ETH zürich

Introducing latent variables in polynomial chaos expansions to surrogate 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:	Proc. 13th International Conference on Structural Safety & Reliability, Tongji University, Shanghai (China), September 13-17
Report Ref.:	RSUQ-2022-009
Arxiv Ref.:	
DOI:	-
Date submitted:	December 16, 2021
Date accepted:	September 13, 2022

Introducing latent variables in polynomial chaos expansions to surrogate stochastic simulators

Xujia Zhu*[1] and Bruno Sudret†[1]

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

September 23, 2022

Abstract

Stochastic simulators are computational models that produce different results when evaluated repeatedly with the same input parameters. In this respect, the model response conditional on the input is a random variable, and thus it is necessary to run the model many times to fully characterize the associated response. Due to the large number of necessary model runs, performing uncertainty quantification or optimization of a costly stochastic simulator is intractable directly. To alleviate the computational burden, we extend polynomial chaos expansions to metamodeling the entire response probability distribution of stochastic simulators. In this novel approach, we introduce a latent variable and an additional noise, on top of the well-defined input variables, to mimic the intrinsic stochasticity of the simulator. We develop a method to construct such a surrogate *without* requiring repeated runs of the simulator for the same input parameters. The performance of the proposed surrogate model is compared with one of the state-of-the-art kernel estimator on an analytical example from mathematical finance.

1 Introduction

Computational models, a.k.a. simulators, are virtual prototypes that represent operational or physical processes through computer simulations. They have been widely used in modern engineering and applied science for analyzing complex systems. Conventional simulators usually depict a deterministic relation between the model input and output, i.e., for a given set of input parameters the model output is a deterministic value. In contrast, various runs of a stochastic simulator for the same input parameters result in different values of the model response. In other words, the output of a stochastic simulator conditional on the input is a random variable. Such a

*zhu@ibk.baug.ethz.ch

†sudret@ethz.ch

stochastic behavior results from some latent variables that are not explicitly taken into account as part of the input. As a result, when the input parameters are fixed but the latent variables vary randomly, the associated model response remains uncertain.

A stochastic simulator can be mathematically defined by

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

where $\mathcal{D}_{\boldsymbol{X}}$ is the domain of definition of the stochastic simulator, and Ω denotes the sample space accounting for the intrinsic stochasticity. The stochastic simulator is a deterministic function of the input vector $\boldsymbol{x}$ and the element ω of the sample space. However, one can only control $\boldsymbol{x}$ but not ω when evaluating the model. Hence, each model run for a given $\boldsymbol{x}$ provides a single realization of the model output corresponding to a particular $\omega \in \Omega$.

Due to the random nature of the model response, it is necessary to repeatedly evaluate a stochastic model for the same input parameters, called *replications*, to characterize the associated output distribution. This leads to an extra dimension of requirement for model evaluations in the context of uncertainty quantification or optimization, where simulators are required to be evaluated for various input values. As a consequence, a large number of model evaluations are necessary, which is impracticable for costly simulators. In this respect, surrogate models shall be constructed with a limited number of model runs to approximate the original model, and therefore replace the latter.

In this field, large efforts have been devoted to estimating some summary quantities of the response distribution, e.g., the mean and variance [1–3], and the quantiles [4–6], as deterministic functions of the input. However, not many methods have been developed to emulate the entire probability distribution. We classify these methods in the literature into three groups.

The first one [7, 8] capitalizes on replications to estimate (parametrically or nonparametrically) the response distribution for various input values. The estimated distributions (based on replications) are represented by an appropriate parametrization. Then, the associated parameters are cast as deterministic functions of the input variables, which can be represented by classical surrogate models already available for deterministic simulator (e.g., polynomial chaos expansions). Since this approach applies a two-step procedure, it calls for many replications to produce surrogate models of a good accuracy [8].

The second approach views a stochastic simulator as a random field indexed by the input variables [9]. As suggested by Eq. (1), the stochastic simulator is a deterministic function of $\boldsymbol{x}$ for a fixed ω, which can be considered as a *trajectory*. This deterministic function can be emulated by conventional deterministic surrogate models. Based on a set of (emulated) trajectories, one can approximate the underlying random field using the Karhunen-Loève expansion. Since this approach can approximate and produce trajectories, the resulting model allows for representing not only the response distribution but also the general dependence structure of the simulator.

