ETH zürich

STOCHASTIC POLYNOMIAL CHAOS EXPANSIONS TO EMULATE STOCHASTIC SIMULATORS

X. Zhu 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-2022-001B
Arxiv Ref.:	https://arxiv.org/abs/2202.03344 [stat.CO] [stat.ME]
DOI:	-
Date submitted:	May 12, 2022
Date accepted:	-

Stochastic polynomial chaos expansions to emulate stochastic simulators

Xujia Zhu*[1] and Bruno Sudret†[1]

[1] *Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland*

May 12, 2022

Abstract

In the context of uncertainty quantification, computational models are required to be repeatedly evaluated. This task is intractable for costly numerical models. Such a problem turns out to be even more severe for stochastic simulators, the output of which is a random variable for a given set of input parameters. To alleviate the computational burden, surrogate models are usually constructed and evaluated instead. However, due to the random nature of the model response, classical surrogate models cannot be applied directly to the emulation of stochastic simulators. To efficiently represent the probability distribution of the model output for any given input values, we develop a new stochastic surrogate model called *stochastic polynomial chaos expansions*. To this aim, we introduce a latent variable and an additional noise variable, on top of the well-defined input variables, to reproduce the stochasticity. As a result, for a given set of input parameters, the model output is given by a function of the latent variable with an additive noise, thus a random variable. As the latent variable is purely artificial and does not have physical meanings, conventional methods (pseudo-spectral projections, collocation, regression, etc.) cannot be used to build such a model. In this paper, we propose an adaptive algorithm which does not require repeated runs of the simulator for the same input parameters. The performance of the proposed method is compared with the generalized lambda model and a state-of-the-art kernel estimator on two case studies in mathematical finance and epidemiology and on an analytical example whose response distribution is bimodal. The results show that the proposed method is able to accurately represent general response distributions, i.e., not only normal or unimodal ones. In terms of accuracy, it generally outperforms both the generalized lambda model and the kernel density estimator.

*zhu@ibk.baug.ethz.ch
†sudret@ethz.ch

1 Introduction

In modern engineering, computational models, a.k.a. simulators, are commonly used to simulate different operational scenarios of complex systems *in silico*. These models help engineers assess the reliability, control the risk, and optimize the system components in the design phase. Conventional simulators are usually deterministic: a given set of input parameters has a unique corresponding model response. In other words, repeated model evaluations with the same input values will always give identical results. In contrast, stochastic simulators return different outcomes of the model response when run twice with the same input parameters.

Stochastic simulators are widely used in engineering and applied science. The intrinsic stochasticity typically represents some uncontrollable effect in the system [1, 2]. For example, in mathematical finance, Brownian motions are commonly introduced to represent stochastic effects and volatility of the stock market [1]. In epidemic simulations, additional random variables on top of the well-defined characteristic values of the population are used to simulate the stochastic spread of a disease [2].

Mathematically, a stochastic simulator can be viewed as a function

$$\begin{aligned} \mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\to \mathbb{R} \\ (\boldsymbol{x}, \omega) &\mapsto \mathcal{M}_s(\boldsymbol{x}, \omega), \end{aligned} \tag{1}$$

where $\mathcal{D}_{\boldsymbol{X}}$ is the domain of the input parameters, and Ω denotes the probability space that represents the internal stochasticity. The latter is due to some latent random variables $\boldsymbol{\Xi}(\omega)$ which are not explicitly considered as a part of the input variables. The stochastic simulator can then be considered as a deterministic function of the input vector $\boldsymbol{x}$ and the latent variables $\boldsymbol{\Xi}$. However, it is assumed that one can only control $\boldsymbol{x}$ but not $\boldsymbol{\Xi}$ when evaluating the model. Hence, when the value of $\boldsymbol{x}$ is fixed but $\boldsymbol{\Xi}$ is generated randomly following the underlying probability distribution, the output remains random.

In practice, each model evaluation for a fixed vector of input parameters $\boldsymbol{x}_0$ uses a particular realization of the latent variables, i.e., a particular $\omega_0 \in \Omega$ that is usually controlled by the random seed. Thus, it provides only one realization of the output random variable. In order to fully characterize the associated distribution of $\mathcal{M}_s(\boldsymbol{x}_0, \cdot)$, it is necessary to repeatedly run the stochastic simulator with the same input parameters $\boldsymbol{x}_0$. The various output values obtained by this procedure are called *replications* in the sequel.

In the context of uncertainty quantification or optimization, various input values should be investigated. To this aim, multiple runs of the simulator are needed for many different inputs and for many replications. This becomes impracticable for high-fidelity costly numerical models. In this context, surrogate models have received tremendous attention in the past two decades. A surrogate model is a proxy of the original model constructed from a limited number of model runs. However, standard surrogate models such as polynomial chaos expansions [3] and Gaussian

processes [4] that have been successfully developed for deterministic simulators are not directly applicable to emulating stochastic simulators due to the random nature of the latter.

In the past decade, large efforts have been dedicated to estimating some summary quantities of the response distribution which are deterministic functions of the input.

For the mean and variance of the response distribution, Ankenman et al. [5] proposed using replications to estimate the mean and variance for various input values. The mean function is represented by a Gaussian process, for which the variance estimated from the replications is cast as a heteroskedastic effect. Marrel et al. [6] modeled both the mean and variance by Gaussian processes. The estimation procedure is similar to the feasible generalized least-squares [7] that consists in alternating between fitting the mean from the data and the variance from the residuals. This approach does not require replications. Binois et al. [8] proposed jointly optimizing the likelihood to represent the mean and variance by Gaussian processes, which is mainly designed for data with replications.

To estimate the quantiles of the response distribution, Koenker and Bassett [9] proposed optimizing the *check function*, which established the quantile regression method. Plumlee and Tuo [10] suggested estimating the quantiles by performing replications and building a Gaussian process from the estimated quantiles. The reader is referred to Torossian et al. [11] for a detailed review.

The methods listed above produce only targeted summary quantities. However, far less literature has been devoted to the emulation of the entire probability distribution function of the response random variable for a given input. Three types of methods can be found in the literature.

Moutoussamy et al. [12] proposed using replications to characterize the response distribution for different input values. Then, the fitted distributions (based on replications) for the discrete input values can be extended to the entire input space by parametric or nonparametric techniques. Since this approach capitalizes on replications for local inference, it is necessary to generate many replications to obtain an accurate surrogate [13], i.e., in the order of $10^3 - 10^4$ [12].

In the second approach, a stochastic simulator is considered as a random field indexed by the input variables [14, 15]. When fixing the internal stochasticity ω in Eq. (1), the stochastic simulator is a mere deterministic function of $\boldsymbol{x}$, called *a trajectory*. This function can be emulated by standard surrogate methods. Collecting different trajectories, one can approximate the underlying random field using Karhunen–Loève expansions. Therefore, it is necessary to fix the internal randomness to apply this approach, which is practically achieved by controlling the random seed.

The third type of methods is referred to as the statistical approach and does not require replications or manipulating the random seed. If the response distribution belongs to the exponential family, generalized linear models [16] and generalized additive models [17] can be efficiently applied. For arbitrary types of response distributions, nonparametric estimators developed in statistics can be applied, namely kernel density estimators [18, 19] and projection estimators [20]. However, nonparametric estimators are known to suffer from the *curse of dimensionality*, which indicates that the necessary amount of data increases drastically with increasing input dimensionality. To

balance between very restrictive parametric assumptions and nonparametric approaches, Zhu and Sudret [21, 22] proposed using generalized lambda distributions to approximate the response distributions. The four distribution parameters are seen as functions of the input and further represented by polynomial chaos expansions. The main limitation of this approach is that it cannot produce multimodal distributions, however.

In this paper, we develop an original approach that directly emulates the functional representation in Eq. (1). More precisely, we extend the classical polynomial chaos expansions to emulating stochastic simulators. We introduce a latent variable and a noise variable to reproduce the random behavior of the model output. We develop an adaptive method to construct such a surrogate model. This novel stochastic surrogate is parametric and shown to be not limited to unimodal distributions.

The remainder of the paper is organized as follows. In Section 2, we first review the standard polynomial chaos representations. In Section 3, we present a novel formulation named *stochastic polynomial chaos expansions* which is meant for stochastic simulators. In Section 4, we present the algorithms to adaptively build such a surrogate from data without the need for replications. We illustrate the performance of the proposed method on a complex analytical example and on case studies from mathematical finance and epidemiology in Section 5. Finally, we conclude the main findings of the paper and provide outlooks for future research in Section 6.

2 Reminder on polynomial chaos expansions

Polynomial chaos expansions (PCEs) have been widely used in the last two decades to emulate the response of deterministic simulators in many fields of applied science and engineering. Consider a deterministic model $\mathcal{M}_d$ which is a function that maps the input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)^T \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$ to the scalar output $y = \mathcal{M}_d(\boldsymbol{x}) \in \mathbb{R}$. In the context of uncertainty quantification, the input vector $\boldsymbol{x}$ is affected by uncertainties and thus modeled by a random vector $\boldsymbol{X}$ with prescribed joint probability density function (PDF) denoted by $f_{\boldsymbol{X}}$. In the sequel, we focus on the case where the input parameters are independent for simplicity. Therefore, the joint PDF is expressed by

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{j=1}^{M} f_{X_j}(x_j), \tag{2}$$

where f_{X_j} is the marginal PDF of the input random variable X_j. Note that in the case where the input vector $\boldsymbol{X}$ has dependent components, it is always possible to transform them into independent ones using the Nataf or Rosenblatt transform [23–25].

Because of the randomness in the input, the model response $Y = \mathcal{M}_d(\boldsymbol{X})$ becomes a random variable. Provided that Y has a finite variance, i.e., $\mathrm{Var}\,[Y] < +\infty$, the function $\mathcal{M}_d$ belongs to

the Hilbert space $\mathcal{H}$ of square-integrable functions with respect to the inner product

$$\langle u, v\rangle_{\mathcal{H}} \overset{\text{def}}{=} \mathbb{E}\left[u(\boldsymbol{X})v(\boldsymbol{X})\right] = \int_{\mathcal{D}_{\boldsymbol{X}}} u(\boldsymbol{x})v(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{3}$$

Under certain conditions on the joint PDF $f_{\boldsymbol{X}}$ [26], the Hilbert space $\mathcal{H}$ possesses a polynomial basis. As a result, $\mathcal{M}_d$ can be represented by an orthogonal series expansion

$$\mathcal{M}_d(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{4}$$

where $c_{\boldsymbol{\alpha}}$ is the coefficient associated with the basis function $\psi_{\boldsymbol{\alpha}}$ that is defined by the multi-index $\boldsymbol{\alpha}$. More precisely, the multivariate basis function $\psi_{\boldsymbol{\alpha}}$ is given by a tensor product of univariante polynomials

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j), \tag{5}$$

where α_j indicates the degree of $\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ in its j-th component x_j, and $\left\{\phi_k^{(j)} : k \in \mathbb{N}\right\}$ is the orthogonal polynomial basis with respect to the marginal distribution f_{X_j} of X_j, which satisfies

$$\mathbb{E}\left[\phi_k^{(j)}(X_j)\,\phi_l^{(j)}(X_j)\right] = \delta_{kl}. \tag{6}$$

In the equation above, the Kronecker symbol δ_{kl} is such that $\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise.

Following Eq. (5), the multivariate polynomial basis is defined from univariate orthogonal polynomials that depend on the corresponding marginal distribution. For uniform, normal, gamma and beta distributions, the associated orthogonal polynomial families are known analytically [27]. For arbitrary marginal distributions, such a basis can be iteratively computed by the *Stieltjes procedure* [28].

