PLS-based adaptation for efficient PCE representation in high dimensions

Iason Papaioannou, Max Ehre, Daniel Straub

Engineering Risk Analysis Group, Technische Universität München, Arcisstr. 21, 80290 München, Germany

Abstract

Uncertainty quantification of engineering systems modeled by computationally intensive numerical models remains a challenging task, despite the increase in computer power. Efficient uncertainty propagation of such models can be performed by use of surrogate models, such as polynomial chaos expansions (PCE). A major drawback of standard PCE is that its predictive ability decreases with increase of the problem dimension for a fixed computational budget. This is related to the fact that the number of terms in the expansion increases fast with the input variable dimension. To address this issue, Tipireddy and Ghanem (2014) introduced a sparse PCE representation based on a transformation of the coordinate system in Gaussian input variable spaces. In this contribution, we propose to identify the projection operator underlying this transformation and approximate the coefficients of the resulting PCE through partial least squares (PLS) analysis. The proposed PCE-driven PLS algorithm identifies the directions with the largest predictive significance in the PCE representation based on a set of samples from the input random variables and corresponding response variable. This approach does not require gradient evaluations, which makes it efficient for high dimensional problems with black-box numerical models. We assess the proposed approach with three numerical examples in high-dimensional input spaces, comparing its performance with low-rank tensor approximations. These examples demonstrate that the PLS-based PCE method provides accurate representations even for strongly non-linear problems.

Keywords: Uncertainty quantification, High dimensions, Polynomial chaos, Basis adaptation, Partial least squares, Surrogate models

1. Introduction

In many domains of science and engineering, one employs models of physical systems that aim at representing accurately the behavior of the underlying system under future conditions. Input parameters of models and future conditions are subject to uncertainties. Uncertainties can be due to limited availability of data, limited understanding of the underlying physical process or the intrinsic randomness of a phenomenon, such as wind or earthquake. Proper quantification of uncertainties and their impact on the performance of the model is paramount for obtaining accurate predictions.

Efficient uncertainty propagation of complex numerical models remains a challenge despite the increase in computer power. The challenge is two-fold: On the one hand, the analysis of complex systems often requires the use of computationally intensive deterministic solvers that are only available as *black boxes*, i.e. one does not have access to core routines of the computer code and hence cannot modify them. On the other hand, output quantities of interest are integrals over the space of uncertain inputs and numerical evaluation of these integrals suffers from the *curse of dimensionality*, i.e. the number of model evaluations increases geometrically with increase of the number of inputs for a fixed target accuracy.

Monte Carlo sampling can be easily coupled with black-box models and resolves the curse of dimensionality, but suffers from slow convergence rates. A possible remedy is to construct a surrogate model of the computationally intensive model using a simple mathematical form and then employ the surrogate model to perform uncertainty propagation. In particular, surrogate models based on polynomial chaos expansions (PCE) [1, 2] have enjoyed extensive application in uncertainty quantification due to their simplicity and guaranteed convergence property, among other reasons. The basic idea of PCE is to project the model output onto a space spanned by multivariate polynomials that are orthogonal with respect to the input probability measure. The projection can be performed by stochastic Galerkin schemes [1, 3, 4], which are intrusive in the sense that they require modification of existing deterministic solvers, or collocation-type methods [5, 6, 7, 8], which are non-intrusive and can be coupled with black-box deterministic solvers.

Non-intrusive PCE approaches estimate the coefficients of the expansion by numerical quadrature, interpolation or regression methods. Both numerical quadrature and interpolation techniques with tensor product grids suffer from the curse of dimensionality, i.e. their rate of convergence deteriorates drastically with increase of the dimension, e.g. [7, 8]. Approaches that employ sparse grids to delay the curse of dimensionality associated with integration/interpolation based on tensor product grids can be found in [9, 8, 10].

Accurate estimation of the PCE coefficients with regression requires an experimental design with size equal to a multiple of the total number of PCE terms. The number of PCE terms in common truncation schemes increases fast with increase of the number of inputs. This implies that in problems with high-dimensional inputs, a large number of model evaluations is required for estimating accurately the PCE coefficients with regression. A possible solution is to construct a sparse polynomial basis through selecting the most significant terms in the PCE. This can be done though regularization techniques, such as the least-angle regression [11] or l_1-minimization, also known as compressive sensing [12, 13]. Adaptive algorithms for determining sparse PCEs based on such techniques can be found in [11, 14, 15].

The quality of the PCE approximation obtained with regression can be potentially increased by choosing an appropriate experimental design set. A common choice is to use samples from the distribution of the input random variables obtained with standard Monte Carlo, stratified or quasi-random sampling schemes [16, 11]. Schemes based on the roots of the orthogonal polynomials are discussed in [6, 17] and randomized versions thereof in [18]. Alternative random sampling schemes that present optimal performance, especially in high-order PCEs, are discussed in [19, 20, 21]. Similar approaches have been proposed for use within compressive sensing-based sparse PCEs methods [13, 22, 23].

Another non-intrusive approach for surrogate modeling with polynomial bases is provided by canonical decompositions [24, 25, 26], a special case of low-rank tensor approximations (LRA). This approach is based on approximating the model response by a linear combination of rank-one approximations, obtained as products of univariate polynomial expansions. The coefficients can be determined adaptively, e.g. with a technique termed alternating least-squares regression [24, 27]. It is demonstrated in [28] that LRA performs better than sparse PCE in moderate dimensional problems and small experimental designs.

Recently, it has been proposed to reduce the number of terms in the classical PCE representation through performing a coordinate transformation in Gaussian space [29]. The transformed basis can be adapted to the output quantity of interest, e.g. through performing a small initial number of model evaluations. For example, the basis adaptation can be informed by evaluating the low-order coefficients with a sparse-grid numerical quadrature. This approach has been combined with compressive sensing for simultaneously determining the basis adaptation as well as the PCE coefficients in [30].

An alternative approach for identifying important directions in the space of uncertain inputs is the active subspace (AS) method [31, 32]. This approach is based on identifying the projection subspace of highest variability through decomposing the covariance matrix of the gradient of the model. Usually this matrix is estimated by a set of samples and corresponding gradient evaluations. Although AS can lead to vast dimensionality reduction, in high-dimensional problems with black box numerical models, the additional computational cost from the numerical evaluation of the gradients might be prohibitive. The AS method has been combined with PCEs in [33], wherein the covariance of the gradient vector is computed based on a low-order PCE.

In this contribution, we propose an approach that computes the basis transformation based on a given experimental design with a technique termed partial least squares (PLS) regression [34]. The proposed approach takes advantage of the structures in the covariance of the input parameters and model response to determine the directions with largest predictive significance in the PCE representation. Unlike the AS method, PLS does not require gradient evaluations at the samples, and hence is ideally suited for application to high-dimensional problems with black box numerical models. The PLS method is widely used in the field of chemometrics, where regression with many variables but only a few observations is common [35, 36]. Here, we employ a nonlinear version of the PLS algorithm [37, 38] and modify it for use with PCE models. The proposed PCE-driven PLS algorithm identifies simultaneously a set of dominant directions and the corresponding PCE coefficients along each direction. The performance of the method is demonstrated with three high-dimensional examples, a linear elastic bar with stochastic axial rigidity, a hysteretic oscillator under random loading and a low-carbon steel plate with stochastic stiffness. The results are compared with the ones obtained by LRA and it is shown that the proposed PLS-based PCE approach performs consistently better than LRA for experimental design sizes in the order of the dimension of the random variable space.

2. PCE representations

Let $\mathbf{X}$ be a random vector with outcome space $\mathbb{R}^n$ and joint PDF $f_{\mathbf{X}}(\mathbf{x})$. Consider the Hilbert space $\mathcal{H}$ of all functions from $\mathbb{R}^n$ to $\mathbb{R}$ with finite mean-square under the probability measure of $\mathbf{X}$. The inner product of two functions $g, h \in \mathcal{H}$ is defined as

$$\langle g, h \rangle_{\mathcal{H}} = \int_{\mathbb{R}^n} g(\mathbf{x}) h(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \ . \tag{1}$$

Consider the random variable $Y = \mathcal{M}(\mathbf{X})$ representing the response of an engineering model and assume that $\mathcal{M} \in \mathcal{H}$. Let $\{h_i(\mathbf{x}), i \in \mathbb{N}\}$ be a complete orthonormal basis of $\mathcal{H}$, thus satisfying

$$\langle h_i, h_j \rangle_{\mathcal{H}} = \delta_{ij} \ , \tag{2}$$

where δ_{ij} is the Kronecker symbol. Since $\{h_i(\mathbf{x}), i \in \mathbb{N}\}$ is a complete basis of $\mathcal{H}$, we can represent every element of $\mathcal{H}$ as a linear combination of the functions $\{h_i(\mathbf{x}), i \in \mathbb{N}\}$. Therefore Y can be expressed as

$$Y = \mathcal{M}(\mathbf{X}) = \sum_{i=0}^{\infty} a_i h_i(\mathbf{X}) \ . \tag{3}$$

Truncating the representation of Eq. (3) after the first L terms, we get the following approximation of Y

$$\hat{Y} = \sum_{i=0}^{L} a_i h_i(\mathbf{X}) \ , \tag{4}$$

which converges to Y in the mean-square sense as $L \to \infty$. We now make the following assumption on the distribution of the vector $\mathbf{X}$.

Assumption 1. *The random vector* $\mathbf{X}$ *follows the independent standard Gaussian distribution.*

In such case, we can construct an orthonormal polynomial basis of $\mathcal{H}$ using products of one-dimensional normalized Hermite polynomials [1]

$$\Psi_{\mathbf{k}}(\mathbf{X}) = \prod_{i=1}^{n} \psi_{k_i}(X_i) \ , \tag{5}$$

where $\{\psi_i(X), i \in \mathbb{N}\}$ are the normalized (probabilist) Hermite polynomials and $\mathbf{k} = (k_1, \dots, k_n) \in \mathbb{N}^n$.

Remark. *In cases where the Gaussian restriction of Assumption 1 does not apply, it is possible to express the random variable Y as a function of an underlying independent Gaussian input through performing an isoprobabilistic transformation [39].*

The p-th order total degree Hermite polynomial chaos expansion (PCE) of Y is the representation of Y on the space spanned by products of Hermite polynomials with total degree up to p

$$\hat{Y}_p = \sum_{|\mathbf{k}| \leq p} a_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{X}) \ , \tag{6}$$

where $|\mathbf{k}| = \sum_{i=1}^{n} k_i$. The total number of terms in Eq. (6) is

$$P = \begin{pmatrix} n+p \\ p \end{pmatrix} . \tag{7}$$

The coefficients $a_{\mathbf{k}}$ are found by projecting $\mathcal{M}(\mathbf{X})$ on the space spanned by $\{\Psi_{\mathbf{k}}, |\mathbf{k}| \leq p\}$. According to the projection theorem, this is equivalent to minimizing the norm of the truncation error of the PCE representation $\|Y - \hat{Y}_p\|_{\mathcal{H}} = \mathrm{E}[(Y - \hat{Y}_p)^2]^{1/2}$. Using a set of samples $\mathcal{X} = \{\mathbf{x}^{(i)}, i = 1, \ldots, N\}$ from the distribution of $\mathbf{X}$ and corresponding model evaluations $\mathcal{Y} = \{y^{(i)} = \mathcal{M}(\mathbf{x}^{(i)}), i = 1, \ldots, N\}$, one can estimate the coefficients $a_{\mathbf{k}}$ through minimizing a sample estimate of $\mathrm{E}[(Y - \hat{Y}_p)^2]$, i.e. through solving

$$\hat{\mathbf{a}} = \underset{\mathbf{a}:\mathbf{a}\in\mathbb{R}^P}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \left[y^{(i)} - \sum_{|\mathbf{k}|\leq p} a_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{x}^{(i)}) \right]^2 . \tag{8}$$

Eq. (8) corresponds to an ordinary least squares (OLS) regression problem [6].

Alternative choices of the experimental design set $\mathcal{X}$ can lead to a weighted least squares problem, e.g. [21]. As seen in Eq. (7), the total number of terms P in the PCE representation increases factorially with increase of either the dimension n or the polynomial degree p. A typical requirement for obtaining reasonable estimates of the regression coefficients in OLS is that $N > 2P$, e.g. [40, 17]. It is noted that the minimum number of experimental points N depends on the sampling scheme used and on the probability measure of $\mathbf{X}$, and could potentially be significantly larger than $2P$. In fact, for standard Monte Carlo sampling it has been shown that stable solutions are obtained at best for $N \sim \mathcal{O}(P\log(P))$ and in some cases for $N \sim \mathcal{O}(P^2)$ [21, 20].

Therefore, for a fixed number of experimental points N, the number of terms in the expansion that can be computed with accuracy through solution of Eq. (8) is limited.

To avoid over-fitting and obtain reliable predictions even for relatively large P, several adaptive approaches have been proposed for selecting the most significant terms in the PCE based on solving regularized regression problems [11, 14, 15]. These approaches result in sparse PCE representations. Although regression-based sparse PCEs perform well in moderate dimensional problems, their applicability in high dimensions ($n > 100$) is limited. This is due to the fact that they require the evaluation of the full set of multi-indices $\mathbf{k}$ of the orthogonal polynomials as well as the assembly and storage of the corresponding Vandermonde matrix. In particular, adaptive methods evaluate the multi-indices for a typically high maximum polynomial degree. For the total degree construction of Eq. (6), this requires a huge computing and storage capability in dimensions > 100, when considering that common algorithms for ordering the multi-indices scale exponentially with the dimension, e.g. [41]. Here, we focus on a different approach for reducing the number of coefficients in the PCE that performs a linear coordinate transformation of the Gaussian parameter space.

