Interpolation methods to compute statistics of a stochastic partial differential equation

Daniela Steffes-lai (1), Eveline Rosseel (2), Tanja Clees (1)

(1) Fraunhofer SCAI, Sankt Augustin, Germany,
(2) Katholieke Universiteit Leuven, Leuven, Belgium

May 11, 2014

Abstract

This paper considers the analysis of partial differential equations (PDE) containing multiple random variables. Recently developed collocation methods enable the construction of high-order stochastic solutions by converting a stochastic PDE into a system of deterministic PDEs. This interpolation method requires that the probability distribution of all random input variables is known a priori, which is often not the case in industrially relevant applications. Additionally, this method suffers from a curse of dimensionality, i.e., the number of deterministic PDEs to be solved grows exponentially with respect to the number of random variables. This paper presents an alternative interpolation method, based on a radial basis function (RBF) metamodel, to compute statistics of the stochastic PDE. The RBF metamodel can be constructed even if the probability distribution of all random variables is not known. Then, a lot of statistic scenarios with different probability distributions of the random variables can be computed with this single metamodel. In order to reduce the model complexity, we present a parameter screening technique which can be combined with an interpolation method to solve a reduced stochastic model. Numerical results of a model problem demonstrate that the RBF metamodel is as fast as a low order collocation approach and achieves a good accuracy. The parameter screening is able to reduce the dimension and, thus, to accelerate the computation of the stochastic solution.

1 Introduction

During the fabrication of products important material and process parameters, geometries and also external influences can vary considerably. These variations can have a substantial influence on the quality of the resulting products. The current trend goes to more accuracy and therefore more robust predictions. Hence, there is a high need for efficient stochastic computations, combined with an efficient analysis of sensitivity and robustness, see, e.g., the review papers [38, 23, 26, 36]. The objective is to model uncertainty from the beginning
of the simulation, and not as criterion afterwards. Mathematically, parameters which are known only approximately can be modeled as stochastic variables or processes. The model takes the form of a partial differential equation (PDE) with coefficients that are random variables or random fields, e.g., in the differential operator, the initial or boundary conditions. These random variables are mostly not well known in practice.

We seek for solutions of a stochastic PDE, called stochastic solution, that contain all possible probability information, including a complete description of the probability density function (pdf). Until recently Monte Carlo simulations were the standard tool for solving stochastic PDEs. Today, new stochastic finite element methods [38, 23] are becoming popular because they enable the computation of high-order stochastic solutions with substantially less computational effort than Monte Carlo simulations. A prominent interpolation variant is the collocation method [11, 22, 39]. This approach transforms the stochastic PDE into a set of decoupled deterministic PDEs. However, this method suffers from the “curse of dimensionality” [4, 12], that is, the computational cost grows exponentially as a function of the number of random variables.

The main focus of this work is the computation of statistics of a stochastic PDE by means of interpolation methods. We investigate an alternative approach to the collocation method, namely a radial basis function (RBF) metamodel, which is accelerated by a singular value decomposition (SVD). The difference to the collocation approach are mainly the construction of the sample points and the interpolation function itself. The accelerated metamodel does not require the probability distribution of the random input variables in the setup phase. The probability distribution of the random input variables is only used to compute statistics of the solution.

Additionally, we perform a sensitivity analysis, which determines the dependence of the solution on the variation of random variables, also called uncertain design-parameters. One goal of a sensitivity analysis is to identify the design-parameters possessing the most influence on the solution. This subject is often also called parameter screening. We show that the parameter screening approach and the accelerated metamodel approach can be combined in order to reduce the curse of dimensionality.

We investigate the computation of statistics of the solution. We are interested in quantifying the robustness of the solution, that is, a small change in the design-parameters should also result in only a small change of the solution. Therefore, we will focus on the median and the 0.68-quantile of the solution. We are not primarily interested in the tails of the pdf, since these represent the mostly improbable cases, or worst case scenarios. The tails of the pdf are investigated in a reliability analysis, which is beyond our scope.

This paper is organized as follows. Section 2 introduces the stochastic partial differential equation to be solved. Section 3 presents a simple and effective approach for parameter screening, together with a discussion of its advantages and limitations. The collocation approach and a RBF metamodel to compute statistics of the solution are presented in Section 4. Section 5 describes the model problem for numerical tests. Numerical results of the RBF metamodel approach
are given and compared to those of the collocation method. We conclude the paper with a discussion and some future work issues in Section 6.

2 Problem definition

Consider a stochastic partial differential equation (sPDE)

\[ L(x, \omega; u(x, \omega)) = b(x, \omega) \quad x \in D, \ \omega \in \Omega, \]
\[ u(x, \omega) = g(x, \omega) \quad x \in \partial D, \ \omega \in \Omega, \]
\[ \frac{\partial u(x, \omega)}{\partial n} = h(x, \omega) \quad x \in \partial N, \ \omega \in \Omega, \]

where \( L \) represents a general – possibly nonlinear – differential operator containing stochastic coefficients, \( D \) is the spatial domain enclosed by a Dirichlet boundary \( \partial D \) and a Neumann boundary \( \partial N \), and \( \Omega \) is a sample space.

We make use of the finite dimensional noise assumption \( [1] \). This assumption states that all sources of randomness can be approximated by a finite (small) number of random variables. This is, e.g., satisfied when the random processes present in the problem have the form of a truncated Karhunen-Loève expansion \( [20] \). We denote the vector of all random variables in (1) by \( \xi := (\xi_1(\omega), \ldots, \xi_L(\omega)) \), with \( L \) the total number of random variables. Each random variable \( \xi_i(\omega) : \Omega \to \Gamma_i \) is a function from the event space \( \Omega \) onto a part of the real axis, \( \Gamma_i \subset \mathbb{R} \), and is characterized by its probability density function \( \rho_i(y_i) \), with \( y_i \in \Gamma_i \). Assuming that all random variables are independent, their joint probability distribution equals \( \rho(y) = \prod_{i=1}^{L} \rho_i(y_i) \), \( y = (y_1, \ldots, y_L) \in \Gamma = \prod_{i=1}^{L} \Gamma_i \). Using these assumptions, we can rewrite the sPDE (1) as a parametric deterministic PDE

\[ L(x, y; u(x, y)) = b(x, y) \quad x \in D, \ y \in \Gamma, \]
\[ u(x, y) = g(x, y) \quad x \in \partial D, \ y \in \Gamma, \]
\[ \frac{\partial u(x, y)}{\partial n} = h(x, y) \quad x \in \partial N, \ y \in \Gamma. \]

In the context of the parameter screening technique presented in the next section, we will call the random variables \( \xi \) design-parameters. The space spanned by these design-parameters is called design-parameter space.