The spectral representation in Eq. (4) involves an infinite sum of terms. In practice, the series needs to be truncated to a finite sum. The standard truncation scheme is defined by selecting all the polynomials whose total degree is small than a given value p, i.e., $\mathcal{A}^{p,M} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M, \sum_{j=1}^{M} \alpha_j \leq p\right\}$. However, this will provide a large number of terms for big values of p and M. A more flexible scheme is the hyperbolic (q-norm) truncation scheme [29]:

$$\mathcal{A}^{p,q,M} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M, \|\boldsymbol{\alpha}\|_q \leq p\right\}, \tag{7}$$

where p is the maximum polynomial degree, and $q \in (0, 1]$ defines the quasi-norm $\|\boldsymbol{\alpha}\|_q = \left(\sum_{j=1}^{M} |\alpha_j|^q\right)^{1/q}$. This truncation scheme allows excluding high-order interactions among the input variables but keeps univariate effects up to degree p. Note that with $q = 1$, we recover the full basis of total degree less than p.

To estimate the coefficients in Eq. (4), one popular approach relies on minimizing the mean-

squared error between the model response and the surrogate model. The basic method applies ordinary least-squares (OLS) with a given set of basis (e.g., defined by a truncation scheme) [30]. In this approach, the model is evaluated on a number of points called the *experimental design* $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\right\}$. The associated model responses are gathered into $\boldsymbol{y} = \left\{y^{(1)}, \dots, y^{(N)}\right\}$ with $y^{(i)} = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right)$. The basis functions (and thus the coefficients) can be arranged by ordering the multi-indices $\{\boldsymbol{\alpha}_j\}_{j=1}^{P}$. The regression matrix $\boldsymbol{\Psi}$ is defined by $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}\left(\boldsymbol{x}^{(i)}\right)$. By minimizing the mean-squared error between the original model and the surrogate on the experimental design, the OLS estimator is given by

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c}\|_2^2 \tag{8}$$

With increasing polynomial degree or input dimension, the number of coefficients increases drastically. As a consequence, a large number of models runs are necessary to guarantee a good accuracy, which becomes intractable for costly simulators. To solve this problem, Blatmann and Sudret [29], Doostan and Owhadi [31], Babacan et al. [32] developed methods to build sparse PCEs by only selecting the most influential polynomials. The reader is referred to the review papers by Lüthen et al. [33, 34] for more details.

3 Stochastic polynomial chaos expansions

3.1 Introduction

Let us now come back to stochastic simulators. It would be desirable to have a spectral expansion such as Eq. (4) for stochastic simulators. Indeed, the standard PCE has numerous features such as close-to-zero-cost model evaluations, and clear interpretation of the coefficients in terms of sensitivity analysis [35]. However, because the spectral expansion in Eq. (4) is a deterministic function of the input parameters, it cannot be directly used to emulate stochastic simulators.

Considering the randomness in the input variables, the output of a stochastic simulator is a random variable. The randomness of the latter comes from both the intrinsic stochasticity and the uncertain inputs. When fixing the input parameters, the model response remains random. For the purpose of clarity, we denote by $Y_{\boldsymbol{x}}$ the random model response for the input parameters $\boldsymbol{x}$ and by Y the model output containing all the uncertainties: following Eq. (1), we have

$$Y_{\boldsymbol{x}} \overset{\text{def}}{=} \mathcal{M}_s(\boldsymbol{x}, \omega), \quad Y \overset{\text{def}}{=} \mathcal{M}_s(\boldsymbol{X}(\omega), \omega). \tag{9}$$

From a probabilistic perspective, $Y_{\boldsymbol{x}}$ is equivalent to the conditional random variable $Y \mid \boldsymbol{X} = \boldsymbol{x}$. Let $F_{Y|\boldsymbol{X}}\left(y \,|\, \boldsymbol{x}\right)$ denote the associated cumulative distribution function (CDF). By using the probability integral transform, we can transform *any* continuous random variable Z to the desired

distribution, that is

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{=} F^{-1}_{Y|\boldsymbol{X}}\left(F_Z\left(Z\right)|\,\boldsymbol{x}\right) \tag{10}$$

where F_Z is the CDF of Z. The equality in Eq. (10) is to be understood *in distribution*, meaning that two random variables on the left- and right-hand side follow the same distribution. In Eq. (10), the right-hand side is a deterministic function of both $\boldsymbol{x}$ and z. As a result, assuming that Y has a finite variance, we can represent this function using a PCE in the $(\boldsymbol{X}, Z)$ space, that is,

$$F^{-1}_{Y|\boldsymbol{X}}\left(F_Z\left(Z\right)|\,\boldsymbol{X}\right) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^{M+1}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{X}, Z\right). \tag{11}$$

For a given vector of input parameters $\boldsymbol{x}$, the expansion is a function of the artificial latent variable Z, thus a random variable

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{=} \sum_{\boldsymbol{\alpha}\in\mathbb{N}^{M+1}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x}, Z\right). \tag{12}$$

Then, we apply a truncation scheme $\mathcal{A}$ (e.g., Eq. (7)) to reduce Eq. (12) to a finite sum

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{\approx} \tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}\left(\boldsymbol{x}, Z\right). \tag{13}$$

Even though Eq. (13) is derived from Eq. (11), it is more general. Equation (10) offers one way to represent the response distribution by a transform of a latent variable. But many other transforms can achieve the same goal. For example, using $Z \sim \mathcal{N}(0,1)$, both $\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})Z$ and $\mu(\boldsymbol{x}) - \sigma(\boldsymbol{x})Z$ can represent the stochastic simulator defined by $Y_{\boldsymbol{x}} \sim \mathcal{N}(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$. Because we are interested in the response distribution, Eq. (13) only requires that the polynomial transform of the latent variable produces a distribution that is close to the response distribution, but the transform does not need to follow Eq. (11) exactly. Note that the latent variable Z is only introduced to reproduce the stochasticity, but it does not allow us to represent the detailed data generating process of the simulator though. In other words, the PCE in Eq. (13) cannot emulate the response for a particular replication, yet it provides a representation of the distribution of $Y_{\boldsymbol{x}}$.

3.2 Potential issues with the formulation in Eq. (13)

Building a PCE by least-squares as presented in Section 2 requires evaluating the deterministic function to surrogate, which, in the case of stochastic simulators, is the left-hand side of Eq. (11). However, it is practically impossible to evaluate such a function, as the response distribution $F^{-1}_{Y|\boldsymbol{X}}$ is unknown. One common way to fit the latent variable model defined in Eq. (13) is maximum likelihood estimation [36, 37]. In this section, we show some potential problems associated with a standard use of this method for building Eq. (13), which calls for a novel fitting algorithm.

According to the definition in Eq. (13), $\tilde{Y}_{\boldsymbol{x}}$ is a function of Z. Denote $f_Z(z)$ the PDF of Z and $\mathcal{D}_Z$ the support of Z. Based on a change of variable [38], we can obtain the PDF of $\tilde{Y}_{\boldsymbol{x}}$, which is

denoted by $f_{\tilde{Y}_{\boldsymbol{x}}}(y;\boldsymbol{x},\boldsymbol{c})$. As a result, the (conditional) likelihood function of the coefficients $\boldsymbol{c}$ for a data point $(\boldsymbol{x}, y)$ is given by

$$l(\boldsymbol{c};\boldsymbol{x},y) = f_{\tilde{Y}_{\boldsymbol{x}}}(y;\boldsymbol{x},\boldsymbol{c}). \tag{14}$$

Now, let us consider an experimental design $\mathcal{X} = \left\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}\right\}$. The stochastic simulator is assumed to be evaluated *once* for each point $\boldsymbol{x}^{(i)}$, yielding $\boldsymbol{y} = \left\{y^{(1)},\dots,y^{(N)}\right\}$ with $y^{(i)} = \mathcal{M}_s\left(\boldsymbol{x}^{(i)},\omega^{(i)}\right)$. Note that here we do not control the random seed, so the model outcomes for different values of $\boldsymbol{x}$ are *independent*. Thus, the likelihood function can be computed by the product of $l\left(\boldsymbol{c};\boldsymbol{x}^{(i)},y^{(i)}\right)$ over the N data points. As a result, the maximum likelihood estimator is given by

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i=1}^{N} \log l\left(\boldsymbol{c};\boldsymbol{x}^{(i)},y^{(i)}\right). \tag{15}$$

Equation (15) commonly serves as a basic approach for fitting parametric statistical models (including stochastic surrogates) [16, 21, 39]. However, the likelihood function of the latent PCE defined in Eq. (13) is unbounded and can reach $+\infty$, making the maximization problem Eq. (15) ill-posed.

To illustrate the issue, let us consider a simple stochastic simulator without input variables, which gives a realization of Y upon each model evaluation. Hence, the surrogate in Eq. (13) contains only the latent variable Z, that is, $\tilde{Y} = g(Z) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(Z)$. For simplicity, let $g(z)$ be a second-degree polynomial expressed by monomials $g(z) = a_1 z^2 + a_2 z + a_3$. Note that there is a one-to-one mapping between monomials and full polynomial chaos basis, so one can map $\boldsymbol{a} = (a_1, a_2, a_3)^T$ to $\boldsymbol{c}$ through a change of basis. Using a change of variable [38], the PDF of $\tilde{Y}$ is

$$f_{\tilde{Y}}(y) = \frac{f_Z(z)}{|g'(z)|}\mathbb{1}_{g(z)}(y), \tag{16}$$

where $\mathbb{1}$ is the indicator function, and g' denotes the derivative of g. For a given y_0, certain choices of $\boldsymbol{a}$ can make any given z_0 with $f_Z(z_0) \neq 0$ satisfy $g(z_0) = y_0$ and $g'(z_0) = 0$:

$$\begin{cases} g(z_0) = y_0 \\ g'(z_0) = 0 \end{cases} \Rightarrow \begin{cases} a_1 z_0^2 + a_2 z_0 + a_3 - y_0 = 0 \\ 2a_1 z_0 + a_2 = 0 \end{cases} \Rightarrow \begin{cases} -z_0^2\, a_1^2 + a_3 - y_0 = 0 \\ a_2 = -2 z_0\, a_1 \end{cases}. \tag{17}$$

The system of equations in Eq. (17) is underdetermined for $\boldsymbol{a}$. Therefore, there are infinite combinations of the coefficients $\boldsymbol{a}$, and therefore of $\boldsymbol{c}$, such that the denominator of Eq. (16) is zero and the numerator is non-zero, which gives $f_{\tilde{Y}}(y_0) = +\infty$. Consequently, the maximum likelihood estimation will always produce a certain vector $\boldsymbol{c}$ that makes the likelihood reach $+\infty$.

As a conclusion, the surrogate ansatz of Eq. (13) can produce non-smooth conditional PDFs with singularity points where $f_{\tilde{Y}_{\boldsymbol{x}}}$ tends to infinity. Consequently, the standard maximum likelihood estimation would fail.

3.3 Formulation of stochastic polynomial chaos expansions

In the previous section, we discussed some potential problems of the model defined in Eq. (13). To regularize the optimization problem in Eq. (15) and smooth out the produced PDFs, we introduce an additive noise variable ϵ, and define the stochastic surrogate as follows:

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{\approx} \tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z\right) + \epsilon, \tag{18}$$

where ϵ is a centered Gaussian random variable with standard deviation σ, i.e., $\epsilon \sim \mathcal{N}(0, \sigma^2)$. With this new formulation, the response PDF of the stochastic surrogate is a convolution of that of the PCE and the Gaussian PDF of ϵ. Let $G_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z\right)$. The PDF of $\tilde{Y}_{\boldsymbol{x}} = G_{\boldsymbol{x}} + \epsilon$ reads

$$f_{\tilde{Y}_{\boldsymbol{x}}}(y) = (f_{G_{\boldsymbol{x}}} * f_\epsilon)(y) = \int_{-\infty}^{+\infty} f_{G_{\boldsymbol{x}}}(y-t) f_\epsilon(t) \mathrm{d}t. \tag{19}$$

Using Hölder's inequality, the above integral is bounded from above by

$$\|f_{G_{\boldsymbol{x}}}\|_1 \, \|f_\epsilon\|_\infty = \|f_\epsilon\|_\infty = \frac{1}{\sigma\sqrt{2\pi}}, \tag{20}$$

meaning that the PDF of $\tilde{Y}_{\boldsymbol{x}}$ and the associated likelihood function are bounded.

