A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction

Sijing Li[a], Zhiwen Zhang[a,*], Hongkai Zhao[b]

[a]*Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China.*
[b]*Department of Mathematics, University of California at Irvine, Irvine, CA 92697, USA.*

Abstract

We propose a data-driven approach to solve multiscale elliptic PDEs with random coefficients based on the intrinsic approximate low dimensional structure of the underlying elliptic differential operators. Our method consists of offline and online stages. At the offline stage, a low dimensional space and its basis are extracted from solution samples to achieve significant dimension reduction in the solution space. At the online stage, the extracted data-driven basis will be used to solve a new multiscale elliptic PDE efficiently. The existence of approximate low dimensional structure is established in two scenarios based on: (1) high separability of the underlying Green's functions; and (2) smooth dependence of the parameters in the random coefficients. Various online construction methods are proposed for different problem setups. We provide error analysis based on the sampling error and the truncation threshold in building the data-driven basis. Finally, we present extensive numerical examples to demonstrate the accuracy and efficiency of the proposed method.

AMS subject classification: 35J08, 35J15, 35R60, 65N30, 65N80, 78M34.

Keywords: multiscale elliptic PDEs with random coefficients; uncertainty quantification (UQ); Green's function; separability; proper orthogonal decomposition (POD); neural network.

1. Introduction

In this paper, we shall develop a data-driven method to solve the following multiscale elliptic PDEs with random coefficients $a(x,\omega)$ and source $f(x,\theta)$,

$$\mathcal{L}(x,\omega)u(x,\omega,\theta) \equiv -\nabla\cdot\big(a(x,\omega)\nabla u(x,\omega,\theta)\big) = f(x,\theta), \quad x\in D, \quad \omega\in\Omega_\omega, \quad \theta\in\Omega_\theta, \tag{1}$$

$$u(x,\omega,\theta) = 0, \qquad x\in\partial D, \tag{2}$$

where $D\in\mathbb{R}^d$ is a bounded spatial domain. We separate the randomness in the coefficient and source, where Ω_ω and Ω_θ to denote the sample spaces for random variables ω and θ respectively and treat them differently as we shall see later. We assume $f(x,\theta)$ to be in $L^2(D)$ and uniform ellipticity of the PDE (see Section 2 for precise definition of the problem).

The problem (1)-(2) can be used to model the flow pressure in porous media such as water aquifer and oil reservoirs, where the permeability field $a(x,\omega)$ is a random field whose exact values are infeasible to obtain in practice due to the low resolution of seismic data.

*Corresponding author
Email addresses: lsj17@hku.hk (Sijing Li), zhangzw@hku.hk (Zhiwen Zhang), zhao@math.uci.edu (Hongkai Zhao)

In recent years, there has been an increased interest in quantifying the uncertainty in systems with randomness, i.e., solving stochastic partial differential equations (SPDEs, i.e., PDEs driven by Brownian motion) or partial differential equations with random coefficients (RPDEs). Uncertainty quantification (UQ) is an emerging research area to address these issues; see [20, 44, 5, 42, 4, 35, 34, 37, 39, 41, 13, 14, 22] and references therein. However, when SPDEs or RPDEs involve multiscale features and/or high-dimensional random inputs, these problems become challenging due to high computational cost.

Recently, some progress has been made in developing numerical methods for multiscale PDEs with random coefficients; see [31, 3, 36, 2, 21, 1, 27, 45, 18, 15] and references therein. For example, data-driven stochastic methods to solve PDEs with random and/or multiscale coefficients were proposed in [12, 45, 28, 29]. They demonstrated through numerical experiments that those methods were efficient in solving RPDEs with many different source functions. However, the polynomial chaos expansion [20, 44] is used to represent the randomness in the solutions. Although the polynomial chaos expansion is general, it is a priori instead of problem specific. Hence many terms may be required in practice for an accurate approximation which induces the curse of dimensionality.

We aim to develop a new data-driven method to solve multiscale elliptic PDEs with randomness in (1) based on intrinsic dimension reduction in two scenarios. In the first case, the coefficient $a(x) \in L^\infty(D)$ is fixed while the random source can vary arbitrarily in $L^2(D)$ with a bounded norm. As long as the domain of observation for $u(x,\theta)$ is disjoint from the support of the source $f(x,\theta)$, the low-dimensional structure of the underlying solution space in the observation domain is implied by the high separability of the Green's function for uniformly elliptic operators [8], which provides the theoretical foundation for hierarchical low-rank approximation to the inverses of FEM matrices and other fast direct inverse solvers. In this case, the curse of the dimension of randomness θ in the source function can be avoided without the need of smooth dependence on the randomness. For the other case, the coefficient $a(x,\omega) \in L^\infty(D)$ varies with smooth dependence on ω while the source function is fixed. Since $u(x,\omega)$ depends smoothly on $a(x,\omega)$ and hence on ω as shown in [16], we show an approximate low dimensional structure in this case as well.

Based on the above observations, our method consists of two stages. In the offline stage, the approximate low dimensional structure is extracted by computing a set of data-driven and problem specific basis from solution samples. For example, the data can be generated by solving (1)-(2) corresponding to a sampling of the coefficient $a(x,\omega)$ and/or source $f(x,\theta)$. Here, different sampling methods can be applied, including Monte Carlo (MC) method and quasi-Monte Carlo (qMC) method. The sparse-grid based stochastic collocation method [11, 43, 35] also works when the dimension of the random variables is moderate. Or the data may come from field measurements directly in practice. Then the low-dimensional structure and the corresponding basis are extracted using model reduction methods, such as the proper orthogonal decomposition (POD) [10, 38, 9], a.k.a. principle component analysis (PCA), e.g., by efficient random algorithms [24] due to the approximate low rank structure. The key point is that once the dimension reduction is achieved, the online stage of computing the solution corresponding to a new coefficient and/or source becomes finding a linear combination of the (few) constructed basis functions to approximate the solution.

However, the map from the input randomness of the PDE to the expansion coefficients of the

solution in terms of the data driven basis can be highly nonlinear. We propose a few possible online strategies (see Section 3). For examples, if the coefficient is in parametric form, one can approximate the nonlinear map from the parameter domain to the expansion coefficients through interpolation or neural network approximation. Or one can apply Galerkin method using the extracted basis to solve (1)-(2) for a new coefficient. In practice, the coefficient or the source function of the PDE may not be available but censors can be deployed to record the solution at certain locations. In this case, one can compute the expansion coefficients of a new solution by least square fitting those measurements at designed locations. We also provide analysis and guidelines for sampling, dimension reduction, and other implementations of our methods.

The rest of the paper is organized as follows. In Section 2, we characterize the low dimensional structure in two scenarios for elliptic PDE (2). In section 3, we describe our new data-driven method and its detailed implementation. In Section 4, we present numerical results to demonstrate the efficiency of our method. Concluding remarks are made in Section 5.

2. Low-dimensional structures in the solution space

2.1. High separability of the Green's function of elliptic operators.

We first consider the scenario of a multiscale elliptic PDE with a random source. Let $\mathcal{L}(x) : V \to V'$ be a uniformly elliptic operator in a divergence form

$$\mathcal{L}(x)u(x) \equiv -\nabla \cdot (a(x)\nabla u(x)) \tag{3}$$

in a bounded Lipschitz domain $D \subset \mathbb{R}^d$, where $V = H_0^1(D)$ and $a(x) \in L^\infty(D)$. The uniformly elliptic assumption means that there exist $a_{\min}, a_{\max} > 0$, such that $a_{\min} < a(x) < a_{\max}$ for all $x \in D$. The contrast ratio $\kappa_a = \frac{a_{\max}}{a_{\min}}$ is an important factor in the stability and convergence analysis. We consider the Dirichlet boundary value problem with a random source $f(x, \theta)$, where θ is some random variable.

$$\mathcal{L}(x)u(x,\theta) = f(x,\theta), \quad \text{in } D, \quad u(x,\theta) = 0, \quad \text{on } \partial D. \tag{4}$$

For all $x, y \in D$, the Green's function $G(x, y)$ for differential operator $\mathcal{L}$ is the solution of

$$\mathcal{L}G(\cdot, y) = \delta(\cdot, y), \quad \text{in } D, \quad G(\cdot, y) = 0, \quad \text{on } \partial D, \tag{5}$$

where $\mathcal{L}$ refers to the first variable $\cdot$ and $\delta(\cdot, y)$ is the Dirac delta function denoting an impulse source point at $y \in D$. The Green's function $G(x, y)$ is the Schwartz kernel of the inverse $\mathcal{L}^{-1}$, i.e., the solution of (4) is represented by

$$u(x,\theta) = \mathcal{L}^{-1}f(x,\theta) = \int_D G(x,y)f(y,\theta)dy. \tag{6}$$

Since the coefficient $a(x)$ is only bounded, the $G(x, y)$ can have a lower regularity, compared with the Green's function associated with the Poisson's equation. In [23], the authors proved

the existence of Green's function for $d \geq 3$ and the estimate $|G(x,y)| \leq \frac{C(d,\kappa_a)}{a_{\min}}|x-y|^{2-d}$, where $C(d,\kappa_a)$ is a constant depends on d and κ_a. For $d=2$ the existence of the Green's function was proved in [17] together with the estimate $|G(x,y)| \leq \frac{C(\kappa_a)}{a_{\min}} \log|x-y|$. Thus, when $\mathcal{L}$ is an uniform elliptic operator, $\mathcal{L}^{-1}$ exists and $||\mathcal{L}^{-1}|| \leq C a_{\min}^{-1}$, where C depends on d and κ_a.

