A spectral surrogate model for stochastic simulators computed from trajectory samples

Nora Lüthen, Stefano Marelli, Bruno Sudret

▶ **To cite this version:**

Nora Lüthen, Stefano Marelli, Bruno Sudret. A spectral surrogate model for stochastic simulators computed from trajectory samples. 2022. hal-03787360

HAL Id: hal-03787360
https://hal.science/hal-03787360

Preprint submitted on 24 Sep 2022

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.

ETH zürich

A SPECTRAL SURROGATE MODEL FOR STOCHASTIC SIMULATORS COMPUTED FROM TRAJECTORY SAMPLES

N. Lüthen, S. Marelli and B. Sudret

CHAIR OF RISK, SAFETY AND UNCERTAINTY QUANTIFICATION
STEFANO-FRANSCINI-PLATZ 5
CH-8093 ZÜRICH

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	-
Report Ref.:	RSUQ-2022-005
Arxiv Ref.:	https://arxiv.org/abs/2207.05653 [stat.CO]
DOI:	-
Date submitted:	July 12, 2022
Date accepted:	-

A spectral surrogate model for stochastic simulators computed from trajectory samples

Nora Lüthen[1], Stefano Marelli[1], and Bruno Sudret[1]

[1] *Chair of Risk, Safety, and Uncertainty Quantification, ETH Zürich, 8093 Zürich, Switzerland*

July 14, 2022

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.

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 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.,

[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).

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, 2021a) 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 determine 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

[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.

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 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 2 we recall the relevant theory and definitions. In Section 3 we present our new stochastic emulator. The proposed method is then applied in Section 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 5. Finally, we draw conclusions and give an outlook on possible further developments in Section 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.

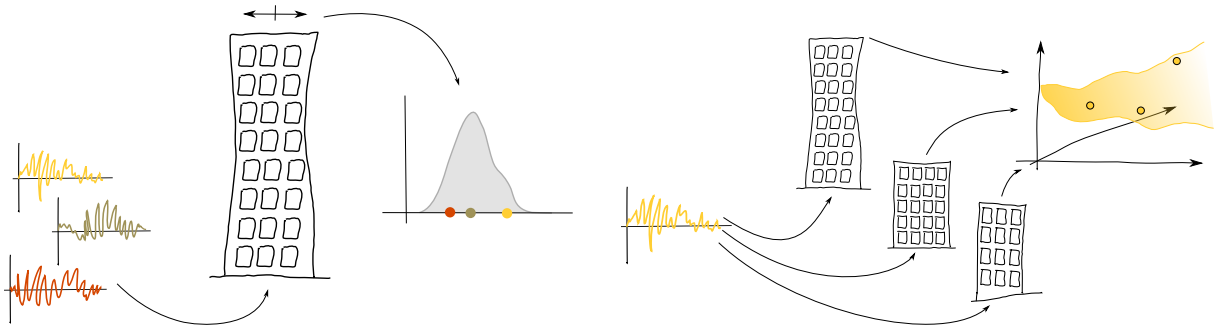
2.1 Stochastic simulators as random fields

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

$$\mathcal{M} : \mathcal{D} \times \Omega \to \mathbb{R}, \tag{1}$$

$$(\boldsymbol{x}, \omega) \mapsto \mathcal{M}(\boldsymbol{x}, \omega). \tag{2}$$

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 Fig. 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 1: *Visual representation of a stochastic simulator when either the input parameters* $\boldsymbol{x}$ *or the random event* ω *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* ω *(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

$$F_{Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_n}}(y_1, \ldots, y_n) = \mathbb{P}\left(Y_{\boldsymbol{x}_1} \leq y_1 \wedge \ldots \wedge Y_{\boldsymbol{x}_n} \leq y_n\right) \tag{3}$$

for *all* $n \geq 1$ and *any* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{D}$. 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

$$c(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(Y_{\boldsymbol{x}} - \mu(\boldsymbol{x}))(Y_{\boldsymbol{x}'} - \mu(\boldsymbol{x}'))\right]. \tag{4}$$

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.

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 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., $L^2_{f_{\boldsymbol{X}}}(\mathcal{D}) = \{g : \mathcal{D} \to \mathbb{R} \mid \mathrm{Var}_{\boldsymbol{X}}[g(\boldsymbol{X})] < +\infty\}$. 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 $a_{\boldsymbol{\alpha}} = \mathbb{E}_{\boldsymbol{X}}[\mathcal{M}(\boldsymbol{X})\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})]$. This representation is called *polynomial chaos expansion*. In practice, a truncated expansion is computed,

$$\mathcal{M}(\boldsymbol{X}) \approx \mathcal{M}^{\mathrm{PCE}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{5}$$

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, 2022).

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 ξ_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

$$\mathcal{M}_{\boldsymbol{x}}(\omega) = \mu(\boldsymbol{x}) + \sum_{k=1}^{\infty}\sqrt{\lambda_k}\xi_k(\omega)\phi_k(\boldsymbol{x}) \tag{6}$$

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, (λ_k,ϕ_k) are solutions to the integral eigenvalue problem

$$\int_{\mathcal{D}} c(\boldsymbol{x},\boldsymbol{x}')\phi_k(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}' = \lambda_k\phi_k(\boldsymbol{x}), \tag{7}$$

and ξ_k is the result of the projection of $\mathcal{M}$ onto the spatial basis

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}}\int_{\mathcal{D}}\mathcal{M}(\boldsymbol{x},\omega)\phi_k(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{8}$$

From Eq. (6) and the properties of ϕ_k and ξ_k it follows immediately that the covariance function can be expressed as

$$c(\boldsymbol{x},\boldsymbol{x}') = \sum_{k=1}^{\infty}\lambda_k\phi_k(\boldsymbol{x})\phi_k(\boldsymbol{x}') \tag{9}$$

(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 ξ_k are standard Gaussian and independent. However, Eq. (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) to (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 3.

In practice, the infinite expansion in Eq. (6) must be truncated. From the orthonormality of $\{\phi_k\}$ it follows from Eq. (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:

$$\frac{\sum_{k=1}^{M}\lambda_k}{\sum_{k=1}^{\infty}\lambda_k} > 1-\epsilon \tag{10}$$

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. (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 3). Detailed calculations are provided in Appendix A.

2.4 Inference of the joint distribution of the Karhunen-Loève random variables

Characterizing the dependent (but uncorrelated) Karhunen-Loève random variables (KL-RV) $\xi_k, k = 1, \dots, M$ 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., 2019a).

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

$$F_{\boldsymbol{Z}}(z_1, \dots, z_M) = C\left(F_{Z_1}(z_1), \dots, F_{Z_M}(z_M)\right), \tag{11}$$

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

$$C(u_1, \dots, u_M) = F_{\boldsymbol{Z}}\left(F_{Z_1}^{-1}(u_1), \dots, F_{Z_M}^{-1}(u_M)\right). \tag{12}$$

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. (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 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 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 $\mathrm{AIC} = 2k - 2\log\mathcal{L}$, 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) $\mathcal{G}(\mu,\beta)$	$\mu\approx-0.4501, \beta\approx0.7797$	0
Gumbel (for minima) $\mathcal{G}_{\min}(\mu,\beta)$	$\mu\approx0.4501, \beta\approx0.7797$	0
Logistic $P(\mu,s)$	$\mu=0, s\approx0.5513$	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 $r=\frac{a(ab+1)}{b-a}, s=\frac{b(ab+1)}{a-b}$	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

[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.

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. (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).