To illustrate the role of the additive noise variable in Eq. (18), let us consider a random variable Y with bimodal distribution to be represented by

$$Y \overset{\mathrm{d}}{\approx} \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(Z\right) + \epsilon, \tag{21}$$

where the latent variable Z follows a standard normal distribution and $\epsilon \sim \mathcal{N}(0, \sigma)$. In the case $\sigma = 0$ (the noise term vanishes), we build the model by applying a standard algorithm such as least-angle regression (LAR) [29] to the probability integral transform $F_Y^{-1}(F_Z(Z))$. When the regularization term ϵ is added, maximum likelihood estimation can be used (see Section 4.1 for details) to construct the surrogate.

Figure 1 shows the original (reference) PDF, and the ones obtained by LAR ($\sigma = 0$) and by the stochastic PCE for two different values of σ. It is observed that the PDF obtained by LAR has singularity points, which confirms the analysis in Section 3.2, whereas the proposed noise term regularizes the PDFs. Moreover, LAR is applied directly to the probability integral transform which in practice is unknown. In contrast, the maximum likelihood estimation does not require knowing the values of Z (in this example, only the realizations of Y are used). Finally, the value of σ affects the accuracy of the model. Hence, σ is an additional parameter of the model that must also be fitted to the data to get the optimal approximation. The fitting procedure is detailed in the next section.

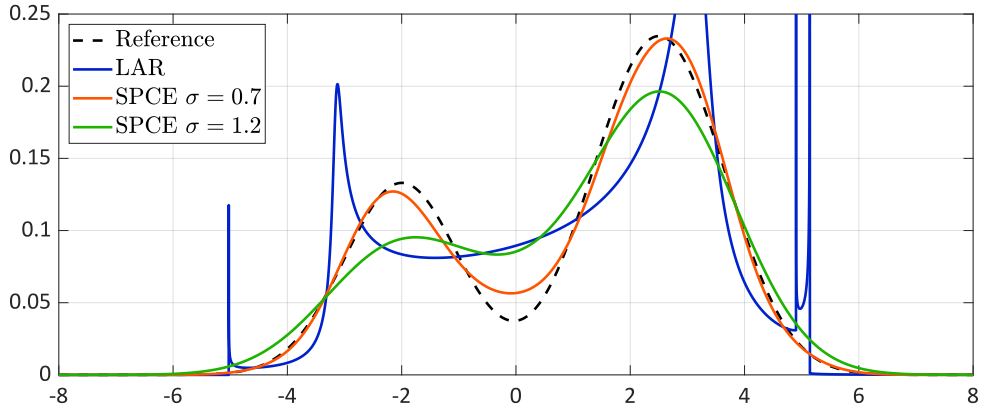


Figure 1: Emulating a bimodal distribution. The blue line corresponds to the result of using LAR to represent directly the probability integral transform (without regularization term). The red and green lines are the results of maximum likelihood estimation for two different values of σ.

4 Fitting the stochastic polynomial chaos expansion

To construct a stochastic PCE defined in Eq. (18), one needs to estimate both the coefficients $\boldsymbol{c}$ and the standard deviation σ of the noise variable. In this section, we present a method to calibrate these parameters from data without replications. Moreover, we propose an algorithm that adaptively selects an appropriate distribution for the latent variable Z and truncation scheme $\mathcal{A}$.

4.1 Maximum likelihood estimation

Let us assume for a moment that the standard deviation σ of the noise variable is given (the estimation of σ will be investigated separately in Section 4.4). From Eq. (18), we see that our surrogate response $\tilde{Y}_{\boldsymbol{x}}$ is the sum of a polynomial function of $(\boldsymbol{x}, z)$ and the noise variable ϵ. Therefore, its PDF can be computed by

$$\begin{aligned} f_{\tilde{Y}_{\boldsymbol{x}}}(y) &= \int_{\mathcal{D}_Z} f_{\tilde{Y}_{\boldsymbol{x}}|Z}(y \mid z) f_Z(z) \mathrm{d}z \\ &= \int_{\mathcal{D}_Z} \frac{1}{\sigma} \varphi\left(\frac{y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)}{\sigma} \right) f_Z(z)\, \mathrm{d}z, \end{aligned} \tag{22}$$

since $\tilde{Y}_{\boldsymbol{x}} \mid Z = z$ is a Gaussian random variable with mean value $\sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)$ and variance σ^2 according to Eq. (18). In this equation, φ stands for the standard normal PDF. Therefore, for a given data point $(\boldsymbol{x}, y)$, the likelihood of the parameters $\boldsymbol{c}$ conditioned on σ reads

$$l(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) = \int_{\mathcal{D}_Z} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z))^2}{2\sigma^2} \right) f_Z(z) \mathrm{d}z. \tag{23}$$

In practice, we can use numerical integration schemes, namely Gaussian quadrature [40], to efficiently evaluate this one-dimensional integral, that is

$$l(\boldsymbol{c};\boldsymbol{x},y,\sigma) \approx \tilde{l}(\boldsymbol{c};\boldsymbol{x},y,\sigma) = \sum_{j=1}^{N_Q} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y-\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x},z_j)\right)^2}{2\sigma^2}\right) w_j, \tag{24}$$

where N_Q is the number of integration points, z_j is the j-th integration point, and w_j is the corresponding weight, both associated to the weight function f_Z. Based on Eq. (24) and the available data $(\mathcal{X}, \boldsymbol{y})$, the PCE coefficients $\boldsymbol{c}$ can be fitted using the maximum likelihood estimation (MLE)

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i}^{N} \log\left(\tilde{l}\left(\boldsymbol{c};\boldsymbol{x}^{(i)},y^{(i)},\sigma\right)\right). \tag{25}$$

The gradient of Eq. (24), and therefore of Eq. (25), can be derived analytically. Hence, we opt for the derivative-based BFGS quasi-Newton method [41] to solve this optimization problem.

4.2 Starting point for the optimization

The objective function to optimize in Eq. (25) is highly nonlinear. As a result, a good starting point is necessary to ensure convergence. According to the properties of the polynomial chaos basis functions, the mean function of a stochastic PCE can be expressed as

$$\tilde{m}(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{E}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \mathbb{E}_{Z,\epsilon}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x},Z)+\epsilon\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_z=0} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{26}$$

where α_z is the degree of the univariate polynomial in Z. Equation (26) contains all the terms without Z, as indicated by $\alpha_z = 0$. We define this set of multi-indices as

$$\mathcal{A}_m = \{\boldsymbol{\alpha}\in\mathcal{A} : \alpha_z = 0\}. \tag{27}$$

Another surrogate $\hat{m}(\boldsymbol{x})$ of the mean function can be obtained by using standard (or sparse) regression to directly fit the following expansion:

$$m(\boldsymbol{x}) \overset{\text{def}}{=} \mathbb{E}\left[Y_{\boldsymbol{x}}\right] \approx \hat{m}(\boldsymbol{x}) \overset{\text{def}}{=} \sum_{\boldsymbol{\alpha}\in\mathcal{A}_m} c^m_{\boldsymbol{\alpha}}\psi(\boldsymbol{x}) \tag{28}$$

The obtained coefficients $\boldsymbol{c}^m$ are used as initial values for the coefficients $\{\boldsymbol{c}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha}\in\mathcal{A}_m\}$ of the stochastic surrogate in the optimization procedure, i.e., $\boldsymbol{c}_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha}\in\mathcal{A}_m$.

For the other coefficients $\{c_{\boldsymbol{\alpha}} : \boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m\}$, we randomly initialize their value.

4.3 Warm-start strategy

Because of the form of the likelihood Eq. (23), the gradient at the starting point can take extremely large values when σ is small. In this case, the optimization algorithm may become unstable and converge to an undesired local optimum. To guide the optimization, we propose a warm-start strategy summarized in Algorithm 1. We generate a decreasing sequence $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_{N_s}\}$ with $\sigma_{N_s} = \sigma$ (the target value). In this paper, we choose the maximum value σ_1 of the sequence as the square root of the *leave-one-out error* ε_{LOO} in the mean fitting procedure (see Appendix A.1 for the explanation of this choice). Then, $\boldsymbol{\sigma}$ is generated equally-spaced in the log-space between $\sqrt{\varepsilon_{\text{LOO}}}$ and σ. Starting with σ_1 which is the largest element of $\boldsymbol{\sigma}$, we build a stochastic PCE based on Eq. (25) with the initial values defined above (the mean function estimation and random initialization). Then, the results are used as a starting point for the construction of the surrogate for σ_2. We repeat this procedure sequentially for each element in $\boldsymbol{\sigma}$ with each new starting point being the results of the previous optimization. Because the standard deviation decreases progressively to the target value and the starting point is updated accordingly, the associated gradient for each optimization prevents extremely big values.

Algorithm 1 Warm-start approach for estimating $\boldsymbol{c}$ with known σ

Input: $(\mathcal{X}, \boldsymbol{y})$, σ, $\mathcal{A}$
Output: Coefficients $\hat{\boldsymbol{c}}$

$\boldsymbol{c}^m, \varepsilon_{\text{LOO}} \leftarrow \text{OLS}(\mathcal{X}, \boldsymbol{y}, \mathcal{A}_m)$ % Estimation of the coefficients of the mean function
$c^0_{\boldsymbol{\alpha}} \leftarrow c^m_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathcal{A}_m$ and randomly initialize $\{c^0_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m\}$
$\boldsymbol{\sigma}_{\log} \leftarrow \text{linspace}\left(\log\left(\sqrt{\varepsilon_{\text{LOO}}}\right), \log(\sigma), N_s\right)$
$\boldsymbol{\sigma} \leftarrow \exp\left(\boldsymbol{\sigma}_{\log}\right)$
for $i \leftarrow 1, \dots, N_s$ **do**
    Solve Eq. (25) to compute $\boldsymbol{c}^i$ using $\boldsymbol{c}^{i-1}$ as initial values
end for
$\hat{\boldsymbol{c}} \leftarrow \boldsymbol{c}^{N_s}$

4.4 Cross-validation

As explained in Section 3.2, the hyperparameter σ cannot be jointly estimated together with the PCE coefficients $\boldsymbol{c}$ because the likelihood function can reach $+\infty$ for certain choices of $\boldsymbol{c}$ and $\sigma = 0$. As a result, σ should be tuned separately from the estimation of $\boldsymbol{c}$.

