Journal of Computational Physics 230 (2011) 7311–7331

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Kernel principal component analysis for stochastic input model generation

Xiang Ma, Nicholas Zabaras *

Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA

ARTICLE INFO

Article history:
Received 17 August 2010
Received in revised form 18 May 2011
Accepted 31 May 2011
Available online 5 July 2011

Keywords:
Stochastic partial differential equations
Data-driven models
Kernel principal component analysis
Non-linear model reduction
Flow in random porous media

ABSTRACT

Stochastic analysis of random heterogeneous media provides useful information only if realistic input models of the material property variations are used. These input models are often constructed from a set of experimental samples of the underlying random field. To this end, the Karhunen–Loève (K–L) expansion, also known as principal component analysis (PCA), is the most popular model reduction method due to its uniform mean-square convergence. However, it only projects the samples onto an optimal linear subspace, which results in an unreasonable representation of the original data if they are non-linearly related to each other. In other words, it only preserves the first-order (mean) and second-order statistics (covariance) of a random field, which is insufficient for reproducing complex structures. This paper applies kernel principal component analysis (KPCA) to construct a reduced-order stochastic input model for the material property variation in heterogeneous media. KPCA can be considered as a nonlinear version of PCA. Through use of kernel functions, KPCA further enables the preservation of higher-order statistics of the random field, instead of just two-point statistics as in the standard Karhunen–Loève (K–L) expansion. Thus, this method can model non-Gaussian, non-stationary random fields. In this work, we also propose a new approach to solve the pre-image problem involved in KPCA. In addition, polynomial chaos (PC) expansion is used to represent the random coefficients in KPCA which provides a parametric stochastic input model. Thus, realizations, which are statistically consistent with the experimental data, can be generated in an efficient way. We showcase the methodology by constructing a low-dimensional stochastic input model to represent channelized permeability in porous media.

1. Introduction

Over the past few decades there has been considerable interest among the scientific community in studying physical processes with stochastic inputs. These stochastic input conditions arise from uncertainties in boundary and initial conditions as well as from inherent random material heterogeneities. Material heterogeneities are usually difficult to quantify since it is physically impossible to know the exact property at every point in the domain. In most cases, only a few statistical descriptors of the property variation or only a set of samples can be experimentally determined. This limited information necessitates viewing the property variation as a random field that satisfies certain statistical properties/correlations, which naturally results in describing the physical phenomena using stochastic partial differential equations (SPDEs).

In the past few years, several numerical methods have been developed to solve SPDEs, such as Monte Carlo (MC) method, perturbation approach [1,2], generalized polynomial chaos expansion (gPC) [3–6] and stochastic collocation method [7–13].

* Corresponding author. Fax: +1 607 255 1222.
E-mail address: zabaras@cornell.edu (N. Zabaras).
URL: http://mpdc.mae.cornell.edu/ (N. Zabaras).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.05.037

However, implicit in all these developments is the assumption that the uncertainties in the SPDEs have been accurately characterized as random variables or random fields through the finite-dimensional noise assumption [11]. The most common choice is the Karhunen–Loève (K–L) expansion [2,3,14,15], which is also known as linear principal component analysis or PCA [16]. Through K–L expansion, the random field can be represented as a linear combination of the deterministic basis functions (eigenfunctions) and the corresponding uncorrelated random variables (random coefficients). The computation of the K–L expansion requires the analytic expression of the covariance matrix of the underlying random field. In addition, the probability distribution of the random variables is always assumed known *a priori*. These two assumptions are obviously not feasible in realistic engineering problems. In most cases, only a few experimentally obtained realizations of the random field are available. Reconstruction techniques are then applied to generate samples of the random field after extracting the statistical properties of the random field through these limited experimental measurements. These processes are quite expensive and numerically demanding if thousands of samples are needed. This leads to the problem of probabilistic model identification or stochastic input model generation, where the purpose is to find a parametric representation of the random filed through only limited experimental data.

To this end, a polynomial chaos (PC) representation of the random field through experimental data was first proposed in [17] and improved in subsequent papers [18–23]. This framework consists of three steps: (1) computing the covariance matrix of the data numerically using the available experimental data; (2) stochastic reduced-order modeling with a K–L expansion to obtain a set of deterministic basis (eigenfunctions) and the corresponding random expansion coefficients (called K–L projected random variables); (3) a polynomial chaos representation of these random coefficients is constructed given the realizations of these coefficients which are calculated from data. These realizations are then used for the estimation of the deterministic coefficients in the PC representation, where several methods have been proposed. In the pioneering work [17], maximum likelihood estimation was used to find the PC representation of the K–L random variables. In [18], a Bayesian inference approach was used instead to construct the PC representation of the random field. However, these two works did not take into account the dependencies between various components of the K–L projected random variables. In [19], the Rosenblatt transformation [24] was used to capture these dependencies and maximum entropy approach together with Fisher information matrix was used for the estimation of the PC coefficients. In [20,21,25], a non-intrusive projection approach with Rosenblatt transformation was developed for the PC estimation. Apart from PC representation, in [26], the distribution of the K–L random variables was assumed directly to be uniform within the range of the realizations of these coefficients from data. Later, in [27], the distribution of the K–L random variables was still assumed to be uniform. However, the range of the uniform distribution was found through enforcing the statistical constraints of the random field and solving the resulting optimization problems. In [28], the uncertain input parameters are modeled as independent random variables, whose distributions are estimated using a diffusion-mixed-based estimator. Except the work in [28], all the previous developments rely heavily on the K–L expansion for the reduced-order modeling. However, the K–L expansion has one major drawback. The K–L expansion based stochastic model reduction scheme constructs the *closest linear subspace* of the high-dimensional input space. In other words, it only preserves the mean and covariance of the random field, and therefore is suitable for multi-Gaussian random fields. But most of the random samples contain essential non-linear structures, e.g. higher-order statistics. This directly translates into the fact that the standard linear K–L expansion tends to over-estimate the actual intrinsic dimensionality of the underlying random field. Hence, one needs to go beyond a linear representation of the high-dimensional input space to accurately access the effect of its variability on the output variables.

To resolve this issue, the authors in [29] proposed a nonlinear model reduction strategy for generating data-driven stochastic input models. This method is based on the manifold learning method, where multidimensional scaling (MDS) is utilized to map the space of viable property variations to a low-dimensional region. Then isometric mapping from this high-dimensional space to a low-dimensional, compact, connected set is constructed via preserving the geodesic distance between data using the IsoMap algorithm [30]. However, this method has two major issues. First, after dimensionality reduction, it only gives us a set of low-dimensional points. It does not give us the inherent patterns (similar to the eigenfunctions as in the K–L expansion) in the embedded random space. Therefore, it cannot provide us a mathematical parametric input model as in the K–L expansion. Here, we are interested in a representation of the form $\mathbf{y} = f(\xi)$, where the vector $\mathbf{y}$ is a realization of a discrete random field, and the vector ξ, of dimension much smaller than the original input stochastic dimension, is a set of independent random variables with a specified distribution. In addition, when new experimental data becomes available, this nonlinear mapping needs to be recomputed. Secondly, the IsoMap algorithm requires the computation of the geodesic distance matrix among data. In general, this matrix may be not well defined for real data. Even if it is well defined, the algorithm is computationally expensive.

Both problems of the K–L expansion and the non-linear model reduction algorithm in [29] can be solved with kernel principal component analysis (KPCA), which is a nonlinear form of PCA [31,32]. KPCA has proven to be a powerful tool as a nonlinear feature extractor of classification algorithm [31], pattern recognization [33], image-denosing [34] and statistical shape analysis [35]. The basic idea is to map the data in the input space to a feature space F through some nonlinear map Φ, and then apply the linear PCA there. Through the use of a suitably chosen kernel function, the data becomes more linearly related in the feature space F. In the context of stochastic modeling, there are two pioneering works [36,37]. In [36], KPCA was used to construct the prior model of the unknown permeability field and then gradient-based algorithm is used to solve the history-matching problem. However, the random expansion coefficients of the linear PCA in the feature space are assumed i.i.d. standard normal random variables. This choice clearly does not capture the statistical information from the data. In [37], KPCA is used in the feature space F for the selection of a subset of representative realizations containing similar properties to the larger set.

Motivated by all the above mentioned works, in this paper, a stochastic non-linear model reduction framework based on KPCA is developed. To be specific, the KPCA is first used to construct the stochastic reduced-order model in the feature space. The random coefficients of the linear PCA are then represented via PC expansion. Because the K–L expansion is performed in the feature space, the resulting realizations lie in the feature space, and therefore, the mapping from the feature space back to the input space is needed. This is called the "pre-image problem" [34,38]. The pioneering work in solving the pre-image problem is Mika's fixed-point iterative optimization algorithm [34]. However, this method suffers from numerical instabilities. It is sensitive to the initial guess and is likely to get stuck in local minima. Therefore, it is not suitable for our stochastic modeling. It is noted that this algorithm was also used in the work [36,37] to find the pre-image. In [38], the authors determine a relationship between the distances in the feature space and the distances in the input space. The MDS is used to find the inverse mapping and thus the pre-image. This method is expensive since it involves a singular value decomposition on a matrix of nearest neighbors. Recently [39], it was shown that Euclidean distances in the feature space correspond to field-dependent distances in the input space. In this work, we propose a new approach to find the pre-image. It is based on local linear interpolation among n-nearest neighbors using only the distances in the input space, which is similar to the method proposed in our earlier work [29].

This paper is organized as follows: In the next section, the mathematical framework of KPCA is considered. In Section 3, the PC representation of the random coefficients is developed. In Section 4, the new pre-image algorithm is introduced. An Example with channelized permeability is given in Section 5. Finally, concluding remarks are provided in Section 6.

2. Kernel principal component analysis of random fields

2.1. Problem definition

Let us define a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with sample space Ω which corresponds to the outcomes of some experiments, $\mathcal{F} \subset 2^{\Omega}$ is the σ-algebra of subsets in Ω and $\mathcal{P} : \mathcal{F} \rightarrow [0,1]$ is the probability measure. Also, let us define D as a two-dimensional bounded domain. Denote $\mathbf{a}(\boldsymbol{x},\omega)$ the random field (property) used to describe and provide a mathematical model of the available experimental data. The random field $\mathbf{a}(\boldsymbol{x},\omega)$ in general belongs to an infinite-dimensional probability space. However, in most cases, the random field is associated with a spatial discretization. Thus we can have a finite-dimensional representation of the random field which can be represented/described as a random vector $\mathbf{y} := (y_1, \ldots, y_M)^T : \Omega \rightarrow \mathbb{R}^M$. M can be regarded as the number of grid blocks in a discretized model. So each y_i, $i = 1,\ldots,M$ is a random variable which represents the random property in each grid block. The dimensionality of the stochastic model is then the length M of this vector. Let $\mathbf{y}_i$, $i = 1,\ldots,N$ be N real column vectors in $\mathbb{R}^M$, i.e. $\mathbf{y}_i \in \mathbb{R}^M$, representing N independent realizations of the random field.

In most cases, M would be a large number. Our problem is to find a reduced-order polynomial chaos representation of this random field that is consistent with the data in some statistical sense. To be specific, we want to find a form $\mathbf{y} = f(\xi)$, where vector ξ, of dimension much smaller than the original input stochastic dimension M, is a set of independent random variables with a specified distribution. Therefore, by drawing samples ξ in this low-dimensional space, we obtain different realizations of the underlying random field.

2.2. Basic idea of KPCA

Fig. 1 demonstrates the basic idea behind nonlinear kernel PCA. Consider a random field $\mathbf{y} = (y_1, y_2)^T \in \mathbb{R}^2$. If $\mathbf{y}$ is non-Gaussian, y_1 and y_2 can be nonlinearly related to each other in $\mathbb{R}^2$. In this case, linear PCA or K–L expansion attempt to fit

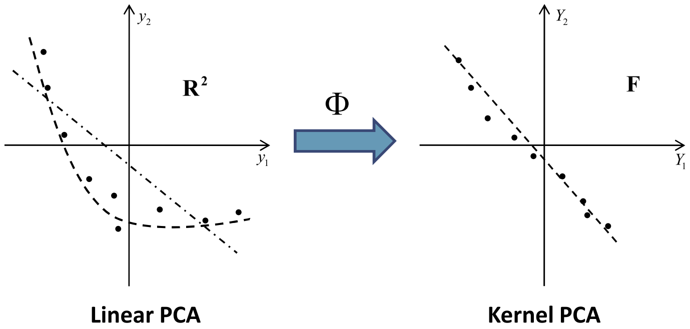

Fig. 1. Basic idea of KPCA. Left: in this non-Gaussian case, the linear PCA is not able to capture the nonlinear relationship among the realizations in the original space. Right: after the nonlinear mapping Φ, the realizations becomes linearly related in the feature space F. Linear PCA or K–L expansion can now be performed in F.

a linear surface such that the reconstruction error is minimized (Fig. 1, left). This clearly results in a poor representation of the original data. Now, consider a nonlinear mapping Φ that relates the input space $\mathbb{R}^M$ to another space F

$$\Phi : \mathbb{R}^M \to F, \quad \mathbf{y} \mapsto \mathbf{Y}. \tag{1}$$

We will refer F as the feature space. In the right figure of Fig. 1, the realizations that are nonlinearly related in $\mathbb{R}^2$ become linearly related in the feature space F. Linear PCA or K–L expansion can now performed in F in order to determine the principal directions in this space.

2.3. PCA in feature space

In this section, the theory of KPCA is briefly reviewed. For a detailed introduction, the authors may refer to [31,38].