Note, if (2) is solved on a spatial discretization with \( M \) degrees of freedom, the discretized solution in node \( i \) is denoted by

\[ u_i(y), \quad (i = 1, \ldots, M). \]

3 Parameter screening

We present a parameter screening method that allows the identification of the design-parameters possessing the most influence on the solution. This is used
to reduce the dimension of the design-parameter space and therefore the model complexity. To investigate the model and the influences of variations of design-parameters onto the solution, a special design of experiments is performed and described in Section 3.1. Section 3.2 describes the sensitivity analysis in detail. We start with the computation of an indicator for nonlinearity. Then we compute a measure which also deals with the linear behavior of design-parameters. The advantages and limitations of the presented parameter screening method are discussed in Section 3.3.

### 3.1 Design of experiments

First, a small design of experiments (DoE) [25] is generated, consisting only of $2L + 1$ sampling points $\{\zeta_1, \ldots, \zeta_{2L+1}\}$, with $L$ the total number of random variables. The sampling points represent variations of the random variables $\xi$, often so called design-parameters. Each random variable is varied separately. The first sample corresponds to the mean values of all random variables. In addition, $2L$ samples are generated by sampling each random variable in a minimum and maximum value, while the other random variables are sampled in their mean value.

Next, deterministic simulations on these sampling points are carried out to compute discrete solutions $u_i(\zeta_1), \ldots, u_i(\zeta_{2L+1}), i = 1, \ldots, M$ on a spatial discretization with $M$ spatial degrees of freedom. We define a discrete, deterministic solution vector as

$$u(\zeta_j) = (u_1(\zeta_j), \ldots, u_M(\zeta_j))^T.$$ 

All solution vectors are grouped together column by column in a matrix $X \in \mathbb{R}^{M \times (2L+1)}$.

$$X := X(\zeta) = [u(\zeta_1) \ldots u(\zeta_{2L+1})].$$

(4)

The rows of $X$ correspond to the $M$ spatial unknowns. $X$ is called data base for the following analysis. Each column of the data base represents a solution to one particular parameter variation.

### 3.2 Sensitivity analysis

A sensitivity analysis determines the dependence of the solution on the variation of design-parameters. It allows one to identify the design-parameters possessing the most influence on the solution [16, 33, 34]. This objective of sensitivity analysis is often called parameter screening. Design-parameters possessing no or only a small influence on the solution are negligible and can be fixed to their mean value. This leads to a reduction of the dimension of the design-parameter space and therefore to a reduction of the model complexity. Sensitivity methods can be divided into local and global methods, for a detailed discussion refer to [33, 34]. A simple local sensitivity method is to approximate partial derivatives of the solution w.r.t. the design-parameters with a finite difference scheme. We apply this method on the data base (4).

We compute $J = (J_{ij})_{i=1,\ldots,M; j=1,\ldots,L}$ with a finite difference scheme as a second order approximation of the Jacobian matrix, where $J_{ij}$ is the sensitivity
measure for design-parameter $j$ in grid point $i$. We use

$$S_j = \sqrt{\sum_{i=1}^{M} J_{ij}^2}$$

(5)
as global measure on the whole grid for estimating the linear influence of the $j$th parameter. The design-parameters are ranked by sorting the sensitivity measures $S_j, j = 1, \ldots, L$ in descending order. The dimension of the design-parameter space is then reduced by retaining only the most sensitive parameters in the model.

The Jacobian matrix only shows the magnitude of the linear influence of each design-parameter onto the solution. To overcome this drawback, additionally an indicator for the nonlinear influence of the design-parameters on the solution is computed. This indicator makes use of the diagonal part of the Hessian matrix; the full Hessian matrix is not computed in order to keep the computation time to a minimum. The diagonal part of the Hessian matrix $\text{diag} H = (\text{diag} H_{ij})$ is analogously approximately computed with a finite difference scheme of second order. We use

$$S^2_j = \sqrt{\sum_{i=1}^{M} \text{diag} H_{ij}^2}$$

(6)
as global measure of the nonlinear influence of the $j$th parameter. We assume that all parameters have the same input range. This can easily be achieved by a normalization of the parameter input space, e.g. to $[-1, 1]$. Then, we are allowed to compare the measures for estimating the linear as well as nonlinear influence of different parameters. If

$$S_j < cS^2_j,$$

(7)

with $c$ appropriately chosen, e.g. $c = \sigma_j$, the dependence of the $j$th design-parameter on the solution should be assumed nonlinear. Additionally, local effects can be analysed by pointwise comparing the entries of the Jacobian and the diagonal part of the Hessian matrix, i.e., if

$$|J_{ij}| < c|\text{diag} H_{ij}|,$$

(8)

the dependence of $j$th design-parameter on the solution in grid point $i$ should be assumed to be nonlinear. The opposite is not true. It is possible that the Hessian matrix is not diagonal dominant, especially it can occur that $\text{diag} H = 0$, but the off-diagonal elements are large.

In order to ensure that the design-parameters behave linear on the solution (i.e. at least the derivatives of second order are small), the full Hessian matrix has to be analysed. This requires the computation of $L(L - 1)/2$ additional entries of the Hessian matrix due to its symmetry. Thus, we need to perform a DoE with $4L(L - 1)/2$ additional simulations to approximate the required second order derivatives w.r.t. the $j$th and $l$th parameter in grid point $i$ with
finite difference schemes of second order. Analogously to the diagonal part, we define a global approximation of the $j,l$th entry of the Hessian matrix. We define a global measure $D$ as

$$D := \sqrt{\sum_{j=1}^{L} J_j - \frac{\sigma}{4}|\alpha_{\text{max}}|},$$

where $\alpha_{\text{max}}$ denotes the maximal eigenvalue of the global approximation of the Hessian matrix and $\sigma$ is the maximum of $\sigma_j, j = 1, \ldots, L$. If $D$ is negative, the dependence of design-parameters on the solution is assumed to be nonlinear, otherwise it is linear.

In summary, the sensitivity analysis computes the linear sensitivity measure $J$ and the indicator for nonlinearity $\text{diag}H$. Only, if the inequalities (7) and (8) do not hold, the full Hessian matrix is computed. A reduced stochastic model will be constructed by replacing nearly-linear design-parameters which have only a small influence on the solution, by their mean value.

### 3.3 Discussion

#### 3.3.1 Advantages

Firstly, the presented DoE-based parameter screening method is independent of the probability distribution of the random variables. This makes the approach very general and useful in industrial relevant applications, because usually the correct probability distribution of the design-parameters are not known.

