti, F., Holst, N., Li, S., and Wu, Q. (2019). Why so many published sensitivity analyses are false: a systematic review of sensitivity analysis practices. *Envir. Model. Soft.*, 114:29–39.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd., Chichester, West Sussex, UK.

Sargsyan, K., Safta, C., Boll, L., Johnston, K., Khalil, M., Chowdhary, K., Rai, P., Casey, T., Zeng, X., and Debusschere, B. (2022). UQTk Version 3.1.2 User Manual. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Sarma, P., Durlofsky, L. J., and Aziz, K. (2008). Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. *Math. Geosci.*, 40(1):3–32.

Schöbi, R., Sudret, B., and Wiart, J. (2015). Polynomial-chaos-based Kriging. *Int. J. Uncertainty Quantification*, 5(2):171–193.

Schwab, C. and Todor, R. (2006). Karhunen–Loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464.

Shao, Q., Younes, A., Fahs, M., and Mara, T. (2017). Bayesian sparse polynomial chaos expansion for global sensitivity analysis. *Comput. Meth. Appl. Mech. Eng.*, 318:474–496.

Shields, M., Deodatis, G., and Bocchini, P. (2011). A simple and efficient methodology to approximate a general non-Gaussian stationary stochastic process by a translation process. *Probabilistic Engineering Mechanics*, 26(4):511–519.

Shields, M. D. and Zhang, J. (2016). The generalization of Latin hypercube sampling. *Reliability Engineering & System Safety*, 148:96–108.

Shin, Y. and Xiu, D. (2016). On a near optimal sampling strategy for least squares polynomial regression. *Journal of Computational Physics*, 326:931–946.

Slot, R. M., Sorensen, J. D., Sudret, B., Svenningsen, L., and Thogersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14:199–222.

Sobol, I. and Gresham, A. (1995). On an alternative global sensitivity estimator. In *Proceedings of SAMO 1995*, pages 40–42, Belgirate.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Modeling & Comp. Exp.*, 1(4):407–414.

Sobol', I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.*, 55(1-3):271–280.

Sobol', I. M. and Kucherenko, S. (2009). Derivative based global sensitivity measures and their link with global sensitivity indices. *Math. Comput. Simul.*, 79(10):3009–3017.

Soize, C. (2010). Identification of high-dimension polynomial chaos expansions with random coefficients for non-Gaussian tensor-valued random fields using partial and limited experimental data. *Computer Methods in Applied Mechanics and Engineering*, 199(33-36):2150–2164.

Soize, C. and Ghanem, R. (2004). Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26(2):395–410.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society: Series B*, 39(1):44–47.

Sudret, B. (2007). *Uncertainty propagation and sensitivity analysis in mechanical models – Contributions to structural reliability and stochastic spectral methods*. Université Blaise Pascal, Clermont-Ferrand, France. Habilitation à diriger des recherches, 173 pages.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Saf.*, 93:964–979.

Sudret, B. and Der Kiureghian, A. (2000). Stochastic finite elements and reliability: a state-of-the-art report. Technical Report UCB/SEMM-2000/08, University of California, Berkeley (173 pages).

Sudret, B. and Mai, C.-V. (2015). Computing derivative-based global sensitivity measures using polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 134:241–250.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Int. Conf. Mach. Learn.*, pages 3319–3328. PMLR.

Szegö, G. (1939). *Orthogonal polynomials*, volume 23. American Mathematical Society.

Timpe, M. L., Veiga, M. H., Knabenhans, M., Stadel, J., and Marelli, S. (2020). Machine learning applied to simulations of collisions between rotating, differentiated planets. *Comput. Astrophys. Cosm.*, 7(2):1–38.

Tipireddy, R. and Ghanem, R. (2014). Basis adaptation in homogeneous chaos spaces. *Journal of Computational Physics*, 259:304–317.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1(Jun):211–244.

Tipping, M. E. and Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics*.

Torossian, L., Picheny, V., Faivre, R., and Garivier, A. (2020). A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety*, 201:106858.

Torre, E., Marelli, S., Embrechts, P., and Sudret, B. (2019). Data-driven polynomial chaos expansion for machine learning regression. *J. Comput. Phys*, 388:601–623.

Tripathy, R. K. and Bilionis, I. (2018). Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*, 375:565–588.

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.