As shown before, kernel PCA can be considered to be a generalization of linear PCA in the feature space F. Thus, all results on linear PCA can be readily generalized for KPCA. Now, assume we are given N number of realizations of the random field $\{\mathbf{y}_1,\ldots,\mathbf{y}_N\}$, where each realization is represented as a high-dimensional column vector $\mathbf{y}_i \in \mathbb{R}^M$ (e.g. M can be considered as the number of grid blocks in the discretization). These realizations are mapped in the feature space F to $\Phi(\mathbf{y}_i)$, $i = 1,\ldots,N$. Denote the mean of the Φ-mapped data by $\overline{\Phi} = \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{y}_i)$ and define the "centered" map $\widetilde{\Phi}$ as

$$\widetilde{\Phi} = \Phi(\mathbf{y}) - \overline{\Phi}. \tag{2}$$

Analogous to linear PCA, we need to find eigenvectors $\mathbf{V}$ and eigenvalues λ of the covariance matrix $\mathbf{C}$ in the feature space, where

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(\mathbf{y}_i)\widetilde{\Phi}(\mathbf{y}_i)^T. \tag{3}$$

The dimension of this matrix is $N_F \times N_F$, where N_F is the dimension of the feature space. Standard linear PCA or K–L expansion involves solving the eigenvalue problem $\mathbf{CV} = \lambda\mathbf{V}$. As explained in [31], N_F could be extremely large. As a result, it will be impossible to compute the $\mathbf{C}$ matrix and solve the eigenvalue problem directly.

Thus, as in [31], a kernel eigenvalue problem, which uses only dot products of vectors in the feature space, is formulated to solve the original eigenvalue problem indirectly. The dot product in the feature space is defined as $\left(\widetilde{\Phi}(\mathbf{y}_i)\cdot\widetilde{\Phi}(\mathbf{y}_i)\right) = \widetilde{\Phi}(\mathbf{y}_i)^T\widetilde{\Phi}(\mathbf{y}_i)$. We first substitute the covariance matrix into the eigenvalue problem $\mathbf{CV} = \lambda\mathbf{V}$ and obtain [36]

$$\mathbf{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(\widetilde{\Phi}(\mathbf{y}_i)\cdot\mathbf{V}\right)\widetilde{\Phi}(\mathbf{y}_i), \tag{4}$$

which implies that all solutions $\mathbf{V}$ with $\lambda \neq 0$ lie in the span of $\widetilde{\Phi}(\mathbf{y}_1),\ldots,\widetilde{\Phi}(\mathbf{y}_N)$. Following the mathematical developments presented in [31], we can expand the solution $\mathbf{V}$ as

$$\mathbf{V} = \sum_{j=1}^{N}\alpha_j\widetilde{\Phi}(\mathbf{y}_j), \tag{5}$$

and the eigenvalue problem is equivalent to

$$(\widetilde{\Phi}(\mathbf{y}_i)\cdot\mathbf{CV}) = \lambda(\widetilde{\Phi}(\mathbf{y}_i)\cdot\mathbf{V}), \quad \forall i = 1,\ldots,N. \tag{6}$$

Now, substituting Eqs. (4) and (5) into Eq. (6), we obtain

$$\frac{1}{N}\sum_{j=1}^{N}\alpha_j\sum_{k=1}^{N}\left(\widetilde{\Phi}(\mathbf{y}_i)\cdot\widetilde{\Phi}(\mathbf{y}_k)\right)\left(\widetilde{\Phi}(\mathbf{y}_k)\cdot\widetilde{\Phi}(\mathbf{y}_j)\right) = \lambda\sum_{j=1}^{N}\alpha_j\left(\widetilde{\Phi}(\mathbf{y}_i)\cdot\widetilde{\Phi}(\mathbf{y}_j)\right), \tag{7}$$

for $i = 1,\ldots,N$. Let us define the $N \times N$ kernel matrix $\mathbf{K}$ which is the dot product of vectors in the feature space F, $\mathbf{K} : K_{i\text{-}j} = (\Phi(\mathbf{y}_i)\cdot\Phi(\mathbf{y}_j))$ and the corresponding centered kernel matrix:

$$\widetilde{\mathbf{K}} : \widetilde{K}_{ij} = \left(\widetilde{\Phi}(\mathbf{y}_i)\cdot\widetilde{\Phi}(\mathbf{y}_j)\right). \tag{8}$$

Define the $N \times N$ centering matrix $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, where $\mathbf{I}$ is the $N \times N$ identity matrix and $\mathbf{1} = [11\ldots1]^T$ is a $N \times 1$ vector. Thus, the centered kernel matrix can be computed as $\widetilde{\mathbf{K}} = \mathbf{HKH}$.

Substituting Eq. (8) into Eq. (7), we arrive at the equation

$$\widetilde{\mathbf{K}}^2\boldsymbol{\alpha} = N\lambda_i\widetilde{\mathbf{K}}\boldsymbol{\alpha}, \quad \forall i = 1,\ldots,N, \tag{9}$$

where $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_N]^T$. This is equivalent to the following kernel eigenvalue problem [31]:

$$\widetilde{\mathbf{K}}\boldsymbol{\alpha} = N\lambda_i\boldsymbol{\alpha}, \quad \forall i = 1,\ldots,N. \tag{10}$$

In the following, for simplicity, we will denote λ_i as the eigenvalues of $\widetilde{\mathbf{K}}$, i.e. the solutions $N\lambda_i$ in Eq. (10). We rewrite Eq. (10) in the following matrix form:

$$\widetilde{\mathbf{K}}\mathbf{U} = \mathbf{\Lambda}\mathbf{U}, \tag{11}$$

where, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of the corresponding eigenvalues and $\mathbf{U} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ is the matrix containing the eigenvectors of the kernel matrix $\widetilde{\mathbf{K}}$, where each column consists of the ith eigenvector $\boldsymbol{\alpha}_i = [\alpha_{i1}, \ldots, \alpha_{iN}]^T$.

Therefore, through Eq. (5), the ith eigenvector of the covariance matrix $\mathbf{C}$ in the feature space can be shown to be [31,38]

$$\mathbf{V}_i = \sum_{j=1}^{N} \alpha_{ij} \widetilde{\Phi}(\mathbf{y}_j). \tag{12}$$

Furthermore, the eigenvector $\mathbf{V}_i$ can be normalized. Since the eigenvectors $\boldsymbol{\alpha}_i$ from the eigenvalue problem Eq. (11) are already normalized, e.g. $\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_i = 1$, the ith orthonormal eigenvector of the covariance matrix $\mathbf{C}$ can be shown to be [31,38]

$$\widetilde{\mathbf{V}}_i = \sum_{j=1}^{N} \tilde{\alpha}_{ij} \widetilde{\Phi}(\mathbf{y}_j), \quad \text{where } \tilde{\alpha}_{ij} = \frac{\alpha_{ij}}{\sqrt{\lambda_i}}. \tag{13}$$

It is ease to verify that Eq. (13) satisfies the orthonormal condition $\widetilde{\mathbf{V}}_i \cdot \widetilde{\mathbf{V}}_i = 1$.

Let $\mathbf{y}$ be a realization of the random field, with a mapping $\Phi(\mathbf{y})$ in F. According to the theory of linear PCA, $\Phi(\mathbf{y})$ can be decomposed in the following way:

$$\Phi(\mathbf{y}) = \sum_{i=1}^{N} z_i \widetilde{\mathbf{V}}_i + \overline{\Phi}, \tag{14}$$

where z_i is the projection coefficient onto the ith eigenvector $\widetilde{\mathbf{V}}_i$:

$$z_i = \widetilde{\mathbf{V}}_i \cdot \widetilde{\Phi}(\mathbf{y}) = \sum_{j=1}^{N} \tilde{\alpha}_{ij} \left(\widetilde{\Phi}(\mathbf{y}) \cdot \widetilde{\Phi}(\mathbf{y}_j) \right). \tag{15}$$

2.4. Computing dot products in feature space

From Eq. (8), it is seen that in order to compute the kernel matrix, only the dot products of vectors in the feature space F are required, while the explicit calculation of the map $\Phi(\mathbf{y})$ does not need to be known. As shown in [31], the dot product can be computed through the use of the kernel function. This is the so called “kernel trick”. Not all arbitrary functions but the Mercer kernels can be used as a kernel function [31]. The kernel function $k(\mathbf{y}_i, \mathbf{y}_j)$ calculates the dot product in space F directly from the vectors of the input space $\mathbb{R}^M$:

$$k(\mathbf{y}_i, \mathbf{y}_j) = \left(\Phi(\mathbf{y}_i) \cdot \Phi(\mathbf{y}_j) \right). \tag{16}$$

The commonly used kernel functions are polynomial kernel and Gaussian kernel.

2.4.1. Kernel for linear PCA

If the kernel function is chosen as polynomial kernel of order one

$$k(\mathbf{y}_i, \mathbf{y}_j) = (\mathbf{y}_i \cdot \mathbf{y}_j), \tag{17}$$

then the linear PCA is actually performed on the sample realizations. It is noted that the use of the kernel matrix to perform the linear PCA is actually the same as the “method of snapshots” which is well known in the area of reduced-order modeling [40]. This method is more computationally efficient than the standard implementation of the K–L expansion as in [17]. Using the kernel matrix, only an eigenvalue problem of size $N \times N$ is needed, whereas in [17] the size of the eigenvalue problem is $M \times M$. In most cases, the number of available experimental data is much smaller than the dimensionality of the data itself.

2.4.2. Kernel for nonlinear PCA

Choosing a nonlinear kernel function leads to performing nonlinear PCA. The most common kernel function is the Gaussian kernel:

$$k(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{2\sigma^2} \right), \tag{18}$$

where $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ is the squared L_2-distance between two realizations. The kernel width parameter σ controls the flexibility of the kernel. A larger value of σ allows more “mixing” between elements of the realizations, whereas a smaller value of σ uses only a few significant realizations. As recommended in [35], a typical choice for σ is the average minimum distance between two realizations in the input space:

$$\sigma^2 = c\frac{1}{N}\sum_{i=1}^{N}\min_{j\neq i}\|\mathbf{y}_i-\mathbf{y}_j\|^2,\quad j=1,\ldots,N, \tag{19}$$

where c is a user-controlled parameter.

2.5. Stochastic reduced-order modeling via KPCA

Since only the dot products of vectors in the feature space are available through use of the kernel function, we first rewrite Eq. (15) in terms of the kernel function [38]:

$$z_i = \sum_{j=1}^{N}\tilde{\alpha}_{ij}\left(\widetilde{\Phi}(\mathbf{y})\cdot\widetilde{\Phi}(\mathbf{y}_j)\right) = \sum_{j=1}^{N}\tilde{\alpha}_{ij}\tilde{k}(\mathbf{y},\mathbf{y}_j), \tag{20}$$

where

$$\begin{aligned}\tilde{k}(\mathbf{y},\mathbf{y}_j) &= \widetilde{\Phi}(\mathbf{y})\cdot\widetilde{\Phi}(\mathbf{y}_j) = (\Phi(\mathbf{y})-\overline{\Phi})\cdot(\Phi(\mathbf{y}_j)-\overline{\Phi}) = k(\mathbf{y},\mathbf{y}_j) - \frac{1}{N}\sum_{i=1}^{N}k(\mathbf{y},\mathbf{y}_i) - \frac{1}{N}\sum_{i=1}^{N}k(\mathbf{y}_i,\mathbf{y}_j) + \frac{1}{N^2}\sum_{p,q=1}^{N}k(\mathbf{y}_p,\mathbf{y}_q)\\ &= k(\mathbf{y},\mathbf{y}_j) - \frac{1}{N}\mathbf{1}^T\mathbf{k}_\mathbf{y} - \frac{1}{N}\mathbf{1}^T\mathbf{k}_{\mathbf{y}_j} + \frac{1}{N^2}\mathbf{1}^T\mathbf{K}\mathbf{1},\end{aligned} \tag{21}$$

with

$$\mathbf{k}_\mathbf{y} = [k(\mathbf{y},\mathbf{y}_1),\ldots,k(\mathbf{y},\mathbf{y}_N)]^T. \tag{22}$$

Let $\tilde{k}_\mathbf{y} = \left[\tilde{k}(\mathbf{y},\mathbf{y}_1),\ldots,\tilde{k}(\mathbf{y},\mathbf{y}_N)\right]^T$. Then using Eq. (21), we can obtain:

$$\tilde{k}_\mathbf{y} = \left[\tilde{k}(\mathbf{y},\mathbf{y}_1),\ldots,\tilde{k}(\mathbf{y},\mathbf{y}_N)\right]^T = \mathbf{k}_\mathbf{y} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{k}_\mathbf{y} - \frac{1}{N}\mathbf{K}\mathbf{1} + \frac{1}{N^2}\mathbf{1}\mathbf{1}^T\mathbf{K}\mathbf{1} = \mathbf{H}\mathbf{k}_\mathbf{y} - \frac{1}{N}\mathbf{H}\mathbf{K}\mathbf{1}. \tag{23}$$

Then Eq. (20) can be written more compactly as

$$z_i = \tilde{\boldsymbol{\alpha}}_i^T\tilde{k}_\mathbf{y} = \tilde{\boldsymbol{\alpha}}_i^T\mathbf{H}\mathbf{k}_\mathbf{y} - \frac{1}{N}\tilde{\boldsymbol{\alpha}}_i^T\mathbf{H}\mathbf{K}\mathbf{1},$$

where $\tilde{\boldsymbol{\alpha}}_i := [\tilde{\alpha}_{i1},\ldots,\tilde{\alpha}_{iN}]^T$. Thus, we can write all the z_i's in a vector form $\mathbf{Z} := [z_1,\ldots,z_N]^T$:

$$\mathbf{Z} = \mathbf{A}^T\mathbf{k}_\mathbf{y} + \mathbf{b}, \tag{24}$$

where $\mathbf{A} = \mathbf{H}\widetilde{\mathbf{U}},\ \mathbf{b} = -\frac{1}{N}\widetilde{\mathbf{U}}^T\mathbf{H}\mathbf{K}\mathbf{1}$ and $\widetilde{\mathbf{U}} = [\tilde{\boldsymbol{\alpha}}_1,\ldots,\tilde{\boldsymbol{\alpha}}_N]$.