One can show the existence of a low dimensional structure in the solution space based on high separability of the underlying Green's function [8] as follows.

Proposition 2.1. *Let $D \subset \mathbb{R}^d$ be a convex domain and X be a closed subspace of $L^2(D)$. Then for any integer $k \in \mathbb{N}$ there is a subspace $V_k \subset X$ satisfying $\dim V_k \leq k$ such that*

$$\operatorname{dist}_{L^2(D)}(u, V_k) \leq C \frac{\operatorname{diam}(D)}{\sqrt[d]{k}} \|\nabla u\|_{L^2(D)}, \quad \text{for all } u \in X \cap H^1(D), \tag{7}$$

where the constant C depends only on the spatial dimension d.

The proof is based on the Poincaré inequality; see [8]. All distances and diameters use the Euclidean norm in $\mathbb{R}^d$ except the distance of functions which uses the $L^2(D)$-norm. In particular, a choice of V_K in Prop. 2.1 is the L^2 projection of piece-wise constant functions defined on a grid with grid size $\frac{\operatorname{diam}(D)}{\sqrt[d]{k}}$ onto X.

Now we present the definition of $\mathcal{L}$-harmonic function on a domain $E \subset D$ introduced in [8]. A function u is $\mathcal{L}$-harmonic on E if $u \in H^1(\hat{E}), \forall \hat{E} \subset E$ with $dist(\hat{E}, \partial E) > 0$ and satisfies

$$a(u,\varphi) = \int_E a(x) \nabla u(x) \cdot \nabla \varphi(x) dx = 0 \quad \forall \varphi \in C_0^\infty(E).$$

Denote the space of $\mathcal{L}$-harmonic functions on E by $X(E)$, which is closed in $L^2(E)$. The following key Lemma shows that the space of $\mathcal{L}$-harmonic function has an approximate low dimensional structure.

Lemma 2.2 (Lemma 2.6 of [8])**.** *Let $\hat{E} \subset E \subset D$ in R^d and assume that $\hat{E}$ is convex such that*

$$dist(\hat{E}, \partial E) \geq \rho \; diam(\hat{E}) > 0, \quad \text{for some constant } \rho > 0.$$

Then for any $1 > \epsilon > 0$, there is a subspace $W \subset X(\hat{E})$ so that for all $u \in X(E)$,

$$dist_{L^2(\hat{E})}(u, W) \leq \epsilon \|u\|_{L^2(\hat{E})}$$

and

$$\dim(W) \leq c^d(\kappa_a, \rho) \big(|\log \epsilon|\big)^{d+1},$$

where $c(\kappa_a, \rho) > 0$ is a constant that depends on ρ and κ_a.

The key property of $\mathcal{L}$-harmonic functions used to prove the above result is the Caccioppoli inequality, which provides the estimate $\|\nabla u\|_{L^2(\hat{\mathbb{E}})} \leq C(\kappa_a, \rho)\|u\|_{L^2(E)}$. In particular, the Green's function $G(\cdot, y)$ is $\mathcal{L}$-harmonic on E if $y \notin E$. Moreover, given two disjoint domains D_1, D_2 in D, the Green's function $G(x,y)$ with $x \in D_1, y \in D_2$ can be viewed as a family of $\mathcal{L}$-harmonic functions on D_1 parameterized by $y \in D_2$. From the above Lemma one can easily deduce the following result which shows the high separability of the Green's function for the elliptic operator (3).

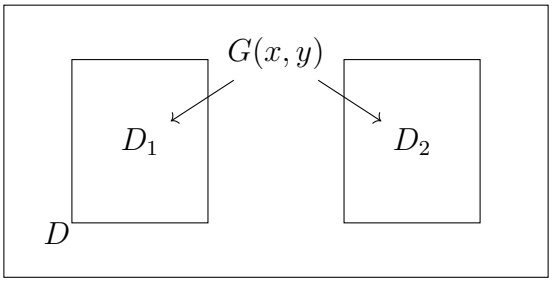


Figure 1: Green's function $G(x,y)$ with dependence on $x \in D_1$ and $y \in D_2$.

Proposition 2.3 (Theorem 2.8 of [8]). *Let $D_1, D_2 \subset D$ be two subdomains and D_1 be convex (see e.g. Figure 1). Assume that there exists $\rho > 0$ such that*

$$0 < \ \operatorname{diam}(D_1) \leq \rho \ \operatorname{dist}(D_1, D_2). \tag{8}$$

Then, for any $\epsilon \in (0,1)$ there is a separable approximation

$$G_k(x,y) = \sum_{i=1}^{k} u_i(x) v_i(y) \quad \text{with } k \leq c^d(\kappa_a, \rho) |\log \epsilon|^{d+1}, \tag{9}$$

so that for all $y \in D_2$

$$\|G(\cdot,y) - G_k(\cdot,y)\|_{L^2(D_1)} \leq \epsilon \|G(\cdot,y)\|_{L^2(\hat{D}_1)}, \tag{10}$$

where $\hat{D}_1 := \{x \in D : 2\rho \ \operatorname{dist}(x, D_1) \leq \operatorname{diam}(D_1)\}$.

The above Theorem shows that there exists a low dimensional linear subspace, e.g., spanned by $u_i(\cdot)$, that can approximate the family of functions $G(\cdot,y)$ well in $L^2(D_1)$ uniformly with respect to $y \in D_2$. Moreover, if $\operatorname{supp}(f(x,\theta)) \subset D_2$, one can approximate the family of solutions $u(x,\theta)$ to (4) by the same space well in $L^2(D_1)$ uniformly. Let

$$u_f(x,\theta) = \int_{D_2} G(x,y) f(y,\theta) dy \tag{11}$$

and

$$u_f^\epsilon(x,\theta) = \int_{D_2} G_k(x,y) f(y,\theta) dy = \sum_{i=1}^{k} u_i(x) \int_{D_2} v_i(y) f(y,\theta) dy. \tag{12}$$

Hence

$$\begin{aligned} \|u_f(\cdot,\theta) - u_f^\epsilon(\cdot,\theta)\|^2_{L^2(D_1)} &= \textstyle\int_{D_1} \left[\int_{D_2} (G(x,y) - G_k(x,y)) f(y,\theta) dy \right]^2 dx \\ &\leq \|f\|^2_{L^2(D_2)} \textstyle\int_{D_2} \|G(\cdot,y) - G_k(\cdot,y)\|^2_{L^2(D_1)} dy \leq C(D_1, D_2, \kappa_a, d) \epsilon^2 \|f\|^2_{L^2(D_2)}, \end{aligned} \tag{13}$$

since $\|G(\cdot,y)\|_{L^2(\hat{D}_1)}$ is bounded uniformly with respect to $y \in D_2$ by a positive constant that depends on D_1, D_2, κ_a, d due to the uniform ellipticity. Note that the low dimensional structure does not need any regularity assumption in $a(x)$. Moreover, dependence of the source on randomness can be arbitrary in terms of dimensionality and regularity.

Remark 2.1. Although, the proof of high separability of the Green's function requires $x \in D_1, y \in D_2$ for two disjoint D_1 and D_2 due to the singularity of the Green's function at $x = y$, the above approximation of the solution u in a domain disjoint with the support of f also works for u in the whole domain even when f is a globally supported smooth function as shown in our numerical tests.

Remark 2.2. Contrary to the elliptic operator, it is shown [19] that the Green's function for the high frequency Helmholtz equation is not highly separable due to fast decorrelation of two Green's functions with well separated (in terms of the wavelength) sources.