Tsilifis, P., Huan, X., Safta, C., Sargsyan, K., Lacaze, G., Oefelein, J. C., Najm, H. N., and Ghanem, R. G. (2019). Compressive sensing adaptation for polynomial chaos expansions. *Journal of Computational Physics*, 380:29–47.

van den Berg, E. and Friedlander, M. P. (2008). Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912.

Vanmarcke, E.-H. and Grigoriu, M. (1983). Stochastic finite element analysis of simple beams. *Journal of Engineering Mechanics (ASCE)*, 109(5):1203–1214.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Venturi, D. (2011). A fully symmetric nonlinear biorthogonal decomposition theory for random fields. *Physica D: Nonlinear Phenomena*, 240(4-5):415–425.

Viana, F. A., Haftka, R. T., and Steffen Jr, V. (2009). Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization*, 39(4):439–457.

Wan, X. and Karniadakis, G. (2006). Beyond Wiener-Askey expansions: handling arbitrary PDFs. *J. Sci. Comput.*, 27:455–464.

Weise, K., Müller, E., Poßner, L., and Knösche, T. R. (2022). Comparison of the performance and reliability between improved sampling strategies for polynomial chaos expansion. *Math. Biosci. Eng.*, 19(8):7425–7480.

Wiener, N. (1938). The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936.

Wipf, D. P. and Rao, B. D. (2004). Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164.

Xiu, D. (2009). Fast numerical methods for stochastic computations: a review. *Comm. Comput. Phys.*, 5(2-4):242–272.

Xiu, D. (2010). *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, Princeton, New Jersey, USA.

Xiu, D. and Hesthaven, J. S. (2005). High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139.

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.

Yamazaki, F. and Shinozuka, M. (1988). Digital generation of non-Gaussian stochastic fields. *Journal of Engineering Mechanics (ASCE)*, 114(7):1183–1197.

Yan, L., Guo, L., and Xiu, D. (2012). Stochastic collocation algorithms using $\ell_1$-minimization. *Int. J. Uncertainty Quantification*, 2(3).

Yang, L., Zhang, D., and Karniadakis, G. E. (2020). Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):A292–A317.

Yang, X., Li, W., and Tartakovsky, A. (2018). Sliced-inverse-regression–aided rotated compressive sensing method for uncertainty quantification. *SIAM/ASA J. Uncertain. Quantif.*, 6(4):1532–1554.

Yin, P., Lou, Y., He, Q., and Xin, J. (2015). Minimization of $\ell_{1-2}$ for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563.

Zentner, I. and Poirion, F. (2012). Enrichment of seismic ground motion databases using Karhunen–Loève expansion. *Earthq. Eng. Struct. Dyn.*, 41(14):1945–1957.

Zhang, J. and Ellingwood, B. (1994). Orthogonal series expansions of random fields in reliability analysis. *Journal of Engineering Mechanics (ASCE)*, 120(12):2660–2677.

Zhang, R., Yang, X., and Dai, H. (2022). A non-Gaussian stochastic model from limited observations using polynomial chaos and fractional moments. *Reliability Engineering & System Safety*, page 108323.

Zhang, Z., Xu, Y., Yang, J., Li, X., and Zhang, D. (2015). A survey of sparse representation: algorithms and applications. *IEEE Access*, 3:490–530.

Zhou, Y., Lu, Z., Cheng, K., and Ling, C. (2019). An efficient and robust adaptive sampling method for polynomial chaos expansion in sparse Bayesian learning framework. *Computer Methods in Applied Mechanics and Engineering*, 352:654–674.

Zhu, X., Broccardo, M., and Sudret, B. (2022). Use of generalized lambda models for seismic fragility analysis. In *Proc. 8th Int. Symp. on Reliability Engineering and Risk Management (ISRERM), Hannover (Germany), September 4-7.*

Zhu, X. and Sudret, B. (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *Int. J. Uncertainty Quantification*, 10(3):249–275.

Zhu, X. and Sudret, B. (2021a). Construction of sparse polynomial chaos surrogate model for simulators with mixed continuous and categorical variables. In *Proc. 4th Int. Conf. Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP), Athens (Greece), June 27-30.* (Talk given by X. Zhu).

Zhu, X. and Sudret, B. (2021b). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA J. Unc. Quant.*, 9(4):1345–1380.

Zhu, X. and Sudret, B. (2021c). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliab. Eng. Sys. Safety*, 214:107815.

Zhu, X. and Sudret, B. (2022). Stochastic polynomial chaos expansions to emulate stochastic simulators. *Int. J. Uncertainty Quantification*. Accepted.

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., and Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81.