Denote $\boldsymbol{\Phi} = [\Phi(\boldsymbol{y}_1),\ldots,\Phi(\boldsymbol{y}_N)]$, which is a matrix of size $N_F \times N$, then we can write:

$$\begin{aligned}\widetilde{\boldsymbol{\Phi}} &= [\widetilde{\Phi}(\mathbf{y}_1),\ldots,\widetilde{\Phi}(\mathbf{y}_N)] = [\Phi(\mathbf{y}_1)-\overline{\Phi},\ldots,\Phi(\mathbf{y}_N)-\overline{\Phi}] = \left[\Phi(\mathbf{y}_1)-\frac{1}{N}\boldsymbol{\Phi}\mathbf{1},\ldots,\Phi(\mathbf{y}_N)-\frac{1}{N}\boldsymbol{\Phi}\mathbf{1}\right]\\ &= \boldsymbol{\Phi}\mathbf{I} - \frac{1}{N}\boldsymbol{\Phi}\mathbf{1}\mathbf{1}^T = \boldsymbol{\Phi}\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right) = \boldsymbol{\Phi}\mathbf{H}.\end{aligned} \tag{25}$$

Thus, Eq. (13) can be rewritten in the following matrix form:

$$\widetilde{\mathbf{V}}_i = \sum_{j=1}^{N}\tilde{\alpha}_{ij}\widetilde{\Phi}(\mathbf{y}_j) = \widetilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}}_i = \boldsymbol{\Phi}\mathbf{H}\tilde{\boldsymbol{\alpha}}_i. \tag{26}$$

Now suppose the eigenvectors are ordered by decreasing eigenvalues and we work in the low-dimensional subspace spanned by the first r largest eigenvectors, where, in general, $r \ll N$. Then the decomposition in Eq. (14) can be truncated after the first r terms as follows:

$$\begin{aligned}\Phi(\mathbf{y}) &= \sum_{i=1}^{N}z_i\widetilde{\mathbf{V}}_i + \overline{\Phi} \approx \sum_{i=1}^{r}z_i\widetilde{\mathbf{V}}_i + \overline{\Phi} = \sum_{i=1}^{r}\boldsymbol{\Phi}\mathbf{H}\tilde{\boldsymbol{\alpha}}_i z_i + \frac{1}{N}\boldsymbol{\Phi}\mathbf{1} = \boldsymbol{\Phi}\left(\mathbf{H}\widetilde{\mathbf{U}}_r\mathbf{Z}_r + \frac{1}{N}\mathbf{1}\right) = \boldsymbol{\Phi}\left(\mathbf{A}_r\mathbf{Z}_r + \frac{1}{N}\mathbf{1}\right) = \boldsymbol{\Phi}\boldsymbol{\beta}\\ &= \sum_{i=1}^{N}\beta_i\Phi(\boldsymbol{y}_i),\end{aligned} \tag{27}$$

where $\boldsymbol{\beta} = \mathbf{A}_r\mathbf{Z}_r + \frac{1}{N}\mathbf{1}$ and β_i is its ith component. It is noted that since only the first r eigenvectors are used, $\widetilde{\mathbf{U}}$ used to calculate $\mathbf{A}_r$ and $\mathbf{Z}_r$ in the above equation only contains the first r columns of the full matrix, i.e. $\widetilde{\mathbf{U}}_r = [\tilde{\boldsymbol{\alpha}}_1,\ldots,\tilde{\boldsymbol{\alpha}}_r]$. Therefore, $\mathbf{A}_r = \mathbf{H}\widetilde{\mathbf{U}}_r$ is a matrix of size $N \times r$ and $\mathbf{Z}_r = [z_1,\ldots,z_r]^T$ is a r-dimensional column vector.

Thus, given N samples from the original stochastic feature space F, we can find an approximate r-dimensional subspace $\widetilde{F}$ of F which is spanned by the orthonormal basis $\widetilde{\mathbf{V}}_i,\ i = 1,\ldots,r$. Similar to K–L expansion, the expansion coefficients $\mathbf{Z}_r$ are a r-dimensional random vector that defines this subspace. By drawing samples of $\mathbf{Z}_r$ from it, we can obtain different realizations of $\Phi(\mathbf{y})$ through Eq. (27). These realizations are expected to capture important statistical information and resemble the inherent pattern of the available samples as much as possible thanks to the properties of the K–L expansion used in the feature space. In other words, the stochastic reduced-order input model in the feature space can be defined as: for any realization $\mathbf{Y} \in F$, we have

$$\mathbf{Y}_r = \sum_{i=1}^{N} \beta_i \Phi(\mathbf{y}_i) = \mathbf{\Phi}\boldsymbol{\beta}, \quad \text{with } \boldsymbol{\beta} = \mathbf{A}_r \boldsymbol{\xi} + \frac{1}{N}\mathbf{1}. \tag{28}$$

Here, the subscript r is to emphasize that the realization $\mathbf{Y}_r$ is reconstructed using only the first r eigenvectors. $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_r]^T$ is a r-dimensional random vector. If the probability distribution of $\boldsymbol{\xi}$ is known, we can then sample $\boldsymbol{\xi}$ and obtain samples of the random field in F.

However, the probability distribution of ξ_i is not known to us. What we know is only the realizations of these random coefficients ξ_i, which can be obtained through Eq. (24) by using the available samples:

$$\boldsymbol{\xi}^{(i)} = \mathbf{A}_r^T \mathbf{k}_{\mathbf{y}_i} + \mathbf{b}_r, \quad i = 1, \ldots, N. \tag{29}$$

Our problem then reduces to identify the probability distribution of the random vector $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_r]^T$, given its N samples $\boldsymbol{\xi}^{(i)} = [\xi_1^{(i)}, \ldots, \xi_r^{(i)}], i = 1, \ldots, N$. A polynomial chaos representation is introduced in the next section for representing each component of the random vector $\boldsymbol{\xi}$ in terms of another random vector with known distribution.

Finally, according to the properties of the K–L expansion [3,15,21] used in the feature space, the random vector $\boldsymbol{\xi}$ satisfies the following two conditions:

$$E[\xi_i] = 0, \quad E[\xi_i \xi_j] = \delta_{ij}\frac{\lambda_i}{N}, \quad i,j = 1, \ldots, r. \tag{30}$$

Therefore, the random coefficients ξ_i are uncorrelated but not independent. In addition, similarly to the K–L expansion, the minimum mean squared error (MSE) of truncated expansion Eq. (28) in the feature space can be shown to be

$$E\left[\|\mathbf{Y} - \mathbf{Y}_r\|^2\right] = \frac{1}{N}\sum_{i=r+1}^{N} \lambda_i, \tag{31}$$

where the norm is defined as the L_2 norm or dot product in the feature space.

Remark 1. For linear PCA, we only need to replace the kernel function in Eq. (24) with the kernel function for linear PCA Eq. (17) and replace $\Phi(\mathbf{y}_i)$ in Eq. (28) with the data $\mathbf{y}_i$ directly.

3. Polynomial chaos representation of the stochastic reduced-order model

The problem is now to identify a random vector $\boldsymbol{\xi} : \widetilde{F} \to \mathbb{R}^r$, given N independent samples $\left\{\boldsymbol{\xi}^{(i)}\right\}_{i=1}^{N}$. For this purpose, we use a polynomial chaos (PC) expansion to represent $\boldsymbol{\xi}$. Several methods have been proposed to compute the expansion coefficients in the resulted PC expansion, such as maximum likelihood [17], Bayesian inference [18], maximum entropy method [19] and non-intrusive projection method [21,25]. As mentioned before, the components of random vector $\boldsymbol{\xi}$ are uncorrelated but not independent. Although Rosenblatt transformation can be used to reduce the problem to a set of independent random variables [20,25], it is computationally expensive for high-dimensional problems. In this work, to reduce the computational cost, we further assume the independence between the components of $\boldsymbol{\xi}$. This is generally not the case in the representation of arbitrary random fields. However, in the works [18,21], it has been numerically verified that this strong hypothesis gives accurate results.

3.1. PC expansion of random variables

The theory and properties of the PC expansion have been well documented in various references [3–5]. In this approach, any random variable with finite variance can be expanded in terms of orthogonal polynomials of specific standard random variables. Since each ξ_i is independent, it can be separately decomposed onto an one-dimensional PC basis of degree p:

$$\xi_i = \sum_{j=0}^{p} \gamma_{ij} \Psi_j(\eta_i), \quad i = 1, \ldots, r, \tag{32}$$

where the η_i are i.i.d. random variables. The random basis functions $\{\Psi_j\}$ are chosen according to the type of random variable $\{\eta_i\}$ that has been used to describe the random input. For example, if Gaussian random variables are chosen then the Askey based orthogonal polynomials $\{\Psi_j\}$ are chosen to be Hermite polynomials, if η_i are chosen to be uniform random variables, then $\{\Psi_j\}$ must be Legendre polynomials [4]. Although the Hermite polynomials are used in this paper, the method developed can be applied to generalized polynomial chaos expansions. The Hermite polynomials are given by

$$\begin{aligned}&\Psi_0(\eta_i)=1,\quad \Psi_1(\eta_i)=\eta_i,\\ &\Psi_{j+1}(\eta_i)=\eta_i\Psi_j(\eta_i)-j\Psi_{j-1}(\eta_i),\quad \text{if } j\geqslant 1.\end{aligned} \tag{33}$$

The above one-dimensional Hermite polynomials are orthogonal with respect the corresponding probability density function (PDF) of the standard normal random variable:

$$E[\Psi_i\Psi_j]=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}\Psi_i(\eta)\Psi_j(\eta)e^{-\frac{\eta^2}{2}}d\eta=i!\;\delta_{ij}. \tag{34}$$

Thus, the PC coefficients can be computed through Galerkin projection:

$$\gamma_{ij}=\frac{E[\xi_i\Psi_j(\eta)]}{E\left[\Psi_j^2(\eta)\right]}=\frac{1}{\sqrt{2\pi}j!}\int_{-\infty}^{+\infty}\xi_i\Psi_j(\eta)e^{-\frac{\eta^2}{2}}d\eta,\quad i=1,\ldots,r,\quad j=0,\ldots,p. \tag{35}$$

A numerical integration is needed to evaluate this integral. However, it is noted that the random variable ξ does not belong to the same stochastic space as η, a non-linear mapping $\Gamma:\eta\to\xi$ is thus needed which preserves the probabilities such that $\Gamma(\eta)$ and ξ have the same distributions. Here, a non-intrusive projection method using empirical cumulative distribution functions (CDFs) of samples, which was developed in [21], is utilized to compute the PC coefficients.

3.2. A non-intrusive projection method for calculating PC coefficients

The non-linear mapping $\Gamma:\eta\to\xi$ can be defined by employing the Rosenblatt transformation [24] as shown below for each ξ_i:

$$\xi_i\overset{d}{=}\Gamma_i(\eta_i),\quad \Gamma_i\equiv F_{\xi_i}^{-1}\circ F_{\eta_i}, \tag{36}$$

where F_{ξ_i} and F_{η_i} denote the CDFs of ξ_i and η_i, respectively. Here, the equalities,"$\overset{d}{=}$" should be interpreted in the sense of distribution such that the PDFs of random variables on both sides are equal. The statistical toolbox of MATLAB provides functions to evaluate the CDF of many standard PC random variables.

However, the marginal CDF of the samples ξ_i is not known and can only be evaluated numerically from the available data. Here, the kernel density estimation approach is used to construct the empirical CDF of ξ_i [41]. Let $\left\{\xi_i^{(s)}\right\}_{s=1}^{N}$ be N samples of ξ_i obtained from Eq. (29). Then the marginal pdf of ξ_i is evaluated as:

$$p_{\xi_i}(\xi)\approx\frac{1}{N}\sum_{s=1}^{N}\frac{1}{\sqrt{2\pi\tau}}\exp\left(-\frac{\xi-\xi_i^{(s)}}{2\tau^2}\right), \tag{37}$$

where the bandwidth τ should be chosen to balance smoothness and accuracy. Then CDF of ξ can be obtained by integrating Eq. (37) and then the inverse CDF is computed. The MATLAB function, `ksdensity`, in statistical tool box is used to find the inverse CDF of ξ_i, where the τ is automatically computed from the information of data.

Now the expectation in the Galerkin projection Eq. (35) is well defined and can be computed in η-space. Then, the coefficients of the PC expansion can be computed using a Gauss–Hermite quadrature:

$$\gamma_{ij}=\frac{1}{\sqrt{2\pi}j!}\int_{-\infty}^{+\infty}\xi_i\Psi_j(\eta)e^{-\frac{\eta^2}{2}}d\eta\approx\frac{1}{\sqrt{\pi}j!}\sum_{k=1}^{N_g}\omega_k\Gamma_i\left(\sqrt{2}\mu_k\right)\Psi_j\left(\sqrt{2}\mu_k\right), \tag{38}$$

where the $\{\omega_k,\mu_k\}_{k=1}^{N_g}$ are integration weights and points of Gauss–Hermite quadrature. It is noted here that a transformation $\eta=\sqrt{2}\mu$ is used since the weight in the Gauss–Hermite quadrature is $e^{-\eta^2}$ while the PDF of Gauss random variable is $\frac{1}{\sqrt{2\pi}}e^{-\eta^2/2}$.