However, this comes with assumptions on the regularity of the trajectories and with the ability to fix the internal randomness, which is practically achieved by controlling the *random seed* of the simulator.

The third approach consists of methods developed in statistics for estimating conditional distributions from real data. As a result, they do not call for replications or controlling the random seed. In fact, a stochastic simulator can be considered as a conditional sampler: one first decides (or samples) the input parameters, and then conditioned on the latter, the simulator samples the associated random model output. As a result, the response distribution is a mere conditional distribution. If the type of the response distribution is known and belongs to the exponential family, one can apply the generalized linear model or the generalized additive model. If the response distribution is arbitrary, nonparametric estimators [10–12] can be used. However, it is well-known that nonparametric estimators suffer from the curse of dimensionality: the accuracy decays fast with increasing dimensionality. To balance between the flexibility and the efficiency, Zhu and Sudret [13] develop the so-called generalized lambda model, which uses the generalized lambda distribution to approximate the response. The distribution parameters as functions of the input are represented by polynomial chaos expansions. This surrogate model is parametric and shown to be flexible, but it is unable to represent multi-model distributions.

To have more flexible parametric surrogate models, we propose to extend the classical polynomial chaos expansions to emulating stochastic simulators in this paper. We introduce a *latent* variable and a *noise* variable to represent the random behavior of the model output. We propose combining the maximum likelihood estimation with cross-validation to construct such a surrogate model.

The paper is organized as follows. We present the general formulation of the stochastic polynomial chaos expansion after a short recap of classical polynomial chaos expansions in Section 2. In Section 3, we present the developed method to build the surrogate model. We illustrate the performance of the proposed method on an analytical example from mathematical finance in Section 4. Finally, we conclude the main findings of the paper and provide outlooks for future research in Section 5.

2 Stochastic polynomial chaos expansions

2.1 Polynomial chaos expansions

Polynomial chaos expansion is a well-established surrogate model for deterministic simulators. Let us consider a deterministic model

$$\begin{aligned} \mathcal{M}_d : \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M &\to \mathbb{R} \\ \boldsymbol{x} &\mapsto \mathcal{M}_d(\boldsymbol{x}). \end{aligned} \tag{2}$$

For uncertainty quantification analysis, the input $\boldsymbol{x}$ is modeled by a random vector $\boldsymbol{X}$ characterized by the joint probability density function (PDF) $f_{\boldsymbol{X}}$. Due to the uncertain input, the model output $Y = \mathcal{M}_d(\boldsymbol{X})$ becomes a random variable.

If Y is a second-order random variable, i.e., $\mathrm{Var}[Y] < +\infty$, $\mathcal{M}_d(\boldsymbol{X})$ belongs to the Hilbert space $\mathcal{H}$ of square-integrable functions, the inner product of which is defined by

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

In this study, we assume that the components of $\boldsymbol{X}$ are mutually independent, which implies that the joint PDF $f_{\boldsymbol{X}}$ is a product of marginal distributions:

$$f_{\boldsymbol{X}} = \prod_{j=1}^{M} f_{X_j}(x_j), \tag{4}$$

where f_{X_j} is the marginal distribution of the j-th component X_j. Under some conditions for each marginal PDF f_{X_j}, the Hilbert space $\mathcal{H}$ is separable with a polynomial basis [14]. Thus, $\mathcal{M}_d$ can be expanded in terms of an orthogonal polynomial basis

$$Y = \mathcal{M}_d(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^M} c_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{5}$$

where $c_{\boldsymbol{\alpha}}$ is the coefficient of the basis function $\psi_{\boldsymbol{\alpha}}$ that is defined by

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(x_j). \tag{6}$$

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

$$\mathbb{E}\left[\phi_k^{(j)}(X_j)\,\phi_l^{(j)}(X_j)\right] = \delta_{kl}, \tag{7}$$

where the Kronecker delta δ_{kl} values 1 if $k = l$ and 0 otherwise.

The expansion in Eq. (5) involves an infinite sum, which is practically truncated to finite terms. The common truncation scheme is to include all the basis functions whose total degree is inferior to a given value p, i.e., $\mathcal{A}^p = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M, \sum_{j=1}^{M} \alpha_j \leq p\right\}$.