Secondly, the DoE needs only $2L + 1$ samples, this is the minimum number of samples needed to compute a second order approximation of the Jacobian as well as the diagonal part of the Hessian matrix. This number of samples is appropriate for representing a linear dependence of each parameter on the solution. As a result, the local sensitivity analysis by means of finite differences requires less computational effort than variance based sensitivity methods, e.g., Sobol indices, for which many simulations are to be performed in Monte Carlo alike algorithms. Additionally, the indicator of nonlinear influence gives an important hint to the nature of design-parameter behavior. If the number of simulations, which can be performed, is not restricted by the simulation time, additionally, the full Hessian matrix and the measure $D$ is computed. This approach ensures that only parameters with a small (the last few parameters in the ranking), linear influence are neglected.

Finally, the presented approach for sensitivity analysis is able to locate local areas on the grid, where the behavior of design-parameters onto the solution is nonlinear. Hence, when solving a sPDE with an interpolation based method, more advanced interpolation methods could be applied in the regions of interest in order to improve the accuracy of the interpolation result.
3.3.2 Limitations

Sensitivity measures computed with partial derivatives around a nominal value are only valid if a very low order Taylor series approximation of the solution is valid. This is typically true only in small regions around the nominal value, i.e., if \( \sigma_j \) is chosen appropriately small. This drawback is somewhat captured with the computation of the indicator of nonlinearity. Note, that interactions of third and higher order can only be computed with more computational effort. Secondly, interactions among design-parameters can not be detected by varying only one parameter at a time. Therefore, this approach requires independent random variables. This limitation can be avoided by computing the full Hessian matrix.

4 Interpolation methods

Let an unknown or computational expensive function \( u \) be determined by a set of \( N \) sample points \( \{\zeta_i\}_{i=1,...,N} \) in the parameter space and corresponding solutions \( \{u(\zeta_i)\}_{i=1,...,N} \). We seek an approximant \( \tilde{u} \) for \( u \), which is faster to evaluate. Additionally, we require

\[
\tilde{u}(\zeta_i) = u(\zeta_i), \ i = 1, \ldots, N, \tag{10}
\]

i.e., the discrete sampling points are interpolated. Note, we write \( u_i \) for \( u(\zeta_i) \). There are a lot of commonly used interpolation methods, for example meta-models, also called response surface models, which are widely used in the field of optimization (for an overview see [25, 13, 18]). Note, for some of these approximation methods, the stronger interpolation condition (10) does not hold. Additionally, there exist special interpolation methods for solving sPDEs, for example the stochastic collocation method. We will use two interpolation methods, the stochastic collocation method and a radial basis function (RBF) metamodel, to compute statistics of the stochastic solution of (1). The stochastic collocation approach is described in Section 4.1. Section 4.2 presents metamodels with radial basis functions accelerated by a principal trends analysis.

4.1 Stochastic collocation finite element method

The stochastic collocation finite element method [1, 39] is a sampling based solution method for stochastic PDEs. In contrast to the Monte Carlo simulation method [14], it achieves a fast convergence rate by a good choice of the multidimensional stochastic collocation points.
4.1.1 Collocation approach

The stochastic collocation method represents the solution discretely by an expansion with multivariate Lagrange polynomials $l_k(y)$,

$$u(x, y) \approx \sum_{k=1}^{N_c} u(x, \zeta_k) l_k(y). \quad (11)$$

The Lagrange polynomials $l_k(y)$ are interpolatory polynomials, defined by a set of multidimensional collocation points, $\{\zeta_1, \ldots, \zeta_{N_c}\}$. Each sample or collocation point $\zeta_k$ consists of $L$ components, $\zeta_k = (\zeta_{k,1}, \ldots, \zeta_{k,L})$, according to the $L$ random variables present in the problem.

In order to obtain a fast convergence of (11), the collocation points $\zeta_k$ are typically constructed as a sparse grid of Gauss or Clenshaw-Curtis points. By using sparse grids, the curse of dimensionality associated with full tensor product collocation grids is reduced. An extended overview of the stochastic collocation method can be found in [2, 3].

The stochastic collocation method proceeds by applying a collocation step, i.e., it requires that the residual vanishes at each collocation point $\zeta_k$,

$$\mathcal{L}\left(x, \zeta_k; \sum_{m=1}^{N_c} u(x, \zeta_m) l_m(\zeta_k)\right) - b(x, \zeta_k) = 0 \quad \forall k = 1, \ldots, N_c. \quad (12)$$

Since Lagrange polynomials satisfy the property that $l_k(\zeta_m) = \delta_{k,m}$, with $\delta_{k,m}$ the Kronecker delta, (12) corresponds to a set of $N_c$ decoupled deterministic PDEs,

$$\mathcal{L}\left(x, \zeta_k; u(x, \zeta_k)\right) - b(x, \zeta_k) = 0 \quad \forall k = 1, \ldots, N_c. \quad (13)$$

These deterministic PDEs (13) are then discretized in the spatial domain; we will denote by $M$ the corresponding number of spatial degrees of freedom.

4.1.2 Computation of statistics

From the deterministic solutions of (13), $u(x, \zeta_1), \ldots, u(x, \zeta_{N_c})$, the statistics of the solution can be obtained. By choosing the collocation points to be a cubature point set [38, 39], the mean $\overline{u}(x)$ and variance $\sigma_u^2(x)$ can be computed as [21]

$$\overline{u}(x) = \sum_{k=1}^{N_c} u(x, \zeta_k) w_k \quad \text{and} \quad \sigma_u^2(x) = \sum_{k=1}^{N_c} u(x, \zeta_k)^2 w_k - \overline{u}(x)^2,$$

where $w_k, k = 1, \ldots, N_c$, are the cubature weights corresponding to the cubature points $\zeta_k$. Other statistics, e.g., quantiles or the probability density function, can be straightforwardly computed after sampling the stochastic solution (11) in the $y$-parameters by a Monte Carlo procedure.
4.1.3 Discussion

**Advantages** As a sampling-based method, the stochastic collocation approach permits a direct reuse of deterministic simulation software. As a consequence, any nonlinearity present in the stochastic PDE does not increase the complexity of the method.

The stochastic collocation method enables one to compute high-order stochastic solutions with a limited number of deterministic simulations. For stochastic elliptic problems, theoretical convergence estimates in [1, 2, 31] indicate an algebraic convergence rate of the solution w.r.t. the number of collocation points, and an exponential rate w.r.t. the polynomial degree in the case of a tensor product collocation point grid of Gauss points.

