Projection pursuit adaptation on polynomial chaos expansions

Xiaoshu Zeng[a,*], Roger Ghanem[a]

[a]*University of Southern California, 210 KAP Hall, Los Angeles, CA90089, United States*

Abstract

The present work addresses the issue of accurate stochastic approximations in high-dimensional parametric space using tools from uncertainty quantification (UQ). The basis adaptation method and its accelerated algorithm in polynomial chaos expansions (PCE) were recently proposed to construct low-dimensional approximations adapted to specific quantities of interest (QoI). The present paper addresses one difficulty with these adaptations, namely their reliance on quadrature point sampling, which limits the reusability of potentially expensive samples. Projection pursuit (PP) is a statistical tool to find the "interesting" projections in high-dimensional data and thus bypass the curse-of-dimensionality. In the present work, we combine the fundamental ideas of basis adaptation and projection pursuit regression (PPR) to propose a novel method to simultaneously learn the optimal low-dimensional spaces and PCE representation from given data. While this projection pursuit adaptation (PPA) can be entirely data-driven, the constructed approximation exhibits mean-square convergence to the solution of an underlying governing equation and is thus subject to the same physics constraints. The proposed approach is demonstrated on a borehole problem and a structural dynamics problem, demonstrating the versatility of the method and its ability to discover low-dimensional manifolds with high accuracy with limited data. In addition, the method can learn surrogate models for different quantities of interest while reusing the same data set.

Keywords: Surrogate modeling, polynomial chaos expansion, projection pursuit, high-dimensional models, dimension reduction, data-driven

1. Introduction

Recent developments in sensing have enabled highly resolved characterization of physical systems, justifying the drive for computational models of increasingly higher resolution and constantly pushing computational resources to the limit of hardware capacity. This computational bottleneck challenge is further exacerbated when uncertainty is acknowledged, which, at least conceptually, necessitates the simultaneous contemplation of a number of plausible realities. Allowing uncertainty to infiltrate computational models, however, can serve, with the proper perspective, to alleviate the computational burden, allowing consistency between numerical accuracy, statistical confidence, and experimental paucity. Response prediction can thus be achieved with the help of statistical surrogate models, which we achieve in the present paper by augmenting machine learning (ML) paradigms with an uncertainty quantification (UQ) dimension. Typically, these surrogate models are trained on replicas obtained by evaluating the physical model over a set of input parameters. Each of these evaluations will typically involve an expensive high-fidelity model, which is typically described by a large number of parameters. This combination of high resolution and high parameterization continues to challenge the accuracy of surrogate modeling. Similarly, in a big data set, the data of various control variables are generated such that comprehensive information on the quantity of interest (QoI) is considered in the predictive model. Usually, the number of data required for prediction with good accuracy increases with the number of parameters. Surrogate modeling in high-dimensional parameter space is thus ubiquitous, remaining at the forefront of predictive science.

UQ is a tool to characterize the uncertainty in specific responses or outputs, given that the input parameters of the system are uncertain [1, 2, 3, 4, 5, 6, 7]. Generally, the uncertainties in the input parameters are represented by random variables that are characterized by their probability distributions. Usually, the supports of the parameters are first provided. This information is sometimes complemented by marginal distributions for each parameter. Only on rare occasions is joint information on the input parameters available at the outset. Such information may be weaved into the statistical model following data-driven updating. In all cases, the output is computed over a set of input parameters that, in principle, spans the support of their joint probability measure in parameter space. In a standard UQ setting, the probabilistic model of the output can be quantified by propagating the uncertainties from the input space through the computational model. Surrogate models, on the other hand, aim to construct a mapping from input parameters to output variables. The ensuing representation can be used as a surrogate to efficiently map the support of the input parameters into the support and distribution of the quantities of interest. In the literature, it is not uncommon

---

*Corresponding author

Email addresses: `xiaoshuz@usc.edu` (Xiaoshu Zeng), `ghanem@usc.edu` (Roger Ghanem)

to use UQ models as an efficient surrogate for further analyses that require repeated evaluation of the computational model, for example, in [8, 9, 10, 11]. The quality of these surrogates is clearly paramount. Polynomial chaos representations [1] can be shown to be faithful mathematical representations that converge in the mean-square sense to the true solution of the underlying governing stochastic equations.

For high-dimensional UQ problems, the standard Monte Carlo (MC) sampling has the attractive feature that its convergence is independent of the number of random parameters, thus can avoid the curse-of-dimensionality. However, it has slow convergence in general, and the rate is dependent on the complexity of the solution being sought. Thus the development of UQ methods such as polynomial chaos expansion (PCE) [1, 12], generalized polynomial chaos expansion (gPCE) [13, 14], Gaussian processes [15, 16, 17], probability density evolution methods [18, 19, 7, 6], manifold methods [20, 21, 22], Wiener path integral technique[23, 24], and others. PCE expands the QoI on a set of multivariate Hermite polynomial bases where the approximation has a mean-squared sense convergence with respect to the maximum order or degree of the polynomial bases. Its solid mathematical foundation, accuracy, simplicity in application, and convergence property gained much attention from science and engineering applications [1, 12, 2, 4]. However, the number of PCE terms in the expansion grows factorially with the parameters' dimension and the maximum polynomial order, leading to the curse-of-dimensionality. In a more common non-intrusive implementation, the PCE coefficients associated with each term are computed as a multi-dimensional integral, where the dimension is equal to the parameters' dimension. One can typically use numerical quadrature [25, 26] to avoid excessive computation arising from the model evaluation to compute the PCE coefficients. Sparse quadrature [27, 28, 29, 26] can reduce the cost without sacrificing accuracy. Another family of methods, for example, least angle regression [30, 31], and compressive sensing [32, 33, 34], propose a sparse representation by expanding QoI on a selection of significant polynomial bases. These methods attenuate the computational burden but still face challenges if the dimension is excessively large or the computational model is expensive. Active subspace [35, 36] is a dimension reduction technique that finds the important directions with the greatest variability by an eigendecomposition of the covariance matrix of the gradient. In practice, the gradient of the QoI usually takes much work to obtain. The basis adaptation method [37] in the framework of PCE is another dimension reduction technique that finds directions adapted to low-dimensional or scalar QoI. The idea is based on rotating the Gaussian input variables with a proper rotation matrix so that the QoI can be represented in a low-dimensional space. The construction of the rotation matrix requires the first-order PCE coefficients, which can be obtained by a first-order pilot PCE. Basis adaptation has been applied to practical engineering problems [38, 39] that reduce the dimension to only a few. This method can have slow convergence, resulting in relatively high reduced dimensions. Thus, the authors developed the accelerated basis adaptation method [40]. The basis adaptation method requires the construction of low-dimensional adapted PCEs, where the PCE coefficients are usually computed via sparse quadrature to reduce computational cost. In addition, the method is adapted to scalar or low-dimensional QoIs.

In ML, dimension reduction techniques, for example, feature selection [41, 42], matrix factorization [43, 44], manifold learning [45, 46], and autoencoder methods [47, 48], have been able to increase the performance when the dimension of the data is high. Among these, the principal component analysis (PCA) is arguably the most widely used method, useful for discovering components that have ranked importance and can capture the data's salient structure. Projection pursuit (PP) [49, 50, 51] is another dimension reduction technique used to find the most "interesting" projections in high-dimensional data that can overcome the curse-of-dimensionality. The low-dimensional interesting projections are found by maximizing a so-called projection index stage-wisely. Different definitions of projection index lead to specific methods, for example, PCA and discriminant analysis. PP has received attention in ML in the recent two decades. For example, in [52], new projection indices were proposed to provide low-dimensional projections for efficient supervised classification. For the classification of complex data, a projection pursuit constructive neural network was proposed to discover the simplest models in [53]. Based on the Jensen-Shannon divergence, a supervised projection pursuit was proposed in [54] for better classification and visualization and proved superior to PCA. In [55], PP was used to develop a novel recurrent neural network for discriminant analysis. These works all belong to the scope of the classification. The combination of statistics and ML methods to improve the prediction performance is not unusual, for example, the Bayesian neural networks [56, 57, 58]. One of the extended ideas of PP is the projection pursuit regression (PPR) [59]. In PPR, the response surface is approximated by a sum of univariate smooth functions of linear combinations, represented by projections of the input parameters (or predictors). The projections in PPR are found in a forward stage-wise strategy until new projections cannot improve the approximation significantly. Since the number of projections can be taken arbitrarily large, with a proper choice of smooth functions, the PPR model is able to approximate any continuous function in the space spanned by the parameters [60]. Such a PPR model is thus a universal approximator. PPR has been applied to many practical applications, for example, in [61, 62, 63, 64].

The basis adaptation in PCE and the PPR have many shared features. They find the important or interesting directions in the input parameter space adapted to the output. The rotation matrix in adaptation corresponds to the projections in PPR. In contrast, the PCE of the adapted variables in adaptation corresponds to the smooth functions of the projected variables. However, these two methods are different in several aspects. The rotation matrix in adaptation is constructed based on a first-order PCE, and the directions are mutually orthogonal. At the same time, the projections in PPR are found one at a time, and no restrictions are applied among the projections. In addition, the functional links input parameters to output response in adaptation are Hermite polynomial bases, while in PPR, one must find proper smooth functions suited for the application. Also, the PCE coefficients

in adaptation are usually found by sparse numerical quadrature (requiring the evaluation of the model on specific parameters) to reduce the computational cost. In contrast, the projections and smooth functions in PPR are discovered based on given data. Then, the advantages of these two methods can be summarized as follows: (1) the adaptation method uses Hermite polynomial bases that can represent any variable in the parameters space, and it has mean-squared convergence to the maximum polynomial order; (2) the PPR model can find projections and smooth functions simultaneously in a data-driven manner without querying the model.

In this paper, we integrate the basis adaptation method in PCE and the PPR model to take advantage of both methods. Specifically, we use PCEs as smooth functions in the PPR model. In a naive integration, the PCEs work as univariate functions on each projected variable. To take advantage of the completeness and convergence properties of the multivariate Hermite polynomial bases, we will maintain the orthogonality of the projections and use a multivariate PCE on all the projected variables in the approximation. Switching from an additive of many univariate models to a multivariate model, in general, will be less cost-efficient. However, the purpose is to find a low-dimensional representation of the output; thus, the number of projections considered is supposed to be small. In such circumstances, seeking a more accurate multivariate representation is reasonable. We refer to the integrated multivariate model as the projection pursuit adaptation (PPA) model. Note that the projections construct the rotation matrix in the PPA model; thus, the multivariate PCE on the projected variables can be transformed back to the original physical space for interpretation purposes. The proposed model will be compared with the basis adaptation and the PPR methods. The paper is organized as follows: Section 2 introduces the basis adaptation method, including the accelerated algorithm, in the framework of PCE; Section 3 first introduces the detailed procedure of the PPR, then derives the integrated PPA model with a detailed procedure also given; the applications on a borehole model and a space structure are shown in Section 4; conclusions are finally presented in Section 5.

2. Basis adaptation on polynomial chaos expansions

2.1. Polynomial chaos expansions

Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with Ω the sample space, $\mathcal{F}$ the σ-algebra on Ω, and $\mathbb{P}$ the probability measure. Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d) \in \mathbb{R}^d$ be a set of independent standard Gaussian variables on Ω. Then, the closed linear span (this allows for the case where $d \to \infty$) of $\boldsymbol{\xi}$ defines a Gaussian Hilbert space, denoted as $\mathcal{H}$. Denoting the space of second order random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ by $L^2(\Omega)$, an inner product on $L^2(\Omega)$ can be introduced as

$$\langle u, v \rangle_{\mathcal{H}} = \mathbb{E}[uv], \quad \forall u, v \in L^2(\Omega), \tag{1}$$

where $\mathbb{E}[\cdot]$ is the mathematical expectation with respect to the probability measure $\mathbb{P}$. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathcal{J}_d := (\mathbb{N}_0)^d$ be a d-dimensional multi-index with its length defined as $|\boldsymbol{\alpha}| = \sum_{i=1}^d \alpha_i$, $h_{\boldsymbol{\alpha}}$ be multivariate Hermite polynomials, and $\psi_{\boldsymbol{\alpha}} = h_{\boldsymbol{\alpha}}/\|h_{\boldsymbol{\alpha}}\|_2$ be the normalized Hermite polynomials, then $\{\psi_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{J}_d\}$ is a complete orthonormal basis on $L^2(\Omega)$. We also let $\mathcal{F}(\mathcal{H})$ denote the σ-algebra generated by $\mathcal{H}$. Then we can represent any random variable $Y \in L^2(\mathcal{F}(\mathcal{H}))$ as the following polynomial chaos expansion (PCE) [1]

$$Y(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{J}_d} Y_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{2}$$

In Eq (2), $Y_{\boldsymbol{\alpha}}$ is the PCE coefficient associated with the basis $\psi_{\boldsymbol{\alpha}}$.