2.2 Latent variable model

Polynomial chaos expansions have been widely used to represent deterministic simulators. Due to the deterministic input-output relation represented in Eq. (5), however, it cannot directly be used to emulate stochastic simulators. In this section, we present the extension of polynomial chaos expansions to a latent variable model.

For the purpose of clarity, we denote $Y_{\boldsymbol{x}}$ the model response of the stochastic simulator for the input parameters $\boldsymbol{x}$. When considering the uncertain input variables, i.e., modeled by a random vector $\boldsymbol{X}$, we define Y as the model output aggregating all the uncertainties from both the input and the intrinsic stochasticity.

In order to mimic the intrinsic stochasticity of stochastic simulators, we include a latent variable Z in the polynomial expansion as follows:

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{=} \sum_{\boldsymbol{\alpha}\in\mathbb{N}^{M+1}} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z). \tag{8}$$

For a given $\boldsymbol{x}$, the expansion on the right-hand side is a function of the latent variable Z, thus a random variable. The equality in Eq. (8) is to be understood *in distribution*, in the sense that two random variables follow the same probability distribution. As a result, the latent variable Z is only introduced to reproduce the randomness. It does not represent the detailed data generation process (involving the intrinsic stochasticity) of the simulator.

Remark. *The latent variable model applies a polynomial transform of a random Z to represent the response of stochastic simulators. This is inspired by the isoprobabilistic transform that is common in structural reliability analysis [15]: we can transform a random variable Z to any desired distribution. Denote $F_{Y_{\boldsymbol{x}}}(y)$ the cumulative distribution function (CDF) of $Y_{\boldsymbol{x}}$. By using the isoprobabilistic transform, we have*

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{=} F^{-1}_{Y_{\boldsymbol{x}}}(F_Z(Z)), \tag{9}$$

where F_Z is the CDF of Z. Here, $F^{-1}_{Y_{\boldsymbol{x}}}(F_Z(Z))$ is a deterministic function of both $\boldsymbol{x}$ and $\boldsymbol{z}$.

Equation (8) can be seemingly interpreted as a representation of Eq. (9). However, Eq. (8) is more general: it does not require the explicit expansion in Eq. (9) but only calls for approximating the response distribution. Because there can be many transforms that achieve the goal, the expansion in Eq. (8) is not unique.

When applying a truncation scheme $\mathcal{A}$ to the polynomial chaos expansion, we introduce an additional noise variable ϵ to represent the approximation error. Thereby, we express the stochastic polynomial chaos expansion for representing stochastic simulators by

$$Y_{\boldsymbol{x}} \overset{\mathrm{d}}{\approx} \tilde{Y}_{\boldsymbol{x}} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z) + \epsilon, \tag{10}$$

where ϵ is a centered Gaussian random variable, i.e., $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$.

The choice of the truncation scheme and the type of the latent variable Z depend on the prior knowledge we may have on the stochastic simulator. In the following example, we set the truncated set $\mathcal{A} = \mathcal{A}^p$ with the maximum degree $p = 3$. The latent variable Z is selected to be standard normal, i.e., $Z \sim \mathcal{N}(0, 1)$.

3 Estimation method

With a given truncation scheme $\mathcal{A}$ and distribution f_Z of the latent variable Z, one needs now to determine the coefficients $\boldsymbol{c}$ and the standard deviation σ of the noise variable to construct the

surrogate model defined in Eq. (10). Because the response distribution is unknown, we cannot emulate directly Eq. (9). In this section, we present a method developed for calibrating these parameters from data without requiring replications.

First, we evaluate the simulator to generate data for the estimation. We consider an *experimental design* $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$. The stochastic model is evaluated *once* for each point $\boldsymbol{x}^{(i)}$, and the associated model response $y^{(i)} = \mathcal{M}_s\left(\boldsymbol{x}^{(i)}, \omega^{(i)}\right)$ is collected in $\mathcal{Y} = \left\{y^{(1)}, \ldots, y^{(N)}\right\}$. The notation $\omega^{(i)}$ means that a different random seed may be used for each run (i.e., we do not control the random seed).