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:

$$\mathcal{T}_r = \left\{ \left(\boldsymbol{x}^{(r,i)}, \mathcal{M}(\boldsymbol{x}^{(r,i)}, \omega^{(r)}) \right) : i = 1, \dots, N_r \right\}, \quad r = 1, \dots, R \tag{13}$$

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 Fig. 2):

1. **Approximate each discrete trajectory** $\mathcal{T}_r$ by a sparse PCE $\mathcal{M}_r^{\mathrm{PCE}}$ in $L^2_{f_{\boldsymbol{X}}}(\mathcal{D})$:

$$\mathcal{M}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{(r)}} a_{\boldsymbol{\alpha}}^{(r)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{14}$$

   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., 2022) and apply the least-angle regression 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

$$\hat{\mu}^{\mathrm{PCE}}(\boldsymbol{x}) = \frac{1}{R}\sum_{r=1}^{R}\mathcal{M}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}}\left(\frac{1}{R}\sum_{r=1}^{R}a_{\boldsymbol{\alpha}}^{(r)}\right)\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{15}$$

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

$$\hat{c}(\boldsymbol{x},\boldsymbol{x}') = \frac{1}{R-1}\sum_{r=1}^{R}\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x})\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}'). \tag{16}$$

Computing the eigenfunctions $\phi(\boldsymbol{x})$ of the associated integral eigenvalue problem in Eq. (7) is equivalent to computing a PCA on the PCE coefficients, i.e., equivalent to solving the following P-dimensional eigenproblem for $\tilde{\boldsymbol{a}}$:

$$\boldsymbol{\Sigma}\boldsymbol{b} = \lambda\boldsymbol{b}, \tag{17}$$

where $\boldsymbol{\Sigma} = \frac{1}{R-1}\tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^T$ (see Appendix 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. (10)).
6. **Compute the realizations** of the KL-RV ξ_i from the sample trajectories by projecting onto the eigenfunctions. Due to the orthonormality of the PCE basis, this can be done analytically (see Appendix 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 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

$$\hat{\mathcal{M}}(\boldsymbol{x},\cdot) = \hat{\mu}(\boldsymbol{x}) + \sum_{k=1}^{K}\sqrt{\lambda_k}\; Z_k(\cdot)\underbrace{\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}} b_{\boldsymbol{\alpha}}^{(k)}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)}_{=\phi_k(\boldsymbol{x})} \tag{18}$$

where $\boldsymbol{Z} = (Z_1, Z_2, \dots, Z_K)$ is a random vector distributed according to the inferred joint distribution $f_{\boldsymbol{\xi}}$.

The full procedure is visualized in Fig. 2.

Using the stochastic emulator constructed in Eq. (18), we can easily compute the following quantities.

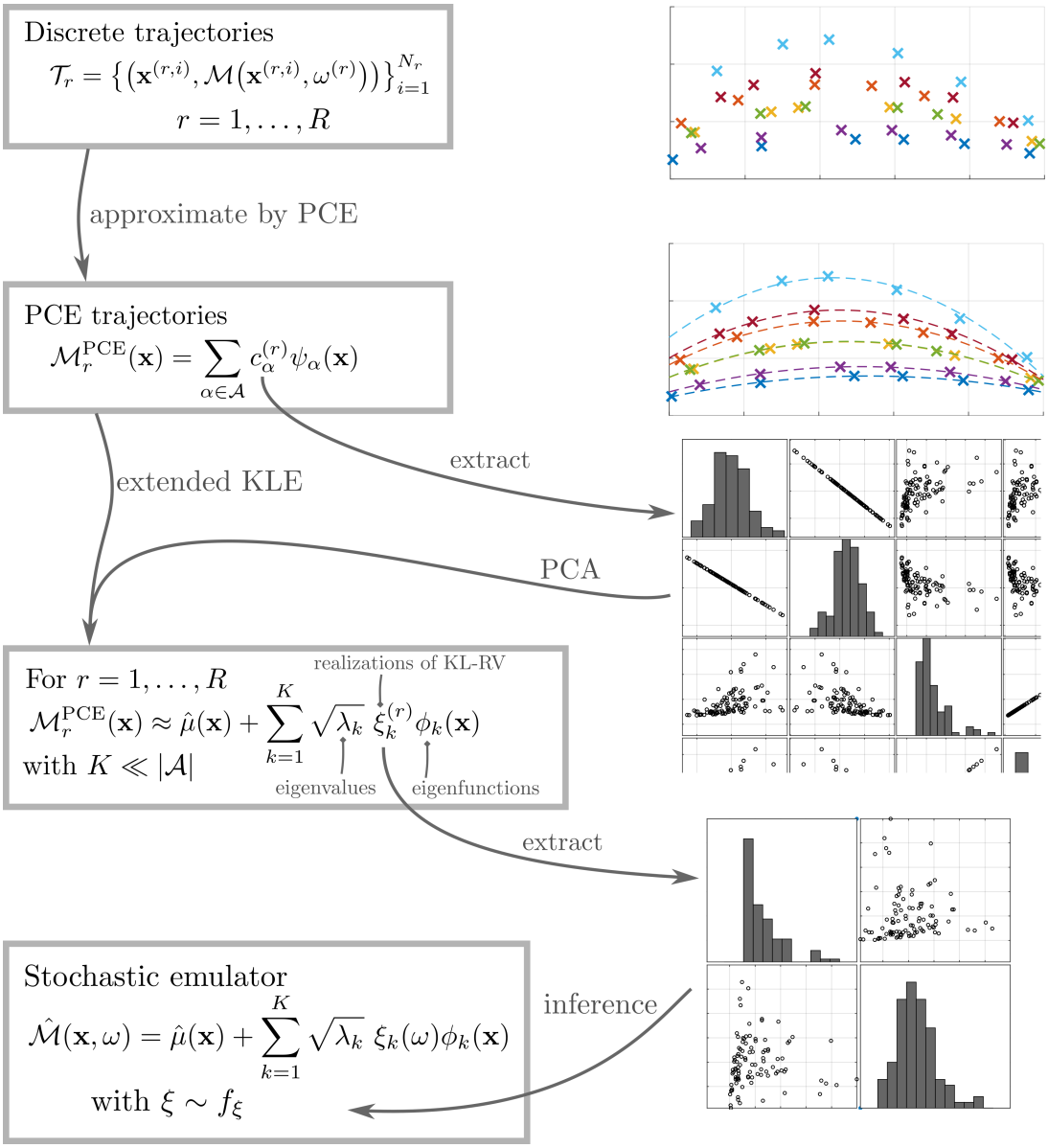


Figure 2: *Sketch of our stochastic emulator, starting with stochastic simulator samples (discrete trajectories) at the top and resulting in the stochastic emulator at the bottom, which is a KLE that includes a probabilistic model of the KL-RV. The sketch is purely for illustration and does not display real data. Note that there are two equivalent ways to arrive at the third box: through extended KLE and through PCA on the coefficients.*

- The mean function $\hat{\mu}$ is given by the sample mean of the approximated trajectories (PCE trajectories), see Eq. (15).
- The covariance function $\hat{c}(\cdot,\cdot)$ can be computed from the KLE eigenfunctions using the truncated version of Eq. (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 ξ_k, and evaluating Eq. (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 4. However, unlike our stochastic emulator in Eq. (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., 2022) 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 Appendix A), which avoids the expensive numerical solution of the integral eigenvalue problem in d dimensions, and instead solves an inexpensive discrete eigenvalue problem.

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 4.1), the borehole model with five hidden (latent) variables (Section 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 (2021b) as a stochastic emulator benchmark model (Section 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)

$$d_{\mathrm{WS}}(Y_1, Y_2) = \|Q_1 - Q_2\|_2 = \sqrt{\int_0^1 (Q_1(u) - Q_2(u))^2 \, \mathrm{d}u}. \tag{19}$$

To measure the global quality of marginal approximation, we consider the quantity

$$\epsilon_{\text{marg}} = \mathbb{E}_{\boldsymbol{X}}\left[\frac{d_{\text{WS}}(\mathcal{M}(\boldsymbol{X},\cdot),\hat{\mathcal{M}}(\boldsymbol{X},\cdot))}{\sigma(\mathcal{M}(\boldsymbol{X},\cdot))}\right], \tag{20}$$

computed by Monte-Carlo integration on a validation set with $N_{\text{val}} = 1,000$ points and $R_{\text{val}} = 10,000$ replications (Zhu and Sudret, 2021a).