2.1. Transformed PCE basis

Consider the following coordinate transformation:

$$\mathbf{Z} = \mathbf{Q}^{\mathrm{T}}\mathbf{X} , \tag{9}$$

where $\mathbf{Q}$ is an $n \times n$ orthogonal matrix, i.e. satisfying $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$. The columns of matrix $\mathbf{Q}$ are orthonormal vectors and define a complete basis in $\mathbb{R}^n$. Therefore, Eq. (9) defines a projection of $\mathbf{X}$ on the coordinate system defined by the columns of $\mathbf{Q}$. Due to the rotational symmetry of the independent standard Gaussian distribution of $\mathbf{X}$, any orthogonal transformation of $\mathbf{X}$ will also be independent standard Gaussian. Therefore, the polynomials defined in Eq. (5) with argument $\mathbf{Z}$ form a complete basis in the transformed space defined by the matrix $\mathbf{Q}$. The p-th order PCE representation of Y on the transformed space reads

$$\hat{Y}_p^{\mathbf{Q}} = \sum_{|\mathbf{k}|\leq p} b_{\mathbf{k}} \Psi_{\mathbf{k}}(\mathbf{Z}) = \sum_{|\mathbf{k}|\leq p} b_{\mathbf{k}} \Psi_{\mathbf{k}}\left(\mathbf{Q}^{\mathrm{T}}\mathbf{X}\right) . \tag{10}$$

The representations of Eq. (10) and Eq. (6) are equivalent, i.e. it is possible to express the coefficients $b_{\mathbf{k}}$ in terms of $a_{\mathbf{k}}$ [29]. The representation of Eq. (10), introduced by Tipireddy and Ghanem [29], provides an additional flexibility in constructing sparse PCE representations through identifying dominant effects in the form of linear combinations of input variables. For example, if a dominant direction in $\mathbf{X}$-space is known, then one possible construction of matrix $\mathbf{Q}$ is to set its first column equal to the dominant direction and determine the remaining columns by the Gram-Schmidt process. In such case, it might be possible to obtain an accurate representation of Y by only including the terms in Eq. (10) for which $\{k_i = 0, i = 2, \dots, n\}$. In [29], it is suggested to determine the dominant direction by estimating the low-order PCE coefficients. As an example, the direction defined by the linear PCE coefficients is $\mathbf{q}_1 \propto [a_{(1,\dots,0)}; \dots ; a_{(0,\dots,1)}]$.

Consider now the case where one has identified a set of m dominant directions. We formalize the process of retaining only the terms in Eq. (10) with non-zero indices in $\mathbf{k}$ for $i \leq m$ by defining a reduced orthogonal matrix $\mathbf{Q}_m$ of dimensions $n \times m$, whose columns correspond to the dominant directions. The corresponding PCE representation reads

$$\hat{Y}_p^{\mathbf{Q}_m} = \sum_{|\mathbf{k}_m| \leq p} b_{\mathbf{k}_m} \Psi_{\mathbf{k}_m} \left(\mathbf{Q}_m^{\mathrm{T}} \mathbf{X} \right) \,, \tag{11}$$

where $\mathbf{k}_m \in \mathbb{N}^m$. In the following section we discuss an approach that determines directions in the input space with high predictive ability based on a set of experimental points. In the subsequent section, we employ this approach to construct the matrix $\mathbf{Q}_m$ and compute the corresponding coefficients $\{b_{\mathbf{k}_m}, \mathbf{k}_m \in \mathbb{N}^m\}$.

3. Partial least squares

Partial Least Squares (PLS) is a modelling technique that attempts to find relations between observable variables using latent variables [34, 35, 36]. This approach was originally developed in the field of chemometrics, where it is often the case that the number of independent variables in an experimental setting is significantly larger than the number of data points, whereas the underlying process is driven by a small number of latent (not directly observable) variables. The basic idea of PLS is to find uncorrelated linear transformations of the original predictor variables that have high covariance with the response variables.

Let $\mathcal{X}$ be an $N \times n$ matrix of samples from the input random vector $\mathbf{X}$ and let $\mathcal{Y}$ be the corresponding $N \times 1$ vector of model responses. It is convenient to assume that both $\mathcal{X}$ and $\mathcal{Y}$ are centered around their means; centering implies the operation $\mathcal{X} \leftarrow \mathcal{X} - \bar{\mathcal{X}}$, where $\bar{\mathcal{X}}$ denotes the arithmetic mean of $\mathcal{X}$. Standard PLS projects the matrix $\mathcal{X}$ to latent components $\mathbf{t}_i$ of dimensions $N \times 1$ by sequentially maximizing the covariance between the response $\mathcal{Y}$ and the latent components. After determining each $\mathbf{t}_i$, it assumes a linear relationship between $\mathbf{t}_i$ and $\mathcal{Y}$ and evaluates the coefficient of $\mathbf{t}_i$ by OLS.

The procedure starts by evaluating the projection to the first latent component $\mathbf{t}_1 = \mathcal{X}\mathbf{w}_1$, where $\mathbf{w}_1$ has dimensions $n \times 1$, by maximizing the covariance between $\mathbf{t}_1$ and $\mathcal{Y}$ under the constraint that $\|\mathbf{w}_1\| = 1$. The corresponding optimization problem is stated as

$$\mathbf{w}_1 = \underset{\mathbf{w}: \mathbf{w} \in \mathbb{R}^n, \|\mathbf{w}\|=1}{\arg\max} \ \operatorname{cov}\left(\mathcal{Y}^{\mathrm{T}} \mathcal{X} \mathbf{w}, \mathcal{Y}^{\mathrm{T}} \mathcal{X} \mathbf{w} \right) \,. \tag{12}$$

The exact solution of Eq. (12) is given by

$$\mathbf{w}_1 = \frac{\mathcal{X}^{\mathrm{T}} \mathcal{Y}}{\|\mathcal{X}^{\mathrm{T}} \mathcal{Y}\|} \,. \tag{13}$$

The regression coefficient of $\mathbf{t}_1$ is then evaluated by OLS as

$$b_1 = \frac{\mathbf{t}_1^{\mathrm{T}} \mathcal{Y}}{\mathbf{t}_1^{\mathrm{T}} \mathbf{t}_1} \,. \tag{14}$$

To obtain the next latent component, the residual matrices $\mathcal{E}$ and $\mathcal{F}$ for the regressor and response matrices, $\mathcal{X}$ and $\mathcal{Y}$, respectively, are evaluated by subtracting from $\mathcal{X}$ and $\mathcal{Y}$ their rank-one approximations based on $\mathbf{t}_1$

$$\mathcal{E} = \mathcal{X} - \mathbf{t}_1 \mathbf{p}_1^{\mathrm{T}} \,, \tag{15}$$

$$\mathcal{F} = \mathcal{Y} - b_1\mathbf{t}_1 \ , \tag{16}$$

where $\mathbf{p}_1$ is the load vector corresponding to $\mathbf{t}_1$ and defines the projection of the rows of $\mathcal{X}$ on the first latent component. It is

$$\mathbf{p}_1 = \frac{\mathcal{X}^{\mathrm{T}}\mathbf{t}_1}{\mathbf{t}_1^{\mathrm{T}}\mathbf{t}_1} \ . \tag{17}$$

The procedure is continued by extracting the next component from the deflated matrices $\mathcal{E}$ and $\mathcal{F}$, until a certain error criterion is satisfied. The latter is usually based on estimates of the mean-square error of the PLS prediction. One possible approximation of this error is obtained through the norm of the residual $\|\mathcal{F}\|$. Alternatively, a more robust estimate can be derived based on cross-validation, e.g. [34]. The PLS process leads to a total of $m \leq n$ latent components or scores $\mathbf{t}_i$ and corresponding weight vectors $\mathbf{w}_i$ and load vectors $\mathbf{p}_i$. The PLS algorithm is summarized in Algorithm 1.

Algorithm 1 PLS algorithm

```
Input Data matrix $\mathcal{X}$ and response matrix $\mathcal{Y}$
Center matrices: $\mathcal{X} \leftarrow \mathcal{X} - \bar{\mathcal{X}}$, $\mathcal{Y} \leftarrow \mathcal{Y} - \bar{\mathcal{Y}}$
Set $\mathcal{E} = \mathcal{X}$, $\mathcal{F} = \mathcal{Y}$, $i = 1$
repeat
    Compute weight: $\mathbf{w}_i = \mathcal{E}^{\mathrm{T}}\mathcal{F}/\|\mathcal{E}^{\mathrm{T}}\mathcal{F}\|$
    Compute score: $\mathbf{t}_i = \mathcal{E}\mathbf{w}_i$
    Compute load: $\mathbf{p}_i = \mathcal{E}^{\mathrm{T}}\mathbf{t}_i/(\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i)$
    Compute regressor: $b_i = \mathbf{t}_i^{\mathrm{T}}\mathcal{F}/\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i$
    Deflate: $\mathcal{E} \leftarrow \mathcal{E} - \mathbf{t}_i\mathbf{p}_i^{\mathrm{T}}$, $\mathcal{F} \leftarrow \mathcal{F} - b_i\mathbf{t}_i$
    $i \leftarrow i + 1$
until change in $\|\mathcal{F}\|$ is smaller than $\epsilon_y$
```

Remark. *The deflation of the response vector $\mathcal{Y}$ in step 9 of Algorithm 1 is not required for computing the PLS weights, i.e. it will not influence the resulting components and regression coefficients [35].*

Remark. *Algorithm 1 is often termed PLS1 in the literature to distinguish it from the PLS algorithm for multivariate outputs, termed PLS2.*

The scores, weights and loads computed by the PLS algorithm, can be gathered in matrices $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_m] \in \mathbb{R}^{N \times m}$, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m] \in \mathbb{R}^{n \times m}$ and $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_m] \in \mathbb{R}^{n \times m}$. The PLS algorithm determines each latent component as a linear combination of columns of the corresponding residual matrix $\mathcal{E}$. However, it is also possible to express $\mathbf{t}_i$ as linear combinations of the data matrix $\mathcal{X}$ [42, 43]. Define the matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ as follows

$$\mathbf{R} = \mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1} \ . \tag{18}$$

The PLS scores can be expressed by projecting $\mathcal{X}$ on the space defined by the columns of matrix $\mathbf{R}$, i.e.

$$\mathbf{T} = \mathcal{X}\mathbf{R} \ . \tag{19}$$

The columns of matrix $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_m]$ can also be obtained through the following recursive relation [35]

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{w}_1 \ , \\ \mathbf{r}_i &= \mathbf{w}_i - \mathbf{r}_{i-1}\left(\mathbf{p}_{i-1}^{\mathrm{T}}\mathbf{w}_i\right) \ . \end{aligned} \tag{20}$$

The approximation $\hat{\mathcal{X}}$ of the data matrix $\mathcal{X}$ using m PLS components can be expressed using the load matrix $\mathbf{P}$ as follows

$$\hat{\mathcal{X}} = \mathbf{T}\mathbf{P}^{\mathrm{T}} \ . \tag{21}$$

3.1. Properties of the PLS matrices

PLS identifies dominant directions in the input space that can be potentially used within the context of PCE in transformed basis. In order to proceed, it is useful to review the properties of the matrices derived by the PLS process.

The PLS algorithm identifies orthogonal directions in the data space, i.e. it holds [35]

$$\mathbf{t}_i^{\mathrm{T}} \mathbf{t}_j = 0 \text{ for } i \neq j \ . \tag{22}$$

This implies that the PLS scores form an orthogonal basis in the space generated by the columns of matrix $\mathcal{X}$. Eq. (22) is a crucial ingredient of the PLS approach; it enables evaluation of the regression coefficients b_i one by one, as in step 8 of Algorithm 1.

The columns of matrices $\mathbf{W}$, $\mathbf{P}$ and $\mathbf{R}$ define directions in the space generated by the rows of matrix $\mathcal{X}$; each direction corresponds to a linear combination of the underlying random variables $\mathbf{X}$. The matrix $\mathbf{R}$ contains the reduced basis in $\mathbf{X}$-space that defines the PLS components, cf. Eq. (19).

The matrix $\mathbf{W}$ is orthogonal, i.e. it holds $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$ [35], and hence its columns form an orthonormal basis in the $\mathbf{X}$-space. However, the same cannot be said for matrices $\mathbf{P}$ and $\mathbf{R}$. Therefore, if the PLS algorithm is applied not for prediction purposes but to determine orthogonal dominant directions then the orthogonal projection matrix $\mathbf{W}$ is often the desired output, e.g. [44]. The matrices $\mathbf{P}$ and $\mathbf{R}$ are not necessarily orthogonal, but their columns are mutually orthogonal, i.e. it holds $\mathbf{P}^{\mathrm{T}}\mathbf{R} = \mathbf{I}$ [43].

In the context of PCE representations in transformed basis, it is possible to define the reduced basis using the orthogonal directions of matrix $\mathbf{W}$, i.e. setting $\mathbf{Q}_m = \mathbf{W}$ in Eq. (11). Projecting the data on the columns of matrix $\mathbf{W}$ would lead to scores $\widetilde{\mathbf{T}} = \mathcal{X}\mathbf{W}$, that differ from the PLS scores given by Eq. (19). This is somewhat suboptimal as the PLS components are determined based on maximizing the covariance with the prediction error obtained from regressing the PLS scores $\mathbf{T}$ (and not $\widetilde{\mathbf{T}}$) with $\mathcal{Y}$. Moreover, unlike the PLS scores $\mathbf{T}$, the scores $\widetilde{\mathbf{T}}$ are not necessarily orthogonal.