3.1 Maximum likelihood estimation

In this first part, we assume that the variance σ^2 of the noise variable ϵ is known. Its estimation will be presented in Section 3.3. According to the definition in Eq. (10), the PDF of $\tilde{Y}_{\boldsymbol{x}}$ can be expressed by

$$\begin{aligned} f_{\tilde{Y}_{\boldsymbol{x}}}(y) &= \int_{\mathcal{D}_Z} f_{Y_{\boldsymbol{x}}|Z}(y \mid z)\, f(z)\, \mathrm{d}z \\ &= f_\epsilon\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)\right) f_Z(z)\, \mathrm{d}z, \end{aligned} \tag{11}$$

where f_Z and f_ϵ are the PDFs of Z and ϵ, respectively. As $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$, f_ϵ has an explicit form, and thus the conditional likelihood for $(\boldsymbol{x}, y)$ becomes

$$\begin{aligned} &l(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) = \\ &\int_{\mathcal{D}_Z} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z)\right)^2}{2\sigma^2}\right) f_Z(z)\mathrm{d}z. \end{aligned} \tag{12}$$

This integral does not have a closed-form solution. We can use Gaussian quadrature [16] to evaluate it numerically:

$$\begin{aligned} &l(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) \approx \tilde{l}(\boldsymbol{c}; \boldsymbol{x}, y, \sigma) \\ &= \sum_{j=1}^{N_Q} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, z_j)\right)^2}{2\sigma^2}\right) w_j, \end{aligned} \tag{13}$$

where N_Q is the number of integration points, and z_j and w_j denote the j-th node and the associated weight, respectively. Based on Eq. (13), we propose using maximum likelihood estimation (MLE) to calibrate the coefficients $\boldsymbol{c}$ from the data $(\mathcal{X}, \mathcal{Y})$

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \sum_{i}^{N} \log\left(\tilde{l}\left(\boldsymbol{c}; \boldsymbol{x}^{(i)}, y^{(i)}, \sigma\right)\right). \tag{14}$$

The derivative of Eq. (13) with respect to $\boldsymbol{c}$, and therefore of the objective function in Eq. (14), can be analytically derived. Hence, we opt for the gradient-based Broyden-Fletcher-Goldfarb-Shanno algorithm [17] to solve Eq. (14).

3.2 Starting point

The optimization problem in Eq. (14) can be non-convex and highly nonlinear. As a result, it is important to initiate the optimization algorithm with a good starting point. Using the properties of the orthogonal polynomials, the mean function of the surrogate $\tilde{Y}_{\boldsymbol{x}}$ can be expressed as

$$\begin{aligned}\mathbb{E}\left[\tilde{Y}_{\boldsymbol{X}}\right] &= \mathbb{E}_{Z,\epsilon}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}, Z) + \epsilon\right] \\ &= \sum_{\boldsymbol{\alpha}\in\mathcal{A},\alpha_z=0} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}),\end{aligned} \tag{15}$$

where α_z is the degree of the basis function in Z. As indicated by $\alpha_z = 0$, the mean function of the surrogate model contains all the terms not involving z. Because the surrogate model aims at representing the distribution of $Y_{\boldsymbol{x}}$, Eq. (15) should approximate the mean function of the model response $\mathbb{E}[Y_{\boldsymbol{x}}]$. Therefore, we can fit the mean function with the basis defined by $\mathcal{A}_m = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_z = 0\}$. The estimated coefficients are then used as a starting point for $\{c_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}_m\}$. For other coefficients defined by $\mathcal{A} \setminus \mathcal{A}_m$, we randomly initialize their value.

To estimate the mean function, we apply the hybrid least-angle regression (LAR) developed in Blatman and Sudret [18]. This algorithm is a sparse solver that selects the most important basis among a candidate set, i.e., $\mathcal{A}_m$. To reduce the number of unknowns in Eq. (14), we set the coefficients to zero for the basis functions that are not selected by the sparse algorithm. In other words, we only estimate the coefficients associated with the basis functions that are either selected by the hybrid LAR for the mean function estimation or defined by $\mathcal{A} \setminus \mathcal{A}_m$.