- The global error between the true covariance function c and the emulated one $\hat{c}$ is computed by

$$\epsilon_{\text{cov}} = \|c - \hat{c}\|_{L^2_{f_{\boldsymbol{X}}}(\mathcal{D})\times L^2_{f_{\boldsymbol{X}}}(\mathcal{D})} \approx \frac{1}{N_{\text{val}}}\left\|C - \hat{C}\right\|_F \tag{21}$$

where C and $\hat{C}$ denote the true and emulated covariance matrices for a validation sample $\{\boldsymbol{x}^{(i)} : i = 1, \ldots, N_{\text{val}}\}$, and $\|\cdot\|_F$ is the Frobenius norm.

4.1 Stochastic Ishigami function

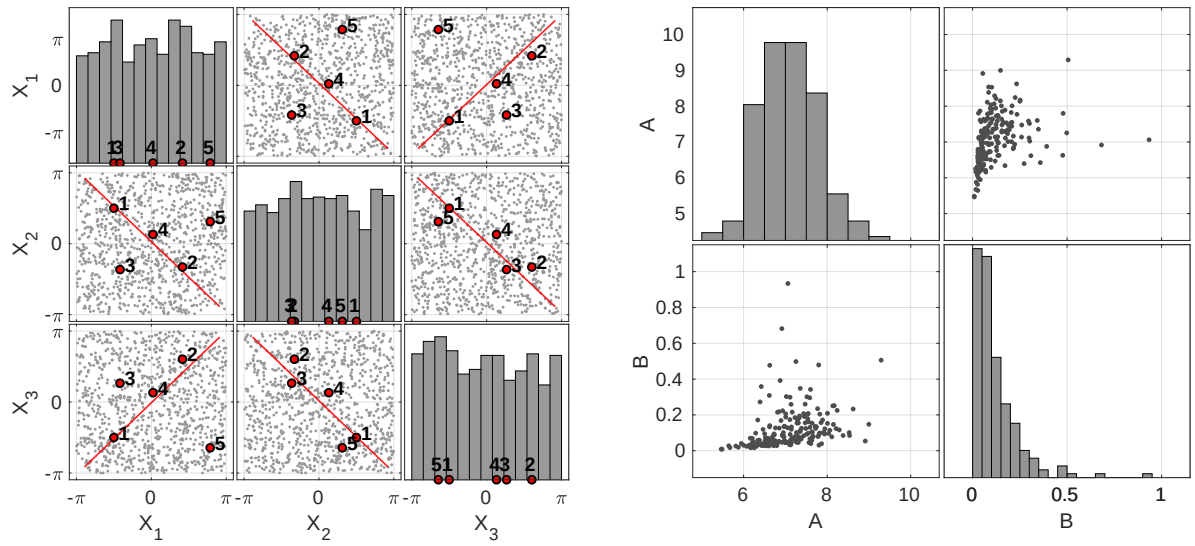
4.1.1 Problem statement

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:

$$f(\boldsymbol{X}; A, B) = \sin(X_1) + A\sin(X_2)^2 + BX_3^4\sin(X_1). \tag{22}$$

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. (22) reveals that the main effect of the group of explicit input parameters ($\mathcal{M}_1$ in Eq. (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 Fig. 3.

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.



(a) *Input space with validation points used for visualization*

(b) *Latent space*

Figure 3: *Samples of the input space (left) and the latent space (right). The red line in Fig. 3a is the trajectory along which the simulator/emulator response is plotted in Fig. 9. The red dots annotated by small numbers denote the five points that are used for visualization in the following. Fig. 3b shows a sample of the latent space (A, B in Eq. (22)).*

4.1.2 Analysis of the KL-RV samples

To illustrate the type of result obtained with our proposed stochastic emulator described in Section 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 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 Fig. 4, 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 3). We see that their inferred joint distribution (Beta and Gumbel marginals, with a Clayton copula) visually matches the validation data well.

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 Fig. 5 for $N = 150$ and $R = 100$. Validation samples from the

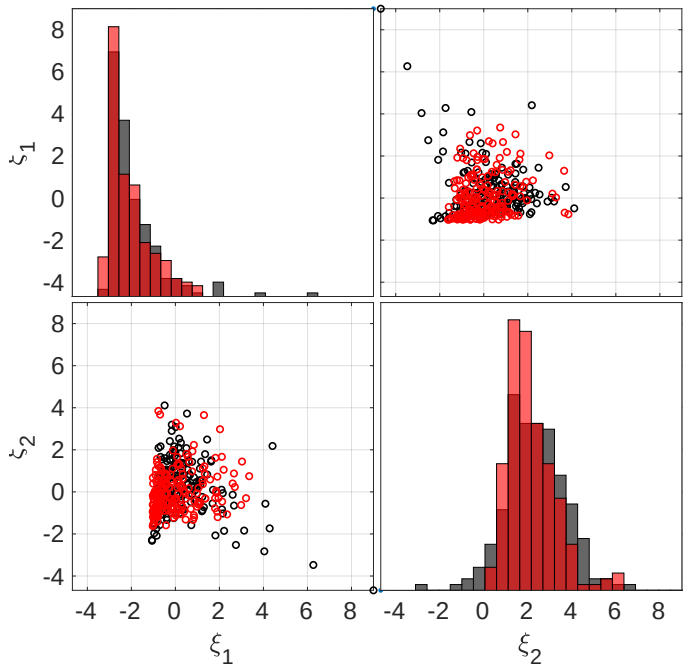


Figure 4: *KL-RV coefficient samples computed from validation trajectories (black) and new samples from the stochastic emulator (red). This is data from one experiment with* $N = 150$ *and* $R = 100$*, max degree* $p = 14$*. Number of validation trajectories and resampled PCE coefficients:* 200 *each. 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* $\lambda_2 = 0.10$*.*

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. (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. (22) and change proportionally with B.

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.

Fig. 6 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.

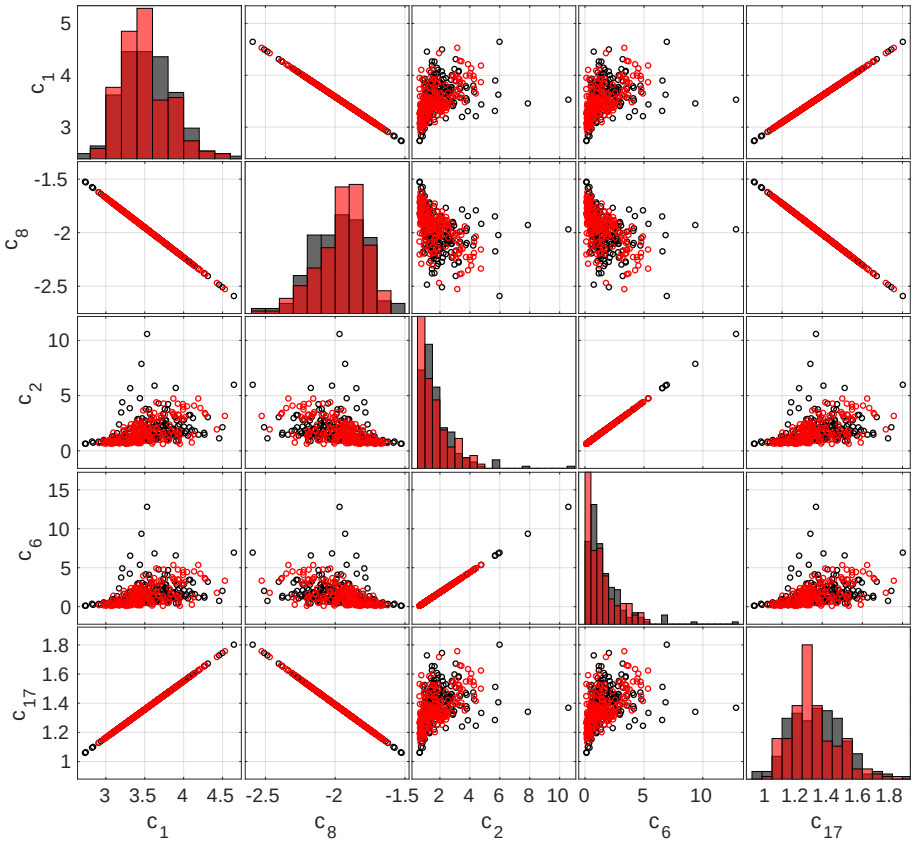


Figure 5: *PCE coefficient samples computed from validation trajectories (black) and new samples from the stochastic emulator (red). This is data from one experiment with $N = 120$ and $R = 100$, max degree $p = 14$. Number of validation trajectories and resampled PCE coefficients: $R_{val} = 200$ each. The PCE coefficients are sorted by mean magnitude, and we only display the largest 5 out of total 75 nonzero coefficients.*

The five selected validation locations $\{\boldsymbol{x}^{(i)}\}_{i=1}^{5}$ in the input space are visualized in Fig. 3a by red dots. Each black (resp. red) point in Fig. 6 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).