To increase the accuracy of a stochastic collocation solution, an extended set of collocation points is required. In the case of nested 1D point sets, e.g., when using Clenshaw-Curtis points, previously obtained deterministic solutions can be reused and only a limited number of additional deterministic simulations need to be performed.

**Limitations** The number of deterministic simulations, i.e., the number of collocation points, increases proportionally to the number of random variables and the required accuracy, which determines the number of 1D collocation points in the interpolation formulae of tensor and sparse grid collocation points. Although sparse grids try to eliminate the curse of dimensionality present in the tensor product approach, the total number of collocation points still grows exceedingly fast in the case of a large number of random variables. Current research aims at further reducing the required number of simulations, for example by taking the anisotropy of the problem into account [30, 7].

4.2 A RBF metamodel accelerated by a principal trends analysis

We present a nonlinear interpolatory metamodel with radial basis functions accelerated by a principal trends analysis. First, we introduce RBF metamodels in Section 4.2.1 and explain principal trends analysis briefly in Section 4.2.2. Then, we describe the algorithm of a RBF metamodel accelerated by a principal trends analysis in Section 4.2.3. Finally, in Section 4.2.4 we show how the algorithm can be used to compute statistics.

4.2.1 RBF metamodels

A radial basis function $\phi : \mathbb{R} \to \mathbb{R}$ is a real-valued function whose values depend only on the distance to the origin, $\phi(x) = \phi(||x||)$, where $||\cdot||$ denotes the Euclidean norm. Table 1 shows some commonly used types of RBFs for $r = ||x||$, [8, 9, 37]. A RBF metamodel is a linear combination of radial basis functions
Table 1: Commonly used types of radial basis functions

| \( \phi(r) \)          | Type                                      |
|------------------------|-------------------------------------------|
| \( r \)                | linear                                    |
| \( r^2 \log r \)      | thin-plate spline                         |
| \( \exp(-\gamma r^2), \gamma > 0 \) | Gaussian                                 |
| \( \sqrt{r^2 + c^2}, c > 0 \) | Multiquadric                             |
| \( r^3 \)             | triharmonic spline                        |
| \( \frac{1}{2}(r^2 + c^2)^{-\frac{1}{2}} \) | inverse Multiquadric                     |

\[ \tilde{u}(\zeta) = \sum_{j=1}^{N} \phi(||\zeta - \zeta_j||)c_j, \quad (14) \]

with coefficients \( c_i \in \mathbb{R} \), which are determined so that \( (10) \) holds, i.e.,

\[ \tilde{u}(\zeta_i) = u_i = \sum_{j=1}^{N} \phi(||\zeta_i - \zeta_j||)c_j, \quad i = 1, \ldots, N. \quad (15) \]

We define an interpolation matrix \( \Phi = (\phi(||\zeta_i - \zeta_j||))_{i,j=1,...,N} \). Note that the interpolation matrix is always nonsingular for the multiquadric function, details on invertibility of the interpolation matrix can be found in [8, 37]. By inverting the interpolation matrix, the result \( \tilde{u} \) can be written as weighted sum \( (29) \)

\[ \tilde{u}(\zeta) = \sum_{i=1}^{N} w_i(\zeta)u_i, \quad (16) \]

with

\[ w_i(\zeta) = \sum_{j=1}^{N} \Phi^{-1}_{ij} \phi(||\zeta - \zeta_j||), \quad (17) \]

where \( \Phi^{-1}_{ij} \) are the entries of \( \Phi^{-1} \). This metamodel can be extended by polynomial detrending, that is by adding a polynomial part

\[ \tilde{u}(\zeta) = \sum_{j=1}^{N} \phi(||\zeta - \zeta_j||)c_j + P(\zeta). \quad (18) \]

Polynomial detrending improves the precision of interpolation and leads to an exact representation of polynomials up to a certain degree, for more details see, e.g., [8].
4.2.2 Principal trends analysis

Principal component analysis (PCA) is a well-known mathematical method for data analysis which provides the principal components of the data, i.e., the dominating dependencies of design-parameters on the solution. PCA is a linear method, that is, it assumes that the data result from a linear transformation of variables.

One way to compute the principal components is by applying a singular value decomposition (SVD) [19]. This decomposition exists for each $M \times N$ matrix $X$, $M \geq N$ and is defined as

$$X = F \Lambda V^T,$$  \hspace{1cm} (19)

where $F = (f_1, f_2, \ldots, f_N) \in \mathbb{R}^{M \times N}$, $V^T = (v_1^T, v_2^T, \ldots, v_N^T) \in \mathbb{R}^{N \times N}$ are orthogonal matrices such that $F^T F = I_N$ and $V^T V = VV^T = I_N$, $\Lambda \in \mathbb{R}^{N \times N}$ is a diagonal matrix with diagonal elements $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$. The number of non-zero singular values determines the rank $r$ of $X$, that is $\lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_N = 0$. The entries $\{\lambda_j\}$ are called the singular values of $X$.

A rank-$k$ approximation $X_k$ of $X$ is defined as

$$X_k = \sum_{i=1}^{k} f_i \lambda_i v_i^T, \quad \text{with } k < r.$$  \hspace{1cm} (20)

Then it holds that $X_k$ is the best rank-$k$ approximation in the Frobenius norm $|| \cdot ||_F$ to the matrix $X$, that is

$$\min_{\text{rank}(Y)=k} ||X - Y||_F^2 = ||X - X_k||_F^2 = \lambda_{k+1}^2 + \cdots + \lambda_r^2.$$  \hspace{1cm} (21)

Therefore, a linear dimension reduction with the SVD proceeds by setting the singular values $\lambda_{k+1}$ to $\lambda_r$ to zero. The approximation error of the resulting reconstruction equals

$$\text{err}_2^2 = \sum_{l=k+1}^{N} \lambda_l^2,$$  \hspace{1cm} (22)

which can be controlled over $k$. Therefore, the SVD gives us a method to approximate the intrinsic dimension and to project the data to that lower dimensional space.

The computation of the SVD of a matrix $X \in \mathbb{R}^{M \times N}$ is generally based on the eigensystem of either $X^T X$ or $XX^T$ [32, 13, 6]. An efficient implementation of the SVD for $M \gg N$ operates block by block so that it is not needed to store the entire matrix in memory. It is based on a computation of the Gram matrix, computing its spectral decomposition, and finally, the $k$ relevant columns of $F$ [29]. We call this approach fastSVD.

4.2.3 RBF metamodel accelerated by principal trends analysis

Algorithm describes the procedure to accelerate the RBF metamodel with a principal trends analysis and to use this for the approximation of a $q$-quantile.