In this paper, we propose applying cross-validation (CV) [39] to selecting the optimal value of σ. More precisely, the data $(\mathcal{X}, \boldsymbol{y})$ are randomly partitioned into N_{cv} equal-sized groups $\{V_k : k = 1, \dots, N_{\text{cv}}\}$ (so-called N_{cv}-fold CV). For $k \in \{1, \dots, N_{\text{cv}}\}$, we pick the k-th group V_k as the validation set and the other $N_{\text{cv}} - 1$ folds denoted by $V_{\sim k}$ as the training set. The latter is used to build a stochastic PCE following Eq. (25) and Algorithm 1, which yields

$$\hat{\boldsymbol{c}}_k(\sigma) = \arg\max_{\boldsymbol{c}} \sum_{i \in V_{\sim k}} \log\left(\tilde{l}\left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)}, \sigma\right)\right). \tag{29}$$

Note that the coefficients depend on the value of σ, and thus we explicitly write them as functions of σ. The validation set V_k is then used to evaluate the *out-of-sample* performance:

$$\mathrm{l}_k(\sigma) = \sum_{i \in V_k} \log\left(\tilde{l}\left(\hat{\boldsymbol{c}}_k(\sigma); \boldsymbol{x}^{(i)}, y^{(i)}, \sigma\right)\right). \tag{30}$$

We repeat this procedure for each group of the partition $\{V_k : k = 1, \ldots, N_{\mathrm{cv}}\}$ and sum up the respective score to estimate the generalized performance, referred to as *CV score* in the sequel. Then, the optimal value of σ is selected as the one that maximizes this CV score:

$$\hat{\sigma} = \arg\max_{\sigma} \sum_{k=1}^{N_{\mathrm{cv}}} \mathrm{l}_k(\sigma). \tag{31}$$

Because of the nested optimization in Eq. (29), the gradient of Eq. (31) is difficult to derive. In this paper, we apply the derivative-free Bayesian optimizer [42] to solving Eq. (31) and search for σ within the range $[0.1, 1] \times \sqrt{\varepsilon_{\mathrm{LOO}}}$. The upper bound of the interval is explained in Appendix A.1. The lower bound is introduced to prevent numerical instabilities near $\sigma = 0$. According to our investigations, the optimal value $\hat{\sigma}$ is always within the proposed interval.

After solving Eq. (31), the selected $\hat{\sigma}$ is used in Eq. (25) with all the available data to build the final surrogate.

Large value of N_{cv} can lead to high computational cost, especially when N is big. In this paper, we choose $N_{\mathrm{cv}} = 10$ for $N < 200$ (small data set), $N_{\mathrm{cv}} = 5$ for $200 \leq N < 1{,}000$ (moderate data set) and $N_{\mathrm{cv}} = 3$ for $N \geq 1{,}000$ (big data set).

4.5 Adaptivity

The method developed in Sections 4.1 and 4.4 allows us to build a stochastic PCE for a given distribution of the latent variable Z and truncated set $\mathcal{A}$ of polynomial chaos basis. In principle, one can choose any continuous probability distribution for the latent variable and a large truncated set. However, in practice, certain types of latent variables may require a lot of basis functions to approximate well the shape of the response distribution. This leads to many model parameters to estimate, which would cause overfitting when only a few data are available. In this section, we propose a procedure to iteratively find a suitable distribution for the latent variable Z and truncation scheme $\mathcal{A}$.

We consider N_z candidate distributions $\boldsymbol{D} = \{D_1, \ldots, D_{N_z}\}$ for the latent variable, N_p degrees $\boldsymbol{p} = \{p_1, \ldots, p_{N_p}\}$ and N_q q-norms $\boldsymbol{q} = \{p_1, \ldots, p_{N_p}\}$ that are used to define the hyperbolic truncation scheme in Eq. (7). Both $\boldsymbol{p}$ and $\boldsymbol{q}$ are sorted in increasing order.

The adaptive procedure is shown in Algorithm 2 and described here. For each type of latent variable and truncation set $\mathcal{A} = \mathcal{A}^{p,q,M}$, we first apply the hybrid LAR algorithm developed by Blatman and Sudret [29] to fitting the mean function $\hat{m}(\boldsymbol{x})$ as shown in Eq. (28). This

algorithm only selects the most important basis among the candidate set $\mathcal{A}_m$ defined in Eq. (27). To reduce the total number of unknowns in the optimization Eq. (25), we exclude from $\mathcal{A}$ the basis functions in $\mathcal{A}_m$ that are not selected by hybrid LAR. In other words, we only estimate the coefficients associated with the basis functions that either have $\alpha_z \neq 0$ or are selected by the hybrid LAR when fitting the mean function $m(\boldsymbol{x})$. Then, we use the methods presented in Sections 4.1 and 4.4 to build a stochastic PCE for $\mathcal{A}$ and record the CV score. The latter is used for model comparisons, and the one with the best CV score is selected as the final surrogate.

Algorithm 2 Adaptive algorithm for building a stochastic PCE

Input: $(\mathcal{X}, \boldsymbol{y})$, $\boldsymbol{D}$, $\boldsymbol{p}$, $\boldsymbol{q}$
Output: D_{opt}, $\mathcal{A}_{opt}$, $\hat{\boldsymbol{c}}$, $\hat{\sigma}$

$\mathrm{l}_{opt} \leftarrow -\infty$
for $i_z \leftarrow 1, \ldots, N_z$ **do**
  Set $Z \sim D_{i_z}$
  for $i_p \leftarrow 1, \ldots, N_p$ **do**
    for $i_q \leftarrow 1, \ldots, N_q$ **do**
      $\mathcal{A} \leftarrow \mathcal{A}^{p_{i_p}, q_{i_q}, M+1}$
      $\mathcal{A}_m \leftarrow \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathcal{A}, \alpha_z = 0\}$, $\mathcal{A}_c \leftarrow \mathcal{A} \setminus \mathcal{A}_m$
      $\mathcal{A}_n \leftarrow$ Hybrid-LAR $(\mathcal{X}, \boldsymbol{y}, \mathcal{A}_m)$ % Selection of the basis for $\hat{m}(\boldsymbol{x})$
      $\mathcal{A} \leftarrow \mathcal{A}_n \cup \mathcal{A}_c$
      Apply the algorithm presented in Sections 4.1 and 4.4 to build a stochastic PCE with $\mathcal{A}$, which gives $\boldsymbol{c}$, σ, and the CV score l_{i_p, i_q} associated with σ.
    end for
  end for
end for
Return the model with the maximum CV score

In order to avoid going through all the possible combinations, we propose a heuristic *early stopping criterion* for both degree and q-norm adaptivity. If two consecutive increases of q-norm cannot improve the CV score, the inner loop for q-norm adaptivity stops. Besides, if the best model (among all the q-norms) of a larger degree decreases the CV score, the algorithm stops exploring higher degrees. Note that the early stopping is only applied to p- and q-adaptivity, but all the candidate distributions are investigated.

In summary, we sketch the overall procedure (presented in Sections 4.1 to 4.5) to adaptively build a stochastic PCE from data in Figure 2.

In the application examples, we choose $N_Z = 2$ possible distributions for the latent variable Z, namely a standard normal distribution $\mathcal{N}(0, 1)$ and a uniform distribution $\mathcal{U}(-1, 1)$. The truncation parameters $\boldsymbol{p}$ and $\boldsymbol{q}$ may be selected according to the dimensionality M of the problem and the prior knowledge on the level of non-linearity. We typically use $\boldsymbol{p} = \{1, 2, 3, 4, 5\}$ and $\boldsymbol{q} = \{0.5, 0.75, 1\}$.

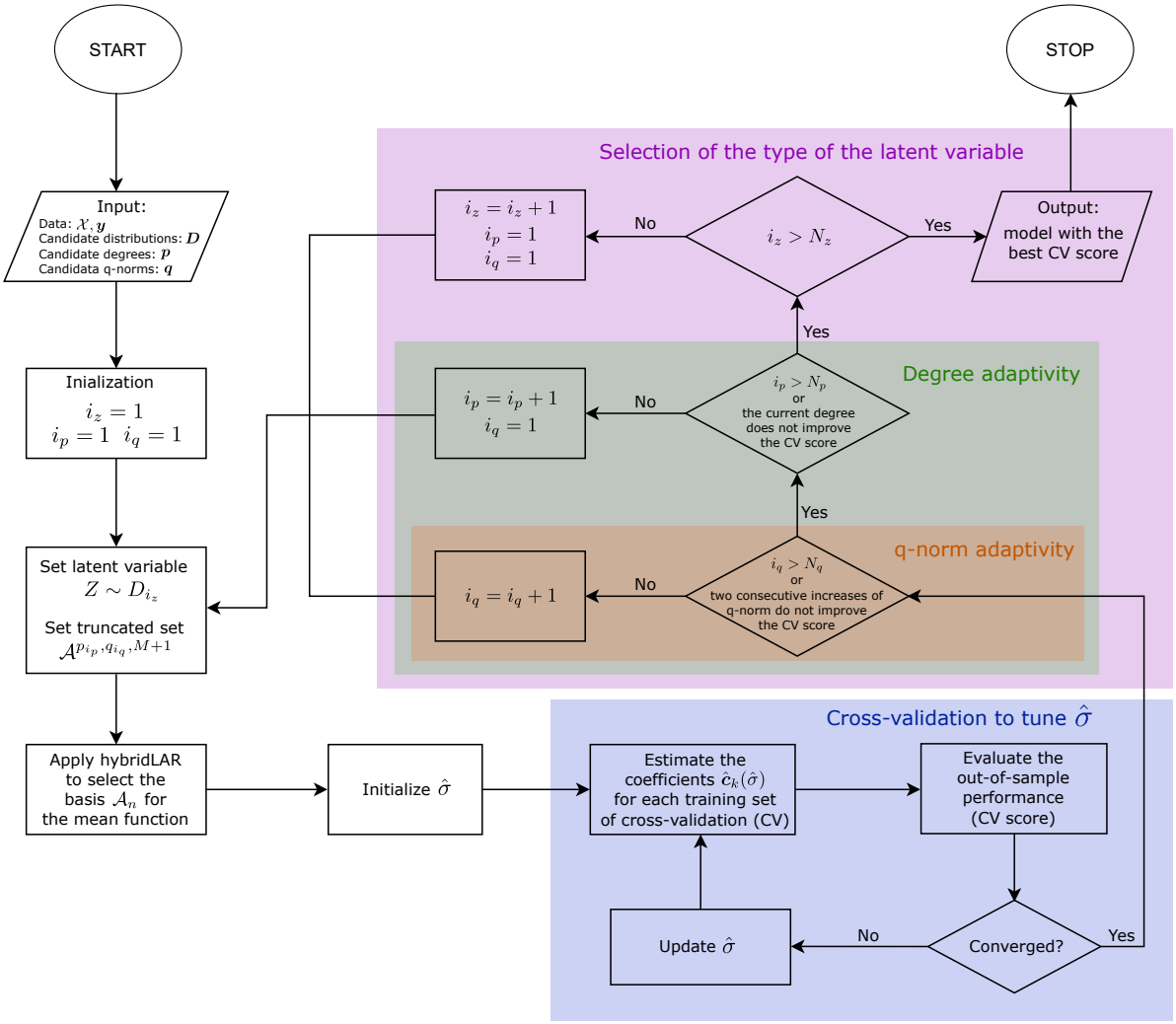


Figure 2: Flow chart of the procedure to adaptively build a stochastic PCE

4.6 Post-processing of stochastic polynomial chaos expansions

In this section, we show how to post-process a stochastic PCE for various analyses. The very feature of this surrogate is that it provides a functional mapping between the input parameters $\boldsymbol{X}$, the latent variable Z, and the noise term ϵ:

$$\tilde{Y} \overset{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{X}, Z \right) + \epsilon, \tag{32}$$

To generate realizations of $\tilde{Y}$, we simply sample $\boldsymbol{X}$, Z and ϵ following their distributions and then evaluate Eq. (32). To obtain samples of $\tilde{Y}_{\boldsymbol{x}}$ for a fixed $\boldsymbol{x}$ (e.g., to plot the conditional distribution), we follow the same procedure with fixed $\boldsymbol{X} = \boldsymbol{x}$. Moreover, Eq. (32) can be easily vectorized for efficient sampling.