3.3. PC representation of the random field

Now bringing together the KPCA and PC expansions, one arrives at the following r-dimensional PC representation of the stochastic random field in the feature space:

$$\mathbf{Y}_r=\sum_{i=1}^{N}\beta_i\Phi(\mathbf{y}_i),\quad \text{with } \boldsymbol{\beta}=\mathbf{A}_r\boldsymbol{\xi}+\frac{1}{N}\mathbf{1}, \tag{39}$$

where the $r\times 1$ column vector $\boldsymbol{\xi}$ is:

$$\boldsymbol{\xi}=\left[\sum_{j=0}^{p}\gamma_{1j}\Psi_j(\eta_1),\ldots,\sum_{j=0}^{p}\gamma_{rj}\Psi_j(\eta_r)\right]^T. \tag{40}$$

Therefore, by drawing i.i.d. samples of standard Gaussian random variables η_i, $i = 1,\ldots,r$, we obtain different realizations of the underlying random field in the feature space F. Now the dimensionality of the stochastic input space is successfully reduced from a large number M to a small number r.

4. The pre-image problem in KPCA

The simulated realizations of the random field from Eq. (39) are in the feature space F. However, we are interested in obtaining realizations in the physical input space. Therefore, an inverse mapping is required as $\mathbf{y} = \Phi^{-1}(\mathbf{Y})$. This is known as the pre-image problem [34,38]. As demonstrated in [34], this pre-image may not exist or even if it exists, it may be not unique. Therefore, we can only settle for an approximate pre-image $\hat{\mathbf{y}}$ such that $\Phi(\hat{\mathbf{y}}) \approx \mathbf{Y}_r$.

4.1. Fixed-point iteration for finding the pre-image

One solution to the pre-image problem is to address this problem as a nonlinear optimization problem by minimizing the squared distance in the feature space between $\Phi(\hat{\mathbf{y}})$ and $\mathbf{Y}_r$:

$$\rho(\hat{\mathbf{y}}) = \|\Phi(\hat{\mathbf{y}}) - \mathbf{Y}_r\|^2. \tag{41}$$

The extremum can be obtained by setting $\nabla_{\hat{\mathbf{y}}}\rho = 0$. For Gaussian kernel Eq. (18), this nonlinear optimization problem can be solved by a fixed-point iteration method [34]:

$$\hat{\mathbf{y}}_{t+1} = \frac{\sum_{i=1}^{N}\beta_i \ \exp\left(-\|\hat{\mathbf{y}}_t - \mathbf{y}_i\|^2/(2\sigma^2)\right)\mathbf{y}_i}{\sum_{i=1}^{N}\beta_i \ \exp\left(-\|\hat{\mathbf{y}}_t - \mathbf{y}_i\|^2/(2\sigma^2)\right)}. \tag{42}$$

As can be easily seen, the pre-image in this case will depend on the initial starting point and the optimization process is likely to be stuck in local minima. In addition, this scheme is numerically unstable and one has to try a number of initial guesses. Therefore, it is not suitable for our stochastic simulation since we need to have a one to one mapping and to find an efficient way for generating a large number of samples.

Furthermore, it is noted that the pre-image obtained in this scheme is in the span of all realizations $\mathbf{y}_i$'s, i.e. it is a linear combination of all the available realizations:

$$\hat{\mathbf{y}} = \sum_{i=1}^{N}\theta_i\mathbf{y}_i, \quad \text{with} \ \sum_{i=1}^{N}\theta_i = 1, \tag{43}$$

where the weights

$$\theta_i = \frac{\beta_i \ \exp\left(-\|\hat{\mathbf{y}}_t - \mathbf{y}_i\|^2/(2\sigma^2)\right)}{\sum_{i=1}^{N}\beta_i \ \exp\left(-\|\hat{\mathbf{y}}_t - \mathbf{y}_i\|^2/(2\sigma^2)\right)}. \tag{44}$$

Due to the exponential term in the Gaussian kernel, the contributions of the realizations typically drop rapidly with increasing distance from the pre-image. In other words, the influence of training realizations with smaller distance to $\hat{\mathbf{y}}$ will tend to be bigger. Therefore, it is reasonable to use only nearest neighbors of the pre-image for local linear interpolation.

4.2. Local linear interpolation for finding the pre-image

As illustrated in the last section, we can find the pre-image using local linear interpolation within n-nearest neighbors. Motivated by our previous work in [29], the Euclidean distances are used as the interpolation weights. Actually, there exists a simple relationship between the feature-space and input-space distance for Gaussian kernel [38,42]. The basic idea of the method is shown in Fig. 2. For an arbitrary realization $\mathbf{Y}_r$ in F, we first calculate its distances to the nearest neighbors in the

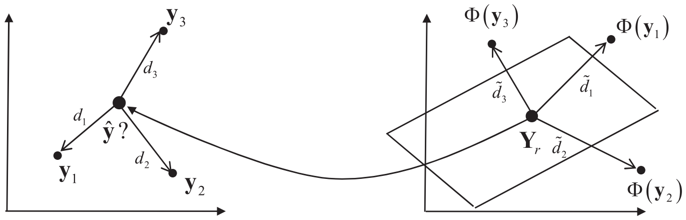

Fig. 2. Basic idea of the proposed pre-image method.

feature space. Then the distances in the input space are recovered. Finally, the input-space distances are used as the local linear interpolation weights.

Given any realization $\mathbf{Y}_r \in F$, we can compute the squared feature distance between $\mathbf{Y}_r$ to the ith mapped data as:

$$\tilde{d}_i^2(\mathbf{Y}_r, \Phi(\mathbf{y}_i)) = \|\mathbf{Y}_r - \Phi(\mathbf{y}_i)\|^2 = \|\Phi(\mathbf{y}_i)\|^2 + \|\mathbf{Y}_r\|^2 - 2\mathbf{Y}_r^T\Phi(\mathbf{y}_i), \tag{45}$$

for $i = 1,\ldots,N$. Recall that for Gaussian kernel, $k(\mathbf{y}_i,\mathbf{y}_i) = 1$ and $\mathbf{Y}_r = \sum_{i=1}^N \beta_i \Phi(\mathbf{y}_i)$. Then each feature distance $\tilde{d}_i^2(\mathbf{Y}_r, \Phi(\mathbf{y}_i)),\ i = 1,\ldots,N$ can be computed in the following matrix form:

$$\begin{aligned}\tilde{d}_i^2 &= \Phi(\mathbf{y}_i)^T\Phi(\mathbf{y}_i) + \mathbf{Y}_r^T\mathbf{Y}_r - 2\mathbf{Y}_r^T\Phi(\mathbf{y}_i) = k(\mathbf{y}_i,\mathbf{y}_i) + (\mathbf{\Phi}\boldsymbol{\beta})^T\mathbf{\Phi}\boldsymbol{\beta} - 2(\mathbf{\Phi}\boldsymbol{\beta})^T\Phi(\mathbf{y}_i) = 1 + \boldsymbol{\beta}^T\mathbf{\Phi}^T\mathbf{\Phi}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\mathbf{\Phi}^T\Phi(\mathbf{y}_i)\\ &= 1 + \boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\mathbf{k}_{\mathbf{y}_i},\end{aligned} \tag{46}$$

for $i = 1,\ldots,N$. Denote the vector $\tilde{\mathbf{d}}^2 = [\tilde{d}_1^2,\ldots,\tilde{d}_N^2]^T$ and we can sort this vector in ascending order to identify the n-nearest neighbors with respect to $\mathbf{Y}_r$, $\Phi(\tilde{\mathbf{y}}_i),\ i = 1,\ldots,n$.

On the other hand, the squared feature distance between the Φ-map of the pre-image $\hat{\mathbf{y}}$ and the ith mapped data point is:

$$\hat{d}_i^2(\Phi(\hat{\mathbf{y}}), \Phi(\mathbf{y}_i)) = \|\Phi(\hat{\mathbf{y}}) - \Phi(\mathbf{y}_i)\|^2 = k(\hat{\mathbf{y}},\hat{\mathbf{y}}) + k(\mathbf{y}_i,\mathbf{y}_i) - 2k(\hat{\mathbf{y}},\mathbf{y}_j) = 2(1 - k(\hat{\mathbf{y}},\mathbf{y}_i)), \tag{47}$$

for $i = 1,\ldots,N$ and where we have used $k(\hat{\mathbf{y}},\hat{\mathbf{y}}) = k(\mathbf{y}_i,\mathbf{y}_i) = 1$ again since a Gaussian kernel is used. Furthermore, the squared input-space distance can be computed from the following equation:

$$k(\hat{\mathbf{y}},\mathbf{y}_i) = \exp\left(-\frac{\|\hat{\mathbf{y}} - \mathbf{y}_i\|^2}{2\sigma^2}\right),$$

from which we obtain

$$d_i^2 = \|\hat{\mathbf{y}} - \mathbf{y}_i\|^2 = -2\sigma^2\log(k(\hat{\mathbf{y}},\mathbf{y}_i)), \tag{48}$$

for $i = 1,\ldots,N$. Substituting the expression of $k(\hat{\mathbf{y}},\mathbf{y}_j)$ from Eq. (47) into Eq. (48), one arrives at

$$d_i^2 = \|\hat{\mathbf{y}} - \mathbf{y}_i\|^2 = -2\sigma^2\log(1 - 0.5\hat{d}_i^2), \tag{49}$$

for $i = 1,\ldots,N$. Because we try to find an approximate pre-image such that $\Phi(\hat{\mathbf{y}}) \approx \mathbf{Y}_r$, it is straightforward to identify the relationship $\tilde{d}_i^2 \approx \hat{d}_i^2$ from Eqs. (45) and (47):

$$\tilde{d}_i^2(\mathbf{Y}_r, \Phi(\mathbf{y}_i)) = \|\mathbf{Y}_r - \Phi(\mathbf{y}_i)\|^2 \approx \|\Phi(\hat{\mathbf{y}}) - \Phi(\mathbf{y}_i)\|^2 = \hat{d}_i^2(\Phi(\hat{\mathbf{y}}), \Phi(\mathbf{y}_i)). \tag{50}$$

Therefore, the squared input-distance between the approximate pre-image $\hat{\mathbf{y}}$ and the ith input data realization can be computed by:

$$d_i^2 = \|\hat{\mathbf{y}} - \mathbf{y}_i\|^2 = -2\sigma^2\log(1 - 0.5\tilde{d}_i^2), \tag{51}$$

for $i = 1,\ldots,N$ and where $\tilde{d}_i^2$ is given by Eq. (46).

Finally, the pre-image $\hat{\mathbf{y}}$ for a feature space realization $\mathbf{Y}_r$ is given by

$$\hat{\mathbf{y}} = \frac{\sum_{i=1}^n \frac{1}{d_i}\tilde{\mathbf{y}}_i}{\sum_{i=1}^n \frac{1}{d_i}}, \tag{52}$$

where $\tilde{\mathbf{y}}_i,\ i = 1,\ldots,n$ are the n-nearest neighbors. It is noted that here we use the n-nearest neighbors in the feature space. However, they are the same as the n-nearest neighbors in the input space since Eq. (51) is monotonically increasing.

Therefore, the pre-image $\hat{\mathbf{y}}$ of an arbitrary realization in the feature space is the weighted sum of the pre-images of the n-nearest neighbors of $\mathbf{Y}_r$ in the feature space, where the nearest neighbors are taken from the samples $\mathbf{y}_i$, $i = 1,\ldots,N$. This local linear interpolation procedure is based on the principle that a small region in a highly curved manifold can be well approximated as a linear patch. It is easily verified that the interpolation weights satisfies Eq. (43). Thus, a unique pre-image can now be obtained using simple algebraic calculations in a single step (no iteration is required) and is suitable for stochastic simulation.

5. Numerical example

In this section, we apply kernel PCA on modeling random permeability field of complex geological channelized structures. This structure is characterized by multipoint statistics (MPS), which expresses the joint variability at more than two locations [43]. Therefore, only mean and correlation structure (two-point statistics) are not enough to capture the underlying uncertainty of the random field and thus linear PCA or standard K–L expansion is expected to fail.

5.1. Generation of experimental samples

In [17], the experimental data was obtained through solution of stochastic inverse problems given several realizations of the system outputs. This method is computationally expensive if a large number of samples is required. To this end, we utilize the reconstruction technique from an available training image to numerically generate samples of the random field [29]. The basic idea is that the training image contains geological structure or continuity information of the underlying random field, and the MPS algorithm creates realizations of the random field honoring this structural relationship [36]. In this example, one of the MPS algorithms, the single normal equation simulation 'snesim' algorithm is used to generate the channelized permeability [43] from the training image shown in Fig. 3. It is a binary image where 1 designates channel sand and 0 designates background mud. Since we are only interested in the geological structure not the value itself, a further assumption is made such that the image is a logarithmic transformation of the original permeability field and the values 1 and 0 are the permeability values themselves.

Using the 'snesim' algorithm, 1000 realizations of a channelized permeability field, of dimension 45×45, are created from the training image in Fig. 3. These serve as the training samples (experimental data) of the random field for the construction of KPCA. Additional set of realizations, which is not included in the training samples, will serve as the test samples to verify the accuracy of the constructed reduced-order model using KPCA algorithm. Some of the realizations created are shown in Fig. 4.