From the data in the off-diagonal scatter plots in Fig. 6, we can compute the sample covariance matrix. However, we can also compute the covariance analytically from the KLE eigenfunctions, using Eq. (9). In Fig. 7, we use the five illustrative points shown in Fig. 3a 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.

In Fig. 8, 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 Fig. 3a) 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

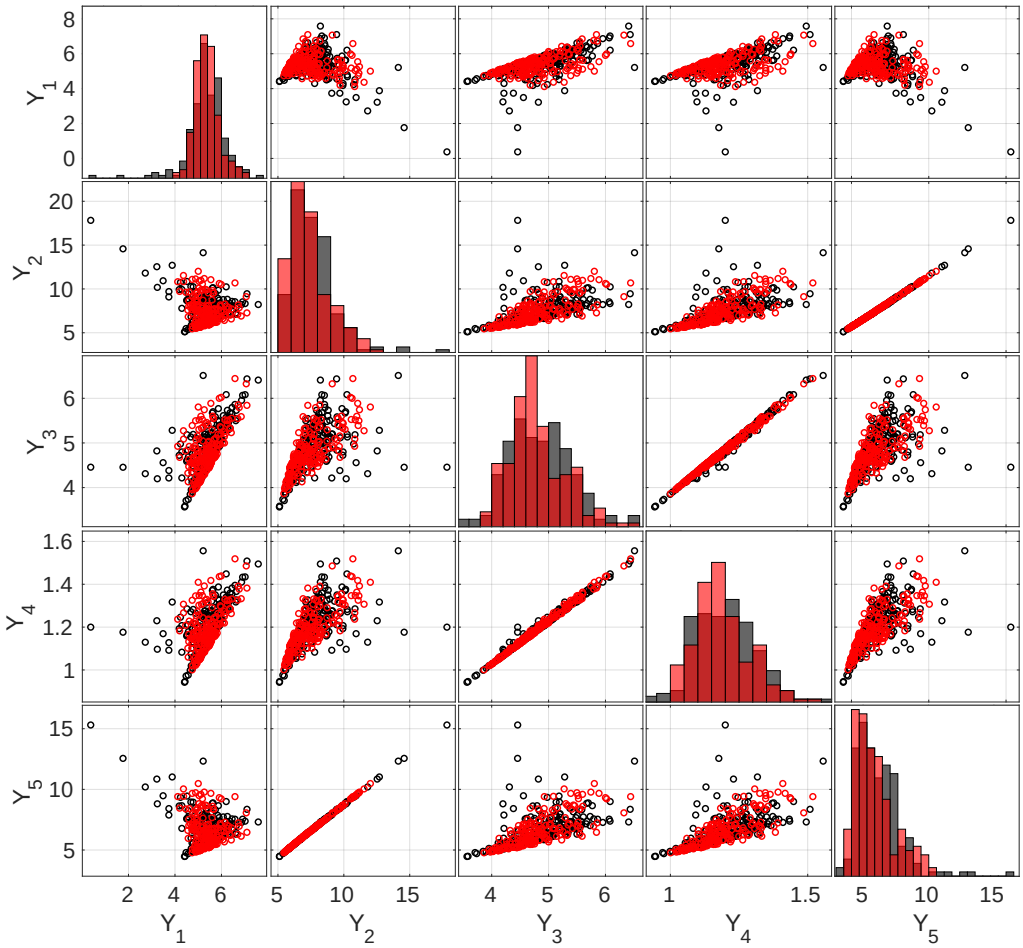


Figure 6: *Scatterplot of output $Y_i = \mathcal{M}(\boldsymbol{x}^{(i)}, \cdot)$ of the stochastic simulator (black) and of output $\hat{Y}_i = \hat{\mathcal{M}}(\boldsymbol{x}^{(i)}, \cdot)$ of the parametric stochastic emulator (red, created from training set with $N = 150$, $R = 100$) for five validation points sampled from the input space. The location of these five points is illustrated in Fig. 3a.*

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.

Finally, to assess visually how well the resampled trajectories match the behavior of the original stochastic model, we plot in Fig. 9 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 Fig. 3a 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 Fig. 9c 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,

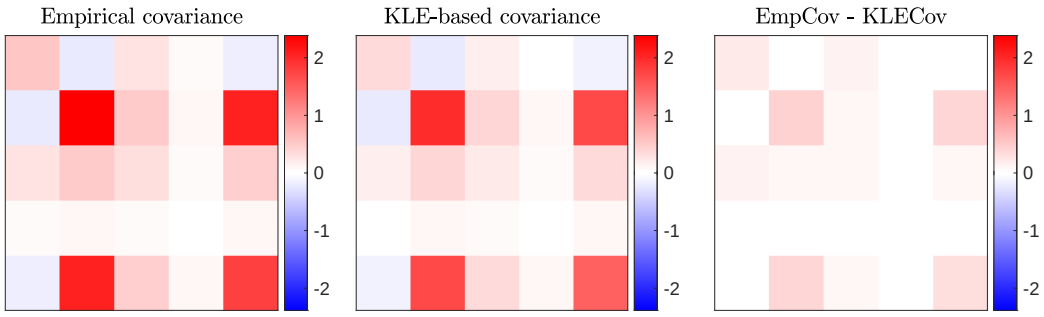
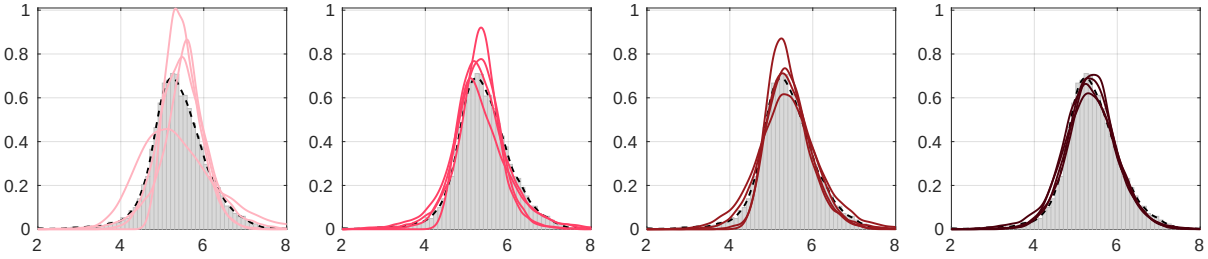


Figure 7: *Covariance approximation for the Ishigami model for the $N_{val} = 5$ validation locations in the input space illustrated in Fig. 3a. Computation based on $N = 150, R = 100$, max degree $p = 14$. KLE-based covariance: computed from eigenfunctions as in Eq. (9). Empirical covariance: based on the validation set comprising $10,000$ trajectories of the stochastic simulator.*