2.2. Low dimensional structures with respect to random coefficients

In the second scenario we consider the following elliptic PDEs with random coefficients:

$$\mathcal{L}(x,\omega)u(x,\omega) \equiv -\nabla \cdot \big(a(x,\omega)\nabla u(x,\omega)\big) = f(x), \quad x \in D, \quad \omega \in \Omega_\omega, \tag{14}$$

$$u(x,\omega) = 0, \qquad x \in \partial D, \tag{15}$$

where $D \in \mathbb{R}^d$ is a bounded spatial domain, Ω_ω is a sample space, and the source function $f(x) \in L^2(D)$. We assume the random coefficient $a(x,\omega)$ in (14) is almost surely uniformly elliptic, namely, there exist $a_{\min}, a_{\max} > 0$, such that

$$P\big(\omega \in \Omega_\omega : a(x,\omega) \in [a_{\min}, a_{\max}], \forall x \in D\big) = 1. \tag{16}$$

In addition, we assume the random coefficient $a(x,\omega)$ is parameterized by r independent random variables. For example, a commonly used affine form is the following,

$$a(x,\omega) = \bar{a}(x) + \sum_{m=1}^{r} a_m(x)\xi_m(\omega), \tag{17}$$

where $\xi_m(\omega)$, $m = 1, ..., r$ are i.i.d. uniform random variables in $[-1, 1]$. The random coefficient (17) can be used in multiscale random elliptic PDEs, such as elliptic PDEs with highly oscillatory and/or high-contrast coefficients.

Once a parametric form of the random coefficient $a(x,\omega)$ is given, computing the solution $u(x,\omega)$ of the problem (14)-(15) defines a solution map from the parameter domain $\boldsymbol{\xi}(\omega) = [\xi_1(\omega), \cdots, \xi_r(\omega)]^T \in \mathcal{U} = [-1, 1]^r$ to the solution space

$$\boldsymbol{\xi}(\omega) \mapsto u(x,\omega) = u(x, \boldsymbol{\xi}(\omega)) \in H_0^1(D), \tag{18}$$

which is a Banach-space-valued function of the random input vector $\boldsymbol{\xi}(\omega)$. With the uniform ellipticity assumption of $a(x, \boldsymbol{\xi}(\omega))$ and its smooth dependence on the parameter $\boldsymbol{\xi}$, the solution $u(x, \boldsymbol{\xi})$ also depends smoothly on the parameters, which can be approximated via polynomial expansion in $\boldsymbol{\xi}$ of the form

$$\sum_{\boldsymbol{\alpha} \in \mathcal{J}_r} u_{\boldsymbol{\alpha}}(x)\boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega), \tag{19}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_r)$ is a multi-index, $\mathcal{J}_r = \{\boldsymbol{\alpha} \,|\, \alpha_i \geq 0, \alpha_i \in \mathbb{N}, 1 \leq i \leq r\}$ is a multi-index set of countable cardinality, and $\boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega) = \prod_{1 \leq i \leq r} \xi_i^{\alpha_i}(\omega)$ is a multivariate polynomial.

In particular, if uniform ellipticity assumption of $a(x, \boldsymbol{\xi})$ has a holomorphic extension to an open set in complex domain that contains the real domain for $\boldsymbol{\xi}$, explicit estimates for the coefficients $u_{\boldsymbol{\alpha}}$ can be established similar to those estimates for the polynomial approximation for an analytic function. From the estimates, the following result for best n-term approximation can be proved (see [16] for details).

Proposition 2.4. *Consider a parametric problem of the form* (14)-(15) *with a random coefficient* (17). *Both the Taylor series and Legendre series of the form* (19) *converges to* $u(x, \boldsymbol{\xi}(\omega))$ *in* $H_0^1(D)$ *for all* $\boldsymbol{\xi}(\omega) \in \mathcal{U}$. *Moreover, for any set* $\mathfrak{J}_r^n$ *of indices corresponding to the* n *largest of* $||u_{\boldsymbol{\alpha}}(\cdot)||_{H_0^1(D)}$, *we have*

$$\sup_{\boldsymbol{\xi}(\omega) \in \mathcal{U}} \left|\left| u(\cdot, \boldsymbol{\xi}(\omega)) - \sum_{\boldsymbol{\alpha} \in \mathfrak{J}_r^n} u_{\boldsymbol{\alpha}}(\cdot) \boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega) \right|\right|_{H_0^1(D)} \leq C \exp(-c n^{1/r}), \tag{20}$$

where $\mathcal{J}_r^n$ *is a subset of* $\mathcal{J}_r$ *with cardinality* $\# \mathcal{J}_r^n = n$, C *and* c *are positive and depend on* r.

Prop.2.4 shows that there exists a linear subspace with dimension at most $O(n \sim (\frac{\log C}{c} + \frac{|\log \epsilon|}{c})^r)$, e.g., spanned by $u_{\boldsymbol{\alpha}}(x)$, $\boldsymbol{\alpha} \in \mathcal{J}_r^n$, that can approximate the solution of (14)-(15) with random coefficient within ϵ error.

The result in Prop.2.4 reveals the existence of approximate low dimensional structures in the solution space of (14)-(15). However, this approximation is obtained by mathematical techniques, which cannot be directly implemented via a computational algorithm. For instance, we cannot perform an exhaustive search over a huge index set to find $\mathfrak{J}_r^n$. Moreover, there may be problem dependent basis that can approximate the solution space more effectively than problem independent polynomial basis, which motivates our data-driven approach explained in Section 3

Remark 2.3. When the coefficient $a(x, \omega)$ is a nonlinear function of a finite number of random variables, one can apply the empirical interpolation method (EIM) [7] to approximately convert $a(x, \omega)$ into an affine form. Thus, low dimensional structures still exist in the solution space. In addition, we refer the reader to [26, 6] for the results of the best n-term polynomial approximation of elliptic PDEs with lognormal coefficients.

Remark 2.4. Although we present the problem and will develop the data-driven method for the elliptic problem (14)-(15) with scalar random coefficients $a(x, \omega)$, our method can be directly applied when the random coefficient is replaced by a symmetric positive definite tensor $a_{i,j}(x, \omega), i, j, = 1, \ldots, d$ with almost surely uniform ellipticity.

2.3. Some existing numerical methods for random elliptic PDEs

For the ease of the reader, we give a short review of existing methods for solving problem (14)-(15) involving random coefficients. There are basically two types of methods. In intrusive methods, one represents the solution of (14) by $u(x, \omega) = \sum_{\boldsymbol{\alpha} \in \mathcal{J}} u_{\boldsymbol{\alpha}}(x) H_{\boldsymbol{\alpha}}(\omega)$, where $\mathcal{J}$ is an index set, and $H_{\boldsymbol{\alpha}}(\omega)$ are certain basis functions (e.g. orthogonal polynomials or wavelet basis functions). Typical examples are the Wiener chaos expansion (WCE) and polynomial

chaos expansion (PCE) method. Then, one uses Galerkin method to compute the expansion coefficients $u_{\boldsymbol{\alpha}}(x)$; see e.g. [20, 44, 5, 33, 30, 34] and references therein. These methods have been successfully applied to many UQ problems, where the dimension of the random input is small or moderate. However, the number of basis functions increases exponentially fast with respect to the dimension of random input, i.e., they suffer from the curse of dimensionality of both the input space and the output (solution) space, because the random basis $H_{\boldsymbol{\alpha}}(\omega)$'s are built *a priori* based on the random variables in $a(x,\omega)$.

In non-intrusive methods, one can use the MC method or qMC method to solve (14)-(15). However, the convergence rate is slow and the method becomes more expensive when the coefficient $a(x,\omega)$ contains multiscale features. Stochastic collocation methods explore the smoothness of the solutions in the random space and use certain quadrature points and weights to compute the solutions [43, 4]. Exponential convergence can be achieved for smooth solutions, but the quadrature points grow exponentially fast as the number of random variables increases. Sparse grids can reduce the quadrature points to some extent [11, 35]. However, the sparse grid method still becomes very expensive when the dimension of randomness is modestly high.