As will become clear in the next section, employing directly the directions of the PLS components, i.e. the columns of matrix $\mathbf{R}$, to define the PCE reduced basis is of particular benefit; it allows estimating simultaneously the reduced basis and PCE coefficients. However, as the matrix $\mathbf{R}$ is in general not orthogonal, the projection of the random variables $\mathbf{X}$ on the columns of $\mathbf{R}$ will not be independent standard Gaussian. This poses a problem to the PCE representation of Eq. (11): the Hermite polynomial basis of Eq (5) will not be orthogonal in the resulting transformed space. That is, if the matrix $\mathbf{R}$ is not orthogonal then the orthogonal polynomial basis need to be determined for the problem at hand, e.g. by application of the Gram-Schmidt process.

Next we look more closely at the properties of matrix $\mathbf{R}$ and determine a condition under which this matrix becomes orthogonal.

Lemma 3.1. *The matrix* $\mathbf{R}^{\mathrm{T}}\mathbf{W}$ *is a lower triangular matrix with unit diagonal elements.*

Proof. The proof follows from the recursive relation of Eq. (20). Eq. (20) can be rewritten as follows

$$\mathbf{r}_i = \mathbf{w}_i + \sum_{j=1}^{i-1} c_j \mathbf{w}_j \ ,$$

where $c_j \in \mathbb{R}$ are coefficients, which are not used here. This gives

$$\mathbf{r}_i^{\mathrm{T}} \mathbf{w}_k = 0 \text{ for } i < k$$

and

$$\mathbf{r}_i^{\mathrm{T}} \mathbf{w}_i = 1 \ ,$$

where we have used that $\mathbf{w}_i^{\mathrm{T}} \mathbf{w}_j = \delta_{ij}$. □

We now consider the special case where the columns of the centered data matrix $\mathcal{X}$ have equal norm and are mutually orthogonal, i.e.

$$\mathcal{X}^{\mathrm{T}}\mathcal{X} = \omega \mathbf{I} \ , \tag{23}$$

where $\omega \in \mathbb{R}_{>0}$ and $\mathbf{I}$ is the $n \times n$ identity matrix.

Proposition 3.1. *If the data matrix $\mathcal{X}$ satisfies Eq.* (23)*, then the matrix $\mathbf{R}$ is orthogonal, i.e.* $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}$.

Proof. From Eq. (22) and expressing each PLS score as $\mathbf{t}_i = \mathcal{X}\mathbf{r}_i$, we get

$$\mathbf{r}_i^{\mathrm{T}} \left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right) \mathbf{r}_j = 0 \text{ for } i \neq k \ .$$

Using Eq (23), we obtain

$$\mathbf{r}_i^{\mathrm{T}}\mathbf{r}_j = 0 \text{ for } i \neq k \ .$$

From Eq. (20) and using Lemma 3.1, we have

$$\mathbf{r}_i^{\mathrm{T}}\mathbf{r}_i = \mathbf{r}_i^{\mathrm{T}}\mathbf{w}_i - \mathbf{r}_i^{\mathrm{T}}\mathbf{r}_{i-1}\left(\mathbf{p}_{i-1}^{\mathrm{T}}\mathbf{w}_i\right) = \mathbf{r}_i^{\mathrm{T}}\mathbf{w}_i = 1 \ .$$

□

Proposition 3.1 implies that if the pair-wise sample correlation of the underlying random variables vanishes and the sample variances of all variables are equal, then the directions identified by PLS, the columns of $\mathbf{R}$, will be orthogonal and will have unit length. One interpretation of this result is that if the underlying variable space is uncorrelated and has equal variances, then the directions identified by PLS should contribute equally to the explained variance of $\mathcal{X}$.

Although Eq. (23) is not necessarily true, under Assumption 1 it is asymptotically true as $N \to \infty$. Moreover, one can generate $\mathcal{X}$ through sampling techniques that aim at approximately satisfying Eq. (23) for finite N. This is achieved by stratification techniques such as Latin hypercube sampling (LHS) [45], or a modified version that aims at minimizing the correlation between samples, e.g. [46].

4. PLS-based PCE

We now discuss how the PLS approach can be used in the context of the PCE representation in transformed basis. We discuss two versions of PLS-based PCE representations; the first is based on the linear PLS algorithm presented in Section 3 and the second is based on a nonlinear version of the PLS algorithm tailored for use with PCE models.

4.1. Linear PLS-based transformation

As mentioned in the previous section, it is possible to build a PCE representation of Y by selecting $Q_m = \mathbf{W}$ or $\mathbf{R}$ in Eq. (11). The matrix of PLS weights $\mathbf{W}$ is orthogonal, while the matrix $\mathbf{R}$ defining the PLS components in $\mathbf{X}$-space is asymptotically orthogonal as $N \to \infty$. Having determined the PLS directions with samples $\mathcal{X} = \{\mathbf{x}^{(i)}, i = 1, \ldots, N\}$ from the distribution of $\mathbf{X}$ and corresponding model evaluations $\mathcal{Y} = \{y^{(i)} = \mathcal{M}(\mathbf{x}^{(i)}), i = 1, \ldots, N\}$, the same samples can be used to determine the PCE coefficients $\{b_{\mathbf{k}_m}, \mathbf{k}_m \in \mathbb{N}^m\}$ in Eq. (11) by OLS regression.

We note that for the first PLS component it holds $\mathbf{r}_1 = \mathbf{w}_1$, cf. Eq. (20). We make the following observation.

Proposition 4.1. *If the reduced basis is determined by a single direction ($m = 1$) as computed by Algorithm 1, then this direction is asymptotically equivalent to the direction defined by the first-order PCE coefficients.*

Proof. The first linear PLS direction $\mathbf{r}_1$ is obtained as follows

$$\mathbf{r}_1 \propto \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \bar{\mathcal{X}})(y^{(i)} - \bar{\mathcal{Y}}) \ ,$$

where $\bar{\mathcal{X}}$ and $\bar{\mathcal{Y}}$ are the arithmetic means of $\{\mathbf{x}^{(i)}, i = 1, \ldots, N\}$ and $\{y^{(i)}, i = 1, \ldots, N\}$, respectively. Under Assumption 1, $\bar{\mathcal{X}} \to 0$ as $N \to \infty$. Therefore as $N \to \infty$, it is

$$\mathbf{r}_1 \propto \sum_{i=1}^{N} \mathbf{x}^{(i)} y^{(i)} \ .$$

The vector of first-order PCE coefficients $\mathbf{a}_1$ is obtained by projecting $Y = \mathcal{M}(\mathbf{X})$ on the linear Hermite polynomials $\{\Psi_{\mathbf{k}}, |\mathbf{k}| = 1\} = \{X_1, \ldots, X_n\}$. It is

$$\mathbf{a}_1 = \mathrm{E}[\mathcal{M}(\mathbf{X})\mathbf{X}] = \mathrm{E}[Y\mathbf{X}] \ .$$

As the samples $\mathbf{x}^{(i)}$ and $y^{(i)}$ follow respectively the distribution of $\mathbf{X}$ and Y, the PLS direction $\mathbf{r}_1$ converges to the one defined by $\mathbf{a}_1$ as $N \to \infty$. □

Proposition 4.1 implies that the linear PLS algorithm with $m = 1$ is asymptotically equivalent to the linear PCE-driven Gaussian adaptation proposed in Section 3.1 of [29].

The linear PLS algorithm does not not always result in an optimal coordinate transformation because the matrices $\mathbf{W}$ and $\mathbf{R}$ are evaluated assuming a linear relationship between the response and each latent variable. Next we discuss an approach that adapts PLS for obtaining directions that are optimal for use in the representation of Eq. (11).

4.2. PCE-driven PLS-based transformation

Several variants of the classical PLS method have been proposed for addressing problems where the underlying process is nonlinear [47]. Here, we employ the approach proposed by Wold et al. [37] and later modified by Baffi et al. [38]. This approach was originally proposed for quadratic models and later extended for use with neural networks [48]. Here, we adapt this approach for use in PCE representations.

The nonlinear PLS algorithm proceeds by obtaining a first approximation of each projection vector $\mathbf{w}_i$ with standard PLS. Thereafter, it assumes a nonlinear relationship between the response and the latent variable $\mathbf{t}_i$, which is fitted by OLS. Within the context of PCE representations, the nonlinear relationship in each latent variable is a one-dimensional Hermite polynomial expansion of order p

$$\hat{\mathcal{M}}_i^p(t) = \sum_{j=1}^{p} \hat{b}_{ij}^p \psi_j(t) \ . \tag{24}$$

For the first latent variable, the PCE regression problem is stated as

$$\mathcal{Y} = \sum_{j=1}^{p} b_{1j}^p \psi_j(\mathbf{t}_1) + \mathbf{e} \ , \tag{25}$$

wherein the operations are performed element-wise, $\mathbf{t}_1 = \mathcal{X}\mathbf{w}_1$ and $\mathbf{e}$ is the vector of regression errors. Eq. (25) is solved for the coefficients $\{b_{1j}^p\}$. The vector $\mathbf{w}_1$ is then modified iteratively by means of a Newton-Raphson linearisation of Eq. (25), i.e. by performing a first-order Taylor series expansion of Eq. (25) with respect to $\mathbf{w}_1$ and then solving it for the increment $\Delta\mathbf{w}_1$. This gives

$$\Delta\mathbf{w}_1 = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-}\mathbf{A}^{\mathrm{T}}\mathbf{e} \ , \tag{26}$$

where $(\cdot)^-$ denotes the generalized inverse of a matrix and $\mathbf{A}$ is the gradient of the PCE model with respect to the weights $\mathbf{A} = \nabla_{\mathbf{w}}\hat{\mathcal{M}}_1^p(\mathcal{X}\mathbf{w})$. Thereafter, $\mathbf{w}_1$ is updated, $\mathbf{w}_1 = \mathbf{w}_1 + \Delta\mathbf{w}_1$, and normalized. The latent component $\mathbf{t}_1$ is then updated, Eq. (25) is fitted anew and the next increment $\Delta\mathbf{w}_1$ is evaluated. This iterative procedure is continued until $\Delta\mathbf{w}_1$ is sufficiently small.

To obtain the next latent component, the residual matrices $\mathcal{E}$ and $\mathcal{F}$ are evaluated by subtracting from $\mathcal{X}$ its rank-one approximations based on $\mathbf{t}_1$ and from $\mathcal{Y}$ its PCE approximation using the first direction $\hat{\mathcal{M}}_1^p(\mathbf{t}_1)$ and the same process is repeated using $\mathcal{E}$ and $\mathcal{F}$ as the new $\mathcal{X}$ and $\mathcal{Y}$.

In order to obtain PLS directions that reflect the nonlinear nature of the underlying process while avoiding over-fitting, we choose the polynomial degree in Eq. (25) for each latent variable by evaluating each latent component for different polynomial degrees $q = \{1, \ldots, p\}$ and retaining the one that results in the smallest modified leave-one-out error ϵ_{LOO}^q; ϵ_{LOO}^q can be evaluated based on a single PCE built using the Vandermonde matrix of the OLS problem [49, 11]. The PCE-driven PLS algorithm is detailed in Algorithm 2.

Algorithm 2 PCE-driven PLS algorithm

```
 1: **Input** Data matrix $\mathcal{X}$ and response matrix $\mathcal{Y}$
 2: Center matrices: $\mathcal{X} \leftarrow \mathcal{X} - \bar{\mathcal{X}}$, $\mathcal{Y} \leftarrow \mathcal{Y} - \bar{\mathcal{Y}}$
 3: Set $\mathcal{E} = \mathcal{X}$, $\mathcal{F} = \mathcal{Y}$, $i = 1$
 4: repeat
 5:     Compute weight: $\mathbf{w}_i^0 = \mathcal{E}^{\mathrm{T}}\mathcal{F}/\|\mathcal{E}^{\mathrm{T}}\mathcal{F}\|$
 6:     for $q \leftarrow 1, p$ do
 7:         Set $\mathbf{w}_i^q = \mathbf{w}_i^0$
 8:         repeat
 9:             Compute score: $\mathbf{t}_i^q = \mathcal{E}\mathbf{w}_i^q$
10:             Fit a 1D PCE of order $q$: $\hat{\mathbf{b}}_i^q \leftarrow$ fit $\left[\mathcal{F} = \sum_{j=1}^q b_{ij}^q \psi_j(\mathbf{t}_i^q) + \epsilon\right]$
11:             Set $\hat{\mathcal{M}}_i^q(t) = \sum_{j=1}^q \hat{b}_{ij}^q \psi_j(t)$
12:             Compute the error: $\hat{\mathcal{F}} = \hat{\mathcal{M}}_i^q(\mathbf{t}_i^q)$; $\mathbf{e} = \mathcal{F} - \hat{\mathcal{F}}$
13:             Compute: $\Delta\mathbf{w}_i^q = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-}\mathbf{A}^{\mathrm{T}}\mathbf{e}$ with $\mathbf{A} = \nabla_{\mathbf{w}}\hat{\mathcal{M}}_i^q(\mathcal{E}\mathbf{w})$
14:             Set: $\mathbf{w}_i^q \leftarrow \mathbf{w}_i^q + \Delta\mathbf{w}_i^q$
15:             Normalize: $\mathbf{w}_i^q \leftarrow \mathbf{w}_i^q/\|\mathbf{w}_i^q\|$
16:         until $\|\Delta\mathbf{w}_i^q\|$ is smaller than $\epsilon_w$
17:         Evaluate the relative leave-one-out error $\epsilon_{\text{LOO}}^q$ as in [11]
18:     end for
19:     Set $\{q_i, \hat{\mathcal{M}}_i^{q_i}, \mathbf{w}_i^{q_i}\}$ as the triple $\{q, \hat{\mathcal{M}}_i^q, \mathbf{w}_i^q\}$ with the smallest $\epsilon_{\text{LOO}}^q$
20:     Compute score: $\mathbf{t}_i^{q_i} = \mathcal{E}\mathbf{w}_i^{q_i}$
21:     Compute load: $\mathbf{p}_i^{q_i} = \mathcal{E}^{\mathrm{T}}\mathbf{t}_i^{q_i}/((\mathbf{t}_i^{q_i})^{\mathrm{T}}\mathbf{t}_i^{q_i})$
22:     Deflate: $\mathcal{E} \leftarrow \mathcal{E} - \mathbf{t}_i^{q_i}(\mathbf{p}_i^{q_i})^{\mathrm{T}}$, $\mathcal{F} \leftarrow \mathcal{F} - \hat{\mathcal{M}}_i^{q_i}(\mathbf{t}_i^{q_i})$
23:     $i \leftarrow i + 1$
24: until change in $\|\mathcal{F}\|$ is smaller than $\epsilon_y$
25: return $\{q_i, \hat{\mathcal{M}}_i^{q_i}, \mathbf{t}_i^{q_i}, \mathbf{w}_i^{q_i}, \mathbf{p}_i^{q_i}\}, i = 1 \ldots, m$.
```