By generating a large number of samples, one can display the distribution of $\tilde{Y}$ and $\tilde{Y}_{\boldsymbol{x}}$ using histograms or kernel density estimation. We can also use the quadrature version in Eq. (24) to get an explicit form of the conditional response distribution of $\tilde{Y}_{\boldsymbol{x}}$.

In addition, because the proposed surrogate model is derived based on PCE, it inherits all the good properties of PCE. In particular, some important quantities can be directly computed by post-processing the PCE coefficients $\boldsymbol{c}$ and the parameter σ, without the need for sampling. Indeed, the mean and variance of $\tilde{Y}$ are given by

$$\mathbb{E}\left[\tilde{Y}\right] = c_{\boldsymbol{0}}, \quad \text{Var}\left[\tilde{Y}\right] = \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \boldsymbol{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2. \tag{33}$$

where $c_{\boldsymbol{0}}$ is the coefficient of the constant function.

As already shown in Eq. (26), for a given value of $\boldsymbol{x}$, the mean of the model response $\tilde{Y}_{\boldsymbol{x}}$ can be computed as

$$\mathbb{E}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \sum_{\boldsymbol{\alpha} \in \mathcal{A}, \alpha_z = 0} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{34}$$

Similarly, we can compute the variance as follows:

$$\text{Var}\left[\tilde{Y}_{\boldsymbol{x}}\right] = \text{Var}_{Z,\epsilon}\left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}} \left(\boldsymbol{x}, Z \right) + \epsilon\right] = \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\alpha}}^2(\boldsymbol{x}) + \sigma^2. \tag{35}$$

4.7 Global sensitivity analysis

In the context of global sensitivity analysis of stochastic simulators [22], various types of Sobol' indices can also be computed analytically for the proposed surrogate model. The *classical Sobol' indices* are defined from the Sobol'-Hoeffding decomposition of the deterministic model given by the stochastic simulator with both the well-defined input variables $\boldsymbol{X}$ and its intrinsic stochasticity as explicit inputs ω, see Eq. (1). Since the surrogate model in Eq. (32) is also a deterministic function of $\boldsymbol{X}$ and the additional variables Z and ϵ, the Sobol' indices can be

efficiently computed from the PCE coefficients, similarly to the classical PCE-based Sobol' indices [35]. For example, the first-order classical Sobol' index of the i-th input X_i is given by

$$S_i \overset{\text{def}}{=} \frac{\text{Var}\left[\mathbb{E}\left[\tilde{Y} \mid X_i\right]\right]}{\text{Var}\left[\tilde{Y}\right]} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_i} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \tag{36}$$

where $\mathcal{A}_i \overset{\text{def}}{=} \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i \neq 0,\, \alpha_j = 0\,, \forall j \neq i\}$. Similarly, one can also calculate higher-order and total Sobol' indices of the model Eq. (32). Let us split the input vector into two subsets $\boldsymbol{X} = (\boldsymbol{X}_{\mathbf{u}}, \boldsymbol{X}_{\sim\mathbf{u}})$, where $\mathbf{u} \subset \{1, \ldots, M\}$ and $\sim\mathbf{u}$ is the complement of $\mathbf{u}$, i.e., $\sim\mathbf{u} = \{1, \ldots, M\} \setminus \mathbf{u}$. The higher-order and total Sobol' indices, denoted by $S_{\mathbf{u}}$ and S_{T_i}, respectively, are given by

$$S_{\mathbf{u}} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \quad S_{T_i} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_i\neq 0} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2 + \sigma^2}, \tag{37}$$

where $\mathcal{A}_{\mathbf{u}} \overset{\text{def}}{=} \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i \neq 0,\, \alpha_j = 0\,, \alpha_z = 0\,, \forall i \in \mathbf{u}, \forall j \in \sim\mathbf{u}\}$. However, as mentioned in Section 3, the surrogate model aims only at emulating the response distribution of the simulator instead of representing the detailed data generation process. Therefore, the indices involving the artificial variables introduced in the surrogate (i.e., Z and ϵ), e.g., the first-order Sobol' index for Z and the total Sobol' index for each component of $\boldsymbol{X}$, do not reveal the nature of the original model [22].

The QoI-based Sobol' indices quantify the influence of the input variables on some quantity of interest of the random model response, e.g., mean, variance, and quantiles [22]. As the mean function in Eq. (26) is a PCE, the associated Sobol' indices can be computed in a straightforward way [35]. Similar to Eq. (36), the first-order index is given by

$$S_i^m \overset{\text{def}}{=} \frac{\text{Var}\left[\mathbb{E}\left[\tilde{m}(\boldsymbol{X}) \mid X_i\right]\right]}{\text{Var}\left[\tilde{m}(\boldsymbol{X})\right]} = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_i} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}, \tag{38}$$

while higher-order and total Sobol' indices of the mean function read

$$S_{\mathbf{u}}^m = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}, \quad S_{T_i}^m = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_i\neq 0} c_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}_m\setminus\mathbf{0}} c_{\boldsymbol{\alpha}}^2}. \tag{39}$$

In addition, the variance function in Eq. (35) is a polynomial. The associated Sobol' indices can be computed by building another PCE to represent Eq. (35) the without error.

5 Numerical examples

In this section, we validate the proposed method on several examples, namely case studies from mathematical finance and epidemiology and a complex analytical example with bimodal response distributions. To illustrate its performance, we compare the results obtained from the stochastic polynomial chaos expansion (SPCE) with two state-of-the-art models that are developed for emulating the response distribution of stochastic simulators. The first one is the generalized lambda model (GLaM). This surrogate uses the four-parameter generalized lambda distribution to approximate the response distribution of $Y_{\boldsymbol{x}}$ for any $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. The distribution parameters, as functions of the inputs, are represented by PCEs (see details in [13, 21]). The second model is based on kernel conditional density estimation (KCDE) [43]. This method uses kernel density estimation to fit the joint distribution $\hat{f}_{\boldsymbol{X},Y}(\boldsymbol{x}, y)$ and the marginal distribution $\hat{f}_{\boldsymbol{X}}(\boldsymbol{x})$. The response distribution is then estimated by

$$f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) = \frac{\hat{f}_{\boldsymbol{X},Y}(\boldsymbol{x}, y)}{\hat{f}_{\boldsymbol{X}}(\boldsymbol{x})} = \frac{\sum_{i=1}^{N} \frac{1}{h_y} K_Y\left(\frac{y-y^{(i)}}{h_y}\right) \prod_{j=1}^{M} \frac{1}{h_j} K_j\left(\frac{x_j - x_j^{(i)}}{h_j}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{M} \frac{1}{h_j} K_j\left(\frac{x_j - x_j^{(i)}}{h_j}\right)}, \tag{40}$$

where K_y and K_j's are the kernels for Y and X_j's, and h_y and h_j's are the associated bandwidths which are hyperparameters selected by a thorough leave-one-out cross-validation [19].

Finally, we also consider a model where we represent the response with a normal distribution. The associated mean and variance as functions of the input $\boldsymbol{x}$ are set to the *true* values obtained from the simulator. Therefore, the accuracy of such an approximation measures how close the response distribution is to the normal distribution. Moreover, this model represents the "oracle" of Gaussian-type mean-variance models, such as the ones presented in Marrel et al. [6] and Binois et al. [8].

To quantitatively compare the various surrogates, we define an error metric between the simulator and the emulator by

$$\varepsilon = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(Y_{\boldsymbol{X}}, \tilde{Y}_{\boldsymbol{X}}\right)\right]}{\mathrm{Var}\left[Y\right]}, \tag{41}$$

where $Y_{\boldsymbol{x}}$ is the model response, $\tilde{Y}_{\boldsymbol{x}}$ denotes that of the surrogate (with the same input parameters as $Y_{\boldsymbol{x}}$), and Y is the model output aggregating all the uncertainties from both the input and the intrinsic stochasticity. d_{WS} is the *Wasserstein distance of order two* [44] between the two probability distributions defined by

$$d_{\mathrm{WS}}^2\left(Y_1, Y_2\right) \overset{\mathrm{def}}{=} \|Q_1 - Q_2\|_2^2 = \int_0^1 \left(Q_1(u) - Q_2(u)\right)^2 \mathrm{d}u, \tag{42}$$

where Q_1 and Q_2 are the quantile functions of random variables Y_1 and Y_2, respectively. The

error metric ε in Eq. (41) is unitless and invariant to shift and scale, i.e.,

$$\frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(aY_{\boldsymbol{X}}+b, a\tilde{Y}_{\boldsymbol{X}}+b\right)\right]}{\mathrm{Var}\left[aY+b\right]} = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d_{\mathrm{WS}}^2\left(Y_{\boldsymbol{X}}, \tilde{Y}_{\boldsymbol{X}}\right)\right]}{\mathrm{Var}\left[Y\right]}. \tag{43}$$

To evaluate the numerator in Eq. (41), we generate a test set $\mathcal{X}_{\mathrm{test}}$ of size $N_{\mathrm{test}} = 1{,}000$ from the input distribution of $\boldsymbol{X}$. The Wasserstein distance is calculated for each point $\boldsymbol{x} \in \mathcal{X}_{\mathrm{test}}$ and then averaged over N_{test}.

We use Latin hypercube sampling (LHS) [45] to generate the experimental design and the test set. The stochastic simulator is evaluated only once for each set of input parameters, i.e., we do not use replications. To study the convergence property of the surrogates, experimental designs of various sizes are investigated. Each scenario is run 20 times with independent experimental designs to account for the statistical uncertainty in the LHS design and also in the internal stochasticity of the simulator. As a result, error estimates for each size of experimental design are represented by box plots constructed from the 20 repetitions of the full analysis.

5.1 Geometric Brownian motion

In the first example, we consider the *Black-Scholes* model that is popular in mathematical finance [1]

$$\mathrm{d}S_t = x_1\, S_t\, \mathrm{d}t + x_2\, S_t\, \mathrm{d}W_t. \tag{44}$$

Equation (44) is a stochastic differential equation used to model the evolution of a stock price S_t. Here, $\boldsymbol{x} = (x_1, x_2)^T$ are the input variables that describe the expected return rate and the volatility of the stock, respectively. W_t is a Wiener process that represents the stochastic behavior of the market. Without loss of generality, we set the initial condition to $S_0 = 1$.