3. Derivation of the new data-driven method

The results in Prop.2.3 and Prop.2.4 show that there exist low dimensional structures in the solution space of multiscale elliptic PDEs with random coefficient and source. Our goal is to use problem-specific and data-driven approaches to achieve a significant dimension reduction. The low dimensional structures in the solution space are extracted directly from the data, e.g., real measurements. The data-driven approach can also allow one to deal with situations where it is difficult to have an accurate full model, e.g., $a(x,\omega)$, or too expensive to solve a large scale problem in real practice. As demonstrated by our experiments, we find that the dimension of the extracted low dimensional space mainly depends on κ_a (namely $a_{\min}$ and $a_{\max}$) and very mildly on the dimension of the random input. Therefore, the curse of dimension can be alleviated. From now on, we use ω to denote randomness in both coefficient a and source f when there is no confusion.

Our method consists of offline and online stages. In the offline stage, we extract the low dimensional structure and a set of data-driven basis functions from solution samples. For example, a set of solution samples $\{u(x,\omega_i)\}_{i=1}^N$ can be obtained from measurements or generated by solving (14)-(15), e.g., with coefficient samples $\{a(x,\omega_i)\}_{i=1}^N$.

Let $V_{snap} = \{u|_{\hat{D}}(x,\omega_1), ..., u|_{\hat{D}}(x,\omega_N)\}$ denote the solution samples, where $\hat{D} \subseteq D$ is a region where the solution is of interest. For instance, in the reservoir simulation one is interested in computing the pressure value $u(x,\omega)$ on a specific subdomain $\hat{D}$. We use POD [10, 38, 9], or a.k.a PCA, to find the optimal subspace and its orthonormal basis functions to approximate V_{snap} to a certain accuracy. Define the correlation matrix $\Sigma = (\sigma_{ij}) \in \mathbb{R}^{N\times N}$ with $\sigma_{ij} =< u(\cdot,\omega_i), u(\cdot,\omega_j) >_{\hat{D}}$, $i,j = 1,\ldots,N$, where $< \cdot,\cdot >_{\hat{D}}$ denotes the standard inner product on $L^2(\hat{D})$. Let the eigenvalues of the correlation matrix be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \ldots \geq \lambda_N \geq 0$ and the corresponding eigenfunctions be $\phi_1(x)$, $\phi_2(x), \ldots, \phi_N(x)$, which will be referred to as data-driven basis functions. The space spanned by the leading K data-driven basis functions has the following approximation property to V_{snap}.

Proposition 3.1.

$$\frac{\sum_{i=1}^{N}\left|\left|u(x,\omega_i)-\sum_{j=1}^{K}<u(\cdot,\omega_i),\phi_j(\cdot)>_{\hat{D}}\phi_j(x)\right|\right|^2_{L^2(\hat{D})}}{\sum_{i=1}^{N}\left|\left|u(x,\omega_i)\right|\right|^2_{L^2(\hat{D})}}=\frac{\sum_{s=K+1}^{N}\lambda_s}{\sum_{s=1}^{N}\lambda_s}. \tag{21}$$

First, we expect a fast decay in the eigenvalues λ_s so that a small set of data-driven basis ($K \ll N$) will be enough to approximate the solution samples well in the root mean square sense. Secondly, based on the existence of low dimensional structure, we expect that the data-driven basis, $\phi_1(x)$, $\phi_2(x), \ldots, \phi_K(x)$, can almost surely approximate the solution $u|_{\hat{D}}(x,\omega)$ well too under some sampling condition (see Section 3.4) by

$$u|_{\hat{D}}(x,\omega) \approx \sum_{j=1}^{K} c_j(\omega)\phi_j(x), \quad \text{a.s. } \omega \in \Omega_\omega, \tag{22}$$

where the data-driven basis functions $\phi_j(x)$, $j = 1, ..., K$ are defined on $\hat{D}$.

The computational costs of the offline stage mainly consist of two parts, if data are generated by simulation: (1) compute solution samples (of global problems); and (2) compute the data-driven basis by the POD method. This is a common nature for many model reduction methods. Effective sampling of solutions (see Section 3.4) and the use of randomized algorithms [24] for the singular value decomposition (SVD) (utilizing the low-rank structure) helps to reduce the offline computation cost.

Remark 3.1. In Prop.3.1 we construct the data-driven basis functions from eigen-decomposition of the correlation matrix associated with the solution samples. Alternatively, we can subtract the mean from the solution samples, compute the covariance matrix, and construct the basis functions from eigen-decomposition of the covariance matrix. In this setting, the data-driven basis functions will be used to approximate the fluctuation of the solution since the mean function is given.

Now the problem is how to find $c_j(\omega)$ through an efficient online process given a new realization of $a(x,\omega)$. We will prescribe several strategies in different setups.

3.1. A nonlinear solution map

Suppose that $a(x,\omega)$ is parameterized by r independent random variables, i.e.,

$$a(x,\omega) = a(x,\xi_1(\omega),...,\xi_r(\omega)). \tag{23}$$

Thus, the solution can be represented as a functional of these random variables as well, i.e., $u(x,\omega) = u(x,\xi_1(\omega),...,\xi_r(\omega))$. Let $\boldsymbol{\xi}(\omega) = [\xi_1(\omega),\cdots,\xi_r(\omega)]^T$ denote the random input vector and $\mathbf{c}(\omega) = [c_1(\omega),\cdots,c_K(\omega)]^T$ denote the vector of solution coefficients in (22). Now, the problem can be viewed as constructing a map from $\boldsymbol{\xi}(\omega)$ to $\mathbf{c}(\omega)$, denoted by $\mathbf{F}: \boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$, which is nonlinear. We approximate this nonlinear map through the sample solution set. Given a set of solution samples $\{u(x,\omega_i)\}_{i=1}^N$ corresponding to $\{\boldsymbol{\xi}(\omega_i)\}_{i=1}^N$, e.g., by solving (14)-(15) with $a(x,\xi_1(\omega_i),...,\xi_r(\omega_i))$, from which the set of data driven basis $\phi_j(x), j = 1, ..., K$

is obtained by using POD method as described above, we can easily compute the projection coefficients $\{\mathbf{c}(\omega_i)\}_{i=1}^N$ of $u|_{\hat{D}}(x,\omega_i)$ on $\phi_j(x)$, $j=1,...,K$, i.e., $c_j(\omega_i) =< u(x,\omega_i), \phi_j(x) >_{\hat{D}}$. From the data set, $F(\boldsymbol{\xi}(\omega_i)) = \mathbf{c}(\omega_i)$, $i=1,...,N$, we construct the map $\mathbf{F}$. Note the significant dimension reduction by reducing the map $\boldsymbol{\xi}(\omega) \mapsto u(x,\omega)$ to the map $\boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$. We provide several ways to construct $\mathbf{F}$, depending on the dimension of the random input vector. More implementation details will be explained in Section 4.

1. Interpolation.
   When the dimension of the random input r is small or moderate, one can use interpolation. In particular, if the solution samples correspond to $\boldsymbol{\xi}$ located on a uniform or sparse grid, standard polynomial interpolation can be used to approximate the coefficient c_j at a new point of $\boldsymbol{\xi}$. If the solution samples correspond to $\boldsymbol{\xi}$ at scattered points or the dimension of the random input r is moderate or high, one can first find a few nearest neighbors to the new point efficiently using the k-d tree algorithm [40] and then use moving least square approximation centered at the new point to approximate the mapped value. See Figure 5 for an example of the map $\mathbf{F}$ based on interpolation.

2. Neural network.
   When the dimension of the random input r is high, interpolation approach becomes expensive and less accurate. We tried a simple neural network with small output dimension (due to the dimension reduction) that seems to provide a satisfactory solution.

For the uniform-grids or sparse-grid based polynomial interpolation approach, the approximation property (error estimate) can be studied based on the regularity of map F, which is smooth with respect to $\boldsymbol{\xi}$ if $a(x,\boldsymbol{\xi}(\omega))$ depends on $\boldsymbol{\xi}$ smoothly. Our numerical results in Section 4 show the moving least square approach and neural network approach are efficient and accurate. However, since the map $\mathbf{F}$ is nonlinear and lives in a high dimensional space, many issues need to be further investigated, such as how to optimally choose the training samples and how to study the approximation property of the map F.

In the online stage, one can compute the solution $u(x,\omega)$ using the constructed map $\mathbf{F}$. For example, given a new realization of $a(x,\xi_1(\omega_i),...,\xi_r(\omega_i))$, we plug $\boldsymbol{\xi}(\omega)$ into the constructed map $\mathbf{F}$ to approximate $\mathbf{c}(\omega) = \mathbf{F}(\boldsymbol{\xi}(\omega))$, which are the projection coefficients of the solution on the data-driven basis. So we can quickly obtain the new solution $u|_{\hat{D}}(x,\omega)$ using Eq.(22), where the computational time is negligible. Once we obtain the numerical solutions, we can use them to compute statistical quantities of interest, such as mean, variance, and joint probability distributions.