Each realization of the random field can be considered as a 2025-dimensional vector. Since each vector consists of only 0 and 1, all the samples occupy only corners of a 2025-dimensional hypercube. Therefore, they are not linearly related and linear PCA will fail. This problem is similar to the image-denoising problem in machine learning community where binary images of digits are used and the performance of KPCA is proved to be superior to that of linear PCA [34,38].

5.2. Comparison between linear PCA and kernel PCA

The kernels Eqs. (17) and (18) are now used to perform linear PCA and kernel PCA, respectively, on the 1000 sample realizations. The parameter c in Eq. (19) is taken to be 10. The kernel matrices are formulated and subsequent eigenvalue problems are solved. The corresponding first 100 normalized eigenvalues and normalized reconstruction mean squared error (MSE) from Eq. (31) are shown in Fig. 5. Here, normalized means, for example, each eigenvalue is divided by the sum of all the eigenvalues. It is interesting to note that the plots for linear PCA and KPCA are nearly the same. This indicates that the non-linear transformation from input space to feature space retains the energy contribution of each mode to the overall energy of the random field. For non-linear data set, the failure of linear PCA means overestimating the intrinsic dimensionality of the data. Therefore, in order to compare the performance from both methods, we need to keep the same number of eigenvectors. A rule of thumb is to choose r such that $\sum_{i=1}^{r}\lambda_i/\sum_{i=1}^{N}\lambda_i$ is sufficiently close to one. However, this rule is not applied here. Since if the reconstruction MSE is sufficiently small, there is no need to keep so many eigenvectors. To this end, only the largest 30 eigenvalues are retained for both linear PCA and KPCA, which corresponds to about 75% energy of the random field. The associated normalized reconstruction MSE is 0.005, which is small enough to ensure an accurate expansion in the input space and feature space.

We next try to reconstruct one of the test samples using both methods. The results are shown in Fig. 6 with different number of eigenvectors retained. By reconstruction, we mean we first project the test samples onto the low-dimensional

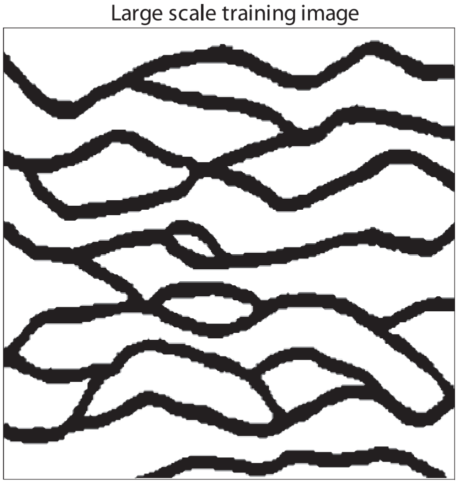

Fig. 3. The large-scale training image for the 'snesim' algorithm. It is a binary image where 1 (black region) designates channel sand while 0 (white regions) designates background mud.

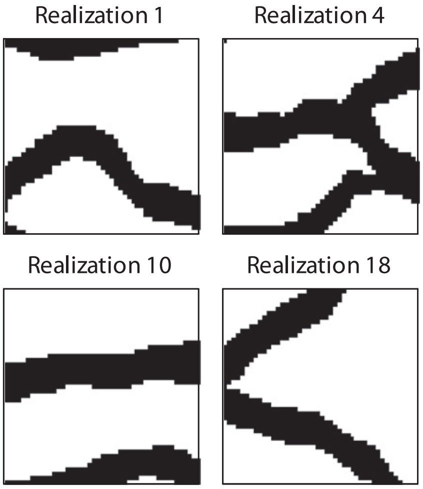

Fig. 4. Some of the realizations created using the ‘snesim’ algorithm.

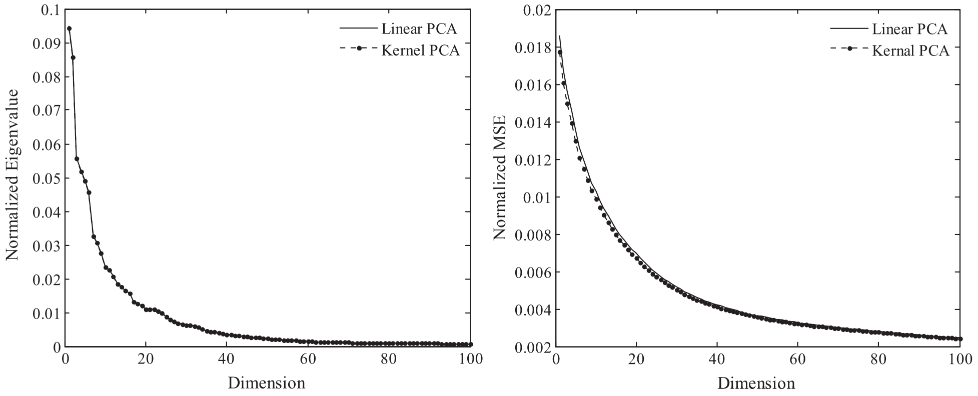

Fig. 5. Plots of the normalized eigenspectrum (left) and MSE (right column) from linear PCA and kernel PCA.

space and compute their coordinates. Then using only the low-dimensional coordinates, the original sample in the input space is reconstructed from linear PCA and kernel PCA pre-image algorithm, respectively. 10 nearest neighbors are used in finding the pre-image. It is seen that using linear PCA, the reconstruction results improve slowly with increasing number of eigenvectors. A large number of eigenvectors are needed to obtain a good result. This is consistent with our previous discussion that the linear PCA overestimates the actual dimensionality of the data in the nonlinear case. However, the reconstructed permeability value is still not correct even with $r = 500$. On the other hand, only 30 eigenvectors are needed to obtain a good reconstruction from KPCA. Increasing the number of eigenvectors will not improve the results since the nearest neighbors have been correctly identified and the reconstruction MSE in the feature space is quite small. The first 9 identified nearest neighbors of the test sample are shown in Fig. 7 with $r = 30$. It is clearly seen that the first 3 neighbors have the geological structures which are most similar to our test sample. This verifies that the introduced pre-image algorithm indeed finds the appropriate nearest neighbors among the data using only the distances in the features space and input space.

To further verify the kernel PCA algorithm, more test samples are reconstructed. To compare its performance with that of linear PCA, 30 eigenvectors are retained in both cases since it was shown before that $r = 30$ is enough for kernel PCA. The results are shown in Fig. 8. Again, kernel PCA consistently captures the geological pattern and the reconstruction samples are more like the original binary images. Fig. 9 also shows the convergence of the reconstruction results of two test samples from Figs. 6 and 8 with increasing number of nearest neighbors. It is shown that using 10 nearest neighbors is enough to obtain satisfying results. If more neighbors are used, the results will become worse since some samples are far from the original sample in Euclidean distance.

In this section, we compared the performance of linear PCA and kernel PCA for reconstruction. Although linear PCA is not optimal, it still more or less provided us the desired geological structure. However, what we are interested in is the generation of realizations from the underlying random field. In the next section, we will show that the linear PCA cannot give us arbitrary realizations with expected geological patterns.

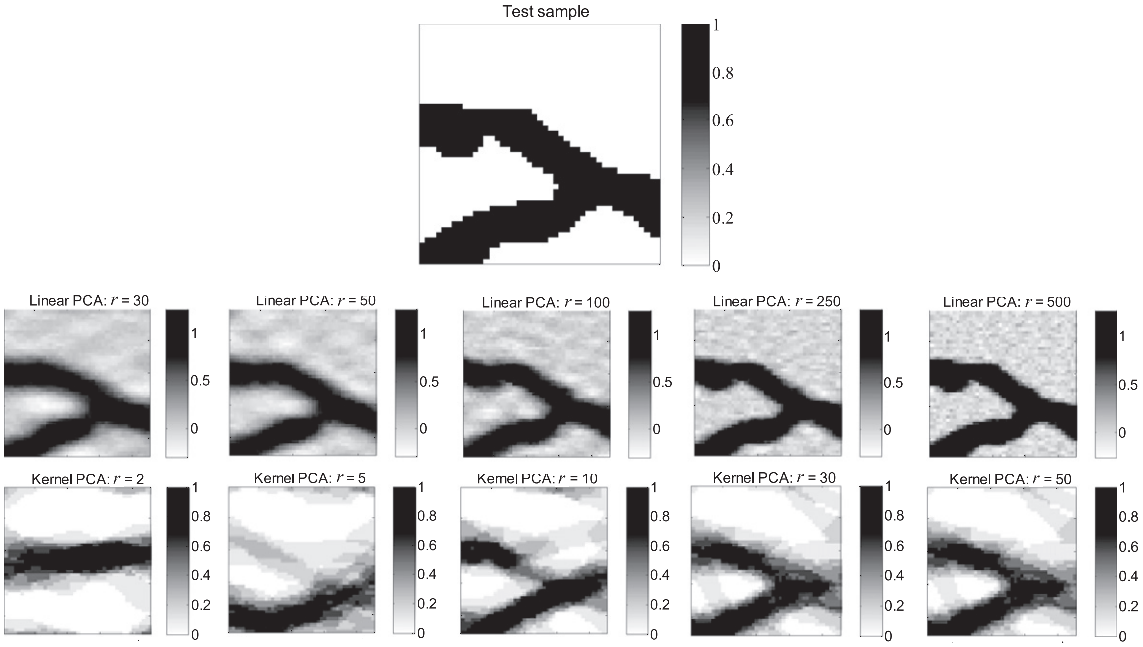

Fig. 6. Reconstruction of a test sample (top figure) using linear PCA (middle row) and kernel PCA (bottom row) with different number of eigenvectors retained.

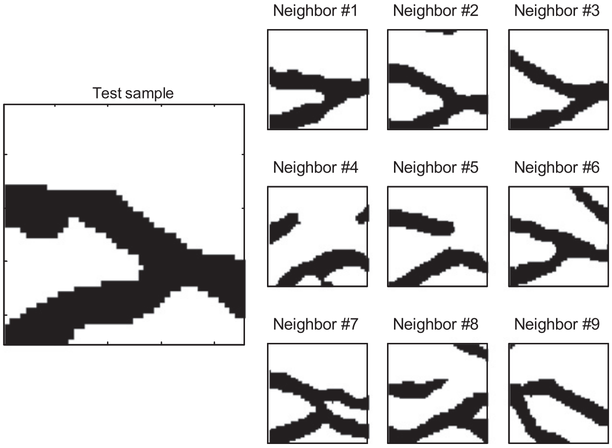

Fig. 7. The first 9 nearest neighbors of the test samples identified in the feature space with $r = 30$.

5.3. PC representation and stochastic sampling in the low-dimensional space

In this section, we construct the PC representation of the random field using the method introduced in Section 3.2. We keep 30 eigenvectors for both linear PCA and kernel PCA. First, the samples $\xi^{(i)}$ of ξ are computed by inserting the 1000 sample realizations $\mathbf{y}_i$ into Eq. (29). Then these samples are used to construct the inverse marginal CDF for each dimension through kernel density estimation. Finally, Eq. (38) is used to calculate the PC coefficients for each dimension. A PC representation of the random channelized permeability is constructed next, which is only of 30 dimensions, compared with the original 2025-dimensional space. By drawing 30 i.i.d. standard normal random variables η_i from this low-dimensional space and substituting them into Eqs. (39) and (40), any realization of the underlying random permeability field can now be obtained in an inexpensive way.

Fig. 10 depicts 4 different marginal PDFs of the initial and identified random variables using the non-intrusive projection method for Hermite chaos with increasing expansion orders using linear PCA. The marginal PDF of initial random variable is obtained by plotting the kernel density estimation of the 1000 sample coefficients $\xi^{(i)}$. The marginal PDF of the identified random variable is obtained by plotting the kernel density estimation of 10,000 PC realizations of ξ. These PC realizations

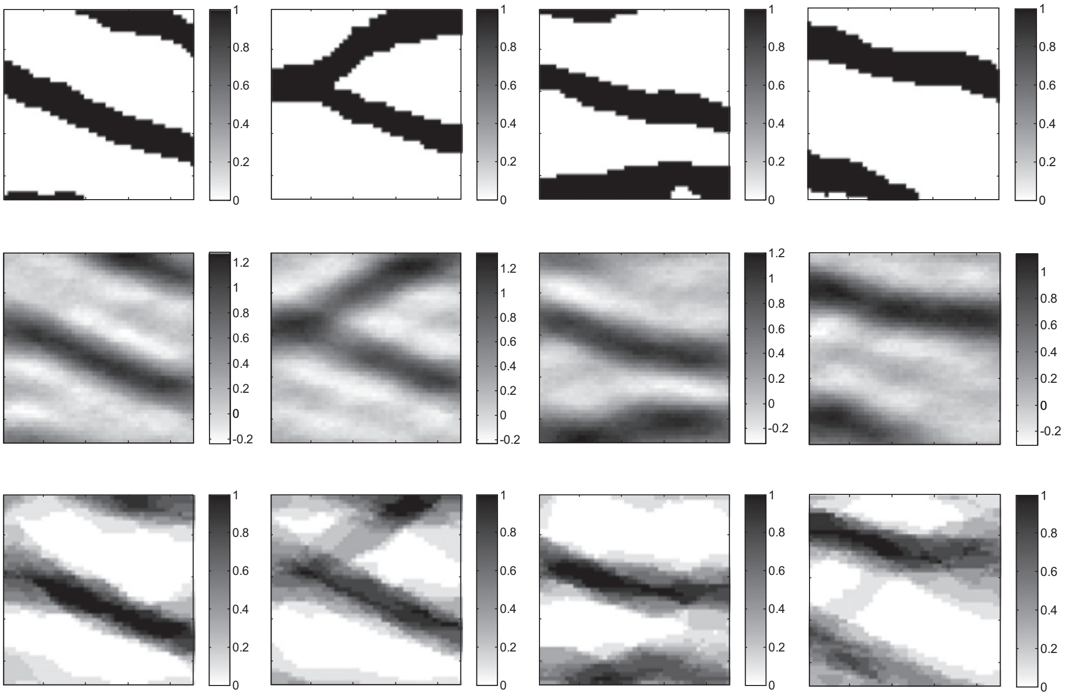

Fig. 8. More reconstruction results of different test samples (top row) using linear PCA (middle row) and kernel PCA (bottom row) with $r = 30$.