(a) $R = 10$ **(b)** $R = 30$ **(c)** $R = 100$ **(d)** $R = 300$

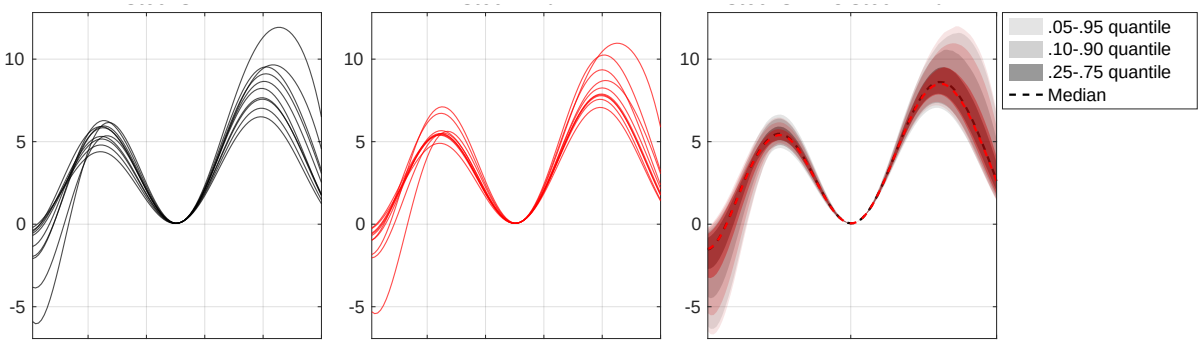
Figure 8: *Prediction of Y-marginal at validation point $\boldsymbol{x}_{val}^{(1)} = (-\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$ for the Ishigami model and $R = \{10, 30, 100, 300\}$ trajectories with the parametric stochastic emulator for 4 independent repetitions. Visualization of predicted marginals by KDE using $10,000$ samples. Number of experimental design points $N = 150$, max degree $p = 14$. The approximation error is in the order $\mathcal{O}(10^{-10})$.*

the emulator should have a smaller variance than the simulator, because terms are missing from Eq. (9) due to truncation.

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 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 Fig. 10a, we display the global convergence of marginal predictions for the parametric stochastic emulator in terms of ϵ_{marg} defined in Eq. (20). Each boxplot represents one experiment (i.e., a specific number of experimental design points and number of trajectories), repeated indepen-



(a) *Stochastic simulator* **(b)** *Stochastic emulator* **(c)** *Comparison of quantiles for both models*

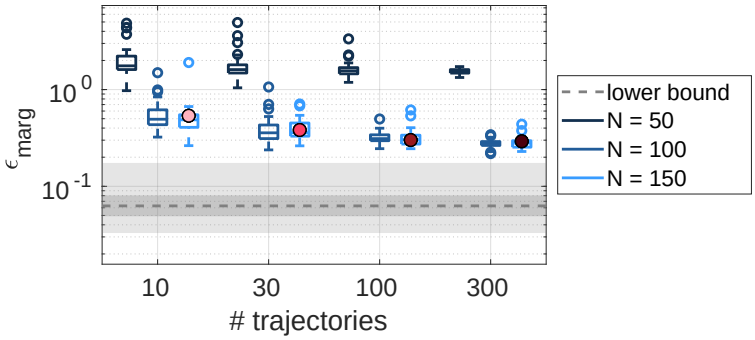
Figure 9: *Visualization of stochastic simulator/emulator response Y when following a 1D slice through the input space, which is illustrated with a red line in Fig. 3a. $N = 150, R = 100$. The left and middle plots show 10 trajectories each. The right plot aggregates the values from $10,000$ trajectories to show quantile information.*

dently 50 times. The value of the averaged and normalized Wasserstein distance ϵ_{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:

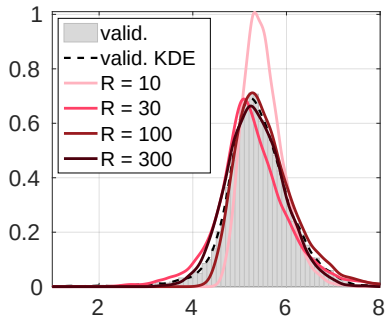
- The averaged and normalized Wasserstein distance is computed based on samples. As a lower bound, we independently sample 100×2 validation sets (each consisting of $R_{\text{val}} = 10,000$ trajectories evaluated at $N_{\text{val}} = 1,000$ points in the input domain; each pair of validation sets shares the same points in the input domain). We then compute the error in Eq. (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 Fig. 10a 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 Fig. 10b. We add the corresponding value of the normalized Wasserstein distance between simulator and emulator prediction as a small colored circle to the plot in Fig. 10a.

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 Fig. 10b).

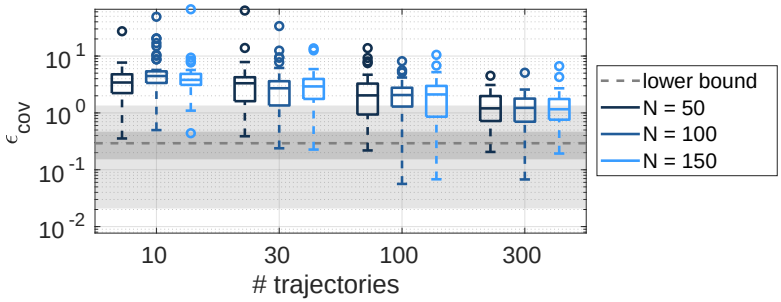
In addition to marginal predictions, our stochastic emulator can also emulate the covariance function, using Eq. (9). Since this equation relies only on the KL eigenfunctions, not on the KL-RV, the choice of inference method in Step 7 of our algorithm in Section 3 does not affect



(a) *Convergence of* ϵ_{marg}



(b) *Marginal at* $\boldsymbol{x}_{val}^{(1)} = (-\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$



(c) *Convergence of* ϵ_{cov}

Figure 10: *Convergence of* ϵ_{marg} *and* ϵ_{cov} *(Eqs. (20) and (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 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* 100×2 *independent MC samples of size* $R_{val} = 10,000$ *generated by the true stochastic simulator. The colored points in Fig. 10a correspond to the results for a single replication and validation point as shown in Fig. 10b and help assess the meaning of the numerical error measures.*

these results. In Fig. 10c we display the convergence of ϵ_{cov} from Eq. (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 Fig. 10a, 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 Fig. 11 we compare the four inference options described in Step 7 of the algorithm in Section 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 ϵ_{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.

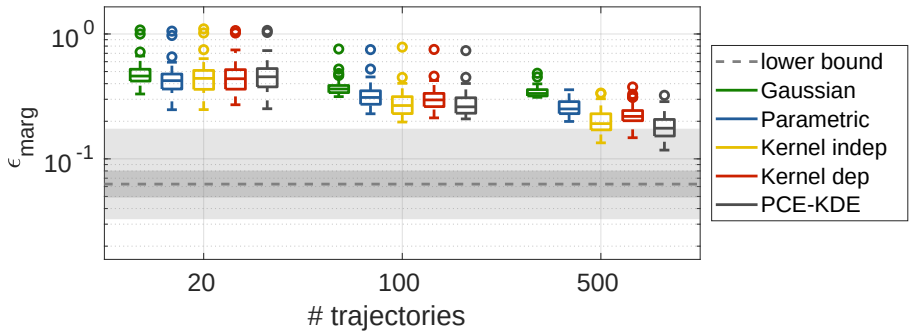


Figure 11: *Convergence of average normalized Wasserstein distance. Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with the method PCE-KDE described in Remark 1. $N = 100$ and $p = 14$. Errors are computed based on $N_{val} = 1,000$, $R_{val} = 10,000$, for 50 replications. The gray areas and dashed line have the same meaning as in Fig. 10.*

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

$$B(R_w, H_u, K_w, R, T_u, T_l, H_l, L) = \frac{2\pi T_u (H_u - H_l)}{\ln(R/R_w)\left(1 + \frac{2LT_u}{\ln(R/R_w) R_w^2 K_w} + \frac{T_u}{T_l}\right)}. \tag{23}$$

Its input random variables and their distributions are provided in Table 2.

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 5 below.