11
Algorithm 1: Approximation of a $q$-quantile with the accelerated metamodel

Require: $\xi$, threshold, $q$
1: $\{\zeta_i\}_{i=1}^{L_S} = \text{DoE}(\xi)$, $X := X(\zeta) = [u(\zeta_1) \ldots u(\zeta_{2L+1})]$. 
2: $\{w_i\}_{i=1}^{N} = \text{metamodel}(\zeta)$, $[F, \Lambda, V^T] = \text{fastSVD}(X, \text{threshold})$
3: $\{\tau_i\}_{i=1}^{L_{\text{add}}} = \text{DoE}(\zeta)$
4: for $j = 1$ to $L_{\text{add}}$ do
5: $\tilde{u}(\tau_j) \leftarrow \text{acceleratedMetamodel}(\tau, w, F, \Lambda, V^T)$
6: $q$-quantile $\leftarrow P_q^2(\tilde{u}, q)$
7: end for

First, a design of experiments (DoE) is generated to construct a data base $X$. Next, a metamodel is set up and a singular value decomposition (SVD) of the database $X$ is computed. Note, that only the weights (17) of the metamodel are computed. These steps, line 1 and line 2, are called the setup phase to construct the metamodel. In the setup phase the probability distribution of all random input variables is not required. Then the SVD of $X$ is used instead of the sample results $X$ itself for the construction of the accelerated metamodel [28]. That is, the SVD is combined with the metamodel weights, see line 5, in order to approximate the solution $u$ for a given parameter set $\tau_j$. This procedure can be written as

$$\tilde{u}(\tau_j) = \sum_{i=1}^{N} u_i w_i(\tau_j) = \sum_{i=1}^{k} f_i \lambda_i v_i^T w_i(\tau_j)$$

(23)

The so-constructed metamodel is evaluated in $L_{\text{add}}$ sampling points in order to compute statistics of the solution by a quasi Monte Carlo approach. The optimal number of sampling points should be large enough and depends on the model problem. In quasi Monte Carlo methods low discrepancy sequences are used as sampling points which lead to a better performance as standard Monte Carlo methods [27, 24]. The computation of statistics is performed in lines 3 to 6 and is described in the next section.

4.2.4 Computation of statistics

To compute statistics of $u$, any method can be used that is based only on samples of the design-parameters. The $q$-quantile for a random variable is defined as the value $x$ such that the probability that the random variable will be less than $x$ is at most $q$, with $q \in [0, 1]$. For example, the 0.9-quantile is the value $x$ such that the probability that the random variable will be less than $x$ is at most 90%. A $q$-quantile is generally estimated as a weighted sum of the order statistics, with a high number of samples, see, e.g., [11]. The problem of such approaches is that all samples must be stored. Therefore, we use the $P^2$ algorithm for dynamic calculation of quantiles, instead of the order statistics, to compute the median and $q$-quantiles of the stochastic solution of (2) in Algorithm 1 (line 6). The $P^2$ algorithm by Jain and Chlamtac stores only five markers, which are updated
as more samples are generated. The five markers are the minimum, the \( q/2 \)-, \( q \)- and \( (q + 1)/2 \)-quantiles and the maximum. The markers are adjusted with a parabolic prediction (\( P^2 \)) formula. Details on the \( P^2 \) algorithm can be found in \[17\]. The best number of sampling points depends on the model problem; usually a high number of samples \( L_{\text{add}} > 1000 \) are required.

In summary, to compute statistics of the solution of a sPDE, we perform a DoE to get \( L_{\text{add}} \) sampling points according to the probability distribution of the random variables, see Algorithm 1 line 3. For each sampling point we approximate the corresponding solution with the accelerated metamodel, see line 5. Note that no additional deterministic solutions of the sPDE by means of simulation are needed. Finally, we update the \( P^2 \) algorithm estimator of the \( q \)-quantile (line 6) in each grid node.

4.2.5 Discussion

Advantages
An advantage of the accelerated RBF metamodel method lies in its flexibility. It does not require the probability distribution of the random input variables in the setup phase. As for the collocation method, the accelerated metamodel is non-intrusive. Simulation code available for the deterministic problem can be reused directly. The choice of the RBF ensures the invertibility of the interpolation matrix, and thus the stability of the method. Secondly, the acceleration by the singular value decomposition decreases the computational time considerably. The SVD computes the principal trends in the data, and, therefore, takes the correlation between the nodal points into account. Due to this reason, fewer sample points may be necessary in order to get the same accuracy as with using a collocation method. Additionally, the accuracy is controllable over the omitted singular values. The computation of statistics of the solution is straightforward. It is possible to compute multiple statistical scenarios with one single metamodel, since the information of probability distributions is used only to compute statistics and not to construct the metamodel.

Limitations
If the random input parameters behave nonlinear on the solution, it is required to increase the number of sampling points (see line 1 of Algorithm 1) to construct a metamodel with appropriate accuracy. This includes the execution of deterministic simulations on these sampling points to compute discrete solutions.

5 Numerical Results

5.1 Overview of the model problem
We consider a steady-state diffusion problem with a random diffusion coefficient, defined on an L-shaped domain \( D \), as illustrated in Fig. 1.

\[
-\nabla \cdot (a(x, u(x, y), y) \nabla u(x, y)) = b(x) \quad x \in D, \ y \in \Gamma.
\]  

(24)
Figure 1: Geometry and deterministic solution of model problem. In the problem shown here, all random fields \(a_i\) are set to their mean values and \(\gamma = 1.0\). The problem is discretized on a mesh with 103479 nodes.

Dirichlet boundary conditions are imposed on the upper and lower right boundaries, and zero Neumann conditions elsewhere. The right-hand side \(b(x)\) is equal to 1. We model \(a(x, y)\) as a nonlinear piecewise random field, consisting of three parts, \(a_1, a_2\) and \(a_3\), defined respectively on the domains \(D_1, D_2\) and \(D_3\):

\[
a(x, u(x, y), y) = \begin{cases} 
a_1(x, y) + \gamma u^2(x, y) & x \in D_1 \\
a_2(x, y) + \gamma u^2(x, y) & x \in D_2 \\
a_3(x, y) + \gamma u^2(x, y) & x \in D_3 
\end{cases}
\] \hspace{1cm} (25)

where \(\gamma\) is a constant for varying the nonlinearity. We assume that each component \(a_i(x, y), i = 1, 2, 3\), has the form of a truncated Karhunen-Loève (KL) expansion \[20\] with respectively \(L_1 = 6\), \(L_2 = 7\) and \(L_3 = 5\) random variables. We assume that all \(L\) random variables are independent, with \(L = L_1 + L_2 + L_3 = 18\).