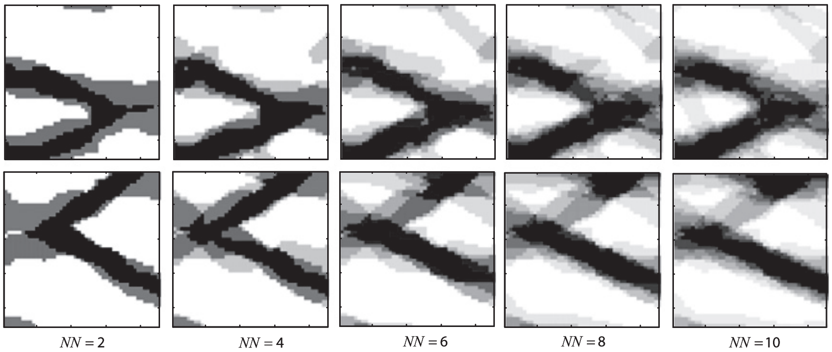

Fig. 9. Reconstruction results with different number of nearest neighbors (NN).

are calculated by generating 10,000 standard normal random vectors and inserting them into Eq. (32). It is seen that a $p = 10$ order expansion is enough to accurately identify the random coefficients. It is also interesting to note that the PC expansion converges slowly for the first two random coefficients ξ_1 and ξ_2. This is because more variance is contained in the larger eigenvectors due to the property of PCA.

Some of the realizations generated through sampling the PC representation of linear PCA are shown in Fig. 11. The failure of linear PCA is more pronounced in this case. The realizations definitely do not reflect the original channelized structure of the permeability field. In addition, the permeability value is not correctly predicted.

Similar results are shown in Figs. 12 and 13 for kernel PCA. However, using kernel PCA, it is seen that the generated realizations clearly show channelized geological structure with correct permeability values.

We introduce the relative errors for statistics of the experimental samples, $\{\mathbf{y}_i\}_{i=1}^{1000}$, and PC realizations, $\left\{\mathbf{y}_i^{(\mathrm{pc})}\right\}_{i=1}^{n_{\mathrm{pc}}}$:

$$e = \frac{\sqrt{\sum_{i=1}^{2025}\left(T_i - T_i^{(\mathrm{pc})}\right)^2}}{\sqrt{\sum_{i=1}^{2025} T_i^2}}, \tag{53}$$

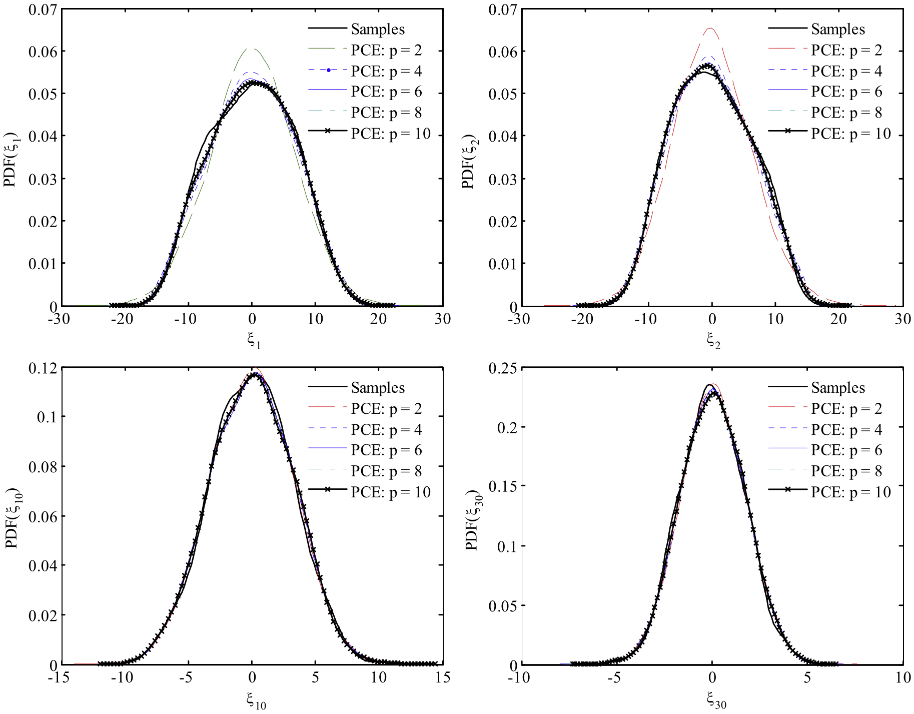

Fig. 10. Four different marginal PDFs of the initial and identified random variables using linear PCA with Hermite chaos.

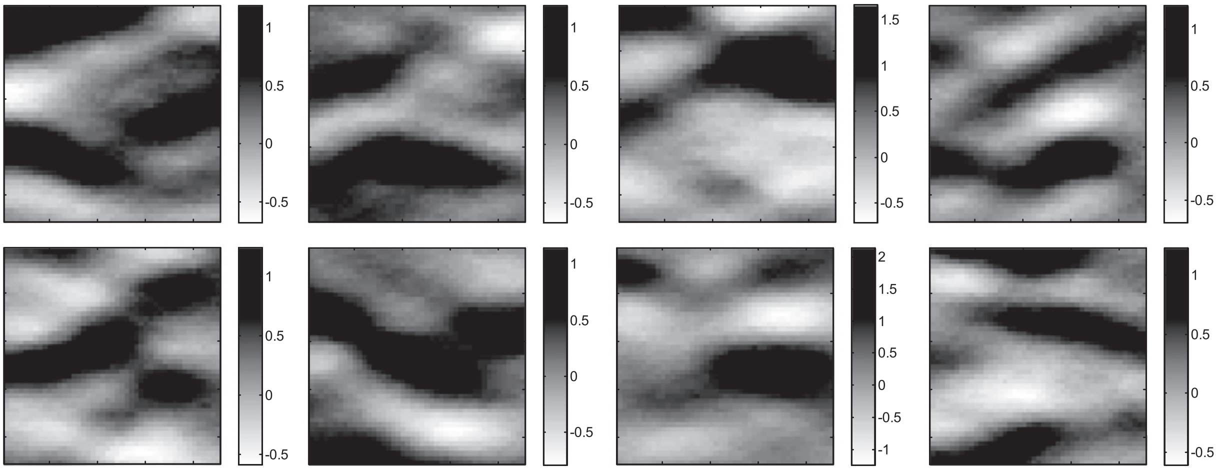

Fig. 11. Realizations of the random field by sampling the corresponding PC representation using linear PCA.

in which T_i represents the appropriate sample statistics of experimental samples $\{\mathbf{y}_i\}_{i=1}^{1000}$ in ith grid block and $T_i^{(\mathrm{pc})}$ represents the corresponding sample statistics of PC realizations $\left\{\mathbf{y}_i^{(\mathrm{pc})}\right\}_{i=1}^{n_{\mathrm{pc}}}$. Ten thousand PC realizations of the random permeability are generated. The results are shown in Table 1. From the table, as expected, the linear PCA gives accurate mean and covariance, which are first- and second-order statistics. On the other hand, the kernel PCA gives better skewness and kurtosis which can be considered as third- and fourth-order statistics, respectively. It is noted that relative error of skewness is 1.006 for linear PCA, while it is 0.114 for kernel PCA. To better understand these results, we also plot in Fig. 14 the marginal PDF (relative frequency diagram) of the permeability at two arbitrarily chosen grid blocks (11,24) and (25,25) from data, KPCA and linear PCA. The PDF is normalized such that the sum of the area is 1. It is obvious that the marginal PDF from binary image only has two bars at 0 and 1. Although the result from KPCA is not exactly the same as that of the original data, it does resemble the main characteristics, namely most of the values are concentrated near 0 which indicates the binary nature of the image. That partially explains why the reconstruction result looks like a binary image more than that of linear PCA does. On the other

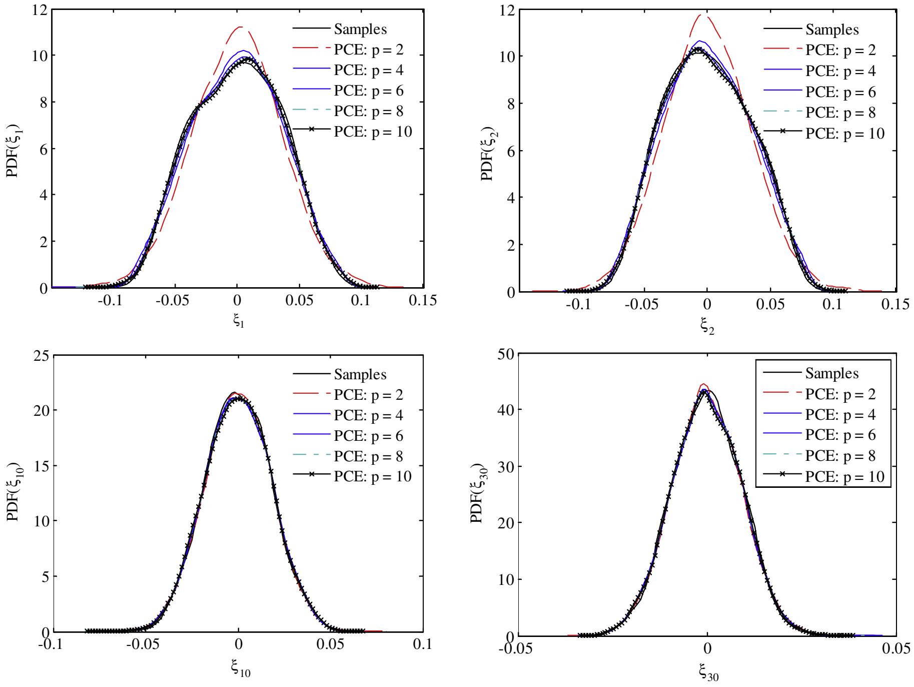

Fig. 12. Four different marginal PDFs of the initial and identified random variables using kernel PCA with Hermite chaos.

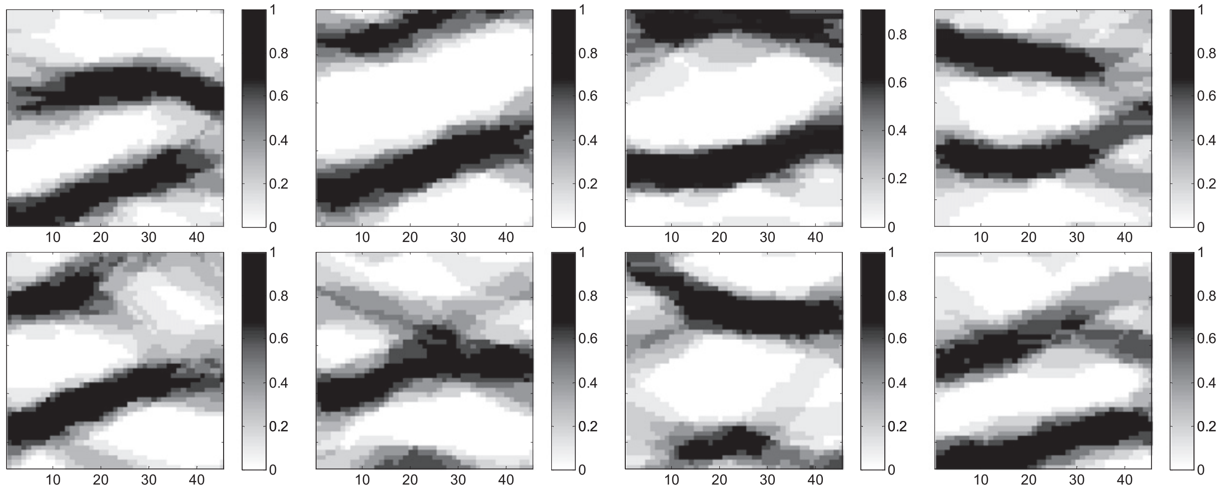

Fig. 13. Realizations of the random field by sampling the corresponding PC representation using kernel PCA.

Table 1
Comparison of statistics between experimental samples and PC samples of the random permeability field.