3.3 Estimation of σ

One may consider estimating σ in the same way as $\boldsymbol{c}$. However, if we include σ into the maximum likelihood estimation Eq. (14), the likelihood function can reach $+\infty$ for $\sigma = 0$ and certain choices of $\boldsymbol{c}$. To see this, let us consider a simple stochastic simulator without input variables, which gives a realization of Y upon each model evaluation. We set $\sigma = 0$, so the noise variable vanishes. The stochastic surrogate model is $\tilde{Y} = g(Z) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(Z)$. Based on a change of variable, the PDF of $\tilde{Y} = g(Z)$ is

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

where $\mathbb{1}$ denotes the indicator function, and g' is the derivative of g. For a given y_0 and z_0 with $f_Z(z_0) > 0$, the two equations $g(z_0) = y_0$ and $g'(z_0) = 0$ are linear in $\boldsymbol{c}$ and under-determined for $\boldsymbol{c}$ having more than 3 components. As a result, we can find a set of the coefficients $\boldsymbol{c}$ such that the denominator of Eq. (16) is equal to 0, and thus the likelihood $f_{\tilde{Y}}(y_0)$ values $+\infty$. The example above shows that ϵ plays the rule of a regularizer: if it is not contained in Eq. (10), the MLE presented in Section 3.1 would fail. Consequently, σ is a hyperparameter and should be fitted separately from $\boldsymbol{c}$.

In this paper, we tune σ by cross-validation [19]. To this end, the available data are randomly split into K equal sized groups $\{V_k : k = 1, \ldots, K\}$. For each group V_k of data, we hold it out as a validation set. The other $K-1$ groups are used to build a surrogate model by solving Eq. (14). The estimated coefficients are denoted by $\hat{\boldsymbol{c}}_k(\sigma)$, where we explicitly express σ as an argument to emphasize the dependence of $\hat{\boldsymbol{c}}$ on σ. The likelihood of the constructed model is evaluated on the validation set V_k

$$\mathbf{l}_k(\sigma) = \sum_{i \in V_k} \log\left(\tilde{l}\left(\hat{\boldsymbol{c}}_k(\sigma), \sigma; \boldsymbol{x}^{(i)}, y^{(i)}\right)\right). \tag{17}$$

We repeat this process for each fold of the partition $\{V_k : k = 1, \ldots, K\}$, which gives K scores from Eq. (17). The optimal σ is selected as the one that maximizes the out-of-sample performance, that is,

$$\hat{\sigma} = \arg\max_{\sigma} \sum_{k=1}^{K} \mathbf{l}_k(\sigma). \tag{18}$$

Equation (18) is a one-dimensional optimization problem for σ. However, the derivative of the objective function is generally difficult to derive due to the nested optimization for $\hat{\boldsymbol{c}}$. Therefore, we choose to use the derivative-free Nelder-Mead method [20] for σ selection.

4 Numerical experiment

To illustrate the performance of the stochastic polynomial chaos expansion (SPCE), we consider an example of the geometric Brownian motion

$$\mathrm{d}S_t = x_1\, S_t\, \mathrm{d}t + x_2\, S_t\, \mathrm{d}W_t, \tag{19}$$

where W_t is a standard Wiener process, and the model parameters $\boldsymbol{x} = (x_1, x_2)^T$ are called the drift and volatility, respectively. Without loss of generality, the initial value at $t = 0$ is set to $S_0 = 1$. Because W_t is a random process that introduces stochasticity into the differential equation, Eq. (19) is a stochastic simulator: S_t is a random process for a given input vector $\boldsymbol{x}$.