Table 2: 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*

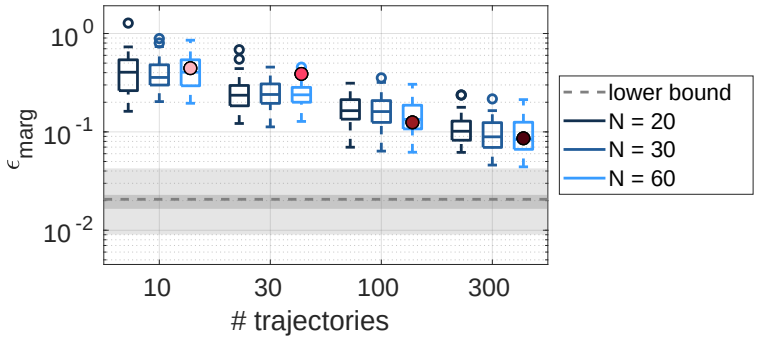
As before, we analyze the global convergence of the marginal and covariance approximation for increasing numbers of input samples and trajectories, and we compare different methods for inferring the distribution of the KL-RV as described in Step 7 of our algorithm (Section 3). For the detailed explanation of the error measures, the setup of the convergence study, and the interpretation of the plots, see Sections 4 and 4.1.

In Fig. 12a, we see that our method converges in both global error metrics (ϵ_{marg} and ϵ_{cov}) towards the rough empirical lower bound indicated by the gray area and dashed line. For ϵ_{marg}, 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.

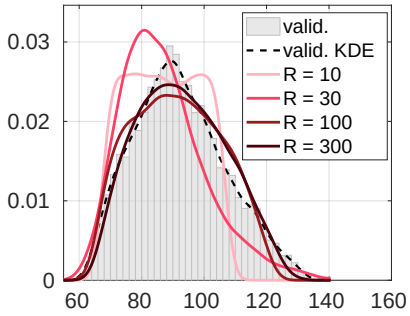
The convergence of the emulated covariance function is displayed in Fig. 12c. As expected, the global error become smaller with an increasing number of trajectories, and approaches the lower bound representing the variability due to the error being computed from samples. Again, the difference in the 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. (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 Fig. 13. 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 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.

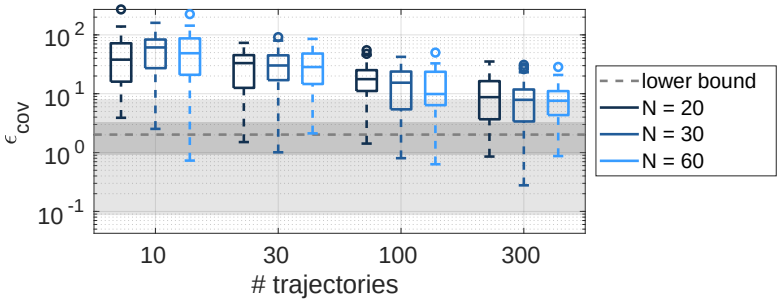
Interestingly, there is no significant difference between the performance of KDE with and without the independence assumption. Here the magnitude of the eigenvalues might offer an explanation: with more than two orders of magnitude difference between λ_1 and λ_2, the dependence between the two random variables ξ_1 and ξ_2 does not influence the resulting predictions as much as the correct identification of the marginal shape of the first KL-RV ξ_1.



(a) *Convergence of* ϵ_{marg}



(b) *Marginal of* $Y_{\boldsymbol{x}_{val}^{(1)}}$



(c) *Convergence of* ϵ_{cov}

Figure 12: *Convergence of the* ϵ_{marg} *and* ϵ_{cov} *(Eqs. (20) and (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* 100×2 *independent MC samples of size* $R_{val} = 10,000$ *generated by the true stochastic simulator. The colored points in Fig. 12a correspond to the results for a single replication and validation point as shown in Fig. 12b and help assess the meaning of the numerical error measures.*

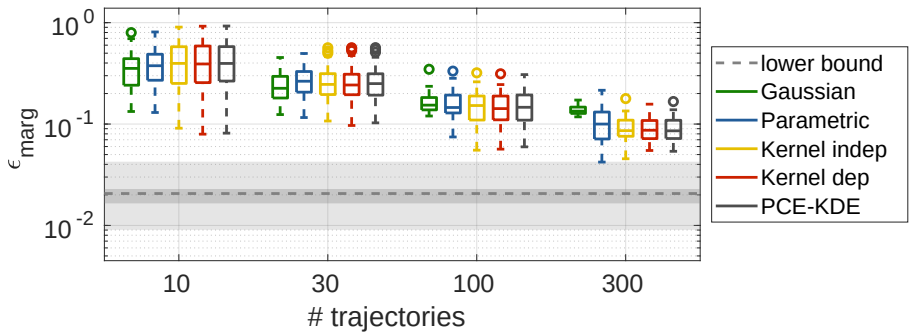


Figure 13: *Convergence of ϵ_{marg}. Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with PCE-KDE described in Remark 1. $N = 60$ and $p = 6$. Errors are computed based on $N_{val} = 1,000$, $R_{val} = 10,000$, and 50 replications. The gray areas and dashed line have the same meaning as in Fig. 12.*

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 ν_t modeled as stochastic process:

$$\mathrm{d}U_t = \mu U_t \mathrm{d}t + \sqrt{\nu_t} U_t \mathrm{d}W_t^{(1)}, \tag{24}$$

$$\mathrm{d}\nu_t = \kappa(\theta - \nu_t)\mathrm{d}t + \sigma\sqrt{\nu_t}\mathrm{d}W_t^{(2)} \tag{25}$$

with two Wiener processes $W_t^{(1)}$ and $W_t^{(2)}$ with correlation coefficient ρ. This model has six uniformly distributed parameters $\boldsymbol{X} = (\mu, \kappa, \theta, \sigma, \rho, \nu_0)$ detailed in Table 3, the bounds of which are calibrated from real data as described in Zhu and Sudret (2021b). The quantity of interest is

$$Y_{\boldsymbol{x}} = U_1(\boldsymbol{X} = \boldsymbol{x}), \tag{26}$$

i.e., the stock price after 1 year. As proposed by Zhu and Sudret (2021b), we set $U_0 = 1$ and use the Euler-Maruyama method to integrate the system of stochastic differential equations (SDEs) and replace ν_t by $\max(\nu_t, 0)$ to avoid negative values of ν_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. (24) and (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 3: Parameters and their distributions for the Heston SDE model.

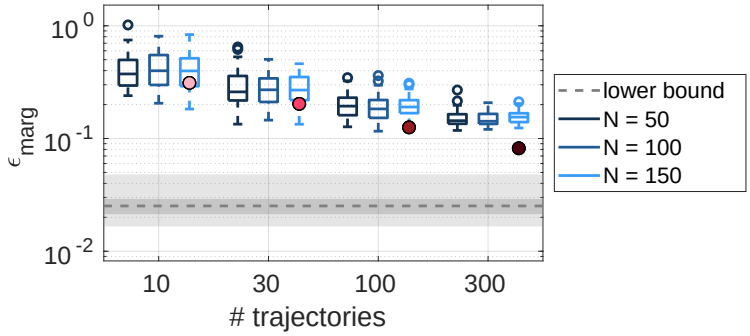
Variable	Distribution	Description
μ	$\mathcal{U}([0, 0.1])$	Expected return rate
κ	$\mathcal{U}([0.3, 2])$	Mean reversion speed of the volatility
θ	$\mathcal{U}([0.02, 0.07])$	Long term mean of the volatility
σ	$\mathcal{U}([0.2, 0.4])$	Volatility of the volatility
ρ	$\mathcal{U}([-1, -0.5])$	Correlation coefficient between $\mathrm{d}W_t^{(1)}$ and $\mathrm{d}W_t^{(2)}$
ν_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 Fig. 14a. 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

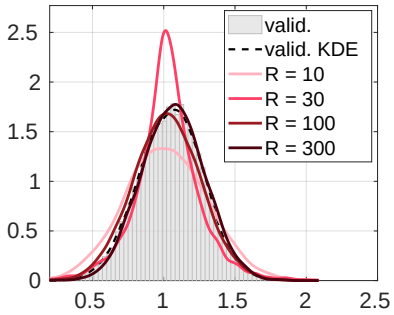
inaccuracy due to the inference of the KL-RV. This, in turn, could be because the PCE approximation is not yet sufficiently accurate (note that the relative validation error is in the order of 10^{-2} for all ED sizes.) Even for the largest number of trajectories, the averaged and normalized Wasserstein 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 Fig. 14a (quantiles and median, respectively). Comparing the boxplots to the colored points corresponding to the marginal estimates illustrated in Fig. 14b, 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 Fig. 14c. 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.