For the numerical experiments, we use an exponential covariance function for \(a_i\), given by

\[
C_{aa}(x, x') = \sigma^2 \exp \left( -\frac{||x - x'||_1}{l_c} \right).
\] \hspace{1cm} (26)

The mean values of \(a_i\) are set to \(\bar{a}_1 = 30\), \(\bar{a}_2 = 5\) and \(\bar{a}_3 = 100\), respectively. Analytical expressions for the corresponding KL-functions are derived in \[40\]. We consider two configurations for creating the KL-expansions of \(a_i\) in (25):

- \(L\) uniformly distributed random variables on \([-\sqrt{3}, \sqrt{3}]\): covariance function \[26\] with correlation lengths \(l_{c,1} = 1\), \(l_{c,2} = 0.5\) and \(l_{c,3} = 1.5\); and variances \(\sigma_1^2 = 100\), \(\sigma_2^2 = 2.25\), \(\sigma_3^2 = 900\).
• $L$ standard normally distributed random variables (mean 0 and variance 1): covariance function (26) with $l_{c,1} = 1$, $l_{c,2} = 0.5$ and $l_{c,3} = 1.5$; and variances $\sigma_1^2 = 9$, $\sigma_2^2 = 0.25$, $\sigma_3^2 = 100$.

The problem is solved on a spatial finite element grid with $M = 12154$ nodes and with $M = 103479$ nodes. In each case, we set the constant $\gamma$ for varying the nonlinearity (25) to 1.0 and 100.0. An example of a deterministic solution is shown in Figure 1.

The nonlinear deterministic PDEs that are obtained by sampling the random variables are solved with a Newton-multigrid method, that is, algebraic multigrid is used to solve a sequence of linearized systems.

When applying the stochastic collocation method, a sparse grid of Clenshaw-Curtis or Gauss points is applied, denoted by ‘ScC’ or ‘Sg’, respectively. The level of the method will be indicated by the number following ‘ScC’ and ‘Sg’.

Since the exact solution of the stochastic PDE is not known, we base the error computation on a reference solution $u_{\text{ref}}$, which is obtained as a high-order stochastic collocation solution. Then, we compare the accuracy of the different methods by computing the absolute difference $\text{diff}_{\text{abs}}(j)$ or the relative difference $\text{diff}_{\text{rel}}(j)$ locally in each grid point as

$$\text{diff}_{\text{abs}}(j) = |\tilde{u}(j) - u_{\text{ref}}(j)|, \quad \text{diff}_{\text{rel}}(j) = \frac{|\tilde{u}(j) - u_{\text{ref}}(j)|}{u_{\text{ref}}(j)}, \quad j = 1, \ldots, M,$$

where $\tilde{u}(j)$ denotes the interpolated result in grid point $j$ and $u_{\text{ref}}(j)$ denotes the reference solution in grid point $j$.

### 5.2 Parameter screening results

We perform a design of experiments as described in Section 3.1 with $2L + 1 = 37$ sampling points. Then, we apply the parameter screening measures of Section 3.2 to the model problem.

The results of the parameter screening in the case of $M = 12154$ nodes are shown in Figure 2. The sensitivity analysis gives similar results for different resolutions of the grid. If we use only 12154 spatial degrees of freedom the indicator for nonlinearity (6) is higher than with a high resolution, but the ranking of the parameters remains the same. Differences in the ranking and the amount of influence of the parameters occur due to the variation of $\gamma$ in the model problem. In detail, Figure 2 shows that in the case of $\gamma = 1.0$ the parameters $y_1, y_2, y_7, y_8, y_9$ and $y_{14}$ are the most influencing ones. The corresponding design-parameter space can be reduced to 6 instead of 18 parameters. In the case of $\gamma = 100.0$ the influence of the parameters at the first few ranks increases, so that the design-parameter space can be reduced to only three parameters, namely $y_1, y_2$ and $y_{14}$. The additional indicator for nonlinearity shows no great nonlinear influence of any parameter. In the cases where the measure (7) is true, that is for $y_{10}$ and $y_{13}$ in the case of $\gamma = 100.0$, we find that $\mathcal{J}$ and $\mathcal{H}$ are almost zero. Therefore these parameters do almost not influence the solution at all.
Figure 2: Parameter screening results. The height of the bars gives the influence of the parameter on the solution, where the global approximations of $J$ and $\text{diag}H$ are computed via (5) and (6) respectively.
To confirm the linear behavior of all random variables, we compute the full Hessian matrix and the measure $D$ \(^{[9]}\) in case of 12154 nodes. Therefore, we have to perform $4L(L-1)/2 = 612$ additional simulations to compute the approximation of the second derivatives. We get

$$\vert \alpha_{\text{max}} \vert = 1.05487, \quad D = 3.60796 > 0,$$

with $\sigma = \sqrt{3}$. This proves the linear behavior of all random variables, at least up to second order approximations.

### 5.3 Comparison of the results: Collocation method and accelerated metamodel

![Figure 3: Precision of reconstructed matrix by fastSVD as a function of $k$.](image)

(a) Approximation error \(^{[22]}\) of the deviation $dX$ in dependence on $k$. (b) Corresponding singular values $\lambda_i$.

(c) Approximation error \(^{[22]}\) of the deviation $dX$ in dependence on $k$. (d) Corresponding singular values $\lambda_i$.

In order to show the efficiency of the accelerated metamodel, we compare the results to a reference solution, since the exact solution of the stochastic PDE is not known. We choose the collocation method as reference solution, since the
collocation method is a usually applied state-of-the-art interpolation approach in the context of stochastic PDEs.

We compute the accelerated metamodel based on the multiquadric RBF, see Table 1 and q-quantiles according to Algorithm 1 for several configurations of the model problem 24. In the following, two cases are presented in detail. First, we analyse the case of 18 uniformly distributed random variables on a mesh with 12154 nodes and nonlinearity coefficient $\gamma = 1.0$. Here, the design-parameter space is reduced to only 6 random variables as a result of the previous parameter screening, see Section 5.2. Therefore, the sample points for the construction of the metamodel also reduce to $2L + 1 = 13$. We set up the accelerated metamodel for this configuration with 6 random variables.

As a second example, we analyse the case of 18 Gaussian distributed random variables on a mesh with 103479 nodes and nonlinearity coefficient $\gamma = 100.0$. Here, the design-parameter space is reduced to 3 random variables as a result of the previous parameter screening and 7 sample points are used for the construction of the accelerated metamodel. In the following, we will always use the metamodels with the reduced design-parameter space, unless stated otherwise.