3.2. Galerkin approach

In the case $\hat{D} = D$, we can solve the problem (14)-(15) on the whole domain D by the standard Galerkin formulation using the data driven basis for a new realization of $a(x,\omega)$.

Once the data driven basis $\phi_j(x)$, $j=1,...,K$, which are defined on the domain D, are obtained from solution samples in the offline stage. Given a new realization of the coefficient $a(x,\omega)$, we approximate the corresponding solution as

$$u(x,\omega) \approx \sum_{j=1}^{K} c_j(\omega)\phi_j(x), \quad \text{a.s. } \omega \in \Omega_\omega, \tag{24}$$

and use the Galerkin projection to determine the coefficients $c_j(\omega)$, $j = 1, ..., K$ by solving the following linear system in the online stage,

$$\sum_{j=1}^{K} \int_D a(x,\omega) c_j(\omega) \nabla\phi_j(x) \cdot \nabla\phi_l(x) dx = \int_D f(x)\phi_l(x)dx, \quad l = 1, ..., K. \tag{25}$$

Remark 3.2. The computational cost of solving the linear system (25) is small compared to using a Galerkin method, such as the finite element method, directly for $u(x,\omega)$ because K is much smaller than the degree of freedom needed to discretize $u(x,\omega)$ in the whole domain.

Note that if $a(x,\omega)$ has the affine form (17), we first compute the terms that do not depend on randomness, including $\int_D \bar{a}(x)\nabla\phi_j(x) \cdot \nabla\phi_l(x)dx$, $\int_D a_m(x)\nabla\phi_j(x) \cdot \nabla\phi_l(x)dx$ and $\int_D f(x)\phi_j(x)dx$, $j, l = 1, ..., K$. Then, we save them in the offline stage. This leads to considerable savings in assembling the stiffness matrix for each new realization of the coefficient $a(x,\omega)$ in the online stage. Of course, the affine form is automatically parameterized. Hence, one can also construct the map $\mathbf{F} : \boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$ as described in the previous Section 3.1.

3.3. Least square fitting from direct measurements at selected locations

In many applications, only samples (data) or measurements of $u(x,\omega)$ is available while the model of $a(x,\omega)$ or its realization is not known. In this case, we propose to compute the coefficients $\mathbf{c}$ by least square fitting the measurements (values) of $u(x,\omega)$ at appropriately selected locations. First, as before, from a set of solutions samples, $u(x_j,\omega_i)$, measured on a mesh $x_j \in \hat{D}, j = 1, \ldots, J$, one finds a set of data driven basis $\phi_1(x_j), \ldots, \phi_K(x_j)$, e.g. using POD. For a new solution $u(x,\omega)$ measured at $x_1, x_2, \ldots, x_M$, one can set up the following least square problem to find $\mathbf{c} = [c_1, \ldots, c_K]^T$ such that $u(x,\omega) \approx \sum_{k=1}^{K} c_k\phi_k(x)$:

$$B\mathbf{c} = \mathbf{y}, \quad \mathbf{y} = [u(x_1,\omega), \ldots, u(x_M,\omega)]^T, B = [\boldsymbol{\phi}_1^M, \ldots, \boldsymbol{\phi}_K^M] \in R^{M\times K}, \tag{26}$$

where $\boldsymbol{\phi}_k^M = [\phi_k(x_1), \ldots, \phi_k(x_M)]^T$.

The key issue in practice is the conditioning of the least square problem (26). One way is to select the measurement (sensor) locations $x_1, \ldots x_M$ such that rows of B are as decorrelated as possible. We adopt the approach proposed in [32], where a QR factorization with pivoting for the matrix of data driven basis is used to determine the measurement locations. More specifically, let $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K] \in R^{J\times K}$, $\boldsymbol{\phi}_k = [\phi_k(x_1), \ldots, \phi_k(x_J)]^T$. If $M = K$, QR factorization with column pivoting is performed on Φ^T. If $M > K$, QR factorization with pivoting is performed on $\Phi\Phi^T$. The first M pivoting indices provide the measurement locations. More details can be found in [32].

3.4. Determine a set of good learning samples

A set of good solution samples is important for the construction of data-driven basis in the offline stage. Since the solution depends on the source linearly with an explicit bound, the analysis is straightforward. Here we provide an error analysis for the coefficient based on the finite element formulation. However, the results extend to general Galerkin formulation. First, we make a few assumptions.

Assumption 3.2. *Suppose $a(x,\omega)$ has the following property: given $\delta_1 > 0$, there exists an integer N_{δ_1} and a choice of snapshots $\{a(x,\omega_i)\}$, $i = 1, ..., N_{\delta_1}$ such that*

$$\mathbb{E}\left[\inf_{1\le i\le N_{\delta_1}} \left|\left|a(x,\omega) - a(x,\omega_i)\right|\right|_{L^\infty(D)}\right] \le \delta_1. \tag{27}$$

Let $\{a(x,\omega_i)\}_{i=1}^{N_{\delta_1}}$ denote the samples of the random coefficient, which form a δ_1-net for the coefficient $a(x,\omega)$. For every realization of $a(x,\omega)$, we can find a coefficient sample $a(x,\omega_i)$ that is close to $a(x,\omega)$ in the norm $||\cdot||_{L^\infty(D)}$. We define this δ_1-net in the sense of the expectation $\mathbb{E}[\cdot]$, which allows us to exclude a small set of outliers.

A good sampling of the solution is important for computational efficiency and accuracy. When the coefficient has the affine form (17), one can verify Asm.3.2 and provide a constructive way to sample snapshots $\{a(x,\omega_i)\}_{i=1}^{N_{\delta_1}}$ if we know the distribution of the random variables $\xi_m(\omega)$, $m = 1, ..., r$, since the linear map from $\boldsymbol{\xi}$ space to the function space of $a(x,\boldsymbol{\xi})$ is explicitly determined by $\overline{a}(x), a_m(x), m = 1, ..., r$. In general, it becomes a sampling problem for $\{a(x,\omega_i)\}$, which may be challenging especially when the dimension of the random variables r is high and/or $a(x,\omega)$ does not have an affine form. However, the Asm.3.2 provides us some insight on how to choose coefficient samples $\{a(x,\omega_i)\}$ in order to obtain a set of accurate data-driven basis functions.

Let $V_h \subset H_0^1(D)$ denote a finite element space that is spanned by nodal basis functions on a mesh with size h and $\tilde{V}_h \subset V_h$ denote the space spanned by the data-driven basis $\{\phi_j(x)\}_{j=1}^K$. We assume the mesh size is fine enough so that the finite element space can approximate the solutions to the underlying PDEs well. For each $a(x,\omega_i)$, let $u_h(x,\omega_i) \in V_h$ denote the FEM solution and $\tilde{u}_h(x,\omega_i) \in \tilde{V}_h$ denote the projection on the data-driven basis $\{\phi_j(x)\}_{j=1}^K$.

Assumption 3.3. *Given $\delta_2 > 0$, we can find a set of data-driven basis, $\phi_1, \ldots, \phi_{K_{\delta_2}}$ such that*

$$||u_h(x,\omega_i) - \tilde{u}_h(x,\omega_i)||_{L^2(D)} \le \delta_2, \ \forall 1 \le i \le K_{\delta_2}, \tag{28}$$

where $\tilde{u}_h(x,\omega_i)$ is the L^2 projection of $u_h(x,\omega_i)$ onto the space spanned by $\phi_1, \ldots, \phi_{K_{\delta_2}}$.

Asm.3.3 can be verified by setting the threshold in the POD method; see Prop.3.1. Now we present the following error estimate.

Theorem 3.4. *Under Assumptions 3.2-3.3, for any $\delta_i > 0$, $i = 1, 2$, we can choose the samples of the random coefficient $\{a(x,\omega_i)\}_{i=1}^{N_{\delta_1}}$ and the threshold in constructing the data-driven basis accordingly, such that*

$$\mathbb{E}\left[\left|\left|u_h(x,\omega) - \tilde{u}_h(x,\omega)\right|\right|_{L^2(D)}\right] \le C\delta_1 + \delta_2, \tag{29}$$

where C depends on $a_{\min}$, $f(x)$, and the domain D.