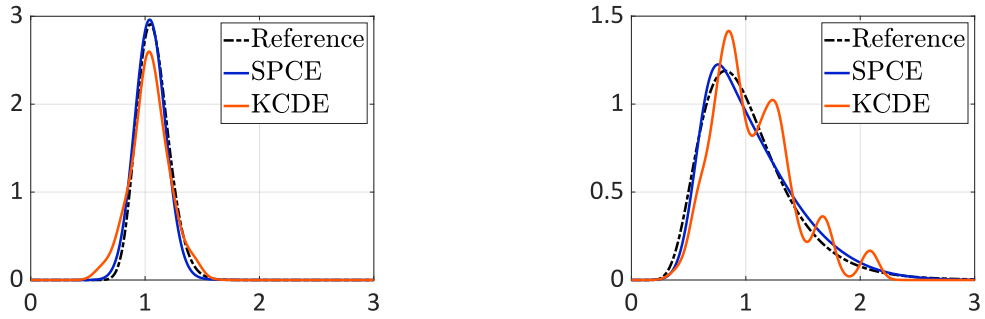
The geometric Brownian motion defined in Eq. (19) is usually used in mathematical finance to model the dynamics of the value S_t of a stock [21]. In this case, x_1 and x_2 correspond to the expected return rate and the volatility of the stock, respectively. We set $X_1 \sim \mathcal{U}(0, 0.1)$ and $X_2 \sim \mathcal{U}(0.1, 0.4)$ to account for the input uncertainty, where the ranges cover the parameters calibrated from real data of the stock market [22].

In this example, we are interested in the value of S_t at $t = 1$, i.e., $Y_{\boldsymbol{x}} = S_1$. The response distribution of $Y_{\boldsymbol{x}} = S_1$ can be derived by solving Eq. (19) using Itô calculus:

$$Y_{\boldsymbol{x}} \sim \mathcal{LN}\left(x_1 - \frac{x_2^2}{2}, x_2\right). \tag{20}$$

As the distribution of $Y_{\boldsymbol{x}}$ is lognormal, we can evaluate the stochastic simulator by sampling directly from the response distribution Eq. (20).

To generate data for surrogate modelling, we use Latin hypercube sampling (LHS) [23] to create the experimental design (ED). The stochastic simulator is evaluated once for each point of the ED. We apply the method developed in Section 3 to build the surrogate. For the purpose of comparison, we consider a state-of-the-art non-parametric kernel conditional density estimator (KCDE) from the package `np` [24] implemented in R.



(a) PDF for $\boldsymbol{x} = (0.07, 0.13)^T$ (b) PDF for $\boldsymbol{x} = (0.02, 0.38)^T$

Figure 1: Comparisons of emulated PDFs, $N = 400$.

Figure 1 shows two response PDFs predicted by the two surrogate models SPCE and KCDE constructed from an experimental design of size $N = 400$. We observe that SPCE gives more accurate predictions: on the left panel Fig. 1a with low volatility (0.13), SPCE represents better the mode and the tails of the response distribution; on the right panel Fig. 1b with high volatility (0.38), KCDE yields spurious oscillations but SPCE stably captures the variation of the response PDF.

To quantitatively assess the performance of the surrogate model, we use the expected normalized Wasserstein distance defined in Zhu and Sudret [13] as an error metric. More precisely, it is defined by

$$e = \mathbb{E}_{\boldsymbol{X}}\left[\frac{d_{\mathrm{WS}}\left(Y_{\boldsymbol{X}}, \tilde{Y}_{\boldsymbol{X}}\right)}{\sqrt{\mathrm{Var}\left[Y\right]}}\right] \tag{21}$$

where $Y_{\boldsymbol{X}}$ is the model response of the simulator, $\tilde{Y}_{\boldsymbol{X}}$ denote that of the emulator, and d_{WS} is the *Wasserstein distance of order two* [25] defined by

$$\begin{aligned} d_{\mathrm{WS}}\left(Y_1, Y_2\right) &\overset{\mathrm{def}}{=} \|Q_1 - Q_2\|_2 \\ &= \sqrt{\int_0^1 \left(Q_1(u) - Q_2(u)\right)^2 \mathrm{d}u}\,, \end{aligned} \tag{22}$$

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

To evaluate the expectation in Eq. (21), we generate a test set $\mathcal{X}_{\mathrm{test}}$ of size $N_{\mathrm{test}} = 1{,}000$ using LHS. The normalized Wasserstein distance is calculated for each point $\boldsymbol{x} \in \mathcal{X}_{\mathrm{test}}$ and then averaged over N_{test}.

Experimental designs of various sizes $N \in \{100; 200; 400; 800; 1{,}600\}$ are investigated to study the convergence property of the surrogate models in comparison. Each scenario is run 20 times

with independent experimental designs to account for uncertainties in the LHS design and the intrinsic stochasticity of the stochastic model. As a consequence, error estimates for each N are represented by box plots.