Remark. *The matrix* $\mathbf{A}$ *required in step 13 of Algorithm 2 can be computed analytically using the properties of the derivatives of the Hermite polynomials [50] and hence does not require additional model evaluations.*

The algorithm returns m quintuples $\{q_i, \hat{\mathcal{M}}_i^{q_i}, \mathbf{t}_i^{q_i}, \mathbf{w}_i^{q_i}, \mathbf{p}_i^{q_i}\}$, with q_i denoting the polynomial degree, $\hat{\mathcal{M}}_i^{q_i}$ the one-dimensional PCE representation, $\mathbf{t}_i^{q_i}$ the scores, $\mathbf{w}_i^{q_i}$ the weights and $\mathbf{p}_i^{q_i}$ the loads of the i-th PLS component. The PLS directions $\mathbf{r}_i^{q_i}$ can then be evaluated through Eq. (18) or Eq. (20). Using the one-dimensional fitted PCEs $\hat{\mathcal{M}}_i^{q_i}$ and the PLS directions $\mathbf{r}_i^{q_i}$, we obtain the following representation

$$\hat{Y}_m^{\text{PLS}} = \hat{\mathcal{M}}_m(\mathbf{X}) = b_0 + \sum_{i=1}^m \hat{\mathcal{M}}_i^{q_i}\left[(\mathbf{r}_i^{q_i})^{\mathrm{T}}\tilde{\mathbf{X}})\right] \ , \tag{27}$$

where $b_0 = \bar{\mathcal{Y}}$ and $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathcal{X}}$. Due to the asymptotic behavior of the matrix $\mathbf{R} = [\mathbf{r}_1^{q_1}, \ldots, \mathbf{r}_m^{q_m}]$, described in Proposition 3.1, and because $\tilde{\mathbf{X}} \to \mathbf{X}$ as $N \to \infty$, the representation of Eq. (27) is asymptotically equivalent to the PCE representation of Eq. (11) with $\mathbf{Q}_m = \mathbf{R}$, for the case where only the main-effects in the transformed coordinate system are considered.

The PCE-driven PLS algorithm identifies simultaneously the PLS directions and the coefficients of the one-dimensional PCEs in each PLS component. A disadvantage of this approach is that the matrix $\mathbf{R}$ defining the PLS components is only asymptotically orthogonal. This implies that for finite N the multivariate Hermite polynomials are not orthogonal with respect to the distribution of the latent variables and hence the polynomial basis loses its optimality. However, as mentioned in Section 3.1, tailored sampling techniques can be used to obtain approximately orthogonal PLS directions for finite N.

An alternative approach would be to employ the orthogonal matrix $\mathbf{W} = [\mathbf{w}_1^{q_1}, \ldots, \mathbf{w}_m^{q_m}]$ obtained from Algorithm 2 to define the orthogonal projection. In such case, the PCE coefficients need to be evaluated anew through setting $\mathbf{Q}_m = \mathbf{W}$ and regressing Eq. (11) with the responses $\mathcal{Y}$.

5. Examples

In this section, we evaluate the proposed method with three numerical examples in high dimensions. We investigate the performance of both the linear PLS-based approach of Section 4.1 and the PCE-driven PLS algorithm of Section 4.2. We compare the linear PLS approach with a single latent component to the linear PCE-driven Gaussian adaptation of Tipireddy and Ghanem (TG) [29], computed with a quadrature-based pseudo-spectral projection, to numerically verify Proposition 4.1. We employ a sufficient number of quadrature points in the TG approach to ensure accuracy of the result. We note that the proposed PLS methods are based on model evaluations at a set of samples from the distribution of the input variables, whereas the TG approach applies numerical quadrature to evaluate the PCE coefficients. Hence, a direct comparison of the computational cost of PLS methods with the TG approach would be difficult to set up[1]. Instead, we compare the performance of the PCE-driven PLS method against polynomial-basis low-rank approximations (LRA), which can be constructed based on the same set of model evaluations. We use the LRA implementation of UQlab [51], which employs alternating least-squares to fit the LRA with an adaptive scheme for the rank selection while considering every polynomial order up to p within the selected ranks. We choose to compare our method to the particular implementation of LRA, as it has been shown that it performs better than sparse PCEs in moderate dimensional problems and small experimental designs [28]. In addition, the computational cost of building an LRA in high dimensions is feasible as its construction is based on products of univariate polynomial expansions. In contrast, sparse PCEs, e.g. based on compressive sensing, require evaluation and storage of the full Vandermonde matrix for high polynomial orders, which is prohibitive in very high dimensions (see the relevant discussion in Section 2). For all methods, we choose a maximum polynomial degree of $p = 10$ and for the LRA a maximum rank of $R = 10$.

The reduced basis identified by the columns of the $\mathbf{R}$-matrix, $\{\mathbf{r}_i^{q_i}, i = 1, \ldots, m\}$, need not be orthogonal for finite sample size N; this compromises the optimality of the Hermite PCE basis as the basis will not be orthogonal in the transformed space. In order to quantify this error, we consider the Gramian of $\mathbf{R}$

$$\mathbf{G}(\mathbf{R}) = \mathbf{R}^{\mathrm{T}}\mathbf{R} \ . \tag{28}$$

If $\mathbf{R}$ is orthogonal, then it is $\mathbf{G}(\mathbf{R}) = \mathbf{I}$. We therefore use $\|\mathbf{I} - \mathbf{G}(\mathbf{R})\|_2$ as the error measure, where $\|\cdot\|_2$ denotes the 2-norm of a matrix.

We compare the proposed method against LRA in terms of the errors in the mean and variance of the output quantity of interest as well as the generalization error err_G. The latter is defined as the mean-square of the residual

$$\mathrm{err}_G = \mathrm{E}[(Y - \hat{Y})^2] \ , \tag{29}$$

[1] The first direction obtained by the linear PLS-based PCE can be viewed as a Monte Carlo approximation of the linear PCE-based transformation of TG. Therefore, the linear PLS-based approach is itself a way of approximating TG.

where $\hat{Y}$ denotes the output response of the surrogate model. An estimate $\widehat{\mathrm{err}}_G$ of err_G can be obtained using a large set of samples $\mathcal{X}_{\mathrm{val}}$, termed validation set.

The experimental designs for both PLS-PCE and LRA are generated via LHS with sample decorrelation, using the built-in Matlab function lhsdesign with correlation criteria; this function iteratively generates samples with LHS to find the ones with the smaller sample correlation. For each example, the analysis is performed 100 times to obtain confidence intervals (CI) on the predictive quantities. Reference solutions are obtained with Monte Carlo simulation with 2×10^5 samples.

5.1. Linear elastic bar

The first example consists of a linear elastic bar of length $L = 1$m, as shown in Figure 1. The displacement of the bar $u(x)$ satisfies the following differential equation

$$-\frac{d}{dx}\left(D(x)\frac{d}{dx}u\right) = q(x) \text{ in } [0, L] \ . \tag{30}$$

The axial resistance of the bar $D(x) = EA(x)$ is described by a homogeneous random field with lognormal marginal distribution with mean $\mu_D = 100$kN and standard deviation $\sigma_D = 10$kN. The autocorrelation function of the underlying Gaussian random field $\ln D$ is $\rho_{\ln D}(\Delta x) = \exp(-|\Delta x|/l)$ with correlation length $l = 0.04$m. The random field $\ln D$ is represented by a Karhunen-Loève (KL) expansion [1] with 100 terms, which captures 95% of the variability of $\ln D$. This leads to an input random vector consisting of 100 independent standard Gaussian random variables. The bar is subjected to a deterministic load $q = 1$kN/m. Eq. (30) is solved by the finite element method with 100 piecewise linear finite elements. The output quantity is the displacement at the tip of the bar $Y = u(L)$.

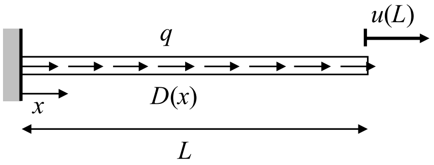

Figure 1: Linear elastic bar with random axial rigidity.

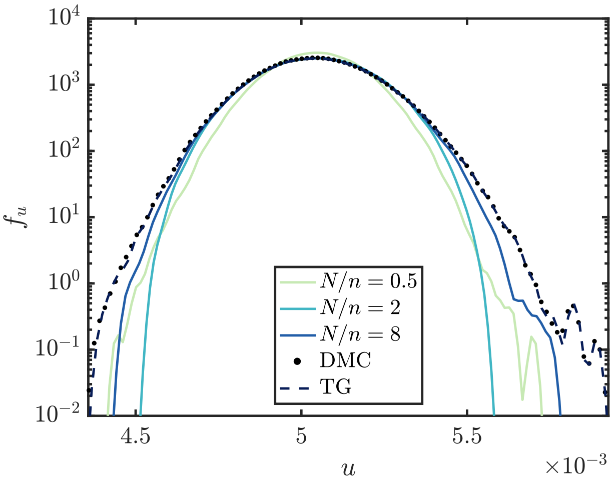

Figure 2: Linear elastic bar: Log-densities of response. Comparison of linear PLS-PCE (with $m = 1$) with various design sizes with the linear PCE-based adaptation of Tipireddy and Ghanem (TG) [29] and direct Monte Carlo (DMC) using 2×10^5 samples.

Figure 2 compares the log-densities obtained with the linear PLS-based PCE with $m = 1$ for increasing experimental design sizes to the one obtained with the linear PCE-based adaptation of TG. The latter compares well with the reference solution computed with direct Monte Carlo, which implies that a single latent component suffices to describe the behavior of the model response. It is also shown that the linear PLS-based PCE approaches the solution of TG as the number of samples N increases, which verifies Proposition 4.1.

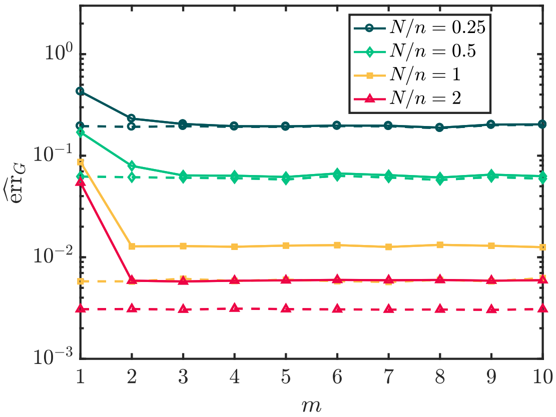

Figure 3: Linear elastic bar: Comparison of the linear PLS-based (solid line) and nonlinear (PCE-driven) PLS-based (dashed line) adaptation at various design sizes and number of latent components.

Figure 3 compares the generalization error obtained with the linear and PCE-driven PLS methods for increasing number of latent components m. It is seen that the PCE-driven PLS method using a single latent component yields consistently lower errors than the linear PLS , whereas using additional components does not improve the results. This is to be expected as the nonlinear PLS algorithm employed within the PCE-driven PLS approach identifies the directions that minimize the residual in the PCE approximation. As the linear PLS assumes a linear relationship between input and output and, hence, is not optimized for use with higher order PCEs, it requires more latent components to capture the behavior of the model.

We now compare the performance of the nonlinear PLS-PCE method with the LRA surrogate based on the same experimental design scheme. Both surrogates capture the response PDF increasingly well as the number of points in the experimental design rises (Figure 4). The PLS-based PCE model exhibits a bias at low experimental designs, which vanishes as the size of experimental design increases. The variability of the PDF estimates obtained by the LRA model is in general larger than the one of the PLS-based PCE model. Figure 5 shows the one-dimensional PCE model along the first latent direction. The nonlinearity (polynomial degree) of the PLS-based PCE model in general increases with the increase of the experimental design size. At $N/n = 2$, the PLS-based PCE model exhibits good convergence while the LRA model still produces occasional outliers (Figure 6). In terms of generalization error, the LRA error is slightly smaller for small experimental designs ($N/n \leq 0.5$). The PLS-based PCE error continuously decreases as we add points to the experimental design while the LRA error seems to stagnate from $N/n = 1$ on (Figure 7, top left). Both methods yield virtually identical mean errors, while the variance error exhibits similar behaviour as the generalization error (Figure 7, bottom right & left). Across all N, the PLS-based PCE reduced space is constructed with $m = 2$ directions, which exhibit relatively low orthogonality error (Figure 7, top right).