In a computational context, a truncated PCE is usually employed where the maximum order of the polynomials is specified. Suppose the maximum order of the PCE is p, then the order p PCE of Y can be written as,

$$\widehat{Y}_p(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{J}_{d,p}} Y_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{3}$$

where $\mathcal{J}_{d,p} \subset \mathcal{J}_d$ is the subset of indices of maximum length p (i.e., $|\boldsymbol{\alpha}| \leq p$). The above expression, $\widehat{Y}_p(\boldsymbol{\xi})$ converges to Y in the mean-squared sense [65] as $p \to \infty$. With abuse of notation, we will henceforth use $Y(\boldsymbol{\xi})$ to denote $\widehat{Y}_p(\boldsymbol{\xi})$. The total number of PCE terms in Eq (3) is

$$P = \binom{d+p}{p} = \frac{(d+p)!}{d!p!}, \tag{4}$$

which grows factorially with the dimension of the parameter space d and the maximum order p. In a non-intrusive implementation of the PCE [1, 25, 26], the PCE coefficients $Y_{\boldsymbol{\alpha}}$ are computed by projecting $Y(\boldsymbol{\xi})$ on the space spanned by the Hermite polynomial basis $\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \boldsymbol{\alpha} \in \mathcal{J}_{d,p}\}$ (referred to as the original space) as

$$Y_{\boldsymbol{\alpha}} = \langle Y, \psi_{\boldsymbol{\alpha}} \rangle, \quad \boldsymbol{\alpha} \in \mathcal{J}_{d,p}. \tag{5}$$

3

The projection can be calculated by, for example, the sampling-based Monte Carlo Simulation (MCS). In this setting, a set of samples $(\Xi, \mathcal{Y}) = \left\{ \left(\boldsymbol{\xi}^{(i)}, Y^{(i)} \right) \right\}_{i=1}^{N}$ is evaluated. These samples are then used to estimate the PCE coefficients $Y_{\boldsymbol{\alpha}}$ by minimizing the squared l^2 norm of the residual

$$\left\| Y - \widetilde{Y} \right\|_2^2 = \sum_{i=1}^{N} \left[Y(\boldsymbol{\xi}^{(i)}) - \sum_{\boldsymbol{\alpha} \in \mathcal{J}_{d,p}} Y_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}^{(i)}) \right]^2 . \tag{6}$$

The optimization problem can be solved by Ordinary Least Squares (OLS). Other methods to choose the sampling set $(\Xi, \mathcal{Y})$ could result in a weighted least squares problem. To obtain a stable and accurate estimation of the PCE coefficients, the number of required samples is typically of order $N \sim \mathcal{O}(P \ln P)$, and in some cases $N \sim \mathcal{O}(P^2 \ln P)$ [66]. With the factorial increased P with respect to d and p, the required computation effort becomes prohibitive when the dimension of parameters is high.

Eq (5) can also be approximated using numerical quadrature [26], where the physical model must be evaluated at specific quadrature points. Usually, the required samples could reduce a lot compared to MC samples. The sparse quadrature rules [28, 29, 27] can reduce the required quadrature points by adopting a sparse sampling algorithm. Other methods propose sparse PCE representations where only the dominant PCE terms are considered in the expansion [30, 32, 33, 67]. These methods partially attenuate the computation burden. However, the challenge of the curse-of-dimensionality remains as the underlying physical problem is mixed with high-dimensionality and a complex computational model.

2.2. Basis adaptation
2.2.1. Rotation

Basis adaptation is a dimension reduction technique for PCE in which the Gaussian input variables are rotated by a proper rotation matrix $\boldsymbol{A}$ such that the QoI can be represented on a low-dimensional space.

Consider a rotation matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ such that $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I}$, and let $\boldsymbol{\eta}$ be the rotated variables defined as

$$\boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{\xi}, \qquad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_{\neg r} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{\eta}_r$ are the first r components, and $\boldsymbol{\eta}_{\neg r}$ are the remaining $(d - r)$ components of $\boldsymbol{\eta}$. Then, instead of expanding Y with respect to the original variable $\boldsymbol{\xi}$, we can equivalently represent Y by the rotated variables $\boldsymbol{\eta}$ as

$$Y^{\boldsymbol{A}}(\boldsymbol{\eta}) = \sum_{\boldsymbol{\beta} \in \mathcal{J}_{d,p}} Y_{\boldsymbol{\beta}}^{\boldsymbol{A}} \psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}) . \tag{8}$$

Same as in the original space, the PCE coefficients $Y_{\boldsymbol{\beta}}^{\boldsymbol{A}}$ can be computed by projecting $Y^{\boldsymbol{A}}(\boldsymbol{\eta})$ on the space spanned by $\{\psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}), \boldsymbol{\beta} \in \mathcal{J}_{d,p}\}$ (referred to as the rotated space) as

$$Y_{\boldsymbol{\beta}}^{\boldsymbol{A}} = \left\langle Y^{\boldsymbol{A}}, \psi_{\boldsymbol{\beta}} \right\rangle, \qquad \boldsymbol{\beta} \in \mathcal{J}_{d,p} . \tag{9}$$

Define

$$\psi_{\boldsymbol{\alpha}}^{\boldsymbol{A}}(\boldsymbol{\xi}) = \psi_{\boldsymbol{\alpha}}(\boldsymbol{A}\boldsymbol{\xi}) = \psi_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) . \tag{10}$$

Then, by the equivalence of Eq (3) and Eq (8), we have

$$Y_{\boldsymbol{\alpha}}^{\boldsymbol{A}} = \sum_{\boldsymbol{\beta} \in \mathcal{J}_{d,p}} Y_{\boldsymbol{\beta}} \left\langle \psi_{\boldsymbol{\beta}}, \psi_{\boldsymbol{\alpha}}^{\boldsymbol{A}} \right\rangle, \qquad Y_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta} \in \mathcal{J}_{d,p}} Y_{\boldsymbol{\beta}}^{\boldsymbol{A}} \left\langle \psi_{\boldsymbol{\beta}}^{\boldsymbol{A}}, \psi_{\boldsymbol{\alpha}} \right\rangle, \qquad \boldsymbol{\alpha} \in \mathcal{J}_{d,p} . \tag{11}$$

The above equation provides the foundation to transfer a PCE model from one representation to another and can be used to compare expansions with respect to different Gaussian variables.

2.2.2. Classical Gaussian adaptation

The rotation changes the variables the QoI expands on without changing the quality of the overall PCE, and without affecting the requisite computational burden. The idea of basis adaptation is to find a proper rotation matrix such that the probabilistic information of the QoI is concentrated on a low-dimensional space to achieve dimension reduction. Therefore, the construction of the rotation matrix becomes crucial.

The rotation matrix is constructed purely based on a first-order PCE expansion on the original space in classical Gaussian adaptation. If the first-order pilot PCE that requires a limited number of model evaluations is constructed, then the rotation matrix is created such that

$$\eta_1 = \sum_{i=1}^{d} Y_{\boldsymbol{e}_i} \xi_i , \tag{12}$$

where $\boldsymbol{e}_i$ is a d-dimensional multi-index with one at the ith location and zeros elsewhere, then $\{Y_{\boldsymbol{e}_i}\}_{i=1}^d$ is the set of first-order PCE coefficients or Gaussian coefficients in the original space. Eq (12) defines the first row of the rotation matrix $\boldsymbol{A}$, with which the first rotated variable captures the complete Gaussian components of the QoI. Next, the other rows of $\boldsymbol{A}$ are defined based on the sensitivity of the original variables. The sensitivities of the original variables $\boldsymbol{\xi}$ are quantified by the absolute values of their Gaussian coefficients. As a result, the second row of $\boldsymbol{A}$ is defined such that η_2 is the most sensitive variable of $\boldsymbol{\xi}$; the third row of $\boldsymbol{A}$ is defined such that η_3 is the second most sensitive variable of $\boldsymbol{\xi}$; and so on. Finally, a Gram-Schmidt procedure is applied to $\boldsymbol{A}$ to make it a rotation matrix. The complete procedure to construct the rotation matrix for the classical basis adaptation is summarized in Algorithm 1.

Algorithm 1: Construction of rotation matrix by classical Gaussian adaptation

---

1 Construct a pilot first-order PCE in the original space to find $\{Y_{\boldsymbol{e}_i}\}_{i=1}^d$;

2 Construct the first row of the rotation matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ such that

$$\eta_1 = \sum_{i=1}^d Y_{\boldsymbol{e}_i} \xi_i \, .$$

3 Rank the first order coefficients by absolute value in descending order and record their indices in the original coordinates as $\{\kappa_j\}_{j=1}^d$;

4 For $j = 2, \ldots, d$ construct the jth row of $\boldsymbol{A}$ such that

$$\eta_j = \xi_{\kappa_{j-1}}. \tag{13}$$

5 Perform Gram-Schmidt procedure on $\boldsymbol{A}$ to make it a rotation matrix.

The Gaussian components are typically the most important information in many applications. Therefore, the first rotated variable η_1 conveys the key information of the QoI. Since the other rotated variables are initially constructed as the most sensitive variables in the original space in descending order, the rotated variables have a descending order of importance to the QoI, motivating the seeking of a reduced representation of the QoI by the first several rotated variables. The rows of the rotation matrix $\boldsymbol{A}$ are sometimes referred to as the adaptation or adapted directions.

Suppose the first r rotated variables are adequate to represent the QoI, Y, then we can write

$$Y^{\boldsymbol{A}_r}(\boldsymbol{\eta}_r) = \sum_{\boldsymbol{\beta} \in \mathcal{J}_{r,p}} Y_{\boldsymbol{\beta}}^{\boldsymbol{A}_r} \psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}_r) \, , \tag{14}$$

where $\boldsymbol{A}_r$ denotes the sub-matrix of the first r rows of $\boldsymbol{A}$; $\mathcal{J}_{r,p}$ is a set of r-dimensional multi-indices. We will refer to Eq (14) as the r-dimensional adapted PCE, or simply r-dimensional adaptation. The PCE coefficient $Y_{\boldsymbol{\beta}}^{\boldsymbol{A}_r}$ can be calculated by projecting $Y^{\boldsymbol{A}_r}(\boldsymbol{\eta}_r)$ on the space spanned by the Hermite polynomial basis $\{\psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}_r), \boldsymbol{\beta} \in \mathcal{J}_{r,p}\}$ (referred to as the reduced rotated space) as

$$Y_{\boldsymbol{\beta}}^{\boldsymbol{A}_r} = \left\langle Y^{\boldsymbol{A}_r}, \psi_{\boldsymbol{\beta}} \right\rangle \, , \qquad \boldsymbol{\beta} \in \mathcal{J}_{r,p} \, . \tag{15}$$

Regardless of the methods used to compute Eq (15), the physical model Y must be evaluated on the original variables $\boldsymbol{\xi}$ instead of $\boldsymbol{\eta}_r$. Thus, the following approximation is employed

$$Y^{\boldsymbol{A}_r}(\boldsymbol{\eta}_r) = Y(\tilde{\boldsymbol{\xi}}) = Y \left(\boldsymbol{A}^T \begin{bmatrix} \boldsymbol{\eta}_r & \boldsymbol{0} \end{bmatrix}^T \right) \, . \tag{16}$$

That is, when projecting $\boldsymbol{\eta}_r$ back to the original coordinates, we assume that $\boldsymbol{\eta}_{\neg r} = \boldsymbol{0}$. The assumption introduces negligible errors when the first r components of $\boldsymbol{\eta}$ are adequate to capture the probabilistic information of the QoI.

The reduced dimension r remains to be determined. A systematic way starts with $r = 1$, where we construct an r-dimensional adaptation. Then, r is sequentially increased. The procedure is stopped when the difference between two adaptations with successive dimensions is within a pre-specified tolerance. The measure of the difference between two adaptations, such as r and $(r+1)$ dimensional adaptations, can be quantified by the distance of either their PCE coefficients or the associated Kernel Density Estimate (KDE) of the probability density functions (PDF) of the QoI. In the former, the r-dimensional adaptation is projected to the $(r+1)$-dimensional space, leading to a representation in the $(r+1)$-dimensional space (same idea as presented in Eq (11)). The projected PCE coefficients can then be compared to the ones of the $(r+1)$-dimensional adaptation. In the latter, many MC samples are first generated based on the constructed r- and $(r+1)$-dimensional adaptations independently. These MC samples are then used to construct two KDE estimations of the PDF of the QoI, which are then compared to find a quantified difference.

In many applications, the classical basis adaptation can identify low-dimensional spaces adapted to the QoI, for example,

[38, 39]. In [40], the authors proposed accelerated basis adaptation methods that can further enhance the accuracy of the reduced dimension r. The accelerated algorithm reduces the dimension when the classical adaptation still requires middle-to-high dimensions.

2.2.3. Accelerated adaptation

In [40], two acceleration methods of basis adaptation are proposed. The first method corrects the low-order information of the adapted PCE by utilizing the first-order pilot PCE constructed in the first step of adaptation, see Algorithm 1. The second method updates the rotation matrix by accounting for the information from higher dimensions and high-order adaptations while the dimension is sequentially increased. We will introduce the ideas behind these two methods briefly. For a detailed algorithm, please refer to [40].