For quantitative comparisons, we consider an additional surrogate model, where we represent the response distribution with a Gaussian distribution. The associated mean and variance as functions of the input $\boldsymbol{x}$ are set to the *true* values of the simulator. Therefore, the error associated to this surrogate model only comes from the model misspecification, which measures how close the response distribution is to normal. Because the true mean and variance are used, this model is the "oracle" of Gaussian-type mean-variance models, such as the ones presented in Marrel et al. [2] and Binois et al. [3].

Figure 2 summarizes the error metric e defined in Eq. (21) with respect to the size of experimental design. Because the response distribution is lognormal, the Gaussian approximation is not exact. The average error of SPCE built on $N = 400$ model runs is similar to the oracle Gaussian approximation, which is merely achieved by KCDE for $N = 1{,}600$. For $N > 400$, SPCE clearly provides more accurate results than the oracle Gaussian model. Compared with KCDE, the average performance of SPCE is always better for all sizes of ED. For large N, namely $N = 1{,}600$, the average error of SPCE is less than half of that of KCDE. Furthermore, SPCE demonstrates much a faster decay of the error.

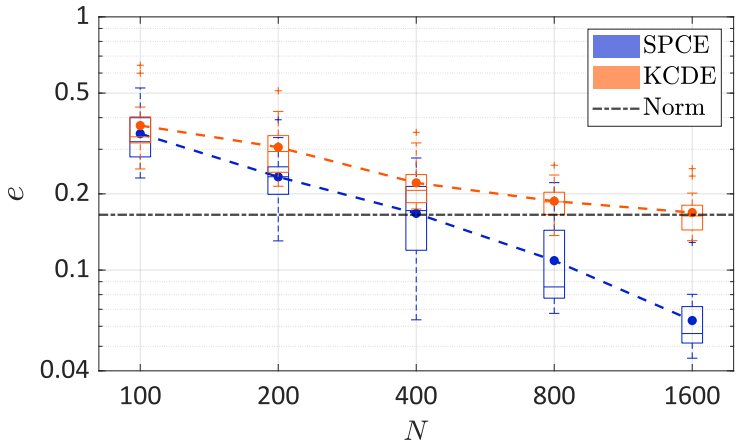


Figure 2: Comparison of the convergence of SPCE and KCDE in terms of the expected normalized Wasserstein distance as a function of the size of the experimental design. The dashed lines denote the average value over 20 repetitions of the full analysis. The black dash-dotted line represents the error of the model assuming that the response distribution is Gaussian with the true mean and variance.

5 Conclusions

In this paper, we extend the classical polynomial chaos expansion to emulating the response distribution of stochastic simulator. The novel surrogate model, called stochastic polynomial

chaos expansions, includes a latent variable in the expansion together with the well-defined input variables. Moreover, we introduce an additive noise, which not only represents the truncation error but also regularizes the likelihood function. We propose using maximum likelihood estimation to calibrate the polynomial coefficients from data. The standard deviation of the noise term is a hyperparameter selected by cross-validation. The developed estimation procedure features no need for replications.

The performance of SPCE is illustrated on an analytical example from mathematical finance. The results show that SPCE is able to well represent the model response with a reasonable number of model runs. In this example with the response distribution being lognomal, SPCE fitted from finite data outperforms the oracle Gaussian approximation. Compared with one state-of-the-art kernel estimator, SPCE yields more accurate results and demonstrates a better convergence rate.

In the current development, the truncation scheme and the distribution of the latent variable are manually selected. We are developing adaptive algorithms that enable an automatic selection procedure for these two quantities from the data. More realistic models will be considered to test the performance of the proposed method. In future research, it would be valuable to develop sparse algorithms to improve the performance of the surrogate model in high-dimensional problems. Finally, some statistical properties of the estimation method (e.g., consistency, asymptotics, error bound) remain to be investigated.

References

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

[2] 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.

[3] 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.

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

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

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

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

[8] 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.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] M. Lemaire. *Structural reliability*. Wiley, 2009.

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

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

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

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

[20] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[21] 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.

[22] 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.

[23] 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.

[24] T. Hayfield and J.S. Racine. Nonparametric Econometrics: The np Package. *J. Stat. Softw.*,

27:1015–1026, 2008.

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