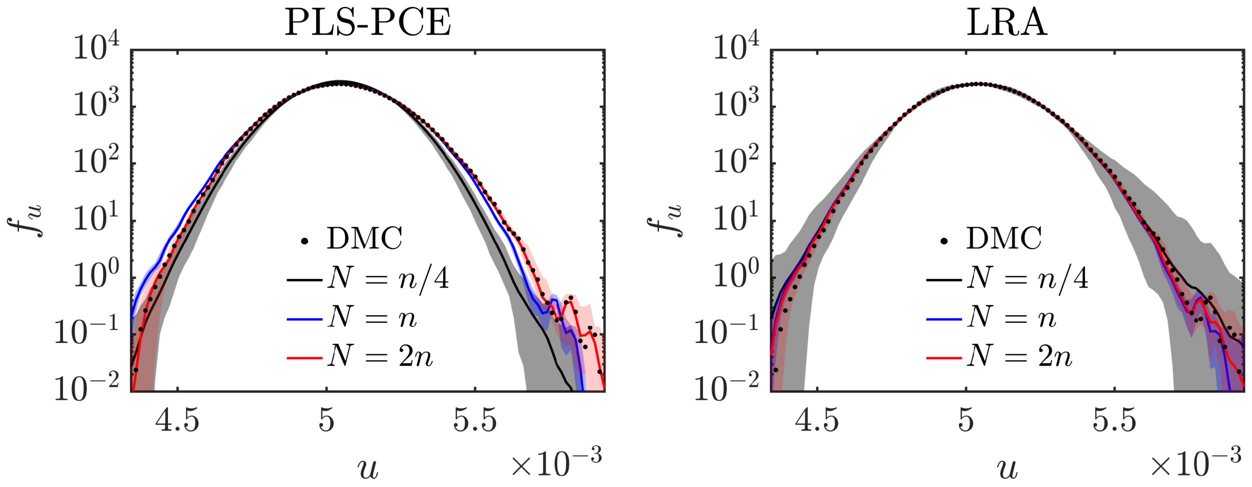

Figure 4: Linear elastic bar: Log-densities of response at various design sizes with 95 % CI. Results of (nonlinear) PLS-PCE and LRA are compared with direct Monte Carlo (DMC) results using 2×10^5 samples.

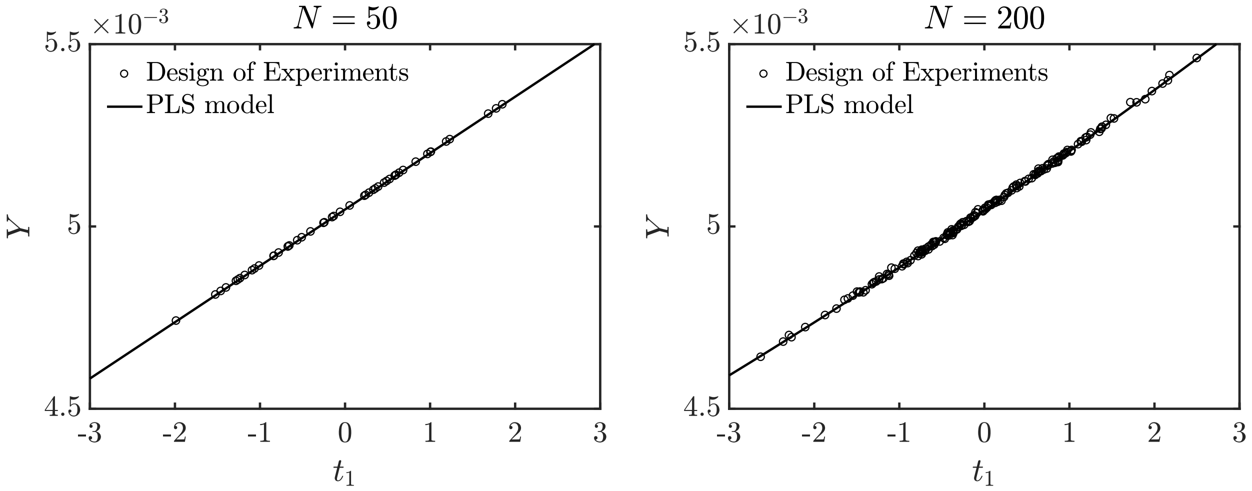

Figure 5: Linear elastic bar: 1-D Surrogate model along first PLS direction t_1 vs. the design of experiments at $N/n = 0.5$ (left) and $N/n = 2$ (right).

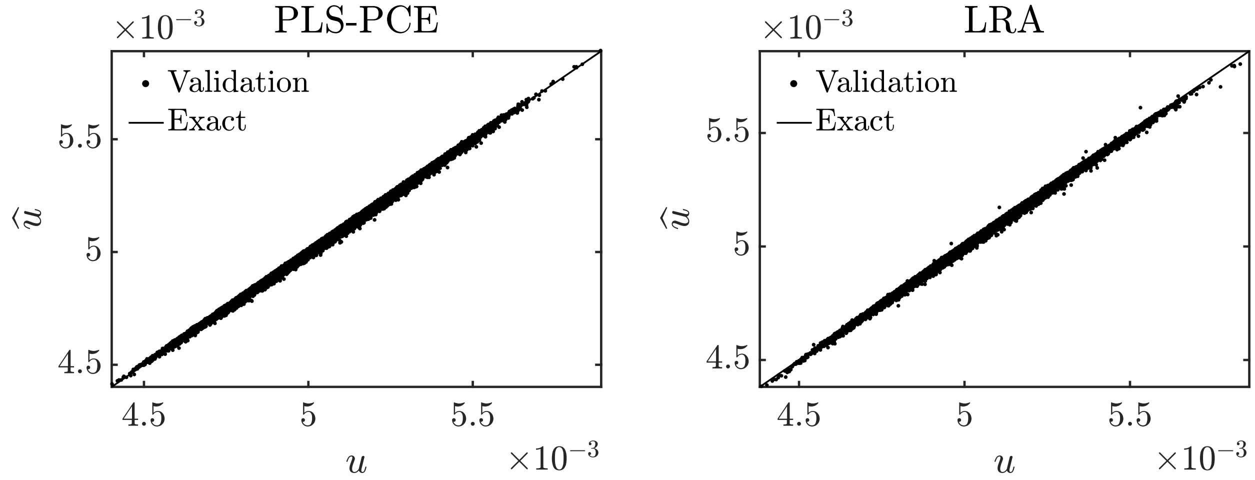

Figure 6: Linear elastic bar: Scatter plots of (nonlinear) PLS-based PCE (left) and LRA (right) response vs original response at $N/n = 2$ using a validation set of 2×10^5 samples.

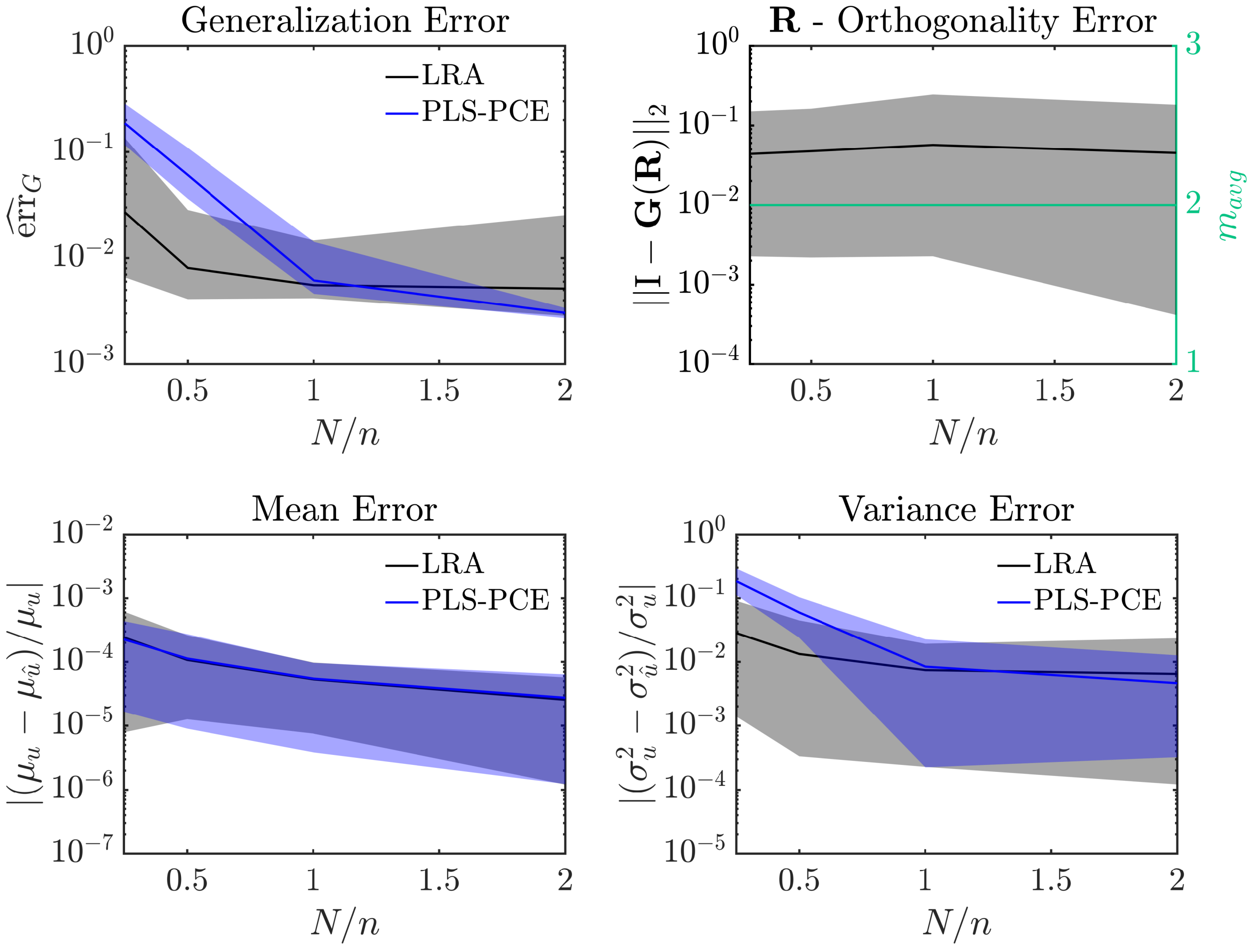

Figure 7: Linear elastic bar: Error measures with 95 % confidence intervals obtained from 100 repeated analyses with different experimental designs; reference solution obtained with 2×10^5 samples. The top right panel includes the average number of latent components (PLS directions) m_{avg}.

5.2. Nonlinear oscillator

The second example, adapted from [52], is a hysteretic oscillator under random loading, defined by the following differential equation:

$$m\ddot{u}(t) + c\dot{u}(t) + k[\alpha u(t) + (1-\alpha)u_y z(t)] = f(t) \ , \tag{31}$$

where $u(t)$, $\dot{u}(t)$ and $\ddot{u}(t)$ denote the displacement, velocity and acceleration of the oscillator. The mass, stiffness and damping of the oscillator are $m = 6 \times 10^4$kg, $k = 5 \times 10^6$N/m, $c = 2m\zeta\sqrt{k/m}$ with $\zeta = 5\%$, and the yielding displacement is $u_y = 0.04$m. The parameter α, which controls the degree of hysteresis is set to $\alpha = 0.1$. The parameter $z(t)$ follows the Bouc-Wen hysteresis law

$$\dot{z}(t) = \frac{1}{u_y}\left[A\dot{u}(t) - \beta|\dot{u}(t)||z(t)|^{\bar{n}-1}z(t) - \gamma\dot{u}(t)|z(t)|^{\bar{n}}\right] \ , \tag{32}$$

with $\beta = \gamma = 0.5$, $A = 1$ and $\bar{n} = 3$. The loading $f(t)$ is a seismic load process modelled by a white noise ground acceleration and discretized in the frequency domain as follows [53]

$$f(t) = -m\sigma\sum_{i=1}^{n/2}[X_i\cos(\omega_i t) + X_{(n/2+i)}\sin(\omega_i t)] \ , \tag{33}$$

where $X_i, i = 1, \ldots, n$, are independent standard Gaussian random variables, $\omega_i = i\Delta\omega$, $\Delta\omega = 30\pi/n$ (the cut-off frequency is $\omega_{\text{cut}} = 15\pi$) and $\sigma = \sqrt{2S\Delta\omega}$, where $S = 0.03\text{m}^2/\text{s}^3$ is the intensity of the white noise. We use $n = 300$ terms in Eq. (33), which leads to an input random vector $\mathbf{X}$ of dimension 300. We are interested in approximating the displacement of the oscillator at $t = 8$s, $u(8\text{s})$.