Suppose the first-order pilot PCE in the original space is constructed as

$$Y(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{J}_{d,1}} Y_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = Y_{\boldsymbol{e}_0} + Y_{\boldsymbol{e}_1}\xi_1 + \cdots + Y_{\boldsymbol{e}_d}\xi_d . \tag{17}$$

Denoting the vector of Gaussian coefficients as $\boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}} = (Y_{\boldsymbol{e}_1}, \ldots, Y_{\boldsymbol{e}_d})$, then the r-dimensional adaptation of order up to one can be expanded as

$$Y^{\boldsymbol{A}_r}(\boldsymbol{\eta}_r) = \sum_{\boldsymbol{\beta} \in \mathcal{J}_{r,1}} Y_{\boldsymbol{\beta}}^{\boldsymbol{A}_r} \psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}_r) = Y_{\boldsymbol{e}_0}^{\boldsymbol{A}_r} + Y_{\boldsymbol{e}_1}^{\boldsymbol{A}_r}\eta_1 + \cdots + Y_{\boldsymbol{e}_r}^{\boldsymbol{A}_r}\eta_r . \tag{18}$$

To measure the difference in the zero and first-order expansions, the PCE (18) can be projected to the original space to obtain the associated PCE coefficients as

$$\widetilde{Y}_{\boldsymbol{e}_0} = Y_{\boldsymbol{e}_0}^{\boldsymbol{A}_r} , \quad \widetilde{Y}_{\boldsymbol{e}_1} = \sum_{i=1}^{r} Y_{\boldsymbol{e}_i}^{\boldsymbol{A}_r} A_{i1} , \quad \ldots , \quad \widetilde{Y}_{\boldsymbol{e}_d} = \sum_{i=1}^{r} Y_{\boldsymbol{e}_i}^{\boldsymbol{A}_r} A_{id} , \tag{19}$$

where A_{ij} denotes the (i, j) component of the rotation matrix $\boldsymbol{A}$. When $r = 1$, the projected coefficients can be computed explicitly as

$$\widetilde{Y}_{\boldsymbol{e}_0} = Y_{\boldsymbol{e}_0}^{\boldsymbol{A}_r} , \quad \widetilde{Y}_{\boldsymbol{e}_1} = \frac{Y_{\boldsymbol{e}_1}^{\boldsymbol{A}_r} Y_{\boldsymbol{e}_1}}{\|\boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}}\|_2} , \quad \ldots , \quad \widetilde{Y}_{\boldsymbol{e}_d} = \frac{Y_{\boldsymbol{e}_1}^{\boldsymbol{A}_r} Y_{\boldsymbol{e}_d}}{\|\boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}}\|_2} . \tag{20}$$

Defining $\widetilde{\boldsymbol{Y}}_{\text{coeff}}^{\boldsymbol{e}} = (\widetilde{Y}_{\boldsymbol{e}_1}, \ldots, \widetilde{Y}_{\boldsymbol{e}_d})$, then the errors of the Gaussian coefficients are quantified as

$$\boldsymbol{\epsilon}_1 = \boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}} - \widetilde{\boldsymbol{Y}}_{\text{coeff}}^{\boldsymbol{e}} = \left(1 - \frac{Y_{\boldsymbol{e}_1}^{\boldsymbol{A}_r}}{\|\boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}}\|_2} \right) \boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}} . \tag{21}$$

We see that the arithmetic errors are proportional to the exact Gaussian coefficients $\boldsymbol{Y}_{\text{coeff}}^{\boldsymbol{e}}$, implying that more significant errors appear in the terms with greater Gaussian coefficients. However, greater Gaussian coefficients imply greater sensitivities in the variable, and greater errors in the terms with greater sensitivity introduce more significant errors in the PCE. As dimension r of the adaptation increases, the errors in Gaussian coefficients decrease and will recover to the exact values when $r = d$. In order to eliminate the errors in the zero and first-order information of the adapted PCE, the authors proposed to correct the zero and first-order PCE coefficients on the adapted space by the exact zero and first-order PCE coefficients on the original space. The idea is accomplished by projecting Eq (17) to the r-dimensional adapted space and obtaining the projected coefficients as

$$\widetilde{Y}_{\boldsymbol{e}_0}^{\boldsymbol{A}_r} = Y_{\boldsymbol{e}_0} , \quad \widetilde{Y}_{\boldsymbol{e}_1}^{\boldsymbol{A}_r} = \sum_{i=1}^{d} Y_{\boldsymbol{e}_i} A_{1i} , \quad \ldots , \quad \widetilde{Y}_{\boldsymbol{e}_r}^{\boldsymbol{A}_r} = \sum_{i=1}^{d} Y_{\boldsymbol{e}_i} A_{ri} . \tag{22}$$

These coefficients are the exact zero and first-order PCE coefficients in the r-dimensional adapted space and can be used to correct the coefficients in Eq (18). Then, the zero and first-order information are guaranteed to be accurate.

For the second method, we noticed that with the increased r in the adaptation, the QoI could be better represented by the adapted PCE. In addition, the r-dimensional adapted PCE retains the information of the first r rotated variables up to order p. However, the rotation matrix in the classical adaptation only considers the first-order information and remains constant even when more probabilistic information becomes available. The second method in [40] proposed to update the rotation matrix $\boldsymbol{A}$ as r is sequentially increased. Specifically, suppose the r-dimensional adaptation from the rotation matrix $\boldsymbol{A}$ is available. In that case, the best one-dimensional adaptation close to the r-dimensional adaptation is discovered from an updated rotation matrix $\boldsymbol{B}$. The one-dimensional adaptation by $\boldsymbol{B}$ can be written as

$$Y^{\boldsymbol{B}_1}(\zeta_1) = \sum_{\boldsymbol{\gamma} \in \mathcal{J}_{1,p}} Y_{\boldsymbol{\gamma}}^{\boldsymbol{B}_1} \psi_{\boldsymbol{\gamma}}(\zeta_1) = \sum_{\boldsymbol{\gamma} \in \mathcal{J}_{1,p}} Y_{\boldsymbol{\gamma}}^{\boldsymbol{B}_1} \psi_{\boldsymbol{\gamma}}^{\boldsymbol{B}_1}(\boldsymbol{\xi}) , \tag{23}$$

where $\boldsymbol{\zeta} = \boldsymbol{B}\boldsymbol{\xi}$ is the new adapted variable by $\boldsymbol{B}$. To compute the PCE coefficients of the new one-dimensional adaptation, one would usually require to evaluate the physical model Y on some parameters. To avoid the additional computation, the evaluation is instead performed on the r-dimensional adapted PCE by $\boldsymbol{A}$, the best available model. For any choice of $\boldsymbol{B}$, one can easily construct the one-dimensional adapted PCE of the QoI. Then, an optimization procedure can be employed to find the best $\boldsymbol{B}$ that minimizes the difference between the one-dimensional adaptation by $\boldsymbol{B}$ and the r-dimensional adaptation by $\boldsymbol{A}$. This method can be combined with the first method that always corrects the zero and first-order PCE coefficients in the adapted PCEs. The combined method is named the sequentially optimized adaptation (SOA) with its workflow in Figure 1. Please refer to [40] for details.
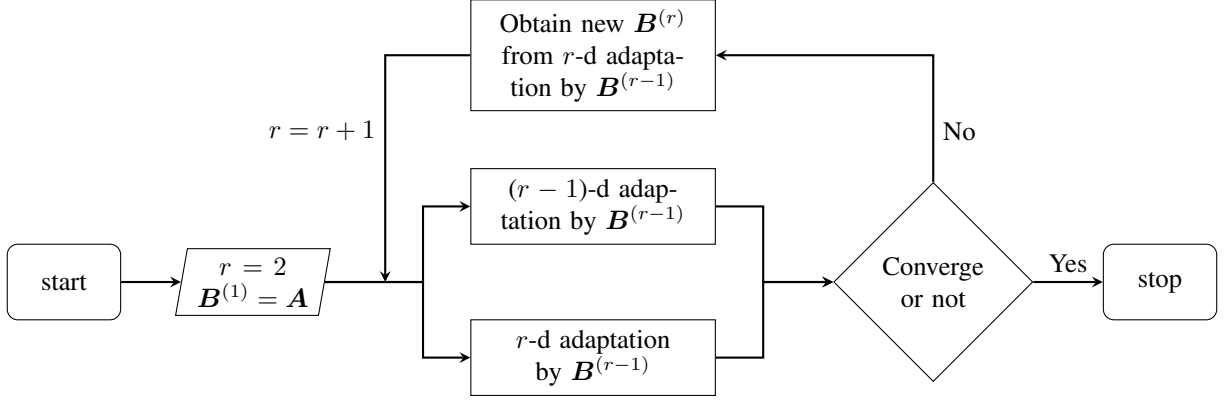
Figure 1: Workflow of the sequentially optimized adaptation method [40].

The basis adaptation is a great tool for dimension reduction in UQ. However, when it comes to computing the adapted PCE coefficients, for example, by Eq (15), we typically use sparse quadrature rules to reduce the number of evaluations while having good accuracy at the same time. In this setting, the model must be evaluated at specific quadrature points. We will next introduce a novel data-driven method that can simultaneously find the optimal rotation matrix and the PCE coefficients.

3. Projection pursuit adaptation

3.1. Projection pursuit regression

Projection pursuit (PP) intends to find the most "interesting" projections in high-dimensional data. These projections are discovered such that they reveal the most information embedded in the data. Therefore, the projections of PP and the adaptation directions by the basis adaptation method have the same purpose but are discovered with different algorithms. The advantages and disadvantages of these two methods are worth exploring. In the sense of adapting input to output, the projection pursuit regression (PPR) was introduced [59], where the projections and the functions from input space to output space are built simultaneously.

In a general supervised learning setting, assume we have a data set $\{(\boldsymbol{x}^i, y^i)\}_{i=1}^N$ with $\boldsymbol{x}^i \in \mathbb{R}^d$ and $y^i \in \mathbb{R}$. Let $\boldsymbol{c}_j$, $j = 1, \ldots, r$ be d-dimensional known vectors. Then the PPR model seeks to approximate y^i as follows

$$y^i \approx \sum_{j=1}^{r} f_j(\boldsymbol{c}_j^T \boldsymbol{x}^i), \tag{24}$$

where $\{f_j : \mathbb{R} \mapsto \mathbb{R}\}$ is a set of univariate smooth functions. This model aims to find $(f_j, \boldsymbol{c}_j)$ such that the approximation fits the data in some optimal sense. Specifically, the following optimization problem is solved to fit such a model with the given data

$$\underset{(f_j, \boldsymbol{c}_j)}{\arg\min} \sum_{i=1}^{N} \left[y^i - \sum_{j=1}^{r} f_j(\boldsymbol{c}_j^T \boldsymbol{x}^i) \right]^2. \tag{25}$$

In this optimization problem, the number of required projections, r, is determined by a stage-wise greedy algorithm where we add a pair of $(f_j, \boldsymbol{c}_j)$ at each stage and stop the procedure if additional information cannot improve the model fitting significantly. For example, suppose we are in stage j and want to find the optimal $(f_j, \boldsymbol{c}_j)$. Let the previously found parameters, $\{(f_l, \boldsymbol{c}_l)\}_{l=1}^{j-1}$ be fixed, then the residual of the ith data point is obtained as

$$\nu^i = y^i - \sum_{l=1}^{j-1} f_l(\boldsymbol{c}_l^T \boldsymbol{x}^i). \tag{26}$$

7

Then, the optimal $(f_j, \boldsymbol{c}_j)$ can be found by solving the following optimization problem

$$\underset{(f_j, \boldsymbol{c}_j)}{\arg\min} \sum_{i=1}^{N} \left[\nu^i - f_j(\boldsymbol{c}_j^T \boldsymbol{x}^i) \right]^2 . \tag{27}$$

When the total residual of the data $\sum_{i=1}^{N} \nu^i$ is converged, we stop the procedure.

An alternative to the greedy procedure is to iteratively update the pairs $(f_k, \boldsymbol{c}_k)$ for $k = 1, \ldots, j$ through backfitting. While this approach could result in significantly fewer terms in the model, the computation cost is increased, and the improvement in predictions is insignificant [60]. In addition, the same performance can usually be achieved by adding more terms to the model. Therefore, we will focus on the stage-wise greedy procedure.

The optimization problem (27) remains to be solved in each stage. The solution can be obtained by an alternating optimization procedure, where the optimization switches between f_j and $\boldsymbol{c}_j$ until converge. Specifically, if the vector $\boldsymbol{c}_j$ is given, then the argument of f_j is a one-dimensional variable. The problem becomes finding the optimal f_j such that the residual is minimized. This can be done by the regression method. On the other hand, assuming f_j is given, one needs to find the optimal $\boldsymbol{c}_j$ to minimize the residual. This can be done by a Gauss-Newton search, a quasi-Newton method that keeps only the zero and first derivatives in a Taylor expansion. Let $\boldsymbol{c}_{j,\text{old}}$ be the current estimate of $\boldsymbol{c}_j$, then we can write the following Taylor expansion

$$f_j(\boldsymbol{c}_j^T \boldsymbol{x}^i) \approx f_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i) + \dot{f}_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)(\boldsymbol{c}_j - \boldsymbol{c}_{j,\text{old}})^T \boldsymbol{x}^i . \tag{28}$$

Substituting Eq (28) into Eq (27) (with f_j given) leads to

$$\underset{\boldsymbol{c}_j}{\arg\min} \sum_{i=1}^{N} \left[\nu^i - f_j(\boldsymbol{c}_j^T \boldsymbol{x}^i) \right]^2 \approx \underset{\boldsymbol{c}_j}{\arg\min} \sum_{i=1}^{N} \dot{f}_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)^2 \left[\left(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i + \frac{\nu^i - f_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)}{\dot{f}_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)} \right) - \boldsymbol{c}_j^T \boldsymbol{x}^i \right]^2 . \tag{29}$$

The optimization problem (29) can be solved by the weighted least squares regression with target $u = \boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i + \frac{\nu^i - f_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)}{\dot{f}_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)}$, regressors $\boldsymbol{x}^i$, weight $\omega = \dot{f}_j(\boldsymbol{c}_{j,\text{old}}^T \boldsymbol{x}^i)^2$, and no bias term. Let $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ denote the diagonal matrix with entries ω, $\boldsymbol{u} \in \mathbb{R}^N$ be the vector with entries u, and use the full data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, then $\boldsymbol{c}_j$ is found by solving the following

$$\underset{\boldsymbol{c}_j}{\arg\min} \|\boldsymbol{u} - \boldsymbol{X} \boldsymbol{c}_j\|_{\boldsymbol{W}}^2 = \left(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X} \right)^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{u} . \tag{30}$$

With an updated $\boldsymbol{c}_j$, we find new projections $\boldsymbol{c}_j^T \boldsymbol{x}^i$ and refit f_j to minimize the residual. Then $\boldsymbol{c}_j$ can be updated again with the new f_j. We repeat the alternating procedure until $(f_j, \boldsymbol{c}_j)$ pair is converged.