Errors	Mean vector	Covariance matrix	Skewness vector	Kurtosis vector
Linear PCA	0.039	0.193	1.006	0.676
Kernel PCA	0.102	0.427	0.114	0.496

hand, as expected, the marginal PDF from linear PCA looks like a Gaussian since it can only capture the mean and variance of the random field which determines a Gaussian distribution. The skewness is a measure of the asymmetry of the probability distribution of the data around mean. We have positive skewness since the tail of the PDF on the right side is longer than that on the left side. However, the skewness of a Gaussian distribution is almost zero since it is a symmetric distribution. Indeed, the skewness at block (25, 25) is 0.792 from data, 0.787 from KPCA and 0.049 from linear PCA. This clearly explains why the

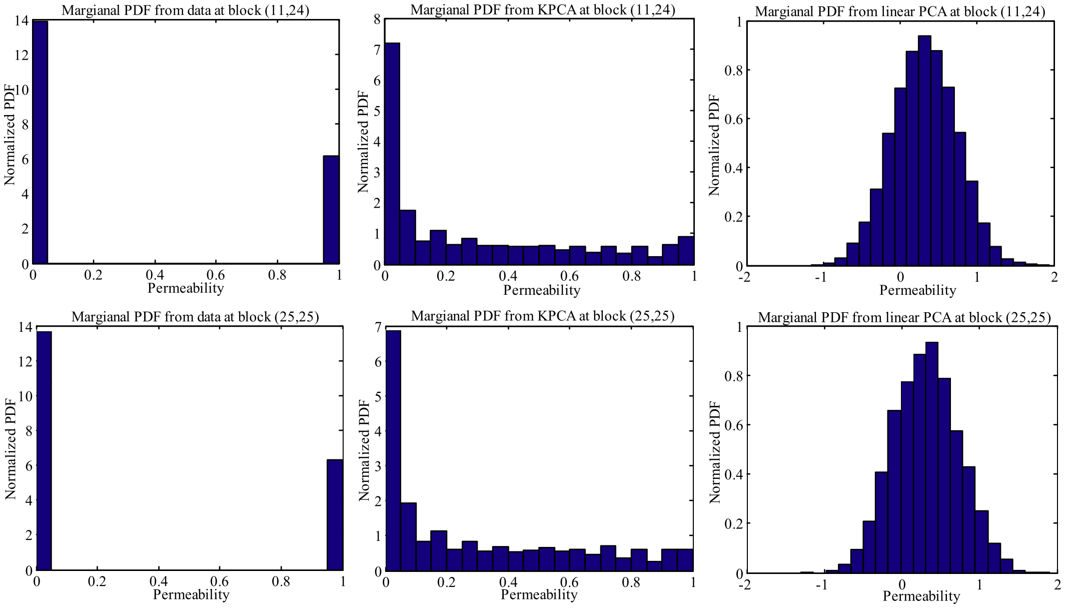

Fig. 14. Marginal PDF of the permeability at grid block (11,24) and (25,25) from data, KPCA and linear PCA, respectively.

error of the skewness vector is significantly different for KPCA and linear PCA. Kurtosis is a measure of the "peakedness" of the probability distribution of the data. Although KPCA does a good job to indicate the peak 0, it does not measure the peak 1 accurately, which is possibly due to the local interpolation Eq. (52) used. On the other hand, linear PCA fails to indicate both values. This explains why the kurtosis error of KPCA is slightly lower than that of linear PCA but is still a little large when compared to data.

Finally, we want to comment on the overall statistics of generated PC realizations. From Table 1, it is seen that the errors from kernel PCA are still big for covariance and kurtosis. However, it is emphasized that we are most interested in the main geological structure of the random field. We cannot obtain exactly the same statistics as in the experimental samples. There are three possible reasons. First, only an approximate pre-image is calculated. This process contributes significant portion of the total error. Second, a PC expansion is used to fit the marginal PDF of the random coefficients. From Fig. 12, it is noted that some experimental samples are from the long tail region of the marginal PDF. Thus, the experimental samples can be considered as the extreme realizations of the underlying random fields. When we sample the corresponding marginal PDF from PC expansion, it is clear that most of the samples are from the high density region. Third, Euclidean distance is used as a distance measure for the calculation which might not be the best choice for binary images. Although KPCA can reveal the linear relation of the mapped data in the feature space, it may not give us the best result in the input space. Therefore, the statistics of the generated PC realizations are expected to be not exactly the same as that of experimental samples. But the PC realizations should capture the main statistical features of the experimental samples. This point is demonstrated in the next section.

5.4. Forward uncertainty propagation with the stochastic input model

In this section, the generated stochastic input model is used as an input to the single-phase flow problem on the domain $[0,1]^2$. The governing equations of the single-phase flow are [44]

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x},\omega) = 0, \quad \boldsymbol{u}(\boldsymbol{x},\omega) = -a(\boldsymbol{x},\omega)\nabla p(\boldsymbol{x},\omega) \quad \forall \boldsymbol{x} \in D, \tag{54}$$

$$\frac{\partial S(\boldsymbol{x},t,\omega)}{\partial t} + \boldsymbol{u}(\boldsymbol{x},t,\omega) \cdot \nabla S(\boldsymbol{x},t,\omega) = 0, \quad \forall \boldsymbol{x} \in D, \ t \in [0,T]. \tag{55}$$

Flow is induced from left-to-right with Dirichlet boundary conditions $p = 1$ on $\{x_1 = 0\}$, $p = 0$ on $\{x_1 = 1\}$ and no-flow homogeneous Neumann boundary conditions on the other two edges. We also impose zero initial condition for saturation $S(x,0) = 0$ and boundary condition $S(x,t) = 1$ on the inflow boundary $\{x_1 = 0\}$. Mixed finite element method developed in [44] is used to solve the above equations with spatial discretization 45×45. So the permeability is defined as a constant in each grid block.

The stochastic permeability is $a = \exp(\mathbf{y})$ if the experimental samples are used or $a = \exp(\mathbf{y}^{n_{pc}})$ if the PC realizations from the stochastic input model are used in the computation. Monte Carlo (MC) simulation is then conducted using both experimental samples directly and generated PC realizations from linear PCA and kernel PCA. Fifty thousand PC realizations are generated by sampling the low-dimensional space. It is noted here for the purpose of better comparison, 8000 experimental samples are generated and used in the MC simulation. However, the stochastic input model is the same as the one used to generate the realizations in the last section, i.e. only the previous 1000 data samples are used in constructing the stochastic input model. In this way, it is able to verify if our model can better resemble the statistics of the underlying random field even with less data which is usually the case in real engineering problem.

The contour plots of mean and standard deviation of the saturation at 0.2 PVI are given in Fig. 15. PVI represents dimensionless time and is computed as $\text{PVI} = \int Q dt / V_p$, where V_p is the total pore volume of the system, which is equal to the area of the domain D here and $Q = \int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n}) ds$ is the total flow rate on the out flow boundary ∂D^{out}. From the figure, it is seen that although the mean saturation values from the three methods are nearly the same, the standard deviation from linear PCA compares better with that of experimental samples than KPCA. This is consistent with the results in Table 1. It is also interesting to note that the contour values from KPCA and linear PCA are almost the same which again verifies the results in Fig. 5 that there is little energy (variance) loss when transforming from input space to feature space and the values of variance are almost the same for both methods when retaining the same number of eigen modes. From this figure, it is obvious that only considering first- and second-order statistics is not enough, which would give us the wrong conclusion that linear PCA works better, thus necessitating further examining the higher-order statistics. The contour plots of skewness and kurtosis of the saturation at 0.2 PVI are given in Fig. 16. As expected from the results in Table 1, the contours of skewness and kurtosis from KPCA compare well with those obtained using experimental samples directly even when only 1000 data are used in the non-linear model construction. On the other hand, linear PCA gives us completely wrong results.

The water cut curves are also given in Fig. 17. The water-cut is defined as $F(t) = \frac{\int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n}) S\, ds}{\int_{\partial D^{\text{out}}} (\boldsymbol{u}_h \cdot \boldsymbol{n})\, ds}$. From the figures, it is seen that we obtain nearly the same mean saturation and mean water cut curves from the experimental samples and the PC realizations. Again, linear PCA gives better variances of the water cut curves than KPCA does. It is difficult to visualize skewness and kurtosis of the water cut curves. Instead, we plot the marginal PDF (relative frequency diagram) at grid block (11,24) in Fig. 18 where it has the highest standard deviation. Similar to Fig. 14, PDF from KPCA looks more like the PDF of experimental data than that of linear PCA. This explains why linear PCA fails to capture the higher-order statistics of the underlying saturation field. Overall, as expected, PCA is better at capturing the first- and second-order statistics of the non-linear related random data while fails to describe the higher-order statistics. On the other hand, KPCA gives satisfying results of the overall statistics and resembles the main patterns of the original experimental samples. Although the standard deviation is not accurate enough, it still captures the same main features as the results from the experimental samples. To be specific, the regions with the highest standard deviation values of saturation are nearly the same in both cases. The shape of the standard deviation of the flow curve also looks similarly in both cases. In addition, the KPCA model is constructed with limited number of

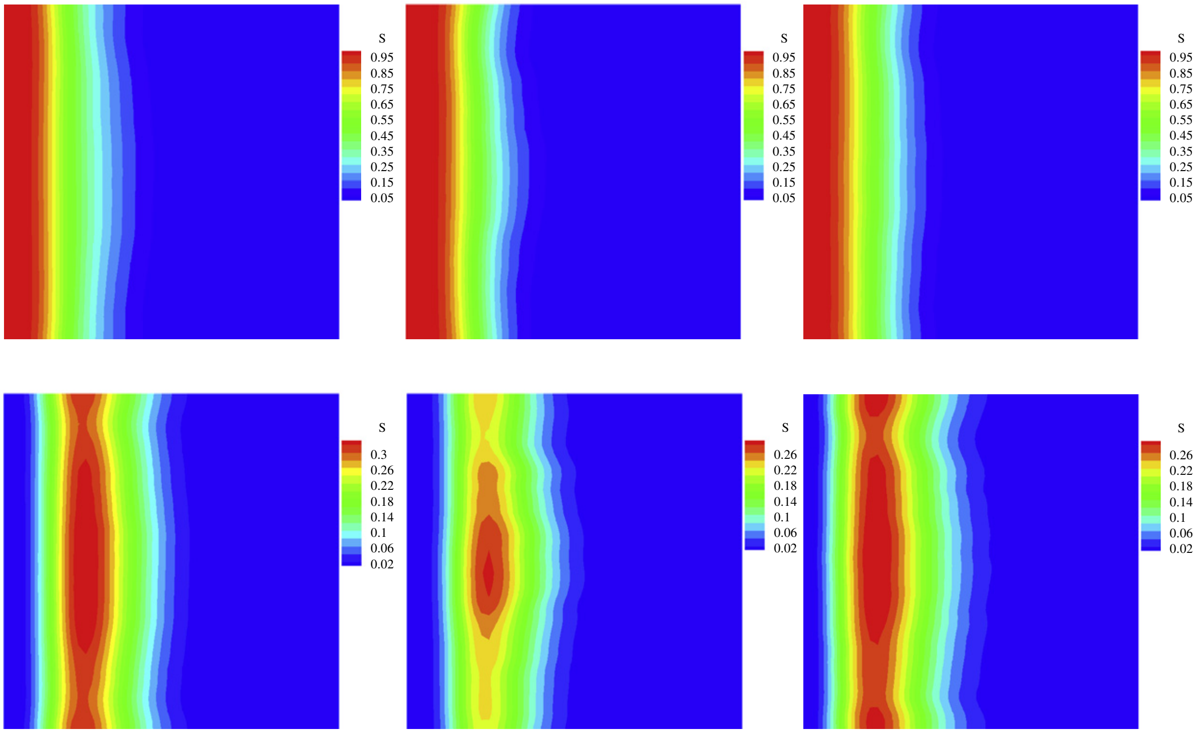

Fig. 15. Contours of mean (top row) and standard deviation (bottom row) of saturation at 0.2 PVI from experimental data (left), kernel PCA (middle) and linear PCA (right), respectively.

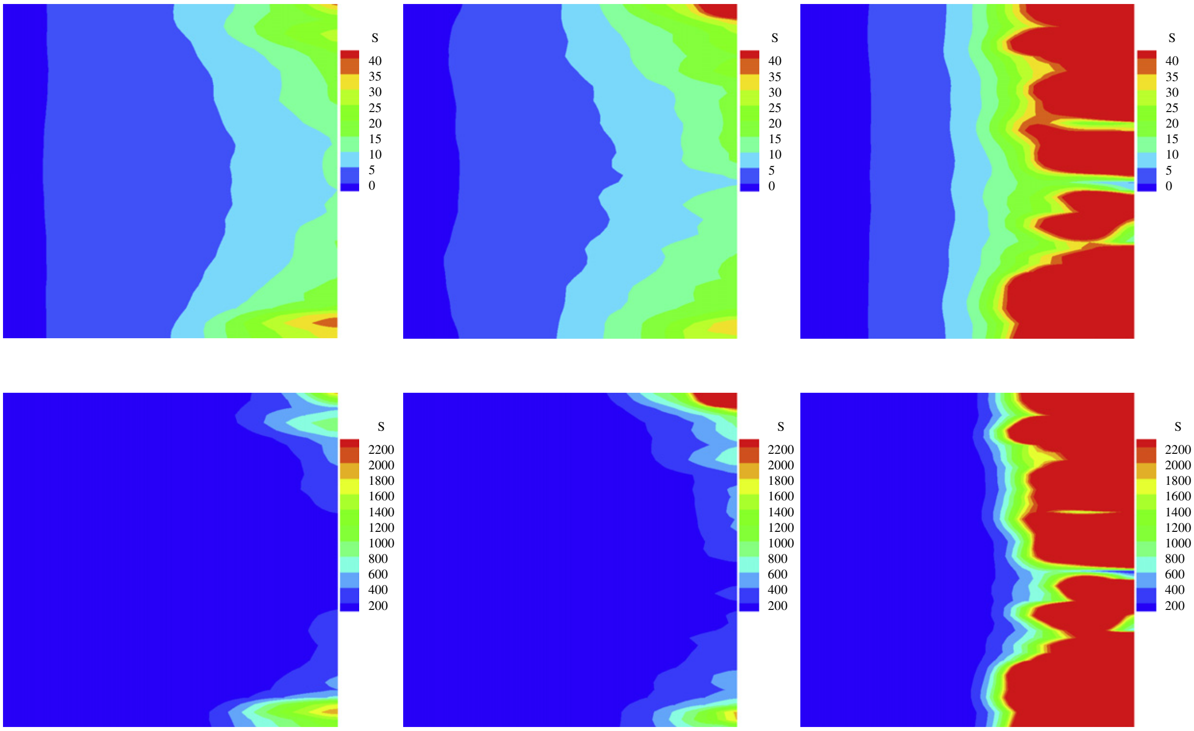

Fig. 16. Contours of skewness (top row) and kurtosis (bottom row) of saturation at 0.2 PVI from experimental data (left), kernel PCA (middle) and linear PCA (right), respectively.