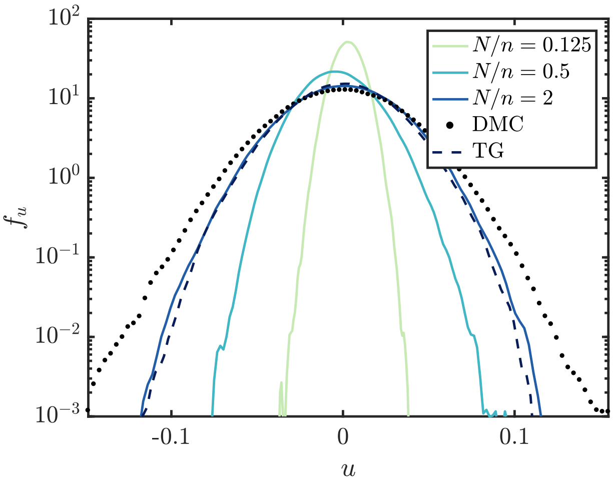

Figure 8: Nonlinear oscillator: Log-densities of response. Comparison of linear PLS-PCE (with $m = 1$) with various design sizes with the linear PCE-based adaptation of Tipireddy and Ghanem (TG) [29] and direct Monte Carlo (DMC) using 2×10^5 samples.

As in example 1, convergence of the linear PLS-based PCE with $m = 1$ to the TG solution is again observed. However, here the TG approach with a single linear PCE-based component cannot capture the behavior of the nonlinear oscillator (Figure 8).

Evidently, the strong nonlinearity present in the hysteretic oscillator model is more challenging to both PLS-based PCE and LRA compared to the bar example. In particular, we were unable to obtain non-diverging surrogate models with the LRA as can be seen from Figure 9. In contrast, PLS-based PCE yields a series of converging surrogate models as N increases. That is, the response PDF is approximated increasingly well and the generalization error decreases monotonously (Figure 11, top left). The polynomial

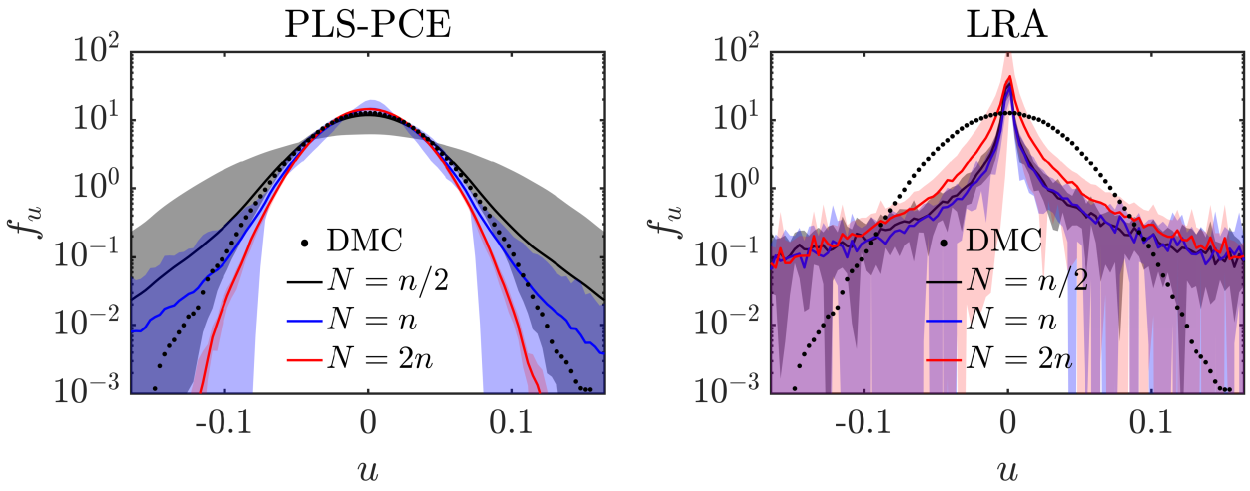

Figure 9: Nonlinear oscillator: Log-densities of response at various design sizes with 95 % CI. Results of (nonlinear) PLS-PCE and LRA are compared with direct Monte Carlo (DMC) results using 2×10^5 samples.

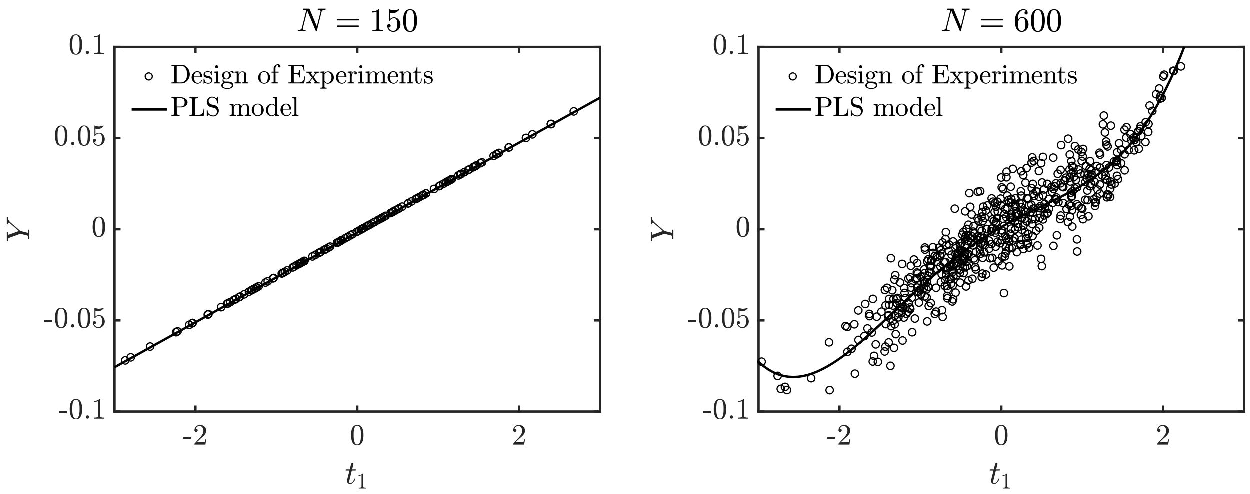

Figure 10: Nonlinear oscillator: 1-D Surrogate model along first PLS direction t_1 vs. the design of experiments at $N/n = 0.5$ (left) and $N/n = 2$ (right).

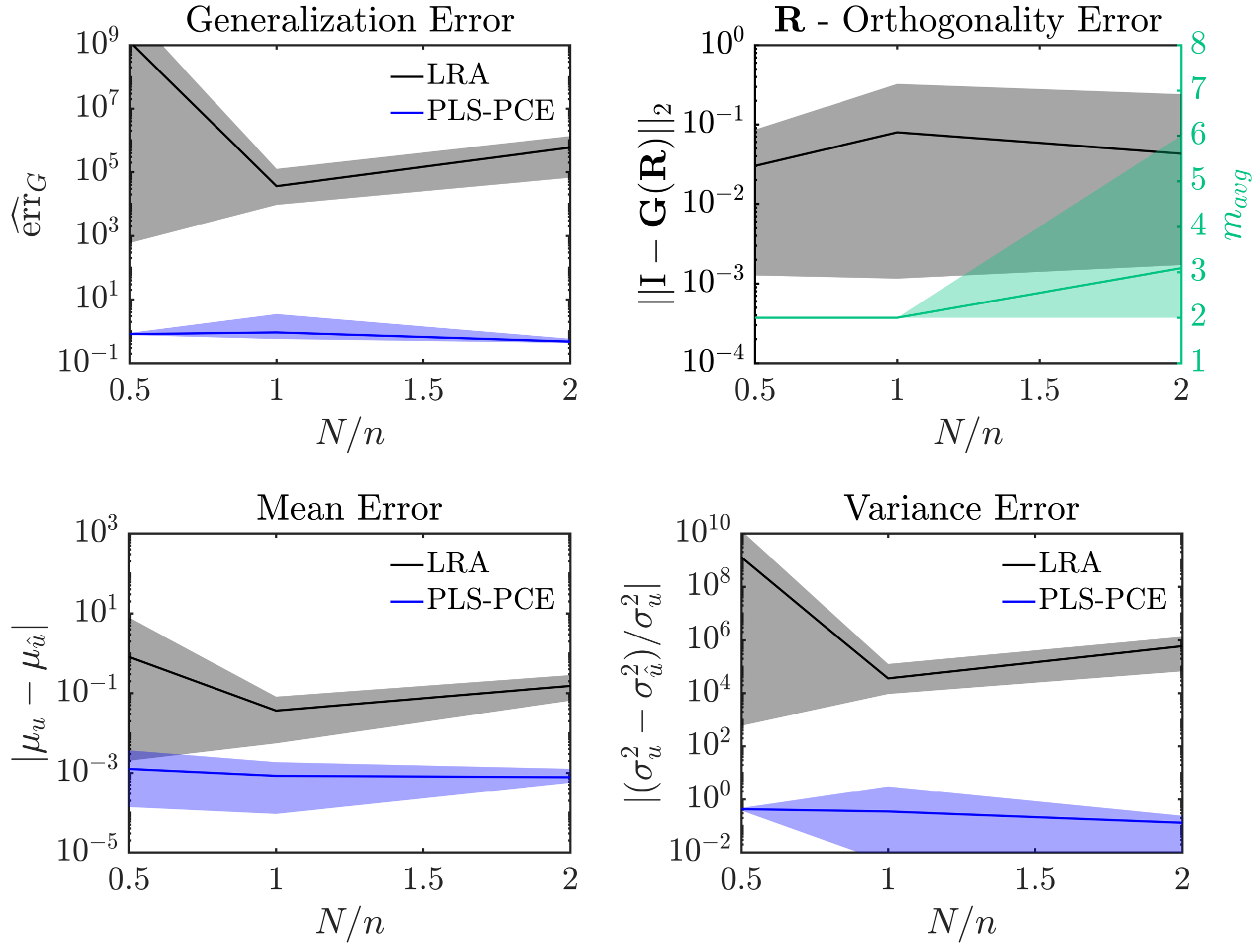

Figure 11: Nonlinear oscillator: Error measures with 95 % confidence intervals obtained from 100 repeated analyses with different experimental designs; reference solution obtained with 2×10^5 samples. The top right panel includes the average number of latent components (PLS directions) m_{avg}.

degree of the one-dimensional PCE identified for the first latent directions increases on average with the size of the experimental design, while the percentage of explained variance by the first latent component decreases (Figure 10). Therefore, the number of PLS components increases with increase of the experimental design size (Figure 11, top right). The response mean is captured well by the PLS-based PCE model even with the smallest investigated N (Figure 11, bottom left). Note, that for this example, Figure 11 depicts the unscaled absolute mean error since $\mu_u = 0$. In this example, depending on the experimental design, between 2–4 reduced space directions are included by the PLS algorithm as $N \geq n$. The orthogonality error increases slightly when $N \geq n$, yet it remains reasonably low for all N. Figure 12 depicts the three Gramian matrices corresponding to $m = \{2, 3, 4\}$. The first two directions which explain most of the output variance enclose angles close to $\pi/2$ in all three cases, whereas combinations of less important directions are responsible for most of the orthogonality error. Thus, using $\mathbf{R}$ to define the transformation does not compromise significantly the optimality of the Hermite polynomial basis.

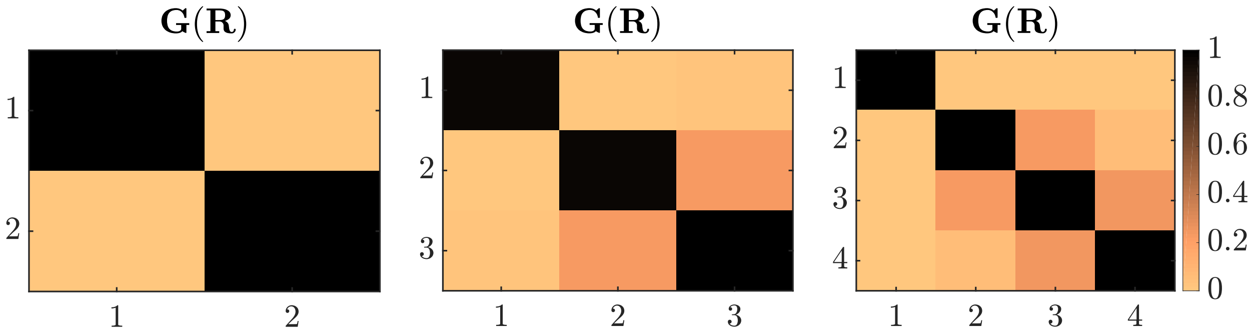

Figure 12: Depiction of Gramian matrices of the nonlinear PLS algorithm resulting from $m = 2$ (left), 3 (center) and 4 (right).