Proof. Given a coefficient $a(x,\omega)$, let $u_h(x,\omega)$ and $\tilde{u}_h(x,\omega)$ be the corresponding FEM solution and data-driven solution, respectively. We have

$$
\begin{aligned}
&\left|\left|u_h(x,\omega)-\tilde{u}_h(x,\omega)\right|\right|_{L^2(D)}\\
\leq&\left|\left|u_h(x,\omega)-u_h(x,\omega_i)\right|\right|_{L^2(D)}+\left|\left|u_h(x,\omega_i)-\tilde{u}_h(x,\omega_i)\right|\right|_{L^2(D)}+\left|\left|\tilde{u}_h(x,\omega_i)-\tilde{u}_h(x,\omega)\right|\right|_{L^2(D)},\\
:=&I_1+I_2+I_3,
\end{aligned}
\tag{30}
$$

where $u_h(x,\omega_i)$ is the solution corresponding to the coefficient $a(x,\omega_i)$ and $\tilde{u}_h(x,\omega_i)$ is its projection. Now we estimate the error term I_1 first. In the sense of weak form, we have

$$
\int_D a(x,\omega)\nabla u_h(x,\omega)\cdot\nabla v_h(x)dx=\int_D f(x)v_h(x),\quad \text{for all}\quad v_h(x)\in V_h,
\tag{31}
$$

and

$$
\int_D a(x,\omega_i)\nabla u_h(x,\omega_i)\cdot\nabla v_h(x)dx=\int_D f(x)v_h(x),\quad \text{for all}\quad v_h(x)\in V_h.
\tag{32}
$$

Subtracting the above two variational formulations (31)-(32), we have, for all $v_h(x)\in V_h$,

$$
\int_D a(x,\omega)\nabla(u_h(x,\omega)-u_h(x,\omega_i))\cdot\nabla v_h(x)dx=-\int_D(a(x,\omega)-a(x,\omega_i))\nabla u_h(x,\omega_i)\cdot\nabla v_h(x).
\tag{33}
$$

Let $w_h(x)=u_h(x,\omega)-u_h(x,\omega_i)$ and $L(v_h)=-\int_D(a(x,\omega)-a(x,\omega_i))\nabla u_h(x,\omega_i)\cdot\nabla v_h(x)$ denote a linear form. Eq.(33) means that $w_h(x,\omega)$ is the solution of the weak form $\int_D a(x,\omega)\nabla w_h\cdot\nabla v_h(x)dx=L(v_h)$, for all $v_h(x)\in V_h$. Therefore, we have

$$
\left|\left|w_h(x)\right|\right|_{H^1(D)}\leq\frac{||L||_{H^1(D)}}{a_{\min}}.
\tag{34}
$$

Notice that

$$
\begin{aligned}
||L||_{H^1(D)}=\max_{||v_h||_{H^1(D)}=1}|L(v_h)|&\leq||a(x,\omega)-a(x,\omega_i)||_{L^\infty(D)}||u_h(x,\omega_i)||_{H^1(D)},\\
&\leq||a(x,\omega)-a(x,\omega_i)||_{L^\infty(D)}\frac{||f(x)||_{H^1(D)}}{a_{\min}}.
\end{aligned}
\tag{35}
$$

Since $w_h(x)=0$ on ∂D, combining Eqns.(34)-(35) and using the Poincaré inequality for $w_h(x)$, we obtain an estimate for the term I_1 as follows,

$$
\begin{aligned}
\left|\left|u_h(x,\omega)-u_h(x,\omega_i)\right|\right|_{L^2(D)}&\leq C_1\left|\left|u_h(x,\omega)-u_h(x,\omega_i)\right|\right|_{H^1(D)}\\
&\leq C_1||a(x,\omega)-a(x,\omega_i)||_{L^\infty(D)}\frac{||f(x)||_{H^1(D)}}{a_{\min}^2},
\end{aligned}
\tag{36}
$$

where C_1 only depends on the domain D. For the term I_3 in Eq.(30), we can similarly get

$$\left|\left|\tilde{u}_h(x,\omega_i) - \tilde{u}_h(x,\omega)\right|\right|_{L^2(D)} \leq C_1 ||a(x,\omega) - a(x,\omega_i)||_{L^\infty(D)} \frac{||f(x)||_{H^1(D)}}{a_{\min}^2}. \tag{37}$$

The term I_2 in Eq.(30) can be controlled according to the Asm.3.3. Combining the estimates for terms I_1, I_2 and I_3 and integrating over the random space, we prove the theorem. □

Corollary 3.5. *If we use Monte Carlo method to compute the expectation in* (29)*, from the proof of Theorem 3.4 we still have*

$$\frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} \left|\left|u_h(x,\omega_j) - \tilde{u}_h(x,\omega_j)\right|\right|_{L^2(D)} \leq C\delta_1 + \delta_2, \tag{38}$$

where N_{MC} is the sample number and C, δ_1 and δ_2 are the same as in Theorem 3.4.

In this paper, we restrict attention to the approximation in the physical space. So we assume the sampling error (i.e., the error of approximation the expectation by a sample mean) is negligible. Theorem 3.4 or Corollary 3.5 indicate that the error between $u_h(x,\omega)$ and its approximation $\tilde{u}_h(x,\omega)$ using the data-driven basis consists of two parts. The first part depends on how well the random coefficient is sampled. While the second part depends on the truncation threshold in constructing the data-driven basis from the solution samples. In practice, a balance of these two factors gives us guidance on how to choose solution samples and the truncation threshold in the POD method to achieve optimal accuracy. Again, the key advantage for our data-driven approach for this form of elliptic PDEs is the low dimensional structure in the solution space which provides a significant dimension reduction.

4. Numerical results

In this section we will present various numerical results to demonstrate the accuracy and efficiency of our proposed data-driven method.

In all of our numerical experiments, we use the same uniform triangulation to implement the standard FEM and choose mesh size $h = \frac{1}{512}$ in order to resolve the multiscale information. We use $N = 2000$ samples in the offline stage to construct the data-driven basis and determine the number of basis K according to the decay rate of the eigenvalues of the correlation matrix $\Sigma = (\sigma_{ij})$ of the solution samples, i.e., $\sigma_{ij} =< u(x,\omega_i), u(x,\omega_j) >$, $i,j = 1,\ldots,N$. Let N_1 and N_2 denote the number of training samples in constructing the nonlinear map $\mathbf{F}$ and the number of testing samples in the online stage, respectively. We will choose $N_1 \ll N_2$.

In the numerical results, the testing error is the error between the numerical solution obtained by our mapping method and the reference solution obtained by the FEM on the same fine mesh used to compute the sample solutions. The projection error is the error between the FEM solution and its projection on the space spanned by data-driven basis, i.e. the best possible approximation error.

4.1. An example with deterministic multiscale coefficients and random sources

First of all, we consider a deterministic multiscale elliptic PDE with a random source defined on a square domain $D = [0,1] \times [0,1]$,

$$\begin{aligned} -\nabla \cdot (a(x,y)\nabla u(x,y,\theta)) &= f(x,y,\theta), \quad (x,y) \in D, \quad \theta \in \Omega_\theta, \\ u(x,y,\theta) &= 0, \qquad (x,y) \in \partial D. \end{aligned} \tag{39}$$

The multiscale coefficient $a(x,y)$ is defined as

$$\begin{aligned} a(x,y) =& 0.1 + \frac{2 + p_1 \sin(\frac{2\pi x}{\epsilon_1})}{2 - p_1 \cos(\frac{2\pi y}{\epsilon_1})} + \frac{2 + p_2 \sin(\frac{2\pi(x+y)}{\sqrt{2}\epsilon_2})}{2 - p_2 \sin(\frac{2\pi(x-y)}{\sqrt{2}\epsilon_2})} + \frac{2 + p_3 \cos(\frac{2\pi(x-0.5)}{\epsilon_3})}{2 - p_3 \cos(\frac{2\pi(y-0.5)}{\epsilon_3})} \\ &+ \frac{2 + p_4 \cos(\frac{2\pi(x-y)}{\sqrt{2}\epsilon_4})}{2 - p_4 \sin(\frac{2\pi(x+y)}{\sqrt{2}\epsilon_4})} + \frac{2 + p_5 \cos(\frac{2\pi(2x-y)}{\sqrt{5}\epsilon_5})}{2 - p_5 \sin(\frac{2\pi(x+2y)}{\sqrt{5}\epsilon_5})}, \end{aligned} \tag{40}$$

where $[p_1, p_2, p_3, p_4, p_5] = [1.98, 1.96, 1.94, 1.92, 1.9]$. We choose $D_1 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{11}{16}, \frac{15}{16}]$ and $D_2 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$. The source function $f(x,y,\theta)$ is a spatially uncorrelated white noise defined on D_2. Note that D_2 is partitioned into a 256×128 fine mesh. In this experiment, $f(x,y,\theta)$ is an independent Gaussian random variable on each mesh.