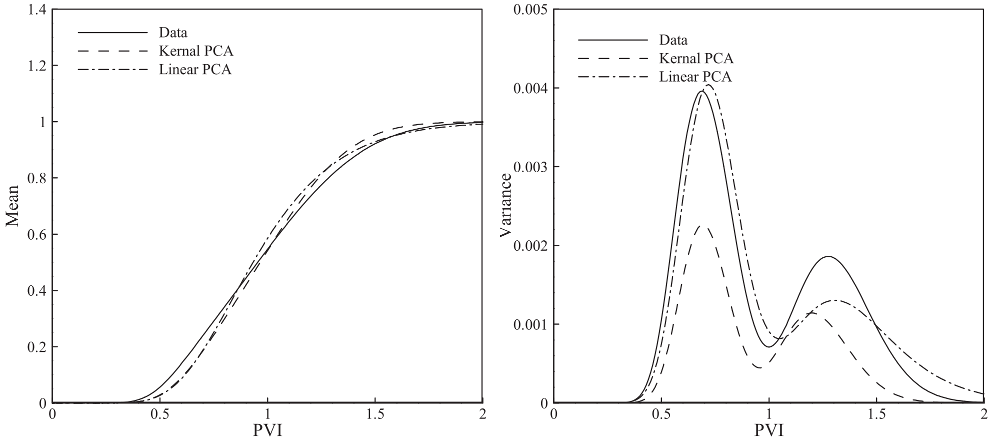

Fig. 17. Comparison of water cut curves: (a) mean; (b) variance.

data but successfully represents the overall statistics of the underlying random field. This is pretty useful since it is possibly expensive in practical engineering problems to obtain a large number of data samples. On the other hand, the proposed stochastic input model provides a fast way to generate many realizations, which are consistent, in a useful sense, with the experimental data.

Finally, we want to comment on the computational time. To generate the 8000 experimental data using the 'snesim' algorithm, it took approximate 10 h on a single processor machine and consumes lots of memory. In fact, 8000 is the highest number we can generate on our machine before running out of memory. On the other hand, to generate 50,000 PC realizations from our reduced-order model, less than 1 minute is needed on the same machine. In addition, the process of generating PC realizations is easily parallelizable, which is used in our MC simulations.

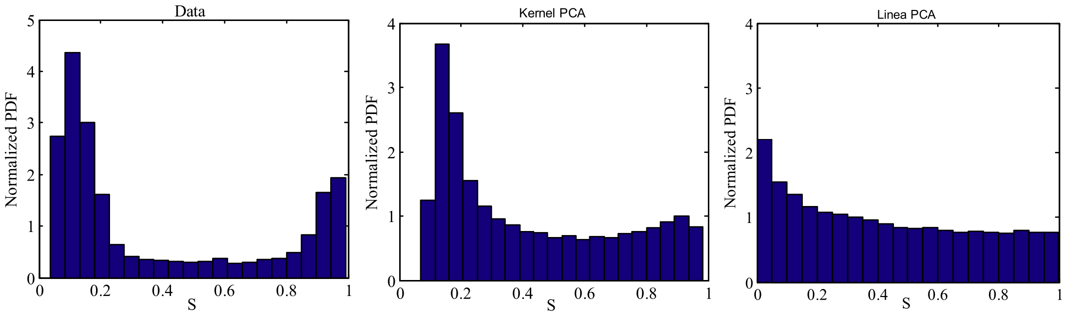

Fig. 18. Comparison of marginal PDF of saturation at grid block (11,24).

6. Conclusion

In this paper, a new parametric stochastic reduced-order modeling method is proposed, which can efficiently preserve higher-order statistics of the random field given limited experimental data. This method relies on the theory of kernel PCA to transform the data into a feature space, in which the data is more linearly related than in the original input space. PC expansion is then utilized to identify the random coefficients in the kernel PCA formulation. A simple approximate pre-image algorithm is also introduced to project the generated PC realizations back to the original input space. A thorough comparison between the linear PCA and kernel PCA on reduced-order modeling of the channelized permeability field is conducted. As expected, kernel PCA gives more realistic realizations which reflects the original channelized geological structure with much less eigenvectors retained. This results in a low-dimensional stochastic input space. On the other hand, linear PCA fails to capture the channelized structure with the same number of eigenvectors due to the nonlinearity of the data. Forward uncertainty quantification is also conducted which shows the introduced stochastic input model indeed captures the main statistical properties of the underlying random field.

As a first step towards the implementation of this method, the independence between the random coefficients was assumed. Although this gives us meaningful results in the numerical examples considered, the effect taking the dependence into account as in [19] using Rosenblatt transformation needs further investigation. In addition, the Euclidean distance is directly used as the distance measure between data. This may be not the best choice for binary images. However, our current development is not limited to Euclidean distance. In fact, any distance can be used to replace the Euclidean distance in the kernel function Eq. (18), for example, the geodesic distance [30] or the Hausdorff distance [45] which is used for object matching in imaging processing. As in [37], it may be interesting to take distances between flow responses as the distance measure between permeability data. This might be more accurate since it incorporates the information of the underlying stochastic system. In this paper, Gaussian kernel is used as a nonlinear kernel and our pre-image algorithm utilizes the properties of this kernel, i.e. $k(\hat{\mathbf{y}}, \hat{\mathbf{y}}) = 1$, which may be not suitable for other kernels since the pre-image is not known *a priori*. It will be interesting to investigate the effect of using polynomial kernel and develop the corresponding pre-image algorithm. All the above mentioned topics are currently under investigation.

Linear PCA has been proved to be effective in various scientific areas. In general, linear PCA is still the first choice for model reduction. KPCA may be less effective than linear PCA when data is linearly-related. However, when simulating random heterogenous materials, most existing experimental data and reconstruction techniques to generate new samples are still based on binary images [46]. It is expected our developed stochastic model to be more useful in this area. In addition, the basic model reduction ideas envisioned in this work are not limited to the generation of viable stochastic input models for property variations. This framework has direct applicability to inverse problems [13], where the generated model can be considered as the prior model of all available properties. Thus, finding the unknown property is only limited to the low-dimensional space and is expected to be more efficient than working in the original high-dimensional space. Furthermore, the generation of a low-dimensional surrogated space provides a prerequisite in stochastic low-dimensional modeling of SPDEs [47], which may have major ramifications in design under uncertainty and stochastic optimization problems. These potentially exciting areas of application of the framework introduced here offer fertile avenues of further research.

Acknowledgments

This research was supported by the US Department of Energy, Office of Science, Advanced Scientific Computing Research, the Computational Mathematics program of the National Science Foundation (NSF) (award DMS- 0809062) and an OSD/AFOSR MURI09 award on uncertainty quantification. The computing for this research was supported by the NSF through TeraGrid resources provided by NCSA under Grant No. TG-DMS090007.

References

[1] D. Zhang, Stochastic Method for Flow in Porous Media: Coping with Uncertainties, Academic Press, San Diego, 2002.

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

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

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

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

[6] X. Ma, N. Zabaras, A stabilized stochastic finite element second-order projection method for modeling natural convection in random porous media, Journal of Computational Physics 227 (2008) 8448–8471.

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

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

[9] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, Journal of Computational Physics 225 (2007) 652–685.

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

[11] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, Journal of Computational Physics 228 (2009) 3084–3113.

[12] X. Ma, N. Zabaras, An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations, Journal of Computational Physics 229 (2010) 3884–3915.

[13] X. Ma, N. Zabaras, An efficient bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method, Inverse Problems 25 (2009) 035013.

[14] G. Lin, A. Tartakovsky, An efficient, high-order probabilistic collocation method on sparse grids for three-dimensional flow and solute transport in randomly heterogeneous porous media, Advances in Water Resources 32 (2009) 712–722.

[15] M. Loève, Probability Theory, fourth ed., Springer-Verlag, Berlin, 1977.

[16] I.T. Jolliffe, Principal Component Analysis, second ed., Springer-Verlag, New York, 2002.

[17] C. Desceliers, R. Ghanem, C. Soize, Maximum likelihood estimation of stochastic chaos representations from experimental data, International Journal for Numerical Methods in Engineering 66 (2006) 978–1001.

[18] R.G. Ghanem, A. Doostan, On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data, Journal of Computational Physics 217 (2006) 63–81.

[19] S. Das, R. Ghanem, J.C. Spall, Asymptotic sampling distribution for polynomial chaos representation from data: a maximum entropy and fisher information approach, SIAM Journal on Scientific Computing 30 (2008) 2207–2234.

[20] S. Das, R. Ghanem, S. Finette, Polynomial chaos representation of spatio-temporal random fields from experimental measurements, Journal of Computational Physics 228 (2009) 8726–8751.

[21] G. Stefanou, A. Nouy, A. Clement, Identification of random shapes from images through polynomial chaos expansion of random level set functions, International Journal for Numerical Methods in Engineering 79 (2009) 127–155.

[22] M. Arnst, R. Ghanem, C. Soize, Identification of Bayesian posteriors for coefficients of chaos expansions, Journal of Computational Physics 229 (2010) 3134–3154.

[23] C. Soize, Identification of high-dimension polynomial chaos expansions with random coefficients for non-gaussian tensor-valued random fields using partial and limited experimental data, Computer Methods in Applied Mechanics and Engineering 199 (2010) 2150–2164.

[24] M. Rosenblatt, Remarks on a multivariate transformation, Ann. Math. Statist. 23 (1952) 470–472.

[25] K. Sargsyan, B. Debusschere, H. Najm, O.L. Maître, Spectral representation and reduced order modeling of the dynamics of stochastic reaction networks via adaptive data partitioning, SIAM Journal on Scientific Computing 31 (2010) 4395–4421.

[26] I. Babuška, K.-M. Liu, R. Tempone, Solving stochastic partial differential equations based on the experimental data, Mathematical Models and Methods in Applied Sciences 13 (2003) 415–444.

[27] B. Ganapathysubramanian, N. Zabaras, Modeling diffusion in random heterogeneous media: data-driven models, stochastic collocation and the variational multiscale method, Journal of Computational Physics 226 (2007) 326–353.

[28] N. Agarwal, N.R. Aluru, A data-driven stochastic collocation approach for uncertainty quantification in mems, International Journal for Numerical Methods in Engineering 83 (2010) 575–597.

[29] B. Ganapathysubramanian, N. Zabaras, A non-linear dimension reduction methodology for generating data-driven stochastic input models, Journal of Computational Physics 227 (2008) 6612–6637.

[30] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

[31] B. Schlkopf, A. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.

[32] B. Schlkopf, A. Smola, Learning with Kernels, MIT Press, 2002.

[33] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[34] S. Mika, S. Bernhard, S. Alexander, M. Klaus-Robert, M. Scholz, G. Ratsch, Kernel PCA and de-noising in feature spaces, Advances in Neural Information Processing Systems, vol. 11, MIT Press, 1999, pp. 536–542.

[35] Y. Rathi, S. Dambreville, A. Tannenbaum, Statistical shape analysis using kernel PCA, in: Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, SPIE, 2006, p. 60641B.

[36] P. Sarma, L.J. Durlofsky, K. Aziz, Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics, Mathematical Geosciences 40 (2008) 3–32.

[37] C. Scheidt, J. Caers, Representing spatial uncertainty using distances and kernels, Mathematical Geosciences 41 (2009) 397–419.

[38] J.-Y. Kwok, I.-H. Tsang, The pre-image problem in kernel methods, IEEE Transactions on Neural Networks 15 (2004) 1517–1525.

[39] D. Venturi, A fully symmetric nonlinear biorthogonal decomposition theory for random fields, Physica D 240 (4–5) (2011) 415–425.

[40] S.S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, International Journal for Numerical Methods in Fluids 34 (2000) 425–448.

[41] A.W. Bowman, A. Azzalini, Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plust Illustrations, Oxford University Press, 1997.

[42] C.K. Williams, On a connection between kernel PCA and metric multidimensional scaling, Machine Learning 46 (2002) 11–19.

[43] S. Strebelle, Conditional simulation of complex geological structures using multiple-point statistics, Mathematical Geology 34 (2002) 1–21.

[44] X. Ma, N. Zabaras, A stochastic mixed finite element heterogeneous multiscale method for flow in porous media, Journal of Computational Physics 230 (2011) 4696–4722.

[45] S. Suzuki, J. Caers, A distance-based prior model parameterization for constraining solutions of spatial inverse problems, Mathematical Geosciences 40 (2008) 445–469.

[46] S. Torquato, Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Springer-Verlag, New York, 2002.

[47] D. Venturi, X. Wan, G.E. Karniadakis, Stochastic low-dimensional modelling of a random laminar wake past a circular cylinder, Journal of Fluid Mechanics 606 (2008) 339–367.