5.3. Steel plate

For a third example, we consider a modified version of the example given in [54], which consists of a low-carbon steel plate of length 0.32 m, width 0.32 m, thickness $t = 0.01$ m, and a hole of radius 0.02 m located at the center. The Poisson ratio is set to $\nu = 0.29$ and the density of the plate is $\rho = 7850$ kg/m^3. The horizontal and vertical displacements are constrained at the left edge. The plate is subjected to a fixed surface load of $q = 96$ MPa, which acts on the right narrow plate side. The Young's modulus $E(x, y)$ is considered uncertain and spatially variable. It is described by a homogeneous random field with lognormal marginal distribution, mean value $\mu_E = 2 \times 10^5$ MPa and standard deviation $\sigma_E = 3 \times 10^4$ MPa. The autocorrelation function of the underlying Gaussian field $\ln E$ is modeled by the isotropic

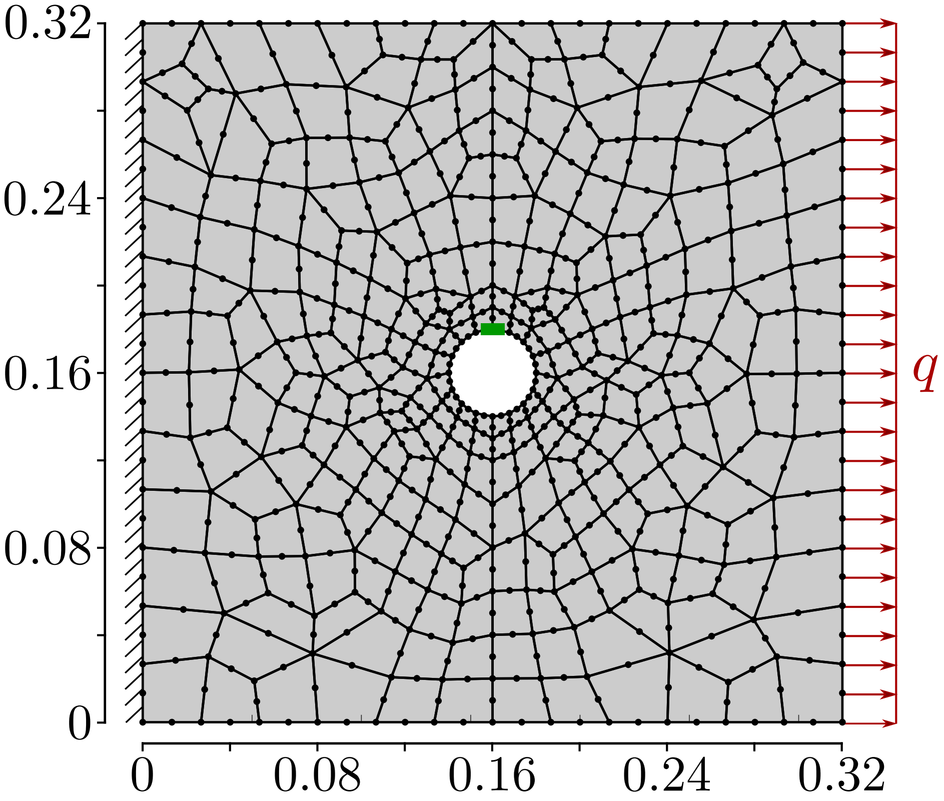

Figure 13: FE-mesh of 2D-plate model. Green marker: Location of maximum first principal stress σ_1.

exponential model, $\rho_{\ln E}(\Delta x, \Delta y) = \exp(-\sqrt{\Delta x^2 + \Delta y^2}/l)$ with correlation length $l = 0.04$m. The random field $\ln D$ is discretized by a KL expansion with $M = 1000$ terms, which yields a global relative variance error of 7%. The stress ($\boldsymbol{\sigma}(x,y) = [\sigma_x(x,y), \sigma_y(x,y), \tau_{xy}(x,y)]^T$), strain ($\boldsymbol{\epsilon}(x,y) = [\epsilon_x(x,y), \epsilon_y(x,y), \gamma_{xy}(x,y)]^T$) and displacement ($\mathbf{u}(x,y) = [u_x(x,y), u_y(x,y)]^T$) fields of the plate are given through elisticity theory, namely the Cauchy-Navier equations [55]. Given the configuration of the plate, the model can be simplified under the plane stress hypothesis, which yields

$$G(x,y)\nabla^2\mathbf{u}(x,y) + \frac{E(x,y)}{2(1-\nu)}\nabla(\nabla\cdot\mathbf{u}(x,y)) + \mathbf{b} = 0 \ . \tag{34}$$

Therein, $G(x,y) := E(x,y)/(2(1+\nu))$ is the shear modulus, and $\mathbf{b} = [b_x, b_y]^T$ is the vector of body forces acting on the plate. Eq. (34) is discretized with a finite-element method. That is, the spatial domain of the plate is discretized into 282 eight-noded quadrilateral elements, as shown in Figure 13. The scalar model output is the first principal plane stress

$$\sigma_1 = 0.5(\sigma_x + \sigma_y) + \sqrt{[0.5(\sigma_x + \sigma_y)]^2 + \tau_{xy}^2}$$

at node 11 (see green marker Figure 13), which is where maximum plane stresses occur typically in this example.

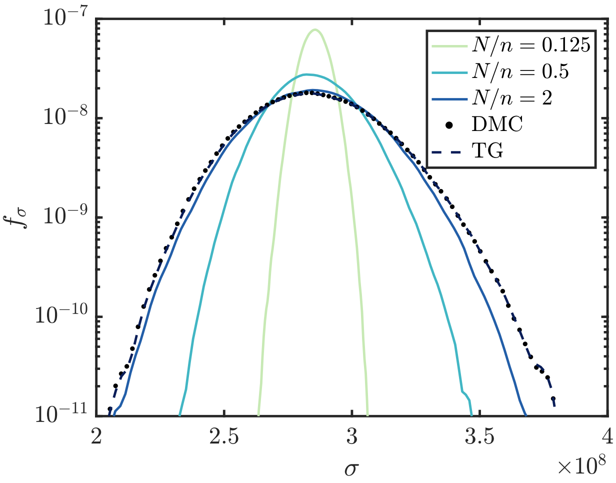

Figure 14: Steel plate: Log-densities of response. Comparison of linear PLS-PCE (with $m = 1$) with various design sizes with the linear PCE-based adaptation of Tipireddy and Ghanem (TG) [29] and direct Monte Carlo (DMC) using 2×10^5 samples.

Figure 14 shows that the TG method with linear PCE-based adaptation performs well for this example. Again, the linear PLS-based PCE with $m = 1$ approaches the TG solution with increase of N.

Comparing the linear and nonlinear PLS-PCE methods, we see again that the nonlinear PCE-driven PLS algorithm gives consistently lower generalization errors with fewer number of latent components than the linear PLS (Figure 15).

Comparing the performance of the nonlinear PLS-PCE and LRA surrogates, we see that both methods are capable of producing a converging approximation of the numerical model as the number of points in the experimental design increases (Figures 16 & 17). Figure 17 indicates that the PLS-based PCE model represents the model response more accurately in the tails. Moreover, the variability associated with the random choice of the experimental design is very small across all N. Similar to the bar example, all PLS-based PCE surrogates are constructed with $m = 2$ with two quasi-orthogonal directions in the reduced space (Figure 18, top right). The PLS-based PCE mean and generalization errors become smaller than the corresponding LRA errors for $N/n = 2$ (Figure 18, bottom & top left).

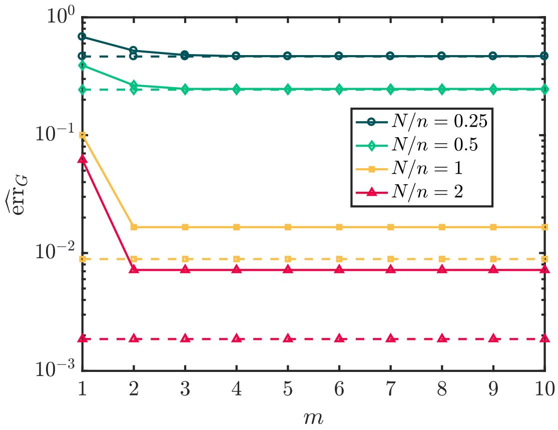

Figure 15: Steel plate: Comparison of the linear PLS-based (solid line) and nonlinear (PCE-driven) PLS-based (dashed line) adaptation at various design sizes and number of latent components.

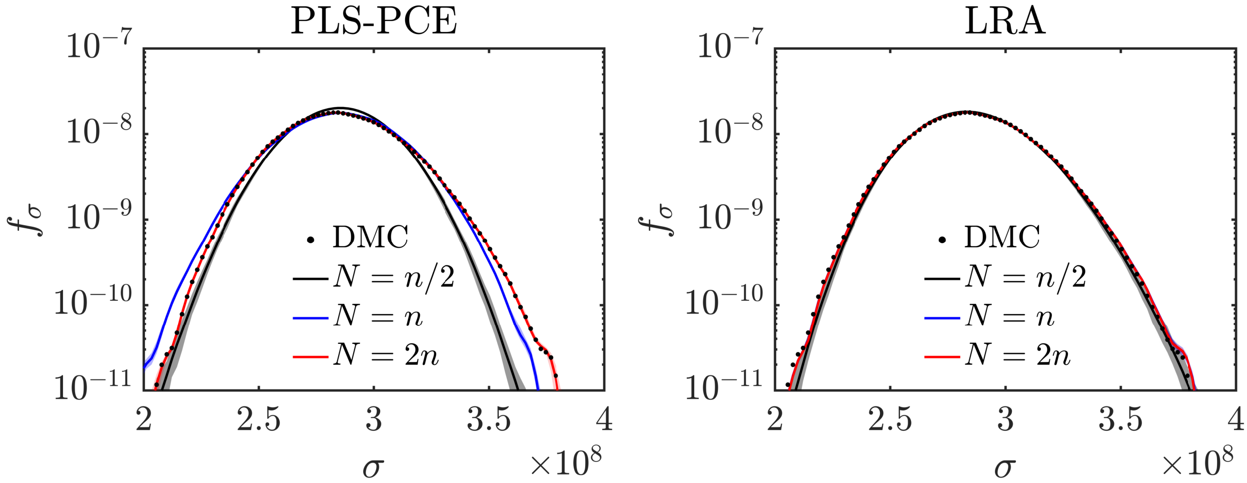

Figure 16: Steel plate: Log-densities of response at various design sizes with 95 % CI. Results of (nonlinear) PLS-PCE and LRA are compared with direct Monte Carlo (DMC) results using 2×10^5 samples.

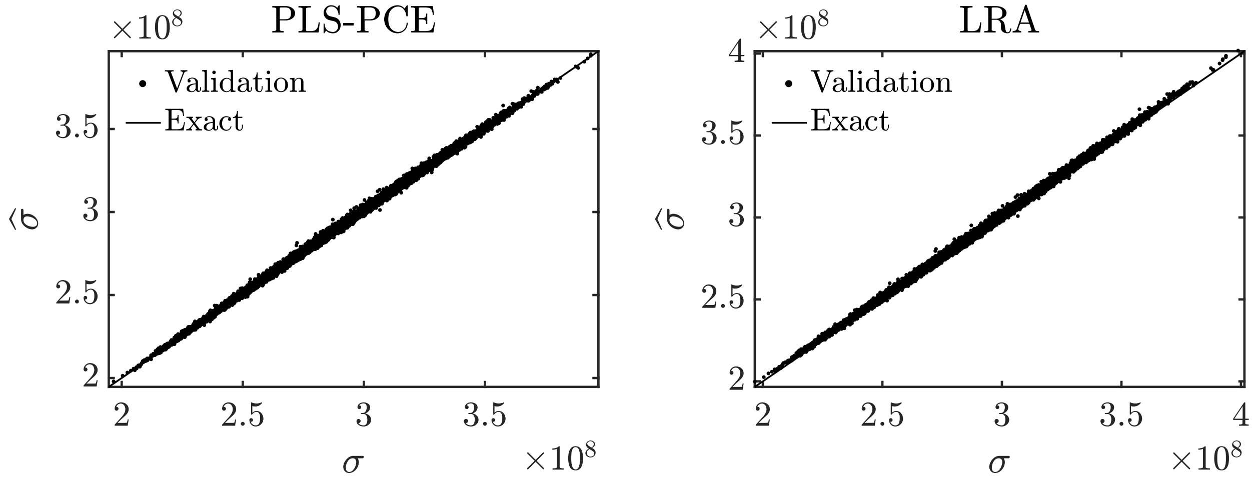

Figure 17: Steel plate: Scatter plots of PLS-based PCE (left) and LRA (right) response vs original response at $N/n = 2$ using a validation set of 2×10^5 samples.