5.3.1 RBF metamodel

We compute the fastSVD for each model problem as described in Section 4.2.2. The precision of the reconstructed matrix as a function of $k$, together with the corresponding singular values $\lambda_l$, is illustrated in Fig. 3 exemplary for a discretization with 103479 nodes and $\gamma = 100.0$. The figures show that the singular values are rapidly decreasing with $k$. Therefore, the error is rapidly decreasing with $k$ as well. Only 7 to 10 singular values in the case of uniformly distributed random variables and 3 singular values in the case of Gaussian distributed random variables are sufficient to get an accurate solution.

Next, the metamodel weights are computed according to (17). To evaluate this metamodel, we perform a new DoE with $L_{add} = 2000$ sampling points $\tau_j, j = 1, \ldots, L_{add}$. The sampling points are randomly chosen and uniformly distributed in the design space. The deterministic solutions of the sPDE according to these sampling points are computed by interpolation with the accelerated metamodel, using formula (23). As an example, the deterministic result for one single sample point $\tau_1 = [\xi_1 = 0.2, \xi_2 = 0.4, \xi_{14} = 0.8, \xi_i = 0.5 \ \forall i \in \{1, 2, 14\}]$ is shown in Figure 4 on the left, on the right side the absolute difference between the interpolation result in this sample point and this deterministic solution is shown. Note that for the interpolation result the metamodel with the reduced parameter space has been used. Figure 4 shows that the interpolation with the accelerated metamodel gives accurate results. With only the reduced parameter space the maximal absolute error is $1.72907e - 05$. The maximal relative error is 0.000122, that is 0.012 percent of the solution value, and this appears only in the region, where the deterministic solution equals almost zero.

The whole set of sample points and corresponding interpolation results is then used to compute several quantiles, for example the median, the 0.68-quantile and the 0.9-quantile.
5.3.2 Collocation method versus accelerated metamodel

We compare the DoE and the interpolation methods with respect to the number of deterministic PDEs to be solved, the computational time and accuracy. The collocation methods and the accelerated metamodel require a different number of sampling points, see Table 2, and therewith a different number of deterministic PDEs to solve.

Table 2: Number of required sampling points as a function of the number of random variables

| random variables | Scc | Scc2 | Sg | Sg2 | accMetamodel |
|------------------|-----|------|----|-----|--------------|
| 18               | 37  | 685  | 37 | 757 | 37           |
| 6                | 13  | 85   | 13 | 109 | 13           |
| 3                | 7   | 25   | 7  | 37  | 7            |

Details of the different components of the computational time are shown in the Tables 3 and 4.

Beside the different computational time for the simulations, the different times for constructing the sampling points for the statistics are standing out. The collocation method uses 2000 sampling points according to the probability distribution of the random variables to reconstruct the stochastic solution [11].
Table 3: Detailed overview of runtime (in sec) on a standard 3GHz Linux PC to compute 0.68-quantile with the stochastic collocation method for the model problem with 18 Gaussian random variables, $\gamma = 100.0$ and a spatial discretization with 103479 nodes.

| Time for                          | Sg                  | Sg2                  |
|-----------------------------------|---------------------|----------------------|
| construction of collocation points| 0.000000E+00       | 0.000000E+00         |
| solution of deterministic PDEs (simulation) | 2.171000E+03     | 4.273300E+04         |
| construction of samples of the collocation solution | 1.919000E+01    | 4.162900E+02         |
| 0.68-quantile                     | 1.022000E+01       | 1.044000E+01         |
| overall runtime                   | 2.200410E+03       | 4.315973E+04         |

Table 4: Detailed overview of runtime (in sec) on a standard 3GHz Linux PC to compute 0.68-quantile with the accelerated metamodel method for the model problem with 18 Gaussian random variables, $\gamma = 100.0$ and a spatial discretization with 103479 nodes.

| Time for                          | acc_metamodel_18   |
|-----------------------------------|---------------------|
| construction of DoE for 18 variables | 0.000000E+00       |
| solution of deterministic PDEs (simulation) | 2.171000E+03     |
| parameter screening               | 3.504000E+00       |
| construction of DoE for 3 variables | 0.000000E+00       |
| fastSVD                            | 1.370000E+00       |
| evaluation of the accelerated metamodel and 0.68-quantile | 1.121000E+01 |
| overall runtime                   | 2.187084E+03       |
We use the same number of sampling points $L_{\text{add}} = 2000$ also for the evaluation of the accelerated metamodel. The tables demonstrate that the parameter screening is very fast, although it computes measures in each grid point. The time for the evaluation of the accelerated metamodel and the 0.68-quantile contains the time for computing the solution of the sPDE with interpolation in the current sampling point using (23), added up for all sampling points and the update of the quantile estimator with the $P^2$ algorithm. Note that we use the same algorithm for the computation of quantiles, namely the $P^2$ algorithm to get a meaningful comparison. Therefore, the higher runtime to compute the 0.68-quantile with the collocation method is only due to the reconstruction of the stochastic solution. Altogether, the runtime of the accelerated metamodel approach is all about the same than the collocation method with Gauss points, level one ($S_g$), and a lot faster than the collocation method with Gauss points, level two ($S_g2$). Note that this comparison only illustrates where the computational effort of each method lies on. Each method may be accelerated by parallelizing components of the method.

![Figure 5](image)

(a) 0.68-quantile, Metamodel vs $S_g$. (b) 0.68-quantile, Metamodel vs $S_g2$.

Figure 5: Absolute difference between the 0.68-quantile obtained with the collocation and accelerated metamodel method. The model problem with 18 Gaussian distributed random variables, discretized with 103479 nodes, $\gamma = 100.0$ and a reduced metamodel with 3 random variables are used.

We compare the metamodel solution against reference solutions $u_{\text{ref}} = S_g$ and $u_{\text{ref}} = S_g2$. However, note that the computed quantile with $S_g$ or $S_g2$ is the exact solution of the constructed polynomial solution approximation, but not the exact solution of the sPDE. Figures 5 and 6 show the absolute and relative differences (27) between the 0.68-quantile computed with the collocation and with the accelerated metamodel method. Some differences occur, especially in the domain where the solution of the sPDE is almost zero. The differences are smaller when compared against $S_g2$. The errors are of course influenced by the
number of sampling points. For comparison each method uses 2000 sampling points, but an evaluation of the accelerated metamodel with more samples may even ameliorate the result.

In summary, the accelerated metamodel performs as fast as the collocation Sg method for the model problem considered. But, its accuracy lies somewhere between Sg and Sg2. Therefore, the numerical results illustrate that the accelerated metamodel approach is an efficient and accurate alternative to the collocation method to compute statistics of a sPDE containing independent random variables. It is especially advantageous when the probability distributions of the random variables are not known a priori.