Although there is no restriction to the smooth functions $\{f_j\}$, it is better if the functions are once differentiable. PCE is a good fit here where each smooth function f_j is a univariate PCE with respect to the projected variable $\boldsymbol{c}_j^T \boldsymbol{x}^i$. In this case, the variables $\boldsymbol{x}$ that can be arbitrarily distributed (based on prior knowledge or from the data) need to be first transformed to standard Gaussian distributions via, for example, inverse transform sampling or Rosenblatt transformation [68]. Then, with a given $\boldsymbol{c}_j$, finding the optimal f_j is equivalent to finding the PCE coefficients of f_j. One of the benefits of using PCEs as smooth functions is that the order of the PCE is adjustable, and we can gradually increase the polynomial orders until convergence is reached. Based on the flexibility of PCE, r can be arbitrarily large to approximate any function in $L^2(\Omega)$, and the computational cost lies in constructing r univariate PCEs.

Note that there is no restriction on the projection in the optimization problem (30), and each projection is discovered based on the current residual. The initial guess of $\boldsymbol{c}_j$ and the data determine where the "optimal" value land. Therefore, there are chances that the weighted least squares find the local optima and stop the stage-wise procedure when the local optima suggest there is not much room for improvement. In addition, there is no restriction on the projections. No coupling effects are considered among the projections since the smooth function f_j only considers information on the jth projected variable. One of the advantages of such construction is that $\{f_j\}$ are univariate functions and are easy to compute with given projections and a given number of samples.

3.2. Integration of projection pursuit and basis adaptation

In the basis adaptation method, we already discovered that many scalar QoI could be represented in a low-dimensional space ([39, 38, 40, 69]). So the question is how to find the optimal low-dimensional space and the PCE on that space. The PPR method is an excellent tool for achieving that in a data-driven setting.

However, the original construction considers univariate representations in PPR with PCE employed as smooth functions. Therefore, it does not take advantage of the good mean-squared convergence property and the ability to represent any variable in the parameter space of the multivariate PCE construction. Accordingly, we will explore using multivariate PCE as a smooth function in PPR. In addition, the projections in this effort are assumed to be orthonormal, which has control of the exploring

space of the next projection based on the current projections. Therefore, in this case, the projections are equivalent to the rows of the rotation matrix, and the smooth function is equivalent to multivariate PCE in the basis adaptation method. Hence, we refer to this method as the projection pursuit adaptation (PPA).

Let $C_r = \begin{bmatrix} c_1 & \dots & c_r \end{bmatrix}^T$, with each row representing a projection and with the projections being mutually orthogonal. Also let z^i denote the projected variables such that

$$z^i = \begin{bmatrix} z_1^i & \dots & z_r^i \end{bmatrix}^T = C_r x^i = \begin{bmatrix} c_1 & \dots & c_r \end{bmatrix}^T x^i, \qquad x^i \in \mathbb{R}^d, \quad z^i \in \mathbb{R}^r \tag{31}$$

The PPA method seeks the following model

$$y^i \approx g_r(z^i), \tag{32}$$

where $g_r(\cdot)$ is r-dimensional PCE. We want to find (g_r, C_r) such that the approximation fits the data well. Unlike PPR, where the regression model is an additive of many univariate functions, the PPA seeks an r-dimensional multivariate model. Same as in PPR, the following optimization problem is solved

$$\underset{(g_r, C_r)}{\arg\min} \sum_{i=1}^{N} \left[y^i - g_r(z^i) \right]^2. \tag{33}$$

We will use a stage-wise greedy strategy to solve the optimization problem. In the rth stage, for example, the previously computed projections C_{r-1} are unchanged, and we want to get the optimal c_r that is appended to C_{r-1}. While for the smooth function, the optimal r-dimensional g_r is computed independently of the previous information in g_{r-1}. That is, the stage-wise greedy strategy finds optimal (g_r, c_r) at each stage. Like PPR, g_r and c_r are computed by an alternating optimization procedure, in which the optimization switches between g_r and c_r until they converge. We can discover the optimal r-dimensional PCE g_r by regression if C_r is given. While for a given g_r, we calculate the optimal c_r that is appended to the previously computed C_{r-1}. The reason for not updating the whole C_r matrix at each stage is to avoid excessive computation and propose a convenient Gauss-Newton search algorithm for optimization. With a fixed g_r, the following problem remains to be solved

$$\underset{c_r}{\arg\min} \sum_{i=1}^{N} \left[y^i - g_r(z^i) \right]^2, \tag{34}$$

which is similar to Eq (27) with f_j fixed. However, the function $g_r(\cdot)$ here is a multivariate function. In order to use the Gauss-Newton search, let $c_{r,\text{old}}$ be the current estimate of c_r, the following multi-dimensional Taylor expansion is applied

$$
\begin{aligned}
g_r(z^i) &= g_r(\begin{bmatrix} z_1^i & \dots & z_{r-1}^i & z_r^i \end{bmatrix}^T) = g_r(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_r \end{bmatrix}^T x^i) \\
&\approx g_r\left(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T x^i \right) + \nabla g_r\left(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T x^i \right) \\
&\quad \times \left(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_r \end{bmatrix}^T - \begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T \right) x^i \\
&= g_r\left(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T x^i \right) + \frac{\partial g_r}{\partial z_{r,\text{old}}^i}\left(\begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T x^i \right) (c_r - c_{r,\text{old}})^T x^i \\
&= g_r(z_{\text{old}}^i) + \frac{\partial g_r(z_{\text{old}}^i)}{\partial z_{r,\text{old}}^i} (c_r - c_{r,\text{old}})^T x^i,
\end{aligned}
\tag{35}
$$

where $\nabla g_r(\cdot)$ denotes the gradient of $g_r(\cdot)$; $z_{\text{old}}^i = \begin{bmatrix} c_1 & \dots & c_{r-1} & c_{r,\text{old}} \end{bmatrix}^T x^i$; $z_{r,\text{old}}^i = c_{r,\text{old}}^T x^i$. In (35), $\{c_j\}$ for $j < r$ are the known previously obtained projections. Although the Taylor expansion is multi-dimensional, we avoid the gradient operations, thus replacing a multi-dimensional regression with a one-dimensional regression by updating only the rth projection. Substituting (35) to (34) leads to the following optimization problem

$$\underset{c_r}{\arg\min} \sum_{i=1}^{N} \left(\frac{\partial g_r(z_{\text{old}}^i)}{\partial z_{r,\text{old}}^i} \right)^2 \left[\left(c_{r,\text{old}}^T x^i + \frac{y^i - g_r(z_{\text{old}}^i)}{\partial g_r(z_{\text{old}}^i)/\partial z_{r,\text{old}}^i} \right) - c_r^T x^i \right]^2, \tag{36}$$

Again, the optimization problem (36) can be solved by the weighted least squares regression with target $\hat{u} = \frac{y^i - g_r(z_{\text{old}}^i)}{\partial g_r(z_{\text{old}}^i)/\partial z_{r,\text{old}}^i}$, regressors x^i, weight $\hat{\omega} = \left(\partial g_r(z_{\text{old}}^i)/\partial z_{r,\text{old}}^i \right)^2$, and no bias term. Let $\hat{W} \in \mathbb{R}^{N \times N}$ denote the diagonal matrix with entries $\hat{\omega}$,

and $\hat{u} \in \mathbb{R}^N$ be the vector with entries $\hat{u}$, then $\boldsymbol{c}_r$ can be found by a solution to the following optimization problem

$$\arg\min_{\boldsymbol{c}_r} \|\hat{\boldsymbol{u}} - \boldsymbol{X}\boldsymbol{c}_r\|_{\hat{\boldsymbol{W}}}^2 = \left(\boldsymbol{X}^T \hat{\boldsymbol{W}} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \hat{\boldsymbol{W}} \hat{\boldsymbol{u}}. \tag{37}$$

Once $\boldsymbol{c}_r$ is computed, we apply a Gram-Schmidt procedure to all the projections to maintain orthogonality among them. Then, with the updated $\boldsymbol{c}_r$, we find new projections of $\boldsymbol{x}^i$ and refit g_r to minimize the residual. Then $\boldsymbol{c}_r$ can be updated again with the new g_r. We repeat the alternating procedure until $(g_r, \boldsymbol{c}_r)$ pair is converged.

The detailed PPA procedure is summarized in Algorithm 2.

Algorithm 2: Projection Pursuit Adaptation (PPA)

1   Let $r = 1$;

2   **while** $\{g_r(\boldsymbol{C}_r \boldsymbol{x}^i)\}$ *is not converged* **do**

3   Initial guess of $\boldsymbol{c}_r$;

4   Project the original variables $\{\boldsymbol{x}^i\}$ to projected variables $\{\boldsymbol{z}^i\}$;

5   Find the optimal g_r (PCE) with respect to $\boldsymbol{z}^i$ by least squares;

6   **while** $(g_r, \boldsymbol{c}_r)$ *is not converged* **do**

7   Compute corresponding $\hat{c}, \hat{\omega}$ in (37);

8   Solve (37) to obtain an updated $\boldsymbol{c}_r$;

9   Perform Gram-Schmidt on $\boldsymbol{C}_r$ to maintain orthogonality;

10   Update the projected variables $\{\boldsymbol{z}^i\}$;

11   Find the updated g_r (PCE) with respect to $\boldsymbol{z}^i$ by least squares;

12      $r = r + 1$.

The PPA method always finds the optimal r-dimensional PCE adapted to the output or the QoI, starting from $r = 1$. The optimal model is in the sense of optimal directions and optimal PCE. Same as in basis adaptation, the dimension r is sequentially increased, resulting in PCE models that can better approximate the data. The procedure is stopped when adding a new projection cannot reduce the residual significantly.

Generally, fitting a multi-dimensional PCE model is much more expensive than fitting multiple univariate PCEs. However, the goal is to propose a representation of the actual model on a low-dimensional manifold. If the reduced dimension is small enough, the increase in computational cost is usually acceptable. In addition, the procedure here is entirely data-driven; no additional model evaluations are required to find the projections and the PCE coefficients.

4. Applications

4.1. Borehole model

The PPA method is first applied on a borehole model (the same model as in [40]) to make a comparison with the classical basis adaptation and the accelerated basis adaptation,

$$f(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{38}$$

which models the water flow through a borehole drilled from the ground surface through two aquifers. In this model we assume the radius of the borehole is $r_w = 0.1$ (m). The other parameters are randomized and listed in Table 1. The QoI $f(\boldsymbol{x})$ is the water flow rate of the borehole (m³/yr).

A previous UQ exploration [40] of the borehole model showed that an accurate probability density function (PDF) of the water flow rate could be obtained through (i) MCS of 3000 samples; (ii) classical basis adaptation of dimension 5 that requires 995 samples evaluated at specific Gauss quadrature points; (iii) accelerated basis adaptation of dimension 3 that required 308 samples evaluated on quadrature points. The maximum PCE order is $p = 3$ in the last two methods, which is verified to be adequate.