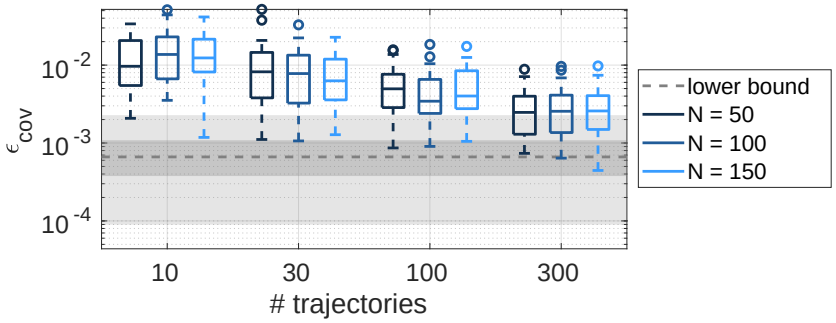
Fig. 15 shows the comparison between the different methods for KL-RV inference (Step 7 of the algorithm in Section 3) as described in Section 4.1.4. All four methods perform comparably. The independent standard Gaussian approximation performs slightly better than the other methods in the case of few 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 3.



(a) *Convergence of* ϵ_{marg}



(b) *Marginal of* $Y_{\boldsymbol{x}_{val}^{(1)}}$



(c) *Convergence of* ϵ_{cov}

Figure 14: *Convergence of* ϵ_{marg} *and* ϵ_{cov} *(Eqs. (20) and (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* 100×2 *independent MC samples of size* $R_{val} = 10,000$ *generated by the true stochastic simulator. The colored points in Fig. 14a correspond to the results for a single replication and validation point as shown in Fig. 14b and help assess the meaning of the numerical error measures.*

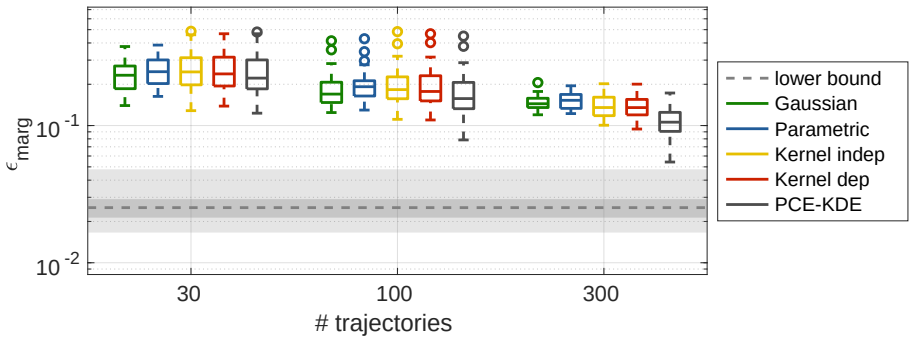


Figure 15: *Convergence of marginals (Wasserstein distance). Comparison of the four different methods for inferring the joint distribution of KL-RV described in Step 7 with PCE-KDE described in Remark 1.* $N = 150$ *and* $p = 5$. *Errors are computed based on* $N_{val} = 1,000$, $R_{val} = 10,000$, *and 50 replications. The gray areas and dashed line have the same meaning as in Fig. 14.*

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. (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 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

$$\mathcal{M}(\boldsymbol{Z}) = m_0 + \sum_{1 \le i \le d} m_i(Z_i) + \sum_{1 \le i < j \le d} m_{i,j}(Z_i, Z_j) + \cdots + m_{1,\dots,d}(Z_1, \dots, Z_d) \tag{27}$$

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. (27) according to whether they involve only input variables, only latent variables, or some variables from both groups. This results in the following decomposition:

$$\mathcal{M}(\boldsymbol{X}, \boldsymbol{\Xi}) = m_0 + \mathcal{M}_1(\boldsymbol{X}) + \mathcal{M}_2(\boldsymbol{\Xi}) + \mathcal{M}_{12}(\boldsymbol{X}, \boldsymbol{\Xi}) \tag{28}$$

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. (28) contains all interaction terms from Eq. (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 $\boldsymbol{\Xi}$:

$$\mathcal{M}(\boldsymbol{X}, \boldsymbol{\Xi}) \approx m_0 + m_1(\boldsymbol{X}) + m_2(\boldsymbol{\Xi}). \tag{29}$$

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. (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. (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 4, e.g., for the case of the borehole model with hidden variables. The deterministic borehole model defined in Eq. (23) has relatively low interaction: despite its nonlinearity, under the input uncertainties in Table 2 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 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., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. *Ins. Math. Eco. 44*(2), 182–198.

Abraham, S., P. Tsirikoglou, J. Miranda, C. Lacor, F. Contino, and G. Ghorbaniasl (2018). Spectral representation of stochastic field data using sparse polynomial chaos expansions. *J. Comput. Phys. 367*, 109–120.

Azzi, S., Y. Huang, B. Sudret, and J. Wiart (2019). Surrogate modeling of stochastic functions – application to computational electromagnetic dosimetry. *Int. J. Uncertainty Quantification 9*(4), 351–363.

Bedford, T. and R. M. Cooke (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 J. Ramsay (1986). Principal components analysis of sampled functions. *Psychometrika 51*(2), 285–311.

Betz, W., I. Papaioannou, and D. Straub (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 B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys 230*, 2345–2367.

Bowman, A. W. and A. Azzalini (1997). *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, Volume 18. OUP Oxford.

Buhmann, M. D. (2000, January). Radial basis functions. *Acta Numerica 9*, 1–38.

Dai, H., Z. Zheng, and H. Ma (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., R. Ghanem, and S. Finette (2009). Polynomial chaos representation of spatio-temporal random fields from experimental measurements. *J. Comput. Phys. 228*(23), 8726–8751.

Desceliers, C., R. Ghanem, and C. Soize (2006). Maximum likelihood estimation of stochastic chaos representations from experimental data. *Int. J. Numer. Meth. Engng. 66*, 978–1001.

Doostan, A., R. Ghanem, and J. Red-Horse (2007). Stochastic model reduction for chaos representations. *Comput. Methods Appl. Mech. Engrg. 196*(37-40), 3951–3966.

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

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

Goan, E. and C. Fookes (2020). Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pp. 45–87. Springer.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (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., J. Racine, and Q. Li (2004). Cross-validation and the estimation of conditional probability densities. *J. Am. Stat. Assoc. 99*(468), 1015–1026.

Harper, W. V. and S. K. Gupta (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., M. Diez, and L. Morino (2006). An extended Karhunen-Loève decomposition for modal identification of inhomogeneous structures. *J. Vib. Acoust. 128*(3), 357–365.

Iooss, B. and M. Ribatet (2009). Global sensitivity analysis of computer models with functional inputs. *Reliab. Eng. Syst. Saf. 94*, 1194–1204.

Joe, H. (2014). *Dependence Modeling with Copulas* (1 ed.). Chapman and Hall/CRC.

Karhunen, K. (1946). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fenn. A1 A1*(37).

Kingma, D. P. and M. Welling (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 R. Cooke (2005). Distribution-free continuous Bayesian belief nets. In *Modern Statistical and Mathematical Methods in Reliability*, pp. 309–322. World Scientific Publishing.

Lataniotis, C., E. Torre, S. Marelli, and B. Sudret (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* (4th ed.), Volume 2. Springer Verlag, New-York.

Lüthen, N., S. Marelli, and B. Sudret (2021). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA J. Unc. Quant. 9*(2), 593–649.

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

Lüthen, N., O. Roustant, F. Gamboa, B. Iooss, S. Marelli, and B. Sudret (2022). 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 B. Sudret (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)*, pp. 2554–2563.

Marrel, A., B. Iooss, S. Da Veiga, and M. Ribatet (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat. Comput. 22*, 833–847.