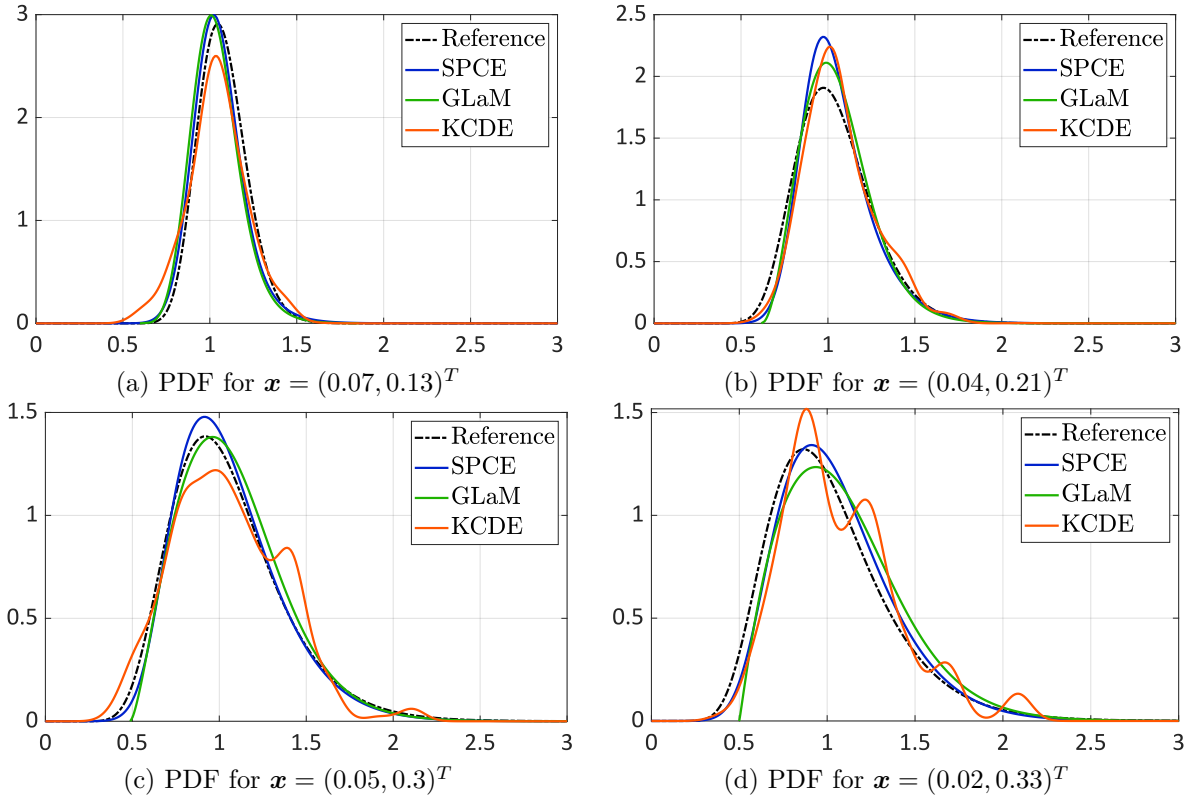
The simulator is stochastic: for a given $\boldsymbol{x}$, the stock price S_t is a stochastic process, where the stochasticity comes from W_t. In this example, we are interested in $Y_{\boldsymbol{x}} = S_1$, which corresponds to the stock value at $t = 1$ year. We set $X_1 \sim \mathcal{U}(0, 0.1)$ and $X_2 \sim \mathcal{U}(0.1, 0.4)$ to represent the uncertainty in the return rate and the volatility, where the ranges are selected based on real data [46].

The solution to Eq. (44) can be derived using Itô calculus [47]: $Y_{\boldsymbol{x}}$ follows a lognormal distribution defined by

$$Y_{\boldsymbol{x}} \sim \mathcal{LN}\left(x_1 - \frac{x_2^2}{2}, x_2\right). \tag{45}$$

As the distribution of $Y_{\boldsymbol{x}}$ is known analytically in this simple example, we can sample directly from the response distribution to get the model output instead of simulating the whole path of S_t.

Figure 3 illustrates four response PDFs predicted by the considered surrogates built on an experimental design of size $N = 400$. We observe that with 400 model runs, both SPCE and



(a) PDF for $\boldsymbol{x} = (0.07, 0.13)^T$ (b) PDF for $\boldsymbol{x} = (0.04, 0.21)^T$

(c) PDF for $\boldsymbol{x} = (0.05, 0.3)^T$ (d) PDF for $\boldsymbol{x} = (0.02, 0.33)^T$

Figure 3: Geometric Brownian motion — Comparisons of the emulated PDFs, $N = 400$.

GLaM accurately represent the variation of the response PDF. Moreover, SPCE better represents the left tail in Fig. 3d. In contrast, KCDE can well approximate the response PDF for low volatility (in Fig. 3a) but exhibits unrealistic oscillations in the case of high volatility.

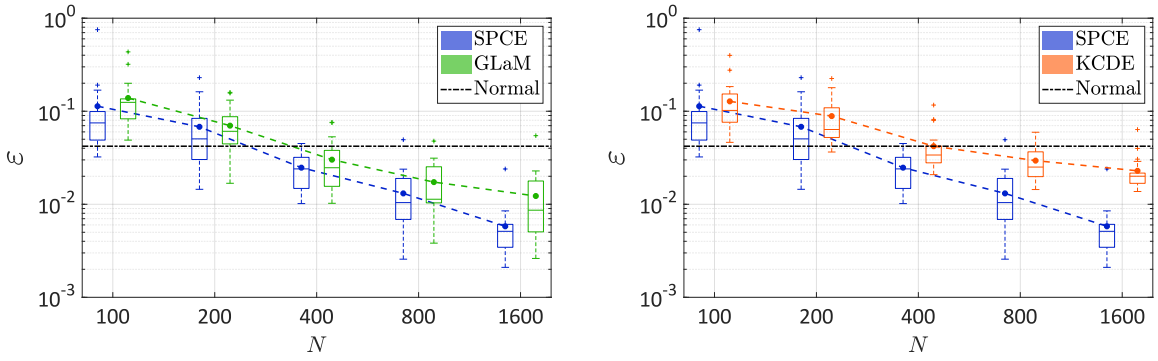


Figure 4: Geometric Brownian motion — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results. The black dash-dotted line represents the error of the model assuming that the response distribution is normal and using the true mean and variance.

For convergence studies, we vary the size of the experimental design $N \in \{100; 200; 400; 800; 1{,}600\}$ and plot the error ε defined in Eq. (41) with respect to N in Fig. 4. In order to show more details, each subfigure in Fig. 4 compares SPCE with one competitor. We observe that the average error of KCDE built on $N = 400$ model runs is similar to the best normal approximation, whereas

both SPCE and GLaM provide smaller errors. Compared with KCDE and GLaM, the average performance of SPCE is always the best for all sizes of experimental design. For large N, namely $N = 1{,}600$, the average error of SPCE is less than half of that of KCDE, and the spread of the error is narrower than that obtained by GLaM.

5.2 Stochastic SIR model

The second example is the stochastic *Susceptible-Infected-Recovered* (SIR) model frequently used in epidemiology [2]. This model simulates the outbreak of an infectious disease which spreads out through stochastic contacts between infected and susceptible individuals. The simulator is a compartmental state-space model. More precisely, a population of P individuals at time t is partitioned into three groups: (1) *susceptible individuals* who have not caught the disease and may be infected by close contact with infectious patients; (2) *infected individuals* who are contaminated and infectious; (3) *recovery individuals* who have recovered from the disease and are immune to future infections. The count of each group is denoted by S_t, I_t, and R_t, respectively. Because no newborn or death is considered, the three quantities satisfy $E_t + I_t + R_t = P$. As a result, any two out of the three counts, e.g., E_t and I_t, can characterize the configuration of the population of size P at time t.

Figure 5 illustrates the dynamics of the model, where the black icons stand for susceptible individuals, the red icons correspond to infected persons, and the blue icons are the ones who have recovered. At time t, the state of the population is given by (S_t, I_t) (the top left panel of Fig. 5). The next configuration depends on two transition channels: infection and recovery. The first channel evolves the system to C_I where one susceptible individual is infected (the bottom left panel of Fig. 5). The recovery channel proceeds to C_R where one infected person recovers (the bottom right panel of Fig. 5). Whether the system evolves to the candidate state C_I or C_R depends on two random variables, T_I and T_R which are the respective transition time of each channel. Both T_I and T_R follow an exponential distribution, yet with different parameters:

$$\begin{aligned} T_I &\sim \text{Exp}(\lambda_I), \quad \lambda_I = \beta \frac{S_t I_t}{P}, \\ T_R &\sim \text{Exp}(\lambda_R), \quad \lambda_R = \gamma I_t, \end{aligned} \tag{46}$$

where β is the contact rate of an infected individual, and γ is the recovery rate. The next configuration of the population is the one that comes first, i.e., for $T_R < T_I$, the system evolves to C_R at $t + T_R$ with $S_{t+T_R} = E_t - 1$ and $I_{t+T_I} = I_t + 1$, and vice versa. We iterates this updating procedure until the time T where $I_T = 0$ corresponding to no remaining infected individual: no infection or recovery can happen, and the outbreak stops. Since the population size is constant and recovered individuals will not be infected again, the outbreak will stop at finite time, i.e., $T < +\infty$. The simulation process described here corresponds to the *Gillespie algorithm* [48].

The input variables of the simulator are the initial conditions S_0 and I_0 and the transitive rates

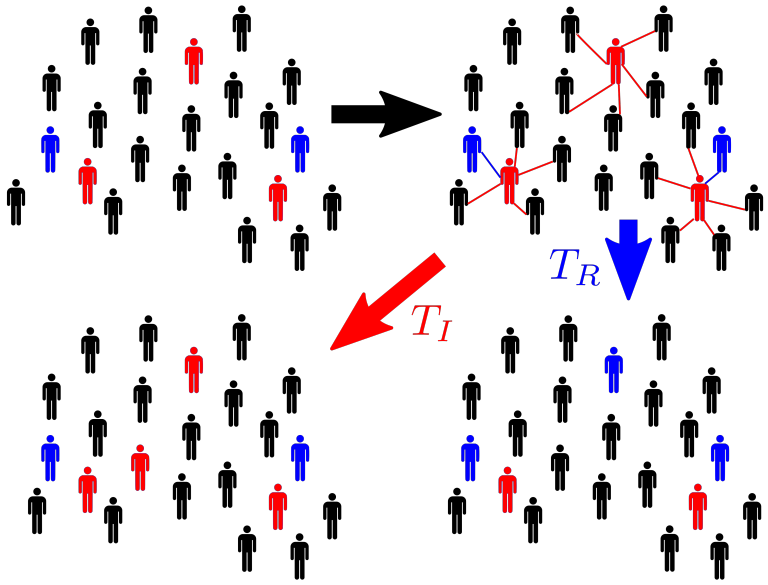


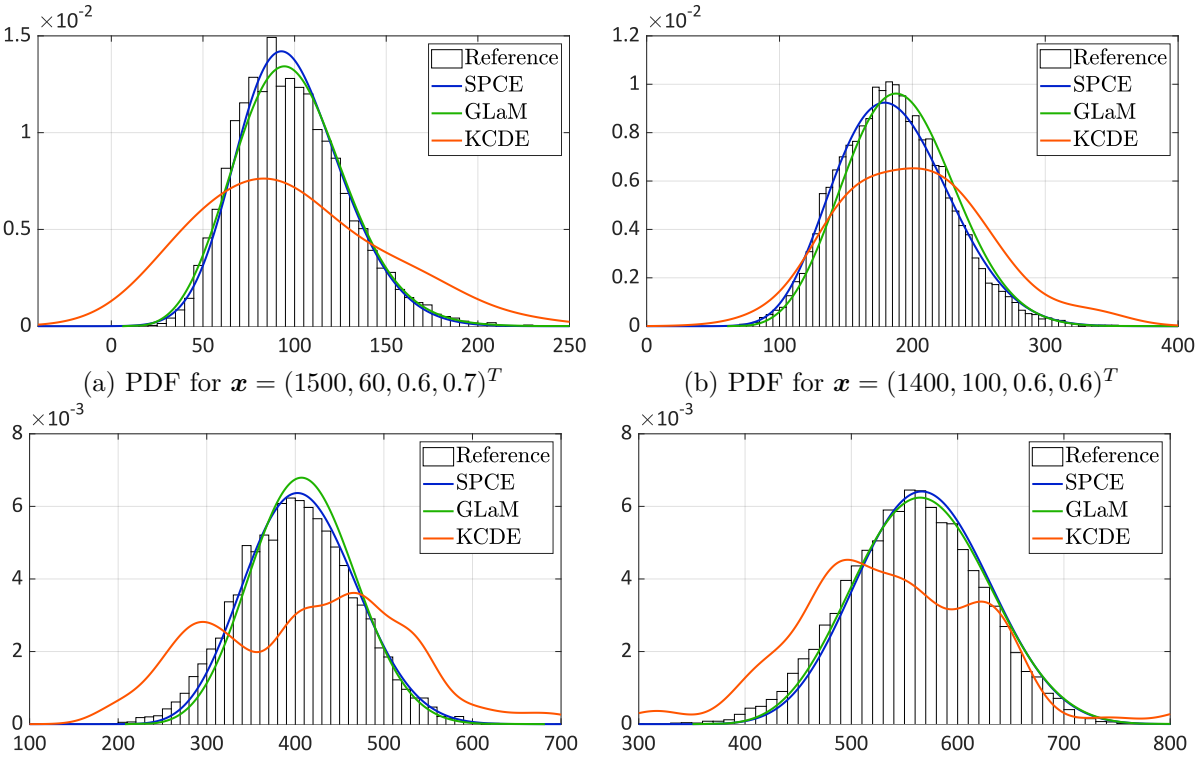
Figure 5: Dynamics of the stochastic SIR model: black icons stand for susceptible individuals, red icons represent infected individuals, and blue icons are the ones that have recovered.