We now apply the data-driven PPA method to learn the adapted projections and the PCE model with respect to the projected variables with a different number of MC samples $\{20, 40, 60, 80, 100, 125, 150, 200\}$. Using the same maximum PCE order $p = 3$, we learn the converged optimally adapted projections and the corresponding optimal PCE models. Then we can generate

Table 1: Random parameters and distributions of the borehole model

Variable	Distribution	Description
r	Lognormal($\mu = 7.71, \sigma = 1.0056$)	Radius of influence (m)
T_u	Uniform(63070, 115600)	Transmissivity of upper aquifer (m^2/yr)
H_u	Uniform(990, 1110)	potentiometric head of upper aquifer (m)
T_l	Uniform(63.1, 116)	transmissivity of lower aquifer (m^2/yr)
H_l	Uniform(700, 820)	potentiometric head of lower aquifer (m)
L	Uniform(1120, 1680)	length of borehole (m)
K_w	Uniform(9855, 12045)	hydraulic conductivity of borehole (m/yr)

many MC samples based on the resulting PCE models and estimate the PDF of the QoI by the KDE. A reference PDF is generated from 100,000 MC simulations of the borehole model by KDE for comparison. For any two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, we define the relative l^2 difference of $\boldsymbol{u}$ with respect to $\boldsymbol{v}$ as the following

$$e = \frac{\|\boldsymbol{u} - \boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2} \tag{39}$$

where $\|\cdot\|_2$ represents the l^2 norm. Then, compared to the reference PDF, we can compute the relative l^2 error of the PDFs generated from PPA with a different number of MC samples. The results are shown in Figure 2. We see that the relative l^2 errors
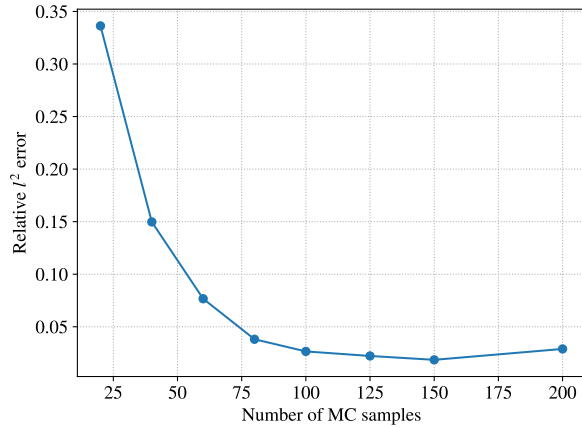
Figure 2: The convergence curve of the relative l^2 error of the PDF obtained from PPA with different number of MC samples for the borehole model.
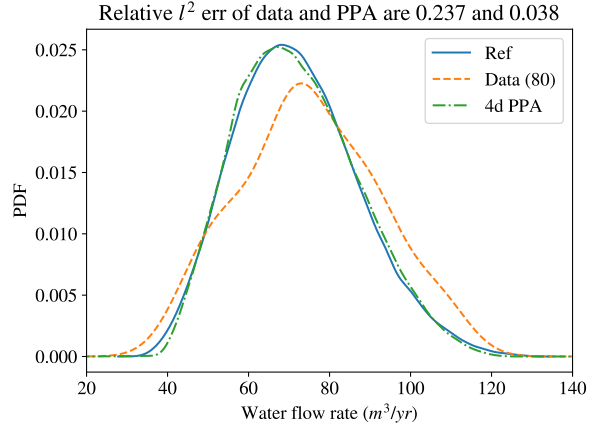
Figure 3: PDF comparison of the PPA method for the borehole model. Three PDFs are compared: the reference from 100,000 MC samples, the data, and the converged PPA model.

are smaller than 5% when the MC samples are greater than 80 and converge with 150 MC samples. Therefore, we show the PDF comparison when 80 MC samples are used to learn the PPA model in Figure 3. In the figure, three PDFs are compared, the reference PDF, the PDF generated by the data, and the PDF by the PPA model. Clearly, the PPA PDF is close to the reference, with a relative l^2 error of 3.8%. In comparison, the PDF generated directly by the data has a large discrepancy from the reference, with an l^2 error of 23.7%.

While for evaluating lower order statistics, the accuracy of this PPA model using 80 MC samples is adequate, other evaluations, such as reliability and failure probabilities, could require more samples. It is clear from Figure (2) that 150 MC samples are sufficient for l^2 convergence, from which convergence in distribution, relevant for tail analysis, follows.

To check the accuracy of the PPA model, Figures 4 and 5 present a comparison of PDFs and CDFs obtained from the reference model, the 150 data, and the PPA model generated from the 150 data. The first observation is that the resulting PDF of the PPA method is even closer to the reference compared to the case of using 80 MC samples. The relative l^2 difference between the PPA model and the reference decreases to 1.9%. The accuracy improvement in the left tail region is noticeable. The second observation is that the PDF generated directly from the data differs from the reference, especially in the left tail region. From the CDF figure, the curve associated with the data is far from the reference, while the result from PPA is consistent with the reference.

Since the PPA method is developed based on the PPR method, a comparison of these two techniques is conducted. PCEs are
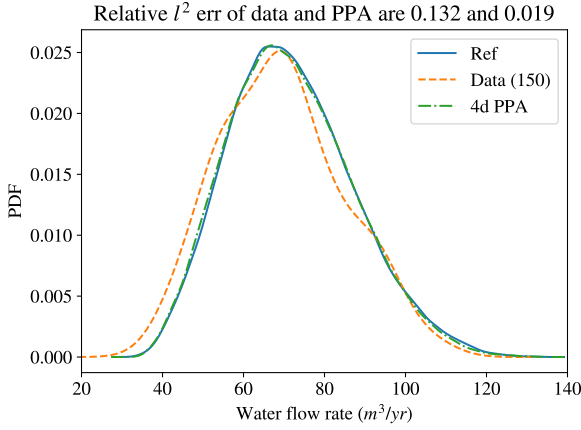
Figure 4: PDF comparison of the PPA method for the borehole model. Three PDFs are compared: the reference from 100,000 MC samples, the data, and the converged PPA model.
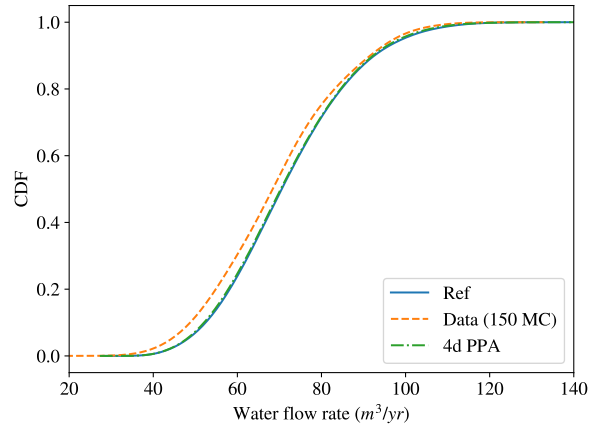
Figure 5: CDF comparison of the PPA method for the borehole model. Three PDFs are compared: the reference from 100,000 MC samples, the data, and the converged PPA model.

used as smooth functions in PPR, leading to an additive model of many univariate PCEs. We apply the PPR method on the 150 MC samples to obtain a converged model with the same stopping criterion as the PPA method. Then, the PPR model is compared to the PPA model. The resulting PDFs and CDFs are shown in Figures 6 and 7. We see that the PPR model also has improved
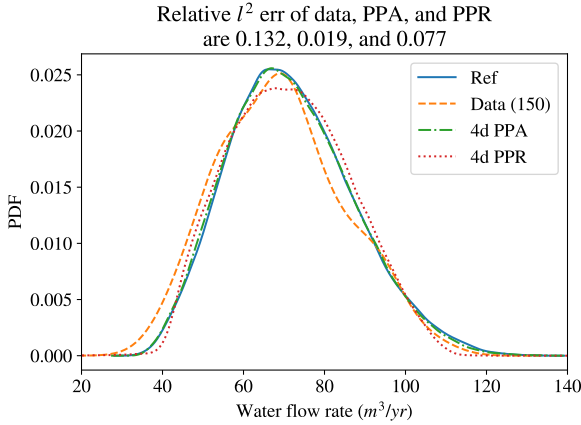
Figure 6: PDF comparison of the PPA method for the borehole model. Three PDFs are compared: the reference from 100,000 MC samples, the data, the converged PPA model, and the converged PPR model.
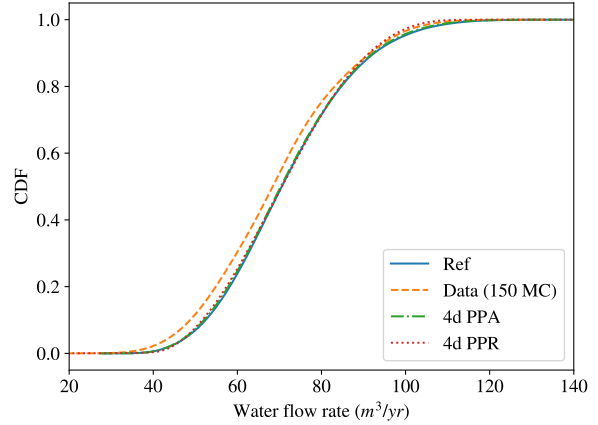
Figure 7: CDF comparison of the PPA method for the borehole model. Three CDFs are compared: the reference from 100,000 MC samples, the data, the converged PPA model, and the converged PPR model.

over the data and is consistent with the reference overall. However, it misses the left and right tail regions, and the PDF peak at around 70 m^3/yr water flow rate. The relative l^2 error of the PPR model is 7.7% compared to 1.9% of the PPA model. The tail regions usually correspond to reliability or failure probability and thus are important in engineering. The CDF comparison presents no clear separation between PPA and PPR. However, we can still see a difference if we examine the right tail. For example, $P(f(\boldsymbol{x}) < 97) = 0.95$ for PPR, while $P(f(\boldsymbol{x}) < 99) = 0.95$ for PPA, suggesting that these two models could make different conclusions regarding the reliability and failure probability.

Compared to MCS, where 3000 samples are required for a converged PDF, we use only 150 samples in PPA to construct a model with the same accuracy. The massive computation saving is because (i) the QoI can be represented on a low-dimensional manifold; (ii) the PPA method can discover the low-dimensional manifold and the function that maps the parameter space to the output space. In this application, the PPA reduced the dimension from 7 to 4, and the number of PCE terms of order 3 has reduced from 120 to 35. Moreover, the number of MC samples required to compute the PCE coefficients accurately will reduce significantly. However, even if we use the accelerated basis adaptation method where the dimension could be reduced to 3, the number of sparse quadrature points is 308, which is still two times greater than 150. In addition, the model is evaluated on specific quadrature points in accelerated basis adaptation to save cost, which in many applications might not be available.

The PPA model can not only be used for UQ but can also be used in model prediction. In the latter, the PPA model can serve as a surrogate model to predict QoI with different input parameters. The PPA model essentially constructed a nonlinear mapping

from input space to output space based on the given data. For any given input parameter, the prediction of the associated output value can be obtained by querying from the PPA model. We call the data used to construct the PPA model training data, which in this case, is the 150 MC samples. To check the accuracy of the PPA model in prediction, we generate 100 MC samples that are different from the training data. We can predict the QoI on these 100 samples using the PPA model. The comparison of the prediction and the reference model on the test set is shown in Figure 8. From the left figure, the PPA prediction (in orange) is
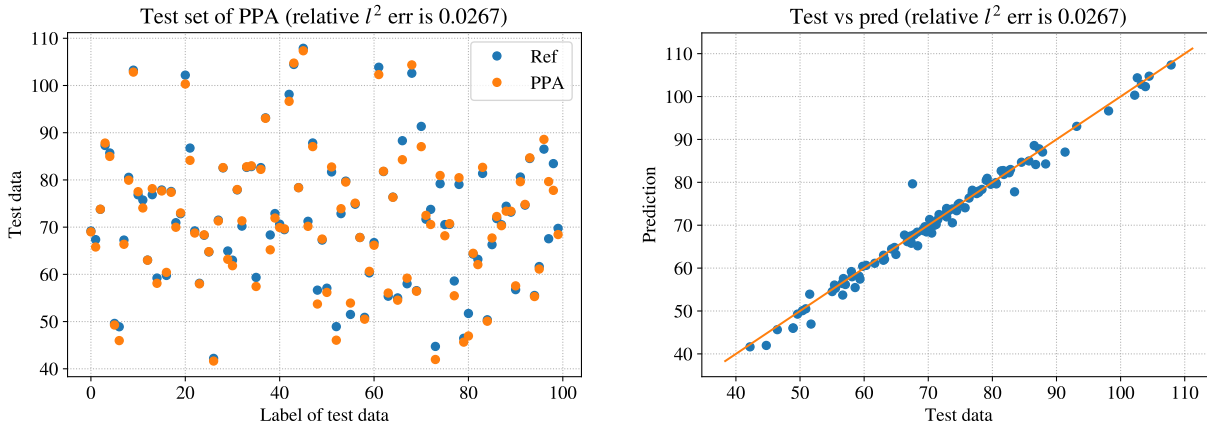
Figure 8: Test data (100) of the PPA method for the borehole model. Figure on left shows the QoI value comparison of reference and prediction on various test data. Figure on the right plot the prediction against the test data where the solid line is of slope 1.

close to the reference (in blue) on the majority of the test data. From the right figure, the scatter dots of prediction against the test set are closely aligned in the unit slope line, meaning that the predictions are close to the reference. The relative l^2 difference of the 100 test data is 2.0%, which is close to the relative l^2 error of the PPA PDF compared to the reference PDF. The results suggest that the prediction is accurate.

To check the robustness of the PPA method, we generated 100 sets of MC samples, each containing 150 samples. Within each set, we apply the PPA method to find a converged PCE model and generate many MC samples and compute the associated KDE estimation of the PDF. The PDFs from these sets are plotted in Figure 9, where the reference PDF in thick black is also presented. Although generated with different MC samples, the PDFs are all close to the reference. To quantify the difference, we
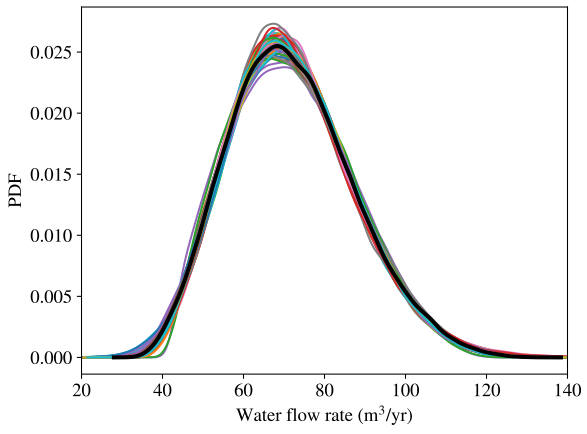
Figure 9: Plot of 100 PDF batch generated by PPA method with different 150 MC samples. The thick black curve represent the reference PDF.
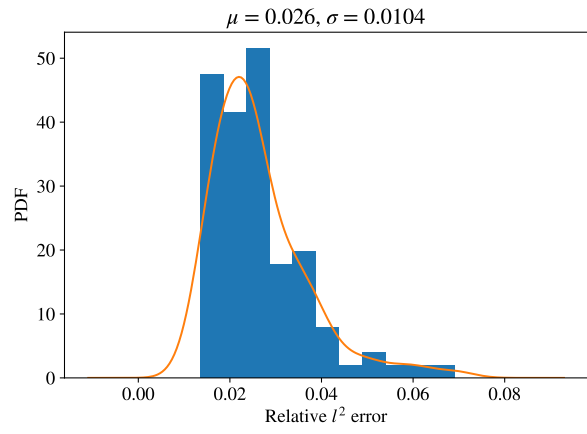
Figure 10: The histogram of relative l^2 errors of the 100 PDFs from Figure 9 compared to the reference PDF.

computed the relative l^2 errors of these 100 PDFs with respect to the reference. The histogram of the relative l^2 errors is shown in Figure 10. The mean value and standard deviation of the relative l^2 errors are 0.026 and 0.0092, respectively. All the relative l^2 errors are less than 5%, and most of the l^2 errors are less than 3%. This means that the variation in the accuracy of the PDF estimation is small when different MC samples are used in the PPA method.