We generate $N = 2000$ samples of the source function $f(x,y,\theta)$. Then, we solve the problem (39) by using FEM and obtain 2000 solution samples $u(x,y,\theta)$. The eigenvalues of the correlation matrix of the solution samples $u(x,y,\theta)|_{D_2}$ is referred to as the eigenvalues of local problem. While the eigenvalues of the correlation matrix of the solution samples $u(x,y,\theta)$ on the whole domain D is referred to as the eigenvalues of global problem.

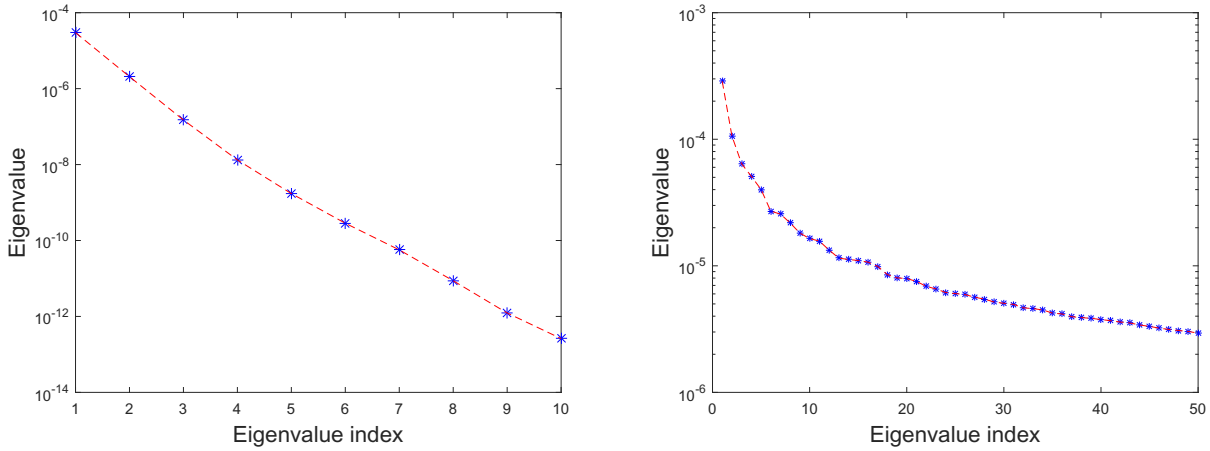
In Figure 2a, we plot the decay properties of the eigenvalues of the local problem. We see the fast decay in the eigenvalues of the correlation matrix, which reveals the existence of low dimensional structure in the solution space implied by Proposition 2.3. We also plot the decay properties of the eigenvalues of the global problem in Figure 2b. First 50 eigenvalues take up 96% of the total sum of the eigenvalues. It means that certain low dimensional structure still exists in the solution space of global problem, however, the dimension of such approximate space is larger than that of the local problem.

We change the distance between D_1 and D_2 and repeat the above experiment. In Figure 3, we plot the decay properties of the local problem. One can see that distance between D_1 and D_2 affects the effective dimension of the approximate solution space, which is embedded in the constant for separability estimate.

4.2. An example with random coefficient

Here we consider a multiscale elliptic PDE with a random coefficient that is defined on a square domain $D = [0,1] \times [0,1]$,

$$\begin{aligned} -\nabla \cdot (a(x,y,\omega)\nabla u(x,y,\omega)) &= f(x,y), \quad (x,y) \in D, \omega \in \Omega_\omega, \\ u(x,y,\omega) &= 0, \qquad (x,y) \in \partial D. \end{aligned} \tag{41}$$



(a) Decay of eigenvalues of local problem.

(b) Decay of eigenvalues of global problem.

Figure 2: The decay properties of the eigenvalues in the problem of Sec.4.1.

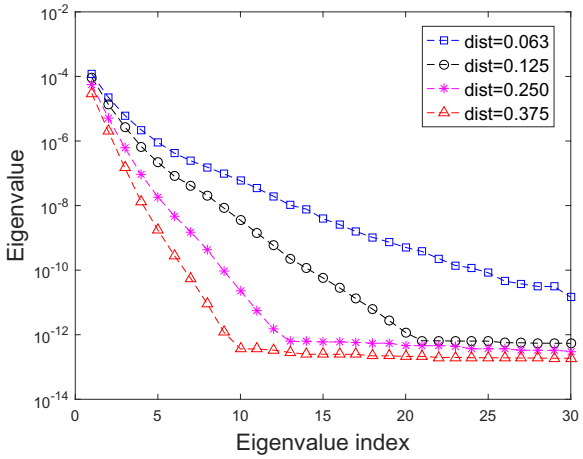


Figure 3: The decay properties of the eigenvalues for different separate distances.

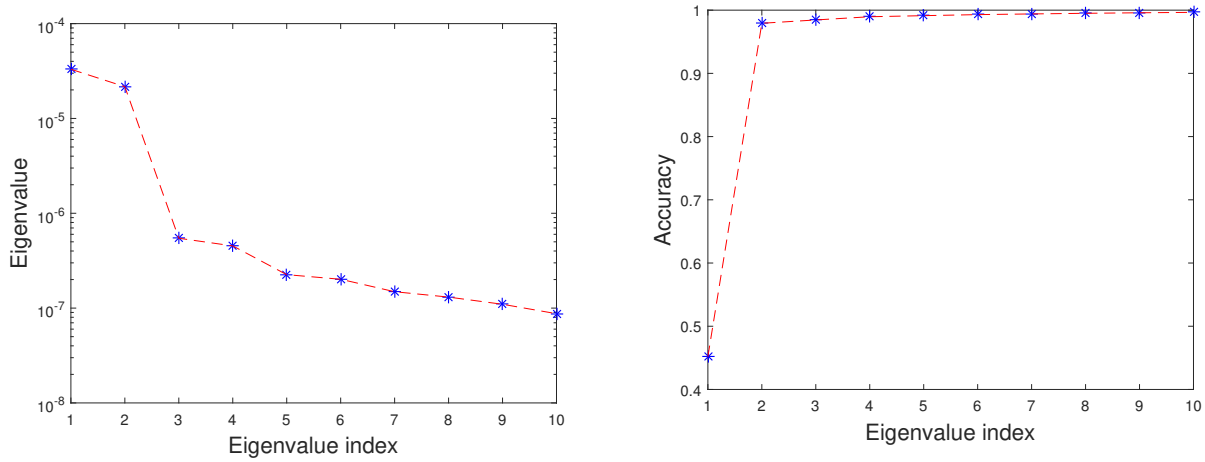
In this example, the coefficient $a(x, y, \omega)$ is defined as

$$
\begin{aligned}
a(x,y,\omega) =& 0.1 + \frac{2 + p_1 \sin(\frac{2\pi x}{\epsilon_1})}{2 - p_1 \cos(\frac{2\pi y}{\epsilon_1})}\xi_1(\omega) + \frac{2 + p_2 \sin(\frac{2\pi(x+y)}{\sqrt{2}\epsilon_2})}{2 - p_2 \sin(\frac{2\pi(x-y)}{\sqrt{2}\epsilon_2})}\xi_2(\omega) + \frac{2 + p_3 \cos(\frac{2\pi(x-0.5)}{\epsilon_3})}{2 - p_3 \cos(\frac{2\pi(y-0.5)}{\epsilon_3})}\xi_3(\omega) \\
&+ \frac{2 + p_4 \cos(\frac{2\pi(x-y)}{\sqrt{2}\epsilon_4})}{2 - p_4 \sin(\frac{2\pi(x+y)}{\sqrt{2}\epsilon_4})}\xi_4(\omega) + \frac{2 + p_5 \cos(\frac{2\pi(2x-y)}{\sqrt{5}\epsilon_5})}{2 - p_5 \sin(\frac{2\pi(x+2y)}{\sqrt{5}\epsilon_5})}\xi_5(\omega), \qquad (42)
\end{aligned}
$$

where $[\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5] = [\frac{1}{47}, \frac{1}{29}, \frac{1}{53}, \frac{1}{37}, \frac{1}{41}]$, $[p_1, p_2, p_3, p_4, p_5] = [1.98, 1.96, 1.94, 1.92, 1.9]$, and $\xi_i(\omega)$, $i = 1, ..., 5$ are i.i.d. uniform random variables in $[0, 1]$. The contrast ratio in the coefficient (42) is $\kappa_a \approx 4.5 \times 10^3$. The source function is $f(x, y) = \sin(2\pi x)\cos(2\pi y) \cdot I_{D_2}(x, y)$, where I_{D_2} is an indicator function defined on $D_2 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$. The coefficient (42) is highly oscillatory in the physical space. Therefore, one needs a fine discretization to resolve the small-scale variations in the problem. We shall show results for the solution to (41) with

coefficient (42) in: (1) a restricted subdomain $D_1 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{11}{16}, \frac{15}{16}]$ away from the support D_2 of the source term $f(x, y)$; and (2) the full domain D.