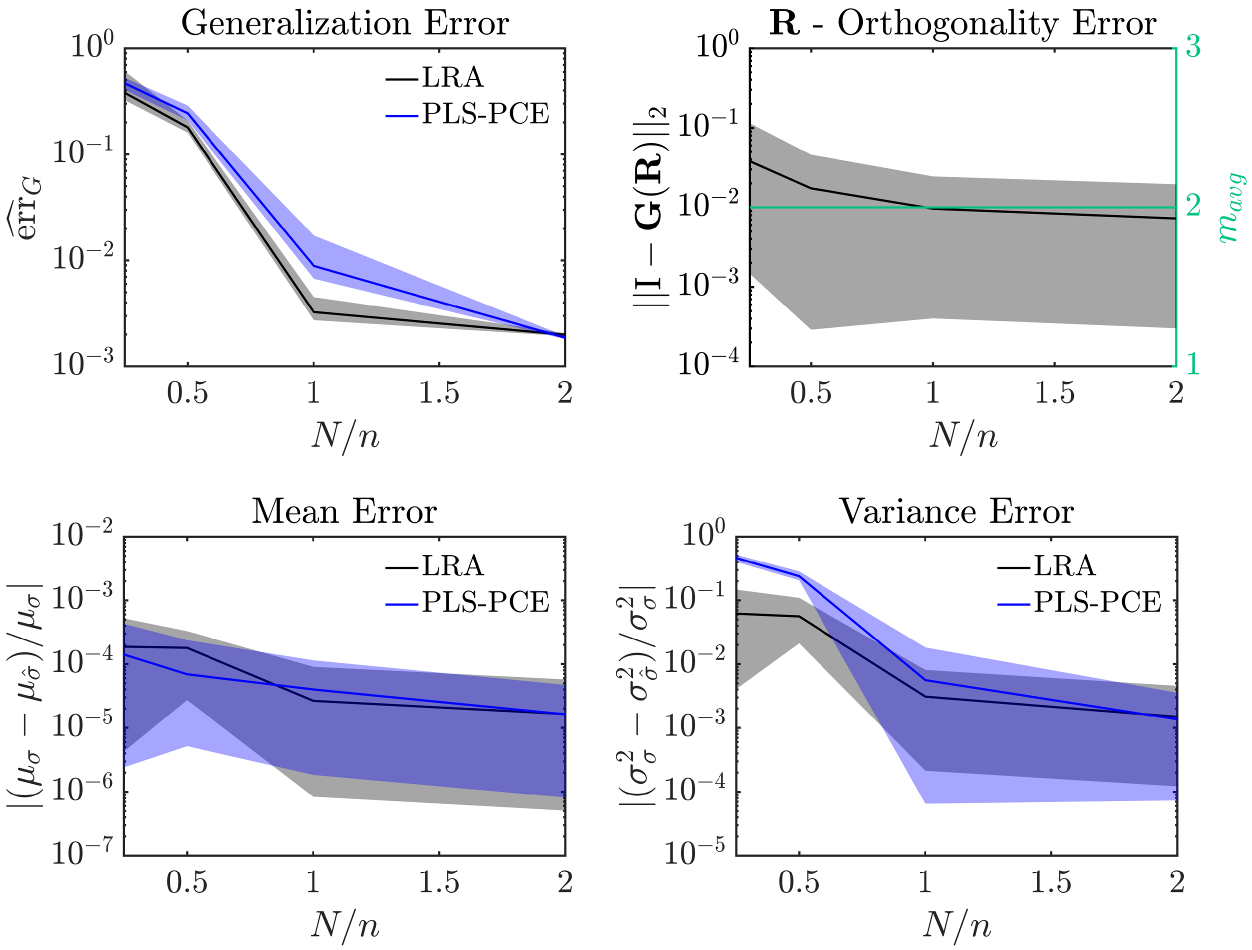

Figure 18: Steel plate: Error measures with 95 % confidence intervals obtained from 100 repeated analyses with different experimental designs; reference solution obtained with 2×10^5 samples. The top right panel includes the average number of latent components (PLS directions) m_{avg}.

6. Concluding remarks

This paper presented a novel sparse polynomial chaos expansion (PCE) representation based on a transformation of the coordinate system in Gaussian space using a number of dominant directions. These directions are identified based on partial least squares (PLS) analysis using a set of experimental points.. Two PLS algorithms were investigated; the standard linear PLS algorithm and a novel PCE-driven nonliear PLS algorithm. It was shown that the linear PLS with a single latent component is asymptotically (as $N \to \infty$) equivalent to the linear PCE-based adaptation of [29]. The proposed PCE-driven PLS algorithm is able to simultaneously determine the dominant directions in input space and the PCE coefficients in the transformed space. Three numerical examples demonstrated the ability of the method to provide accurate estimates of the moments and PDF of quantities of interest in problems with high-dimensional input spaces, provided that the behavior of the model is governed by a small number of latent variables. The method presented comparable performance to the low-rank tensor approximation (LRA) surrogate in moderately nonlinear problems for experimental design sizes in the order of the input dimension. In addition, the proposed method was able to provide an accurate representation of the response of a highly nonlinear oscillator, whereas the LRA model was unable to obtain converging results for the same example.

6.1. Discussion on computational complexity

The linear PLS algorithm (Algorithm 1) has complexity approximately $\mathcal{O}(m\ n\ N)$, with m being the number of latent components. Solution of the OLS problem in the reduced space has complexity $\mathcal{O}(P_m^2\ N)$, with P_m denoting the number of terms in the total degree expansion in reduced space. The complexity of the nonlinear PCE-driven PLS algorithm is dominated by the Newton-Raphson iteration (step 13 in Algorithm 2), which has approximate cost $\mathcal{O}(n^2\ N)$ (for $N > n$). This step is repeated at most l times, with l denoting the maximum number of Newton-Raphson iterations. Since the nonlinear PLS process is repeated for each candidate polynomial degree and each latent component, the computational complexity of the algorithm is approximately $\mathcal{O}(m\ p\ l\ n^2\ N)$, with p being the maximum polynomial degree considered. Although the cost of Algorithm 2 is often much higher than the one of Algorithm 1, it remains polynomial in the dimension n. It is noted that the computational complexity of ordinary or regularized regression-based PCE in high dimensions is governed by the complexity of the algorithm used to order the multi-indices of the orthogonal polynomials, which is asymptotically $2^{\mathcal{O}(n)}$. This does not pose a problem to PLS-PCE methods as typically $m << n$. Hence, the proposed algorithms are orders of magnitude more efficient than sparse PCEs based on compressive sensing as $n \to \infty$. It is also worth mentioning that the two PLS-PCE methods have low memory requirements, as opposed to regularized regression algorithms that require storage of the full Vandermonde matrix of dimensions $N \times P$, with P increasing factorially with n for a total degree construction.

6.2. Outlook

The size of the experimental design for obtaining accurate predictions depends on the nonlinearity of the problem and on its effective dimension, i.e. the number of latent variables that have significant effect on the behavior of the output quantity. One possible future improvement of the method is to estimate the size of the experimental design adaptively, for example based on a cross validation error criterion. In the present implementation, the experimental design is generated with Latin hypercube sampling with sample decorrelation, to minimize the error in the orthogonality of the polynomials in transformed space. The performance of alternative sampling schemes will be investigated in future studies. Additionally, the method can potentially be extended to the case of multivariate output quantities. This could be enabled by application of the multivariate version of the PLS method, the PLS2 algorithm.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) in the framework of the priority program SPP 1886 by grant STR 1140/6-1.

References

[1] R. Ghanem, P. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, Berlin, 1991.

[2] D. Xiu, G. E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM Journal on Scientific Computing 24 (2002) 619–644.

[3] D. Xiu, G. E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, Journal of Computational Physics 187 (2003) 137–167.

[4] H. G. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, Computer Methods in Applied Mechanics and Engineering 194 (2005) 1295–1331.

[5] M. A. Tatang, W. Pan, R. G. Prinn, G. J. McRae, An efficient method for parametric uncertainty analysis of numerical geophysical models, Journal of Geophysical Research: Atmospheres 102 (1997) 21925–21932.

[6] M. Berveiller, B. Sudret, M. Lemaire, Stochastic finite element: A non intrusive approach by regression, European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 15 (2006) 81–92.

[7] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal on Numerical Analysis 45 (2007) 1005–1034.

[8] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys 2 (2007) 293–309.

[9] D. Xiu, J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM Journal on Scientific Computing 27 (2005) 1118–1139.

[10] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal on Numerical Analysis 46 (2008) 2309–2345.

[11] G. Blatman, B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression, Journal of Computational Physics 230 (2011) 2345–2367.

[12] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, Journal of Computational Physics 230 (2011) 3015–3034.

[13] L. Yan, L. Guo, D. Xiu, Stochastic collocation algorithms using l_1-minimization, International Journal for Uncertainty Quantification 2 (2012).

[14] K. Sargsyan, C. Safta, H. N. Najm, B. J. Debusschere, D. Ricciuto, P. Thornton, Dimensionality reduction for complex models via Bayesian compressive sensing, International Journal for Uncertainty Quantification 4 (2014).

[15] J. D. Jakeman, M. S. Eldred, K. Sargsyan, Enhancing l_1-minimization estimates of polynomial chaos expansions using basis selection, Journal of Computational Physics 289 (2015) 18–34.

[16] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2010) 183–197.

[17] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliability Engineering & System Safety 93 (2008) 964–979.

[18] T. Zhou, A. Narayan, D. Xiu, Weighted discrete least-squares polynomial approximation using randomized quadratures, Journal of Computational Physics 298 (2015) 787–800.

[19] J. Hampton, A. Doostan, Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression, Computer Methods in Applied Mechanics and Engineering 290 (2015) 73–97.

[20] A. Narayan, J. Jakeman, T. Zhou, A christoffel function weighted least squares algorithm for collocation approximations, Mathematics of Computation 86 (2017) 1913–1947.

[21] A. Cohen, G. Migliorati, Optimal weighted least-squares methods, SMAI Journal of Computational Mathematics 3 (2017) 181–203.

[22] J. Hampton, A. Doostan, Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies, Journal of Computational Physics 280 (2015) 363–386.

[23] J. D. Jakeman, A. Narayan, T. Zhou, A generalized sampling and preconditioning scheme for sparse approximation of polynomial chaos expansions, SIAM Journal on Scientific Computing 39 (2017) A1114–A1144.

[24] A. Doostan, G. Iaccarino, A least-squares approximation of partial differential equations with high-dimensional random inputs, Journal of Computational Physics 228 (2009) 4332–4345.

[25] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems, Archives of Computational Methods in Engineering 17 (2010) 403–434.

[26] A. Doostan, A. Validi, G. Iaccarino, Non-intrusive low-rank separated approximation of high-dimensional stochastic models, Computer Methods in Applied Mechanics and Engineering 263 (2013) 42–55.

[27] M. Chevreuil, R. Lebrun, A. Nouy, P. Rai, A least-squares method for sparse low rank approximation of multivariate functions, SIAM/ASA Journal on Uncertainty Quantification 3 (2015) 897–921.

[28] K. Konakli, B. Sudret, Reliability analysis of high-dimensional models using low-rank tensor approximations, Probabilistic Engineering Mechanics 46 (2016) 18–36.

[29] R. Tipireddy, R. Ghanem, Basis adaptation in homogeneous chaos spaces, Journal of Computational Physics 259 (2014) 304–317.

[30] P. Tsilifis, X. Huan, C. Safta, K. Sargsyan, G. Lacaze, J. C. Oefelein, H. N. Najm, R. G. Ghanem, Compressive sensing adaptation for polynomial chaos expansions, 2018.

[31] P. Constantine, E. Dow, Q. Wang, Active subspace methods in theory and practice: Applications to kriging surfaces, SIAM Journal on Scientific Computing 36 (2014) A1500–A1524.

[32] P. Constantine, M. Emory, J. Larsson, G. Iaccarino, Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet, Journal of Computational Physics 302 (2015) 1–20.

[33] P. A. Tsilifis, Gradient-informed basis adaptation for Legendre chaos expansions, Journal of Verification, Validation and Uncertainty Quantification 3 (2018) 011005.

[34] S. Wold, A. Ruhe, H. Wold, W. Dunn, III, The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses, SIAM Journal on Scientific and Statistical Computing 5 (1984) 735–743.

[35] A. Höskuldsson, PLS regression methods, Journal of Chemometrics 2 (1988) 211–228.

[36] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometrics and Intelligent Laboratory Systems 58 (2001) 109–130.

[37] S. Wold, N. Kettaneh-Wold, B. Skagerberg, Nonlinear PLS modeling, Chemometrics and Intelligent Laboratory Systems 7 (1989) 53–65.

[38] G. Baffi, E. B. Martin, A. Morris, Non-linear projection to latent structures revisited: the quadratic PLS algorithm, Computers & Chemical Engineering 23 (1999) 395–411.

[39] M. Rosenblatt, Remarks on a multivariate transformation, The Annals of Mathematical Statistics 23 (1952) 470–472.

[40] P. C. Austin, E. W. Steyerberg, The number of subjects per variable required in linear regression analyses, Journal of Clinical Epidemiology 68 (2015) 627–636.

[41] M. Bardet, J.-C. Faugère, B. Salvy, On the complexity of the F5 Gröbner basis algorithm, Journal of Symbolic Computation 70 (2015) 49–70.

[42] I. S. Helland, On the structure of partial least squares regression, Communications in Statistics-Simulation and Computation 17 (1988) 581–607.

[43] A. Phatak, S. De Jong, The geometry of partial least squares, Journal of Chemometrics: A Journal of the Chemometrics Society 11 (1997) 311–338.

[44] X.-Q. Zeng, G.-Z. Li, Incremental partial least squares analysis of big streaming data, Pattern Recognition 47 (2014) 3726–3735.

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

[46] A. B. Owen, Controlling correlations in Latin hypercube samples, Journal of the American Statistical Association 89 (1994) 1517–1522.

[47] R. Rosipal, Nonlinear partial least squares an overview, in: Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques, IGI Global, 2011, pp. 169–189.

[48] G. Baffi, E. Martin, A. Morris, Non-linear projection to latent structures revisited (the neural network PLS algorithm), Computers & Chemical Engineering 23 (1999) 1293–1307.

[49] O. Chapelle, V. Vapnik, Y. Bengio, Model selection for small sample regression, Machine Learning 48 (2002) 9–23.

[50] M. Abramowitz, I. A. Stegun, Handbook of mathematical function: with formulas, graphs and mathematical tables, Dover Publications, 1965.

[51] S. Marelli, B. Sudret, UQLab: A framework for uncertainty quantification in Matlab, in: Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), pp. 2554–2563.

[52] Z. Wang, J. Song, Cross-entropy-based adaptive importance sampling using von Mises-Fisher mixture for high dimensional reliability analysis, Structural Safety 59 (2016) 42–52.

[53] M. Shinozuka, G. Deodatis, Simulation of stochastic processes by spectral representation., Applied Mechanics Reviews 44 (1991) 191–204.

[54] P.-L. Liu, K.-G. Liu, Selection of random field mesh in finite element reliability analysis, Journal of Engineering Mechanics 119 (1993) 667–680.

[55] C. Johnson, Numerical solution of partial differential equations by the finite element method, Dover Publications, 2009.