β and γ. We are interested in the total number of newly infected individuals during the outbreak without counting the initial infections, which is an important quantity in epidemics management [8]. This can be calculated by the difference between the number of susceptibles at time 0 and T, i.e., $Y = S_0 - S_T$. Because each updating step in Eq. (46) depends on two latent variables T_I and T_R, the simulator is stochastic. Moreover, the total number of latent variables is also random.

In this case study, we set $P = 2{,}000$. To account for different scenarios, the input variables $\boldsymbol{X} = \{S_0, I_0, \beta, \gamma\}$ are modeled as $S_0 \sim \mathcal{U}(1{,}200\,,\, 1{,}800)$, $I_0 \sim \mathcal{U}(20, 200)$, and $\beta, \gamma \sim \mathcal{U}(0.5, 0.75)$. The uncertainty in the first two variables is due to the lack of knowledge of the initial condition. The two transitive rates β, γ are affected by possible interventions such as quarantine and increase of medical resources.

Figure 6 illustrates the response PDF for four different sets of input parameters. Because of the transition process in Eq. (46), no analytical closed-form distribution of $Y_{\boldsymbol{x}}$ can be derived. Therefore, we use 10^4 replications for each input values to obtain the reference histograms. The surrogate models are trained on an experimental design of size $N = 1{,}600$ (without any replications). We observe that the four PDFs are unimodal. The reference histogram in Fig. 6a is slightly right-skewed, while the others in Fig. 6 are symmetric. SPCE and GLaM produce similar predictions of the PDF which are very close to the reference histograms. In comparison, KCDE overestimates the spread of the distributions in. Moreover, the KCDE prediction has non-negligible probability for unrealistic negative values in Fig. 6a. Besides, it exhibits relatively poor shape representations with spurious wiggles in Fig. 6c and Fig. 6d.

Figure 7 compares the performance of the surrogates built on various sizes of experimental design $N \in \{200; 400; 800; 1{,}600; 3{,}200\}$. To evaluate the error defined in Eq. (41), the reference distribution for each $\boldsymbol{x}$ is given by the empirical distribution of 10^4 replications. The oracle



(a) PDF for $\boldsymbol{x} = (1500, 60, 0.6, 0.7)^T$

(b) PDF for $\boldsymbol{x} = (1400, 100, 0.6, 0.6)^T$

(c) PDF for $\boldsymbol{x} = (1700, 140, 0.55, 0.55)^T$

(d) PDF for $\boldsymbol{x} = (1600, 180, 0.7, 0.6)^T$

Figure 6: Stochastic SIR — Comparisons of the emulated PDFs, $N = 1{,}600$.

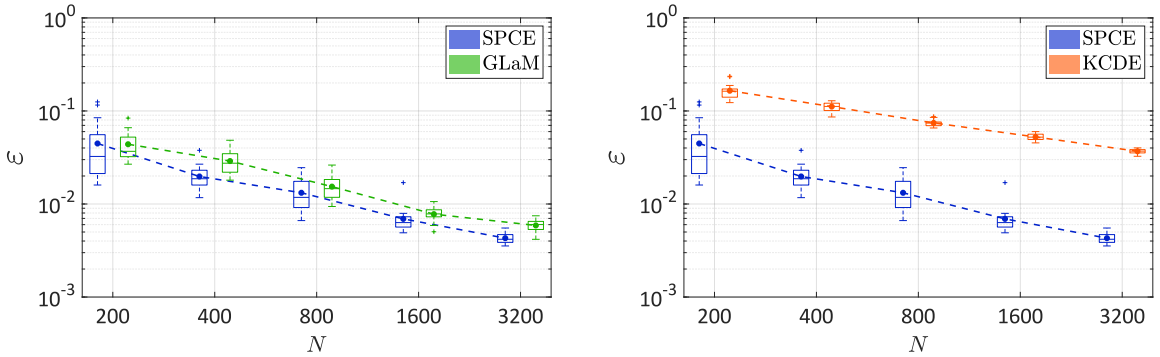


Figure 7: Stochastic SIR — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis, whereas the box plot summarize the 20 results. The Gaussian model that assumes the response distribution being normal with the mean and variance estimated from 10^4 replications yields an error of 6×10^{-4}, which is not plotted in the figure.

normal approximation gives an error of 6×10^{-4} which is smaller than any of the surrogates in consideration. Note that this model is not built on the training data but using the mean and variance from the 10^4 replications for each test point. This implies that the response distribution is close to normal. We do not include this error in Fig. 7 to not loose detailed comparisons of the surrogate models. Figure 7 reveals a poor performance of KCDE in this case study. This is because the example is four-dimensional, and KCDE is a kernel-based method which is known to suffer from the *curse of dimensionality*. In contrast, SPCE and GLaM are flexible parametric models, and both provide a much smaller error than KCDE for all values of N. Compared with GLaM, SPCE yields a similar spread of the error but demonstrates better average performance for $N \geq 400$.

5.3 Bimodal analytical example

The response distributions of the previous two examples are unimodal. In the last example, we consider a complex analytical example to test the flexibility of the stochastic polynomial chaos expansion. For this purpose, we directly define the response distribution to approximate as

$$f_{Y|X}(y \mid x) = 0.5\,\varphi\left(1.25\,y - (5\sin^2(\pi \cdot x) + 5x - 2.5)\right) + 0.75\,\varphi\left(1.25\,y - (5\sin^2(\pi \cdot x) - 5x + 2.5)\right) \tag{47}$$

where φ stands for the standard normal PDF. This response PDF is a mixture of two Gaussian PDFs with weights 0.6 and 0.8. The mean function of each component distribution depends on the input variable x. Let $X \sim \mathcal{U}(0, 1)$. With different realization of X, the two components change their location accordingly. Figure 8 illustrates a data set generated by $N = 800$ model runs and the mean function of each component of Eq. (47) which varies nonlinearly with respect to the input. It is clear that the resulting conditional distribution is bimodal for small ($x \lesssim 0.2$) and large values of x ($x \gtrsim 0.8$), whereas it is unimodal in between.

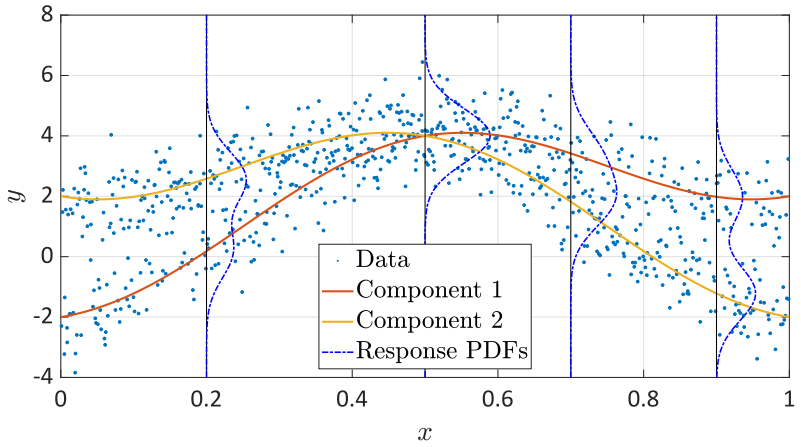
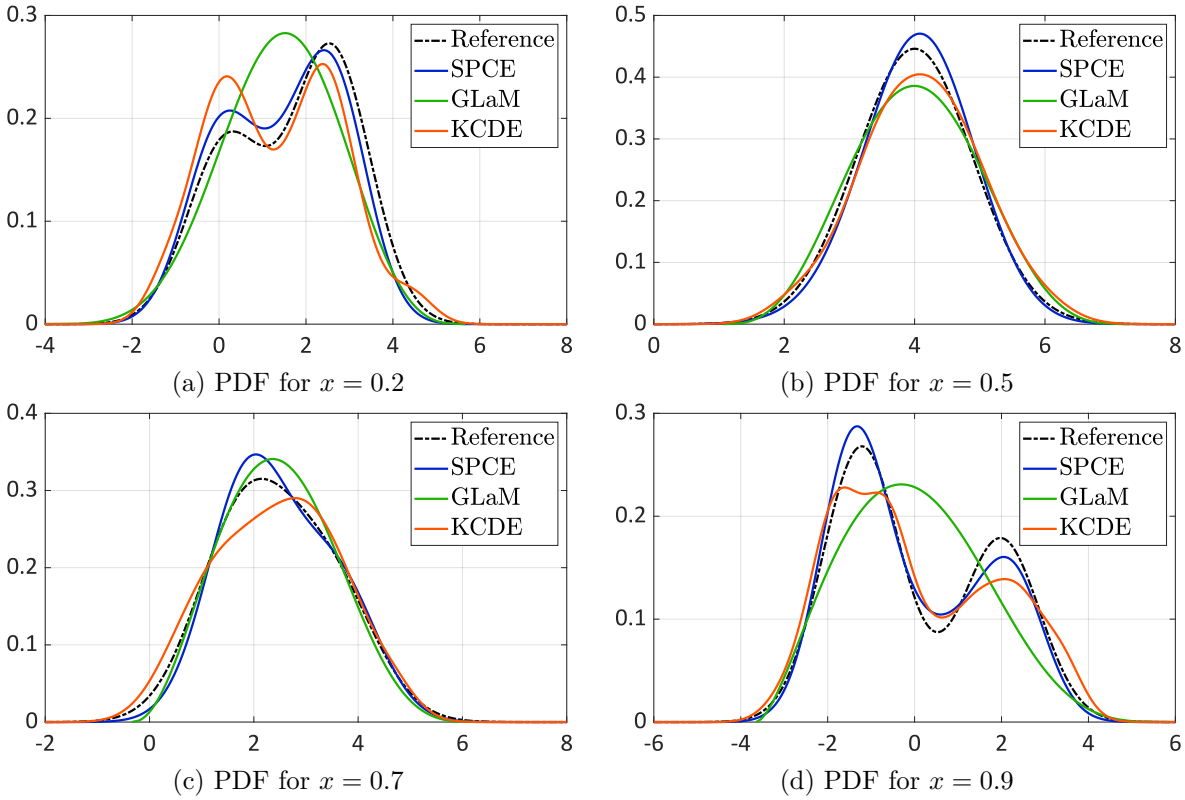


Figure 8: Bimodal analytical example — Illustration of the model with an experimental design of $N = 800$

Figure 9 compares the response PDF estimated by the surrogates built on the experimental

design of Fig. 8 ($N = 800$) for four different values of x. We observe that small values of x yield a bimodal distribution with the higher mode on the right. With x increasing, the two modes merge and form a unimodal distribution at $x = 0.5$. Then, the two modes separate again, which leads to bimodal distributions with the higher mode on the left. This shape variation can also be observed from Fig. 8.

As opposed to the previous two examples, GLaM cannot represent this evolution, since generalized lambda distributions cannot produce multimodal distributions. In contrast, SPCE and KCDE capture well the bimodality and also the shape variation. Moreover, in Fig. 9c the higher mode is moving to the left, which is a feature not exhibited by KCDE but correctly captured by SPCE.