Moutoussamy, V., S. Nanty, and B. Pauwels (2015). Emulators for stochastic simulation codes. *ESAIM: Math. Model. Num. 48*, 116–155.

Nagel, J., J. Rieckermann, and B. Sudret (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., O. P. Le Maître, and O. M. Knio (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* (2nd ed.), Volume 139 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

Pearce, M., M. Poloczek, and J. Branke (2022). Bayesian optimization allowing for common random numbers. *Oper. Res. 0*(0), 1–16.

Perrin, T., O. Roustant, J. Rohmer, O. Alata, J. Naulin, D. Idier, R. Pedreros, D. Moncoulon, and P. Tinard (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliab. Eng. Syst. Saf. 211*, 107522.

Poirion, F. and I. Zentner (2013). Non-Gaussian non-stationary models for natural hazard modeling. *Appl. Math. Model. 37*(8), 5938–5950.

Poirion, F. and I. Zentner (2014). Stochastic model construction of observed random phenomena. *Prob. Eng. Mech. 36*, 63–71.

Rahman, S. (2020, January). A spline chaos expansion. *SIAM/ASA Journal on Uncertainty Quantification 8*(1), 27–57.

Raisee, M., D. Kumar, and C. Lacor (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 B. Silverman (2005). *Functional data analysis* (2nd ed.). Springer Series in Statistics. Springer Science and Business Media.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian processes for machine learning* (Internet ed.). Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Stat. Sci. 4*, 409–435.

Sakamoto, S. and R. Ghanem (2002). Simulation of multi-dimensional non-Gaussian non-stationary random fields. *Prob. Eng. Mech. 17*(2), 167–176.

Schwab, C. and R. Todor (2006). Karhunen–Loève approximation of random fields by generalized fast multipole methods. *J. Comput. Phys. 217*(1), 100–122.

Shields, M., G. Deodatis, and P. Bocchini (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 G. Deodatis (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 B. Schölkopf (2004). A tutorial on support vector regression. *Stat. Comput. 14*, 199–222.

Sobol, I. and A. Gresham (1995). On an alternative global sensitivity estimator. In *Proceedings of SAMO 1995*, Belgirate, pp. 40–42.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Modeling & Comp. Exp. 1*(4), 407–414.

Torossian, L., V. Picheny, R. Faivre, and A. Garivier (2020). A review on quantile regression for stochastic computer experiments. *Reliab. Eng. Syst. Saf. 201*, 106858.

Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019a). Data-driven polynomial chaos expansion for machine learning regression. *J. Comput. Phys 388*, 601–623.

Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019b). A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas. *Prob. Eng. Mech. 55*, 1–16.

Torre, E., S. Marelli, and B. Sudret (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 M. C. Jones (1995). *Kernel smoothing*, Volume 60 of *Monographs on Statistics and Applied Probability*. London New York: Chapman and Hall, Boca Raton.

Xiu, D. (2010). *Numerical methods for stochastic computations – A spectral method approach*. Princeton University press.

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

Yamazaki, F. and M. Shinozuka (1988). Digital generation of non-Gaussian stochastic fields. *J. Eng. Mech. 114*(7), 1183–1197.

Zhang, J. and B. Ellingwood (1994). Orthogonal series expansions of random fields in reliability analysis. *J. Eng. Mech. 120*(12), 2660–2677.

Zhu, X. and B. Sudret (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 B. Sudret (2021a). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA J. Unc. Quant. 9*(4), 1345–1380.

Zhu, X. and B. Sudret (2021b). Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliab. Eng. Sys. Safety 214*, 107815.

Zhu, X. and B. Sudret (2022). Stochastic polynomial chaos expansions to emulate stochastic simulators. *Int. J. Uncertainty Quantification*. Accepted.

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 Appendix 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 Appendix A.2, we show that the realizations of the random KLE coefficients can be determined analytically.

Let for $r = 1, \ldots, R$

$$\tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \tilde{a}_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{30}$$

be the centered PCE trajectory computed from discrete evaluations of the trajectory $\mathcal{T}_r$ (Eq. (13)). The sample covariance function is a polynomial given by

$$\hat{c}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{R-1} \sum_{r=1}^{R} \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}) \tilde{\mathcal{M}}_r^{\mathrm{PCE}}(\boldsymbol{x}'). \tag{31}$$

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:

$$\langle \hat{c}(\boldsymbol{x}, \cdot), \phi_i(\cdot) \rangle_{L^2_{f_{\boldsymbol{X}}}(\mathcal{D})} = \int_{\mathcal{D}} \hat{c}(\boldsymbol{x}, \boldsymbol{x}') \phi_i(\boldsymbol{x}') f_{\boldsymbol{X}}(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' = \lambda_i \phi_i(\boldsymbol{x}). \tag{32}$$

Since $\hat{c}$ is polynomial, also the eigenfunctions will be polynomials and can be represented in terms of the PCE basis as follows:

$$\phi_i(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} b_{\boldsymbol{\alpha}}^{(i)} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}). \tag{33}$$

Solving Eq. (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

$$\sum_{\beta\in\mathcal{A}}\left(\frac{1}{R-1}\sum_{r=1}^{R}\tilde{a}^r_\beta\underbrace{\left(\sum_{\alpha\in\mathcal{A}}\tilde{a}^r_\alpha b_\alpha\right)}_{=(\tilde{\boldsymbol{a}}^r)^T\boldsymbol{b}\in\mathbb{R}}\right)\psi_\beta(x)\overset{!}{=}\lambda\sum_{\beta\in\mathcal{A}}b_\beta\psi_\beta(x).$$

Because the PCE basis functions ψ_β are of different orders, we can rewrite this into matrix form:

$$\frac{1}{R-1}\sum_{r=1}^{R}\begin{pmatrix}\tilde{a}^r_{\beta_1}(\tilde{\boldsymbol{a}}^r)^T\\ \tilde{a}^r_{\beta_2}(\tilde{\boldsymbol{a}}^r)^T\\ \vdots\\ \tilde{a}^r_{\beta_P}(\tilde{\boldsymbol{a}}^r)^T\end{pmatrix}\boldsymbol{b}=\left(\frac{1}{R-1}\sum_{r=1}^{R}\tilde{\boldsymbol{a}}^r\ (\tilde{\boldsymbol{a}}^r)^T\right)\boldsymbol{b}=\left(\frac{1}{R-1}\ \tilde{\boldsymbol{a}}\ \tilde{\boldsymbol{a}}^T\right)\boldsymbol{b}\overset{!}{=}\lambda\boldsymbol{b},\tag{34}$$

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. (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: $\phi_i(\boldsymbol{x})=\sum_{\boldsymbol{\alpha}\in\mathcal{A}}b^{(i)}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$. 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.

A.2 Analytical computation of the realizations of the KL-RV

Let λ_i be an eigenvalue and

$$\phi_i(x)=\sum_{\boldsymbol{\alpha}\in\mathcal{A}}b^{(i)}_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\tag{35}$$

the associated eigenfunction expressed in the PCE basis. The projection of the PCE trajectories onto the KLE basis is given by

$$\begin{aligned}\xi^r_i&=\frac{1}{\sqrt{\lambda_i}}\int_{\mathcal{D}}\tilde{\mathcal{M}}^{\mathrm{PCE}}_r(\boldsymbol{x})\phi_i(x)f_{\boldsymbol{X}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\\&=\frac{1}{\sqrt{\lambda_i}}\sum_{\boldsymbol{\alpha}\in\mathcal{A}}\tilde{a}^r_{\boldsymbol{\alpha}}b^{(i)}_{\boldsymbol{\alpha}}=\frac{1}{\sqrt{\lambda_i}}(\tilde{\boldsymbol{a}}^r)^T\boldsymbol{b}^{(i)}\ \in\mathbb{R}.\end{aligned}$$

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

$$\boldsymbol{\Xi}=\left(\mathrm{diag}\left(\frac{1}{\sqrt{\lambda_1}},\dots,\frac{1}{\sqrt{\lambda_M}}\right)\boldsymbol{b}\right)^T\tilde{\boldsymbol{a}}.\tag{36}$$