4.2. Structural Dynamics

The second application considers a space structure subjected to impulse load in the Z direction. The computational model is shown in Figure 11. The model can be separated into upper and lower parts connected by a three-point mounting pedestal. The
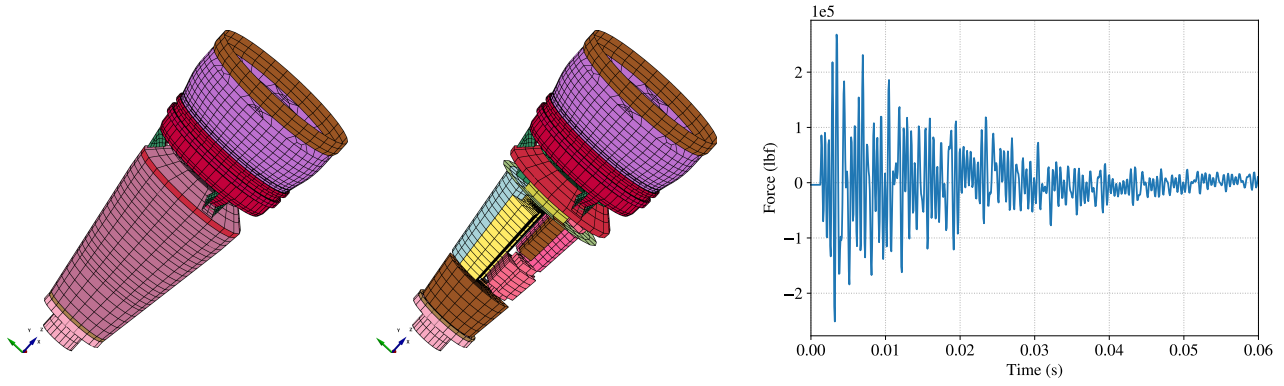
Figure 11: The overall finite element model of the space structure (left); the finite element model of the space structure without the outer shell and shock absorption block (middle); time history of the impulse load (right).

upper part is open and is subjected to impulse load with the time history shown in Figure 11. The lower part contains an outer shell, a solid shock absorption block, and essential components that can be seen in the middle figure of Figure 11. The effects of the impulse loading are transferred from the upper part to the lower parts through the three-point mounting pedestals. Therefore, they can affect the functionality of the essential components in the lower part. In this application, we are particularly interested in one essential part maximum acceleration and maximum velocity responses. The time history of the acceleration and velocity of the essential part is shown in Figure 12.
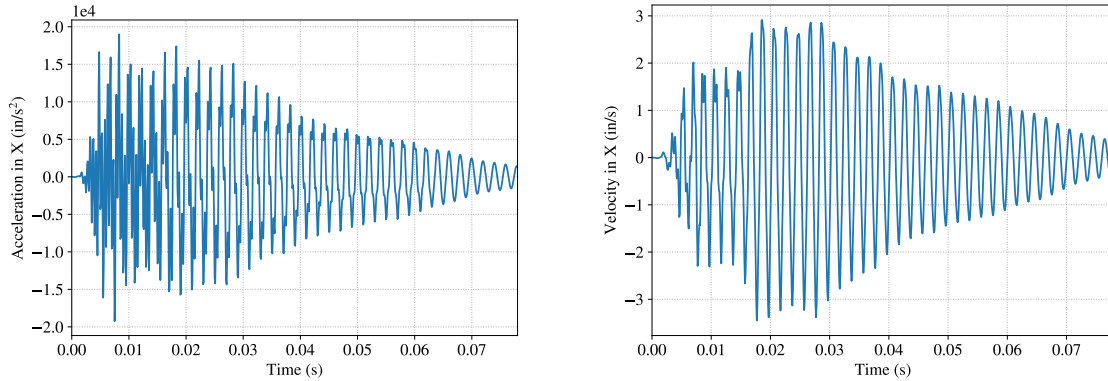
Figure 12: Time history of acceleration (left) and velocity (right) of the essential part under the impulse loading shown in Figure 11.

4.2.1. Uncertainty quantification and prediction

The maximum acceleration and velocity can be directly read from the response histories in a deterministic setting. However, the material properties of the model are almost inevitably subjected to variations, especially for some special materials. The randomness of the materials can affect the results a lot. Therefore, in this application, some material properties are randomized, see Table 2. Note that the randomization here differs from the model in [40]. In this space model, many structural components are modeled in two parts, a solid element part and a shell element part that wraps around the solid part. In this paper, we randomized the shell and solid elements, while in [40], only shell elements are randomized. However, a shared random germ is used for a single structure component, leading to a random dimension of 24, the same as in [40].

In the first calculation, the QoI is the maximum X acceleration along the time. Several methods can be applied to solve the forward UQ problem. Since the dimension is relatively high, a MCS requires hundreds of thousands of samples to reach a converged result. If a 3rd order PCE model is to be built, we can use a level 3 sparse Smolyak quadrature to reduce the cost; however, even in this case, the required quadrature points are $24,449$, which is still too expensive to compute. Therefore, we considered using basis adaptation and accelerated basis adaptation to solve the forward UQ problem. A first-order pilot PCE is first built from 49 level 1 quadrature points. Then, the rotation matrix adapted to the QoI is built for the classical adaptation and can be utilized to construct adaptations with sequentially increased dimensions. We update the rotation matrix for the accelerated algorithm whenever new information is fed into the model. The generated PDFs of the classical basis adaptation are shown in Figure 13, where we see that it converges at dimension 7. The required quadrature points are 2731. In contrast, the generated

Table 2: Random parameters and distributions of the space model

Variable	Distribution	Mean	Coefficient of variation
Modulus of elasticity of 3 major components of the upper parts	Lognormal	1.6×10^7 psi	10%
Stiffness of the 3 point mounting pedestals	Lognormal	5.7×10^7 lbs/in	20%
Modulus of elasticity of 18 components in the lower part	Lognormal	14 with 1.0×107 psi; 2 with 9.76×107 psi; 1 with 2.93×107 psi; 1 with 8829.0 psi	20%

PDFs of the accelerated basis adaptation are shown in Figure 14, where we see that it converges at dimension 6. The required quadrature points are 1708. Then, for these two methods, the total model evaluation on the specific quadrature points is 2780 and 1757, respectively. The required samples are much fewer than PCE and MC methods.
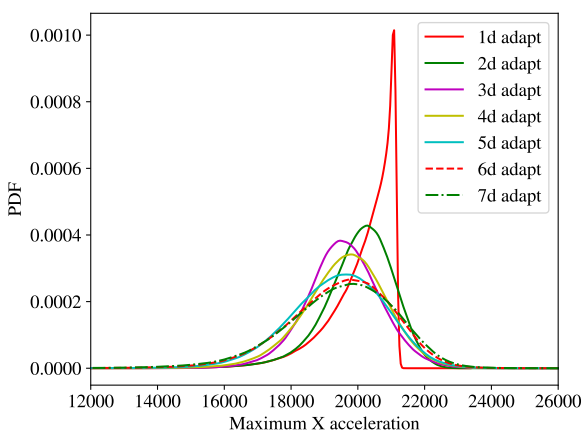
Figure 13: Classical basis adaptation of the space structure.
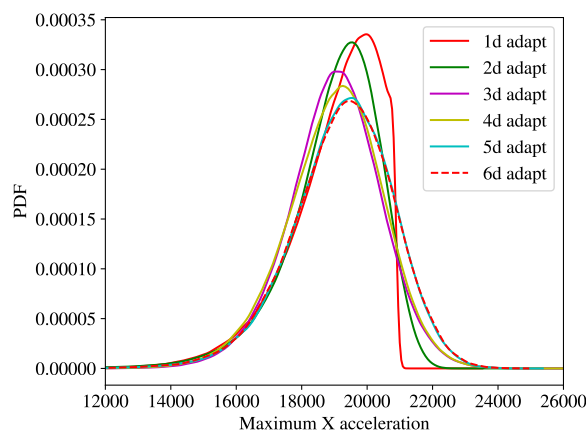
Figure 14: Accelerated basis adaptation method of the space structure.

When model evaluation on specific quadrature points is unavailable, the above method will require much more MC samples to get accurate PCE models. However, the number of samples in practice is usually limited, and no more samples can be generated. Therefore, we can apply the PPA method introduced in this paper to use the available data to solve the forward UQ problem.

As in the borehole model, we first use increasing MC samples in the data set to train converged PPA models. Then, once the PPA models were obtained, we generated many MC samples based on the model and compared the resulting PDF with the reference PDF obtained from the accelerated basis adaptation method. The relative l^2 errors of the PPA model obtained from different numbers of MC samples are shown in Figure 15. In this model, the relative l^2 errors are smaller than 5% when the MC samples are greater than 700 and converge with 1100 MC samples. The number of MC samples required to obtain a converged PPA model is much greater than in the borehole model. Note that in practice, the reference PDF might not be available. In that case, one could compare the PDFs of the PPA model with different MC samples to check if they converge. For most engineering applications, the accuracy of the PPA model with 700 MC samples is good enough. The PDF comparison, in this case, is shown in Figure 16. We see that the PPA PDF is close to the reference with a relative l^2 error of 3.5%, while the PDF generated directly by the data has apparent discrepancies from the reference, the l^2 error of which is 10.3%.

To perform precise analyses, we will choose 1100 MC samples for further analysis. Figures 17 and 18 compare PDFs and CDFs obtained from different models. We see that the resulting PDF of the PPA method is even closer to the reference than the case of using 700 MC samples. The relative l^2 difference between the PPA PDF and the reference PDF has been reduced to 1.9%. The accuracy is improved in all support of the QoI. In addition, the PDF generated directly from the data is different from the reference, especially in the left tail region. Also, comparing the PDF generated from 1100 data in Figure 17 and the PDF generated from 700 data in Figure 16, the data increase only slightly improves the PDF, implying a slow convergence of the MCS. From the CDF figure, the curve associated with the data slightly deviates from the reference, while the result from PPA almost overlaps with the reference.

For this model, we also present the comparison to the PPR model, the additive of univariate PCE models. The same 1100 MC samples are used for PPA and PPR. The results of PDFs and CDFs are shown in Figures 19 and 20. We see that the PPR
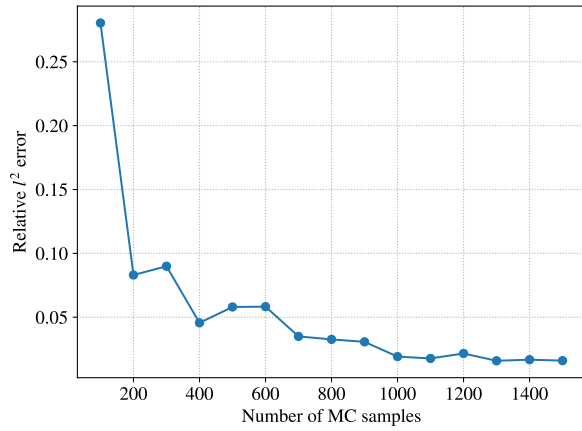
15

Figure 15: The convergence curve of the relative l^2 error of the PDF obtained from PPA with different MC samples for the space structure.
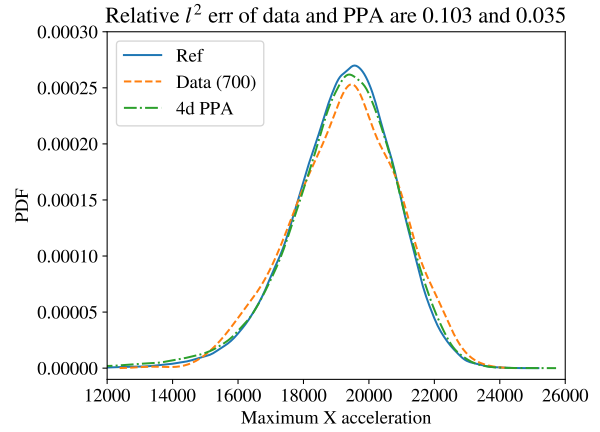

Figure 16: PDF comparison of the PPA method for the space structure. Three PDFs are compared: the reference, the data, and the converged PPA model.
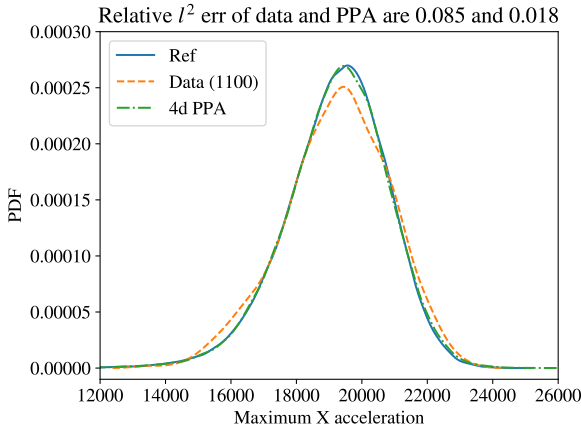

Figure 17: PDF comparison of the PPA method for the space structure. Three PDFs are compared: the reference, the data, and the converged PPA model.
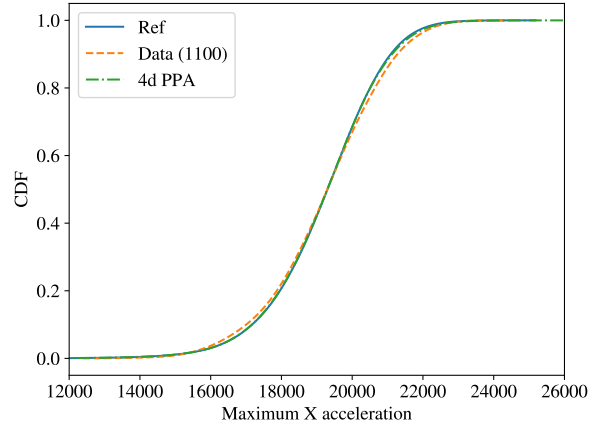

Figure 18: CDF comparison of the PPA method for the space structure. Three CDFs are compared: the reference, the data, and the converged PPA model.

model has improved in the left half of the PDF compared to the data. However, it gives unsatisfactory predictions in the right tail regions. The relative l^2 error of the PPR model is 11.2% compared to 1.8% of the PPA model. From the CDF comparison, we can also clearly see that the PPR model gives inaccurate reliability information on the right tail.