(a) PDF for $x = 0.2$

(b) PDF for $x = 0.5$

(c) PDF for $x = 0.7$

(d) PDF for $x = 0.9$

Figure 9: Bimodal analytical example — Comparisons of the emulated PDFs, $N = 800$.

Quantitative comparisons for $N \in \{100; 200; 400; 800; 1{,}600\}$ in Fig. 10 confirm our observation in Fig. 9. Because of the bimodality, GLaM provides the least accurate approximation. When increasing N, it converges slowly to the same error as the best normal approximation which is clearly outperformed by the best two surrogates: SPCE and KCDE for $N \geq 800$. Both SPCE and KCDE show a consistent decay of the error. Only when a few samples $N = 100$ are available does KCDE provide stabler estimates (the spread of the error is small) and better average performance. For $N \geq 200$, SPCE yields more accurate results and exhibits an overall faster rate of convergence. In summary, this example demonstrates that SPCE can represent bimodal distributions with a high accuracy.

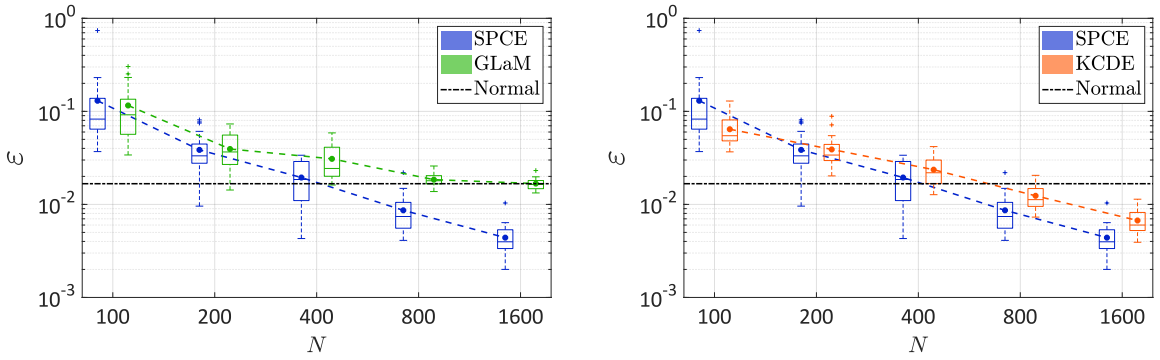


Figure 10: Bimodal analytical example — Comparison of the convergence of the surrogate models. The dashed lines denote the average value over 20 repetitions of the full analysis. The black dash-dotted line represents the error of the model assuming that the response distribution is normal with the true mean and variance.

6 Conclusions

In this paper, we present a novel surrogate model called stochastic polynomial chaos expansions (SPCE) to emulate the response distribution of stochastic simulators. This surrogate is an extension of the classical polynomial chaos expansions developed for deterministic simulators. In order to represent the intrinsic stochasticity of the simulator, we combine a latent variable with the well-defined inputs to form a polynomial chaos representation. In addition, we introduce an additive Gaussian noise as a regularizer. We propose using the maximum likelihood estimation for calibrating the coefficients $\boldsymbol{c}$ of the polynomial basis. The standard deviation σ of the noise variable is a hyperparameter that regularizes the optimization problem for the polynomial coefficients $\boldsymbol{c}$ and is tuned by cross-validation to avoid overfitting. The cross-validation score is also used as a model selection criterion to choose an appropriate truncation scheme for the polynomial chaos expansion in an adaptive manner, and the most suitable distribution for the latent variable. As seen from the presentation and the application examples, the proposed method does not require replications.

The performance of the developed method is illustrated on examples from mathematical finance and epidemiology and on an analytical example showcasing a bimodal response distribution. The results show that SPCE is able to well approximate various response distributions whether unimodal or not, with a reasonable number of model runs.

Using an appropriate error measure defined in Eq. (41), SPCE is compared with the generalized lambda model (GLaM) and one state-of-the-art kernel conditional density estimator (KCDE). In the first two examples where the response distribution is unimodal, SPCE noticeably outperforms KCDE and provides slightly more accurate results than GLaM which is known for its flexibility for representing unimodal distributions. In the last example featuring bimodal distributions which cannot be well approximated by generalized lambda distributions, SPCE can still capture the complex shape variation and yields smaller errors than KCDE. All in all, SPCE generally performs as the best against the various competitors considered in this study.

Applications of the proposed method to complex engineering problems, such as wind turbine design [49] and structural dynamics [50], should be considered in future investigations. Statistical properties (e.g., consistency and asymptotics) of the maximum likelihood estimation used in SPCE remains to be studied. This will allow for assessing the uncertainty in the estimation procedure.

Finally, the proposed approach has been validated so far only for problems with small to moderate dimentionality. To improve the efficiency and performance of SPCE in high dimensions, models that have a general sparse structure (not only regarding the mean function) are currently under investigations.

Acknowledgments

This paper is a part of the project "Surrogate Modeling for Stochastic Simulators (SAMOS)" funded by the Swiss National Science Foundation (Grant #200021_175524), whose support is gratefully acknowledged.

A Appendix

A.1 Upper bound

In this section, we demonstrate that the leave-one-out error obtained from fitting the mean function Eq. (28) provides an upper bound for σ^2.

Taking the expectation of Eq. (35) with respect to $\boldsymbol{X}$, it holds

$$\mathbb{E}\left[\mathrm{Var}\left[\tilde{Y}\,\middle|\,\boldsymbol{X}\right]\right] = \mathbb{E}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}^2\psi_{\boldsymbol{\alpha}}^2(\boldsymbol{X}) + \sigma^2\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathcal{A}_m} c_{\boldsymbol{\alpha}}^2 + \sigma^2. \tag{48}$$

The leave-one-out error $\varepsilon_{\mathrm{LOO}}$ in the mean-fitting process is an estimate of $\mathbb{E}\left[(\hat{m}(\boldsymbol{X}) - Y_{\boldsymbol{X}})^2\right]$ [51]. The latter can be decomposed as

$$\begin{aligned}\mathbb{E}\left[(\hat{m}(\boldsymbol{X}) - Y_{\boldsymbol{X}})^2\right] &= \mathbb{E}\left[(\hat{m}(\boldsymbol{X}) - m(\boldsymbol{X}) + m(\boldsymbol{X}) - Y_{\boldsymbol{X}})^2\right] \\ &= \mathbb{E}\left[(\hat{m}(\boldsymbol{X}) - m(\boldsymbol{X}))^2\right] + \mathbb{E}\left[\mathrm{Var}\left[Y \mid \boldsymbol{X}\right]\right]\end{aligned}. \tag{49}$$

Aiming at approximating $Y_{\boldsymbol{x}}$ with $\tilde{Y}_{\boldsymbol{x}}$, we have $\mathbb{E}\left[\mathrm{Var}\left[Y \mid \boldsymbol{X}\right]\right] \approx \mathbb{E}\left[\mathrm{Var}\left[\tilde{Y}\,\middle|\,\boldsymbol{X}\right]\right]$. Hence, $\varepsilon_{\mathrm{LOO}}$ provides an upper bound for Eq. (48) and therefore for σ^2.

References

[1] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, New Jersey, 2005.

[2] T. Britton. Stochastic epidemic models: a survey. *Math. Biosci.*, 225:24–35, 2010.

[3] R. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, Mineola, 2nd edition, 2003.

[4] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts, Internet edition, 2006.

[5] B. Ankenman, B.L. Nelson, and J. Staum. Stochastic Kriging for simulation metamodeling. *Oper. Res.*, 58:371–382, 2010.

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

[7] J. M. Wooldridge. *Introductory Econometrics: A Modern Approach*. Cengage Learning, 5th edition, 2013.

[8] M. Binois, R. B. Gramacy, and M. Ludkovski. Practical heteroscedastic Gaussian process modeling for large simulation experiments. *J. Comput. Graph. Stat.*, 27:808–821, 2018.

[9] R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46:33–50, 1978.

[10] M. Plumlee and R. Tuo. Building accurate emulators for stochastic simulations via quantile Kriging. *Technometrics*, 56:466–473, 2014.

[11] L. Torossian, V. Picheny, R. Faivre, and A. Garivier. A review on quantile regression for stochastic computer experiments. *Reliab. Eng. Sys. Safety*, 201, 2020.

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

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

[14] S. Azzi, B. Sudret, and J. Wiart. Surrogate modeling of stochastic functions - application to computational electromagnetic dosimetry. *Int. J. Uncertainty Quantification*, 9:351–363, 2019.

[15] N. Lüthen, S. Marelli, and B. Sudret. Surrogates of stochastic simulators using trajectories. *Prob. Eng. Mech.*, 2022. (in preparation).

[16] P. McCullagh and J. Nelder. *Generalized Linear Models*, volume 37 of *Monographs on*

Statistics and Applied Probability. Chapman and Hall/CRC, 1989.

[17] T. Hastie and R. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, 1990.

[18] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*, volume 66 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, 1996.

[19] P. Hall, J. Racine, and Q. Li. Cross-validation and the estimation of conditional probability densities. *J. Amer. Stat. Assoc.*, 99:1015–1026, 2004.

[20] S. Efromovich. Dimension reduction and adaptation in conditional density estimation. *J. Amer. Stat. Assoc.*, 105:761–774, 2010.

[21] X. Zhu and B. Sudret. Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA J. Unc. Quant.*, 9:1345–1380, 2021.

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

[23] A. Nataf. Détermination des distributions dont les marges sont données. *C. R. Acad. Sci. Paris*, 225:42–43, 1962.

[24] M. Rosenblatt. Remarks on a multivariate transformation. *Ann. Math. Stat.*, 23:470–472, 1952.

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

[26] O. G. Ernst, A. Mugler, H. J. Starkloff, and E. Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Math. Model. and Num. Anal.*, 46:317–339, 2012.

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

[28] W. Gautschi. *Orthogonal polynomials: computation and approximation*. Oxford University Press, 2004.

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

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

[31] A. Doostan and H. Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.*, 230(8):3015–3034, 2011.

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

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

[34] N. Lüthen, S. Marelli, and B. Sudret. A benchmark of basis-adaptive sparse polynomial chaos expansions for engineering regression problems. *Int. J. Uncertainty Quantification*, 2021. (submitted).

[35] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Safety*, 93:964–979, 2008.

[36] B. S. Everitt. *An Introduction to Latent Variables Models*. Chapman & Hall, 1984.

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

[38] J. Jacod and P. Protter. *Probability Essentials*. Springer, 2nd edition, 2004.

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

[40] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.

[41] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition, 1987.

[42] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

[43] T. Hayfield and J.S. Racine. Nonparametric Econometrics: The np Package. *J. Stat. Softw.*, 27:1015–1026, 2008.

[44] C. Villani. *Optimal transport, old and new*. Cambridge Series in Statistical and Probabilistic Mathematics. Springer, Cambridge, 2000.

[45] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[46] K. Reddy and V. Clinton. Simulating stock prices using geometric Brownian motion: Evidence from Australian companies. *Australasian Accounting, Business and Finance Journal*, 10(3):23–47, 2016.

[47] S. Shreve. *Stochastic Calculus for Finance II*. Springer, New York, 2004.

[48] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–2361, 1977.

[49] I. Abdallah, C. Lataniotis, and B. Sudret. Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators – application to extreme loads on wind turbines. *Prob. Eng. Mech.*, 55:67–77, 2019.

[50] C. V. Mai, K. Konakli, and B. Sudret. Seismic fragility curves for structures using non-

parametric representations. *Frontiers Struct. Civ. Eng.*, 11(2), 2017.

[51] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* Springer, 2014.