6 Discussion and future work

The paper deals with the computation of statistics of a sPDE by means of interpolation methods. We present an alternative approach to the collocation method, namely the accelerated RBF metamodel. The different methods were demonstrated on a nontrivial model problem. Numerical examples show the high potential of this new interpolation approach, which is very fast and needs only a small set of deterministic solutions of the sPDE. The great advantage of the presented method is its flexibility, it makes only few assumptions and does not require the exact probability distribution of the random input variables for construction. It is possible to compute multiple statistical scenarios with one single metamodel. Therefore, the approach will be applicable in most industrially relevant applications.
Secondly, this paper deals with the curse of dimensionality. The presented parameter screening approach can be used in order to reduce the model complexity. This generally accelerates the computation of the solution of the sPDE. Here, the parameter screening approach is used with the accelerated metamodel approach. But, it is also possible to combine the parameter screening with the collocation method to get an accurate solution in less computational time. This will result in the construction of an adaptive sparse collocation method.

Future work will be on extending the parameter screening and accelerated metamodel approach to more nonlinear cases. This will lead to a better approximation of the deterministic solution and, thus, the $q$-quantiles. Also, there is some work to do in order to find the optimal number of sampling points required for the computation of a $q$-quantile.

References

[1] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45:1005–1034, 2007.

[2] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Rev.*, 52(2):317–355, 2010.

[3] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic spectral galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In *Spectral and high order methods for partial differential equations*, volume 76 of *Lecture notes in computational science and engineering*, pages 43–62. 2011.

[4] Bellman, R. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[5] M. Berry. Large Scale Sparse Singular Value Computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[6] M. Berry, D. Mezher, P. B., and A. Sameh. Parallel computation ot the singular value decomposition. Technical report, Institut national de recherche en informatique et en automatique, 2003.

[7] M. Bieri. A sparse composite collocation finite element method for elliptic sPDEs. *SIAM J. Numer. Anal.*, 49:2277–2301, 2010.

[8] M. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.

[9] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *ACM SIGGRAPH’01 Proceedings of the 28th annual
conference on Computer graphics and interactive techniques, pages 67 – 76. ACM, Los Angeles, CA, USA, 2001.

[10] T. Clees, I. Nikitin, L. Nikitina, and C.-A. Thole. Analysis of bulky crash simulation results: Deterministic and stochastic aspects. In Lecture Notes: Advances in Intelligent and Soft Computing Series. Springer, 2012.

[11] H. David and H. Nagaraja. Order statistics. John Wiley & Sons, 3rd edition, 2003.

[12] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. AMS Conference ‘Math Challenges of the 21st Century’, 2000. Available from http://www-stat.stanford.edu/ donoho/.

[13] H. Fang, M. Rais-Rohani, Z. Liu, and M. Horstemeyer. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. Computers and Structures, 83:2121–2136, 2005.

[14] G. S. Fishman. Monte Carlo: concepts, algorithms and applications. Springer-Verlag, New York, USA, 1996.

[15] G. H. Golub and C. F. Van Loan. Matrix computations. Johns Hopkins University Press, London, UK, 3rd edition, 1996.

[16] D. Hamby. A review of techniques for parameter sensitivity analysis of environmental models. Environmental Monitoring and Assessment, 32:135 – 154, 1994.

[17] R. Jain and I. Chlamtac. The p2 algorithm for dynamic calculation of quantiles and histograms without storing observations. Communications of the ACM, 28(10):1076 – 1085, 1985.

[18] J. P. C. Kleijnen. An overview of the design and analysis of simulation experiments for sensitivity analysis. European Journal of Operational Research, 164:287 – 300, 2005.

[19] J. Lee and M. Verleysen. Nonlinear Dimensionality Reduction. Springer, 2007.

[20] M. Loève. Probability theory. Springer, New York, USA, 1977.

[21] G. Loeven, J. Witteveen, and H. Bijl. Probabilistic collocation: an efficient non-intrusive approach for arbitrary distributed parametric uncertainties. In AIAA, Aerospace sciences meeting and exhibit, 2007, Reno, Nevada, number AIAA 2007-317, 2007.

[22] L. Mathelin, M. Y. Hussaini, and T. A. Zang. Stochastic approaches to uncertainty quantification in CFD simulations. Numerical Algorithms, 38:209–236, 2005.
[23] H. G. Matthies. Stochastic finite elements: computational approaches to stochastic partial differential equations. *Z. Angew. Math. Mech.*, 88(11):849–873, 2008.

[24] W. J. Morokoff and R. E. Caflisch. Quasi-monte carlo integration. *Journal of Computational Physics*, 122:218 – 230, 1995.

[25] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley and Sons, 3rd edition, 2009.

[26] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annu. Rev. Fluid. Mech.*, 41:35–52, 2009.

[27] H. Niederreiter. Quasi-monte carlo methods and pseudo-random numbers. *Bulletin of the american mathematical society*, 84:957 – 1041, 1978.

[28] I. Nikitin, L. Nikitina, and T. Clees. Nonlinear metamodeling of bulky data and applications in automotive design. In M. Guenther, editor, *Progress in industrial mathematics at ECMI 2010*, volume 17 of *Mathematics in Industry*, pages 295 – 301. Springer, 2012.

[29] I. Nikitin, L. Nikitina, and T. Clees. Stochastic analysis and nonlinear metamodeling of crash test simulation and their application in automotive design. In F. Columbus, editor, *Computational Engineering: Design, Development and Applications*. Nova Science, New York 2011.

[30] F. Nobile, R. Tempone, and C. Webster. An anisotropic sparse grid stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Num. Anal.*, 46(5):2411–2442, 2008.

[31] F. Nobile, R. Tempone, and C. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Num. Anal.*, 46:2309–2345, 2008.

[32] W. Press, S. Teukolsky, W. Vetterling, and F. B.P. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1988–1992.

[33] H. Rabitz, M. Kramer, and D. Dacol. Sensitivity analysis in chemical kinetics. *Ann. Rev. Phys. Chem.*, 34:419 – 61, 1983.

[34] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis - the primer*. John Wiley & Sons, 2008.

[35] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55:271 – 280, 2001.
[36] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93:964 – 979, 2008.

[37] H. Wendland. *Scattered data approximation*. Cambridge University Press, 2005.

[38] D. Xiu. Fast numerical methods for stochastic computations: a review. *Commun. Comput. Phys.*, 5(2-4):242–272, 2009.

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

[40] D. Zhang and Z. Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loève and polynomial expansions. *Journal of Computational Physics*, 194:773–794, 2004.