In the case where the PPA model serves as a surrogate model to predict QoI with different input parameters, we generate 200 MC samples that are different from the training data. We can predict the QoI on these 200 samples using the PPA model. The comparison of the prediction and the reference model on the test set is shown in Figure 21. From the left figure, the PPA prediction is close to the reference on most of the test data. From the right figure, although several data have relatively worse predictions, the scatter dots of prediction against the test set are closely aligned in the unit slope line, meaning that the predictions are close to the reference. The relative l^2 difference of the 200 test data is 3.08%, which is close to the relative l^2 error of the PPA PDF compared to the reference PDF. The results suggest that the prediction is accurate.

We see that the PPA model has reduced the dimension from 24 to 4 with only 1100 MC samples. The accuracy of the PPA model is comparable to the models proposed by basis adaptation and accelerated basis adaptation methods. The latter two methods adopted the sparse quadrature rule to compute the PCE coefficients and reduce computational costs. Even in that case, the required quadrature points are 2780 and 1757, respectively, which is still greater than 1100. Another observation is that the reduced dimension discovered by PPA is less than that of the basis adaptation methods, which suggests that the low-dimensional manifold found by PPA is better at revealing the probabilistic information of the QoI.

4.2.2. Multi-QoI capability

Another valuable feature of the data-driven essence of PPA is that the data can be reused for different QoIs. One of the limitations of the basis adaptation methods is that each rotation matrix is associated with a scalar QoI. If the QoI is changed, the
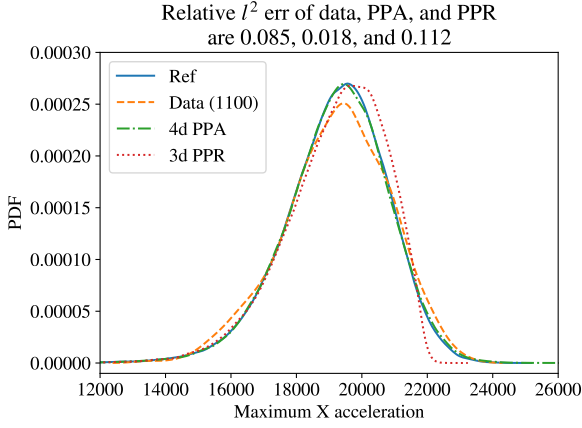
16

Figure 19: PDF comparison of the PPA method for the space structure. Three PDFs are compared: the reference, the data, the converged PPA model, and the converged PPR model.
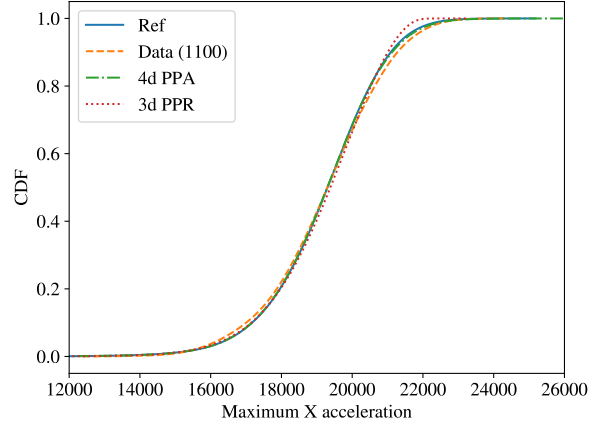
Figure 20: CDF comparison of the PPA method for the space structure. Three CDFs are compared: the reference, the data, the converged PPA model, and the converged PPR model.
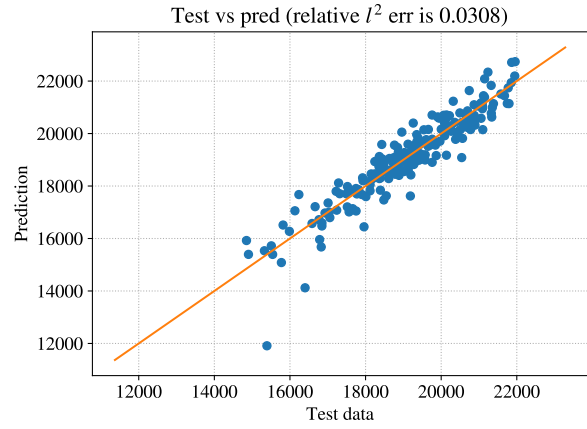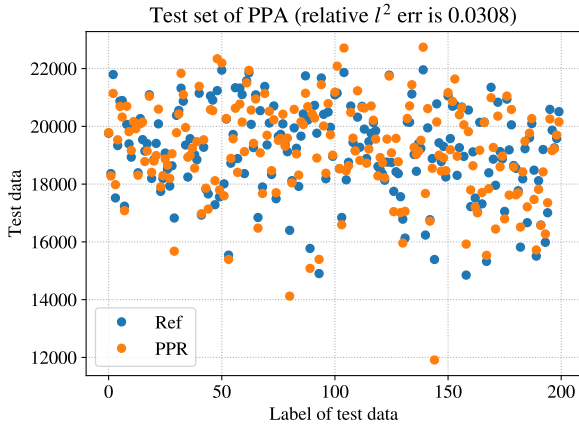

Figure 21: Test data (200) of the PPA method for the space structure. Figure on left shows the QoI value comparison of reference and prediction on various test data. Figure on the right plot the prediction against the test data where the solid line is of slope 1.

whole procedure must be repeated to discover a low-dimensional space for the new QoI. This means that new model evaluations are required to compute the PCE coefficients of the newly adapted model. In the PPA, however, the data we are starting from can be reused to train a new model associated with any new QoI. The only additional computation lies in the training process, which is usually much easier, especially when a low-dimensional manifold is discovered. To test its capability, we use the same data and try to train a model for the maximum velocity of the space structure.

In this case, we are not showing the results from basis adaptation and accelerated basis adaptation. However, the accelerated basis adaptation method provides a reference model of the maximum velocity from a 6-dimensional PCE.

Starting from 100 MC samples in the data set, we gradually increase more MC samples. First, a converged PPA model is trained for each data set expressed in PCE. Then, we generate many MC samples based on the PPA model and obtain its PDF by KDE. Finally, the PDFs of the PPA from the data with different samples are compared to the reference PDF. Then, a convergence curve of the relative l^2 error can be obtained; see Figure 22. For the maximum velocity, the PPA model has an l^2 error close to 5% when the number of samples in the training data set is 800. Again, in most engineering applications, the accuracy is already enough. The PDF comparison, in this case, is shown in Figure 23. Clearly, the PDF generated by the PPA model is close to the reference with an l^2 error of 5.4%, while the PDF generated from the data directly has an l^2 error of 19.2%, a large discrepancy from the reference.

To perform precise analyses, we chose 1200 MC samples in the data set (where the PPA model is converged) for further analysis. The PDF and CDF comparisons of different models are shown in Figures 24 and 25. We see that the PDF obtained from the PPA model is even closer to the reference than when 800 data are used. The relative l^2 error has decreased from 5.4% to 2.7%. However, for the PDF obtained from the data, the l^2 error barely changes from the 800 data case to the 1200 data case. A significant discrepancy still exists even if the number of samples has increased. This signifies the slow convergence of the
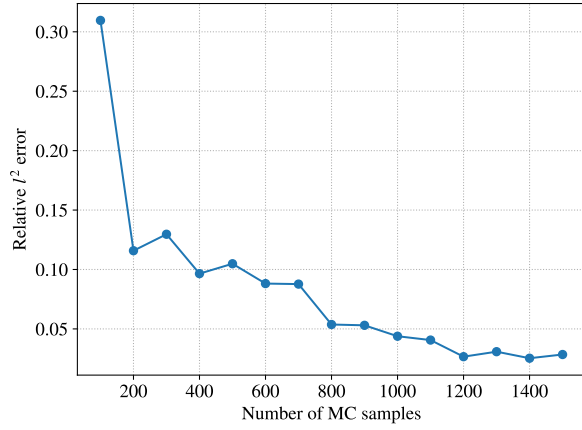
Figure 22: The convergence curve of the relative l^2 error of the PDF obtained from PPA with different MC samples for the space structure when QoI is the maximum velocity.
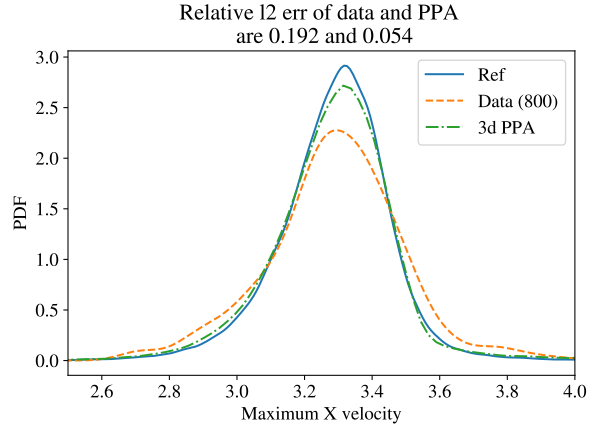

Figure 23: PDF comparison of the PPA method for the space structure when QoI is the maximum velocity. Three PDFs are compared: the reference, the data, and the converged PPA model.
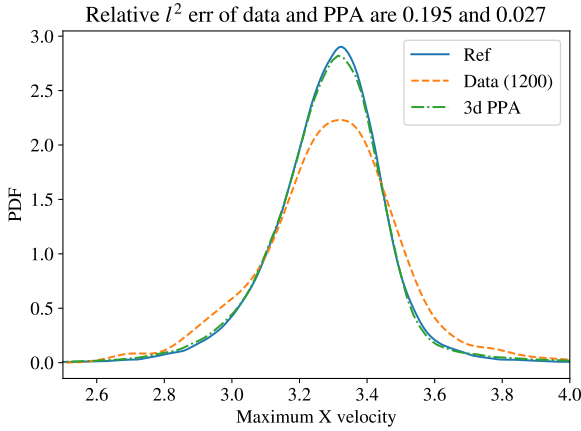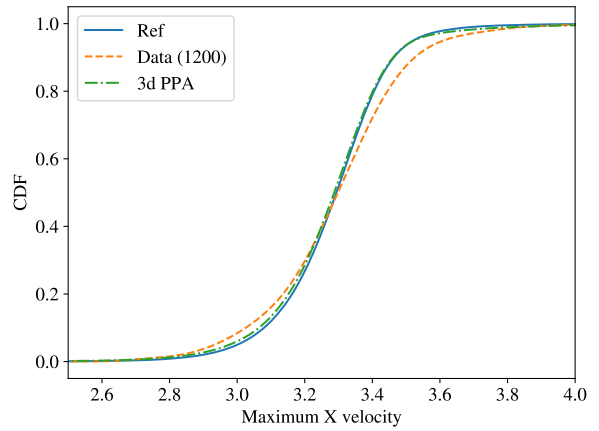

Figure 24: PDF comparison of the PPA method for the space structure when QoI is the maximum velocity. Three PDFs are compared: the reference, the data, and the converged PPA model.

Figure 25: CDF comparison of the PPA method for the space structure when QoI is the maximum velocity. Three CDFs are compared: the reference, the data, and the converged PPA model.

traditional MC methods. From Figure 25, we also see that the CDF of the PPA model is consistent with the reference, while the CDF from the data is far from the reference.

Again, we can use the PPA model as a surrogate to predict QoI with different input parameters. For example, we generate 200 MC samples that differ from the training data as the test set. We can predict the QoI on these 200 samples using the PPA model. The comparison of the prediction and the reference model on the test set is shown in Figure 26. From the left figure, the PPA prediction is close to the reference on most of the test data. This can be seen more evident in the right figure, where the scatter dots of prediction against the test set are closely aligned in the unit slope line, meaning that the predictions are close to the reference. The relative l^2 difference of the 200 test data is 2.9%, which is close to the relative l^2 error of the PPA PDF compared to the reference PDF. The results suggest that the prediction is accurate.

5. Conclusions

The paper proposed a novel, highly accurate, and efficient surrogate model for high-dimensional computational models and high-dimensional data. Specifically, we proposed a data-driven projection pursuit adaptation method (PPA) that can discover optimal low-dimensional space and polynomial chaos expansions (PCE) to represent the quantity of interest (QoI) accurately. The method has the following significances. First, the dimension reduction capability and optimal PCE representation are based on a-priori-specified data. No additional model evaluations are required. Second, the method can discover even a lower-dimensional localization than other techniques, using even less data. Third, the required data are independent samples, which have less restriction and are relatively easier to obtain than, for example, quadrature points. Lastly, the method has the multi-QoI capability as the same data set can be reused to learn different representations for different QoIs.
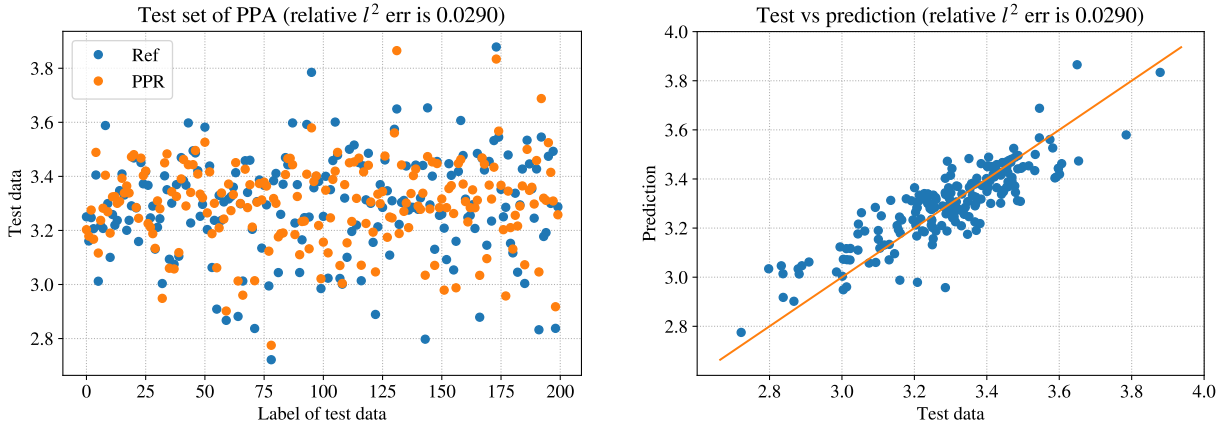
Figure 26: Test data (200) of the PPA method for the space structure when QoI is the maximum velocity. Figure on left shows the QoI value comparison of reference and prediction on various test data. Figure on the right plot the prediction against the test data where the solid line is of slope 1.

The method is developed for uncertainty quantification (UQ) and prediction. In UQ, the basis adaptation method in PCE and its accelerated algorithm present the potential to serve as a general dimension reduction technique. Although versatile, sparse quadrature is often used to construct the PCE models to reduce the computational cost, which requires evaluating the physical model on specific quadrature points. The projection pursuit regression (PPR) model approximates the response for high-dimensional data by a sum of smooth functions of some projected variables. The optimal projections and the smooth functions are found simultaneously from the given data. However, the convergence and capability to represent any variable in the admissible space in PPR depend on the choice of smooth functions. In this paper, we combine the advantages of these two methods and propose the novel PPA method for surrogate modeling of high-dimensional models or data. Specifically, we use multivariate PCE as a smooth function and approximate the response by a multivariate PCE on the projected variables. A stage-wise greedy strategy determines the number of projections, where we simultaneously compute the optimal projection and the optimal multivariate PCE at each stage and stop the procedure if additional information does not improve the approximation significantly. The PPA model inherits the convergence property and the capability to represent any variable in the admissible space from PCE. The applications of a borehole model and a space structure model demonstrate that the PPA model can find low-dimensional spaces to represent the responses accurately. In addition, the number of samples required to obtain accurate results in PPA is even less than the accelerated basis adaptation, where the latter uses sparse quadrature points. The space structure example also shows that the method can be applied to multiple QoIs using the same data set.

Acknowledgments

References

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

[2] R. Ghanem, J. Red-Horse, Propagation of probabilistic uncertainty in complex physical systems using a stochastic finite element approach, Physica D: Nonlinear Phenomena 133 (1999) 137–144.

[3] B. Sudret, Uncertainty propagation and sensitivity analysis in mechanical models–contributions to structural reliability and stochastic spectral methods, Habilitationa diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France 147 (2007) 53.

[4] H. N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Annual review of fluid mechanics 41 (2009) 35–52.

[5] C. Soize, Uncertainty quantification, Springer, 2017.

[6] J. Chen, X. Zeng, Y. Peng, Probabilistic analysis of wind-induced vibration mitigation of structures by fluid viscous dampers, Joural of Sound and Vibration 409 (2017) 287–305.

[7] X. Zeng, Y. Peng, J. Chen, Serviceability-based damping optimization of randomly wind-excited high-rise buildings, The structural design of tall and special buildings 26 (2017) e1371.

[8] Y. M. Marzouk, H. N. Najm, L. A. Rahn, Stochastic spectral methods for efficient Bayesian solution of inverse problems, Journal of Computational Physics 224 (2007) 560–586.

[9] Y. M. Marzouk, H. N. Najm, Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, Journal of Computational Physics 228 (2009) 1862–1902.

[10] Y. Marzouk, D. Xiu, A stochastic collocation approach to Bayesian inference in inverse problems (2009).

[11] B. Sudret, S. Marelli, J. Wiart, Surrogate models for uncertainty quantification: An overview, in: 2017 11th European conference on antennas and propagation (EUCAP), IEEE, 2017, pp. 793–797.

[12] R. Ghanem, Ingredients for a general purpose stochastic finite elements implementation, Computer Methods in Applied Mechanics and Engineering 168 (1999) 19–34.

[13] D. Xiu, G. E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM journal on scientific computing 24 (2002) 619–644.

[14] D. Xiu, G. E. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Computer methods in applied mechanics and engineering 191 (2002) 4927–4948.

[15] D. J. MacKay, et al., Introduction to gaussian processes, NATO ASI series F computer and systems sciences 168 (1998) 133–166.

[16] M. Seeger, Gaussian processes for machine learning, International journal of neural systems 14 (2004) 69–106.

[17] I. Bilionis, N. Zabaras, Multi-output local gaussian process regression: Applications to uncertainty quantification, Journal of Computational Physics 231 (2012) 5718–5746.

[18] J. Li, J. Chen, Stochastic Dynamics of Structures, John Wiley & Sons, 2009.

[19] J. Chen, J. Yang, J. Li, A GF-discrepancy for point selection in stochastic seismic response analysis of structures with uncertain parameters, Structural Safety 59 (2016) 20–31.

[20] C. Soize, R. Ghanem, Probabilistic learning on manifolds constrained by nonlinear partial differential equations for small datasets, Computer Methods in Applied Mechanics and Engineering 380 (2021) 113777.

[21] H. Zhang, J. Guilleminot, L. J. Gomez, Stochastic modeling of geometrical uncertainties on complex domains, with application to additive manufacturing and brain interface geometries, Computer Methods in Applied Mechanics and Engineering 385 (2021) 114014.

[22] D. G. Giovanis, M. D. Shields, Data-driven surrogates for high dimensional models using gaussian process regression on the grassmann manifold, Computer Methods in Applied Mechanics and Engineering 370 (2020) 113269.

[23] I. Kougioumtzoglou, P. Spanos, An analytical wiener path integral technique for non-stationary response determination of nonlinear oscillators, Probabilistic Engineering Mechanics 28 (2012) 125–131.

[24] A. F. Psaros, I. Petromichelakis, I. A. Kougioumtzoglou, Wiener path integrals and multi-dimensional global bases for non-stationary stochastic response determination of structural systems, Mechanical Systems and Signal Processing 128 (2019) 551–571.

[25] 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.

[26] O. Le Maitre, O. M. Knio, Spectral methods for uncertainty quantification: with applications to computational fluid dynamics, Springer Science & Business Media, 2010.

[27] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Doklady Akademii Nauk 4 (1963) 240–243.

[28] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical algorithms 18 (1998) 209–232.

[29] E. Novak, K. Ritter, Simple cubature formulas with high polynomial exactness, Constructive approximation 15 (1999) 499–522.

[30] 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.

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

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

[33] 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) 63–93.

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

[35] P. G. 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.

[36] P. G. Constantine, Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies, volume 2, SIAM, 2015.

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

[38] C. Thimmisetty, P. Tsilifis, R. Ghanem, Homogeneous chaos basis adaptation for design optimization under uncertainty: Application to the oil well placement problem, Ai Edam 31 (2017) 265–276.

[39] Z. G. Ghauch, V. Aitharaju, W. R. Rodgers, P. Pasupuleti, A. Dereims, R. G. Ghanem, Integrated stochastic analysis of fiber composites manufacturing using adapted polynomial chaos expansions, Composites Part A: Applied Science and Manufacturing 118 (2019) 179–193.

[40] X. Zeng, J. Red-Horse, R. Ghanem, Accelerated basis adaptation in homogeneous chaos spaces, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114109.

[41] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.

[42] K. P. Murphy, Machine learning: a probabilistic perspective, MIT press, 2012.

[43] T. Howley, M. G. Madden, M.-L. O'Connell, A. G. Ryder, The effect of principal component analysis on machine learning accuracy with high dimensional spectral data, in: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2005, pp. 209–222.

[44] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, T. Baker, Analysis of dimensionality reduction techniques on big data, IEEE Access 8 (2020) 54776–54788.

[45] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[46] T. Lin, H. Zha, Riemannian manifold learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 796–809.

[47] K. Han, Y. Wang, C. Zhang, C. Li, C. Xu, Autoencoder inspired unsupervised feature selection, in: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2018, pp. 2941–2945.

[48] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML workshop on unsupervised and transfer learning, JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.

[49] J. H. Friedman, J. W. Tukey, A projection pursuit algorithm for exploratory data analysis, IEEE Transactions on computers 100 (1974) 881–890.

[50] P. J. Huber, Projection pursuit, The annals of Statistics (1985) 435–475.

[51] J. H. Friedman, Exploratory projection pursuit, Journal of the American statistical association 82 (1987) 249–266.

[52] E.-K. Lee, D. Cook, S. Klinke, T. Lumley, Projection pursuit for exploratory supervised classification, Journal of Computational and graphical Statistics 14 (2005) 831–846.

[53] M. Grochowski, W. Duch, Projection pursuit constructive neural networks based on quality of projected clusters, in: International Conference on Artificial Neural Networks, Springer, 2008, pp. 754–762.

[54] A. Barcaru, Supervised projection pursuit–a dimensionality reduction technique optimized for probabilistic classification, Chemometrics and Intelligent Laboratory Systems 194 (2019) 103867.

[55] T. Grear, C. Avery, J. Patterson, D. J. Jacobs, Molecular function recognition by supervised projection pursuit machine learning, Scientific reports 11 (2021) 1–15.

[56] A. Olivier, M. D. Shields, L. Graham-Brady, Bayesian neural networks for uncertainty quantification in data-driven materials modeling, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114079.

[57] L. Yang, X. Meng, G. E. Karniadakis, B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data, Journal of Computational Physics 425 (2021) 109913.

[58] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, S. Wahl, Leveraging uncertainty information from deep neural networks for disease detection, Scientific reports 7 (2017) 1–14.

[59] J. H. Friedman, W. Stuetzle, Projection pursuit regression, Journal of the American statistical Association 76 (1981) 817–823.

[60] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, volume 2, Springer, 2009.

[61] G. Qianjian, Y. Jianguo, Application of projection pursuit regression to thermal error modeling of a cnc machine tool, The International Journal of Advanced Manufacturing Technology 55 (2011) 623–629.

[62] F. Ferraty, A. Goia, E. Salinelli, P. Vieu, Functional projection pursuit regression, Test 22 (2013) 293–320.

[63] M. Durocher, F. Chebana, T. B. Ouarda, A nonlinear approach to regional flood frequency analysis using projection pursuit regression, Journal of Hydrometeorology 16 (2015) 1561–1574.

[64] H.-Y. Cui, Y. Zhao, Y.-N. Chen, X. Zhang, X.-Q. Wang, Q. Lu, L.-M. Jia, Z.-M. Wei, Assessment of phytotoxicity grade during composting based on eem/parafac combined with projection pursuit regression, Journal of hazardous materials 326 (2017) 10–17.

[65] S. Janson, et al., Gaussian Hilbert Spaces, Cambridge University Press, 1997.

[66] A. Cohen, G. Migliorati, Optimal weighted least-squares methods, The SMAI journal of computational mathematics 3 (2017) 181–203.

[67] C. Schwab, C. J. Gittelson, Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs, Acta Numerica 20 (2011) 291–467.

[68] M. Rosenblatt, Remarks on a multivariate transformation, The annals of mathematical statistics 23 (1952) 470–472.

[69] P. Tsilifis, R. G. Ghanem, Reduced Wiener chaos representation of random fields via basis adaptation and projection, Journal of Computational Physics 341 (2017) 102–120.