In Figure 4, we show the decay property of eigenvalues. Specifically, we show the magnitude of the eigenvalues in Figure 4a and the ratio of the accumulated sum of the leading eigenvalues over the total sum in Figure 4b. These results and Prop.3.1 imply that a few leading eigenvectors will provide a set of data-driven basis that can approximate all solution samples well.



(a) Decay of eigenvalues.

(b) $1 - \sqrt{\sum_{j=n+1}^{N} \lambda_j / \sum_{j=1}^{N} \lambda_j}$, $n = 1, 2, \ldots$.

Figure 4: The decay properties of the eigenvalues in the local problem of Sec.4.2.

After we construct the data-driven basis, we use the polynomial interpolation to approximate the map $\mathbf{F} : \boldsymbol{\xi} \mapsto \mathbf{c}(\boldsymbol{\xi})$. Notice that the coefficient of (42) is parameterized by five i.i.d. random variables. We partition the random space $[\xi_1(\omega), \xi_2(\omega), \cdots, \xi_5(\omega)]^T \in [0, 1]^5$ into a set of uniform grids in order to construct the map $\mathbf{F}$. Here we choose $N_1 = 9^5$ samples. We can choose other sampling strategies, such as sparse-grid points and Latin hypercube points, for moderate- or high-dimensional cases. In Figure 5, we show the profiles of the first two data-driven basis functions ϕ_1 and ϕ_2 and the plots of the maps $c_1(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ and $c_2(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ with fixed $[\xi_3, \xi_4, \xi_5]^T = [0.25, 0.5, 0.75]^T$. One can see that the data-driven basis functions contain multiscale features, while the maps $c_1(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ and $c_2(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ are smooth with respect to ξ_i, $i = 1, 2$. The behaviors of other data-driven basis functions and other maps are similar (not shown here). Once we get the map $\mathbf{F}$, the solution corresponding to a new realization $a(x, \boldsymbol{\xi}(\omega))$ can be computed easily by finding $\mathbf{c}(\boldsymbol{\xi})$ and plugging in the approximation (22).

In Figure 6, we show the mean relative L^2 and H^1 errors of the testing error and projection error, where $N_2 = 10N_1$. For the experiment, only four data-driven basis are needed to achieve a relative error less than 1% in L^2 norm and less than 2% in H^1 norm. Moreover, the numerical solution obtained by our mapping method is close to the projection solution, which is the best approximation of the reference solution by the data-driven basis. This result also indicates that the nonlinear map $\mathbf{F}$ is a smooth function and has been approximated well by the uniform grids based polynomials interpolation.

We also study the approximation property of the nonlinear map $\mathbf{F}$ based on different uni-

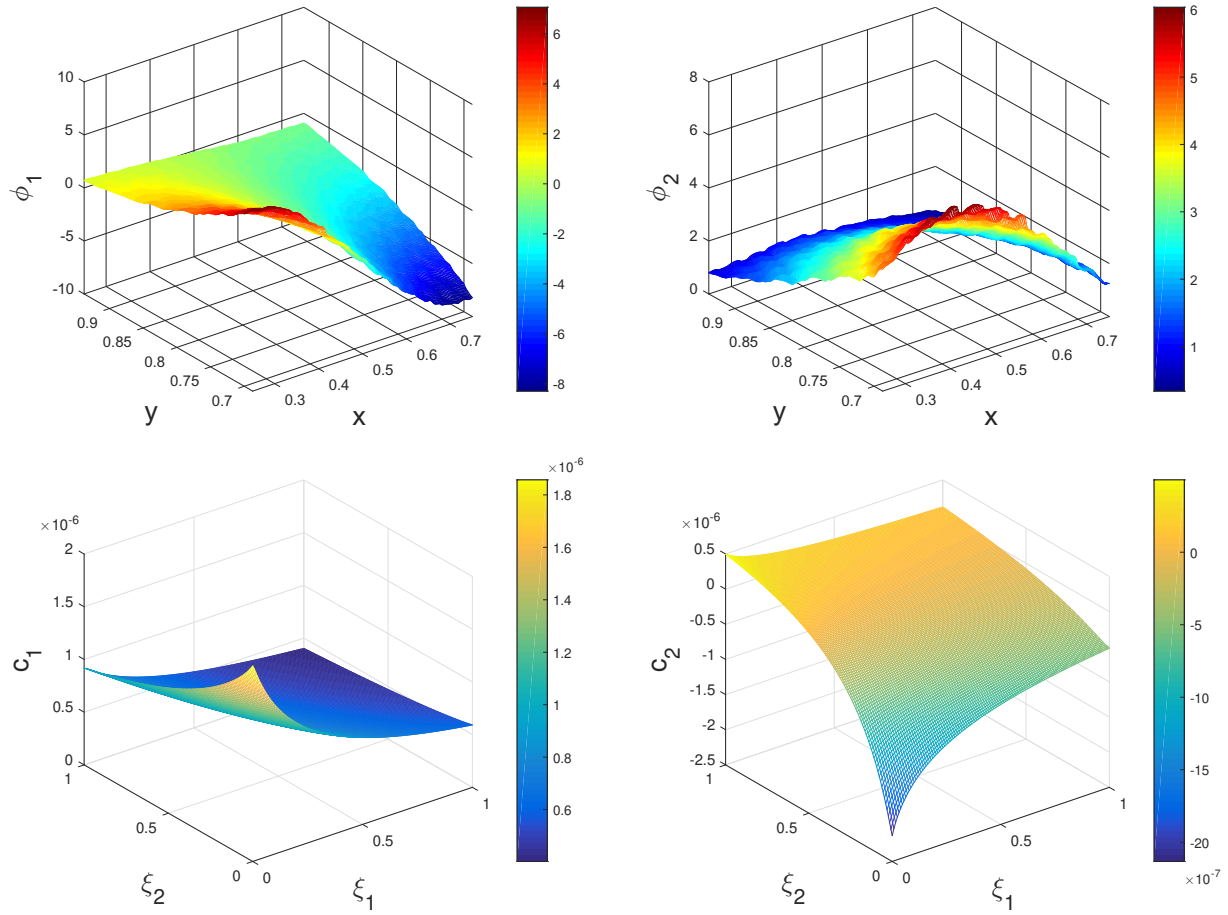


Figure 5: Top: profiles of data-driven basis ϕ_1 and ϕ_2. Bottom: profiles of the maps $c_1(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ and $c_2(\xi_1, \xi_2; \xi_3, \xi_4, \xi_5)$ with fixed $[\xi_3, \xi_4, \xi_5]^T = [0.25, 0.5, 0.75]^T$.

form grids. Specifically, we partition the random space $[\xi_1(\omega), \xi_2(\omega), \cdots, \xi_5(\omega)]^T \in [0,1]^5$ into different uniform grids with $N_1 = 5^5, 6^5, 7^5, 8^5, 9^5$ samples and use the polynomial interpolation to construct the map $\mathbf{F}$. Figure 7 shows the mean relative L^2 and H^1 errors of the testing errors and the project error (which does not depend on the grid partition), where $N_2 = 10^6$. We observe a convergence behavior in constructing the map $\mathbf{F}$ if we increase the partition number in the uniform grids.

The standard FEM takes 0.82 second to compute one solution. In the offline stage of our method, we need to compute N solution samples to construct POD basis and N_1 solution samples to construct the nonlinear map $\mathbf{F}$. The random SVD method takes 1.2 seconds to compute the POD basis. In the online stage of our method, the CPU time is almost negligible. For instance, when the number of basis is $K = 10$, it takes about 0.0022 second to compute one solution. In Figure 8, we compare the CPU time of the FEM and our method (including both stages) as a function of the number of new solutions computed in the online stage. This result shows that our method is very efficient if one needs to solve many forward problems for (41) (e.g. in the context of solving an inverse problem by using the Bayesian method).

In Figure 9, we show the accuracy of the proposed method when we use different number

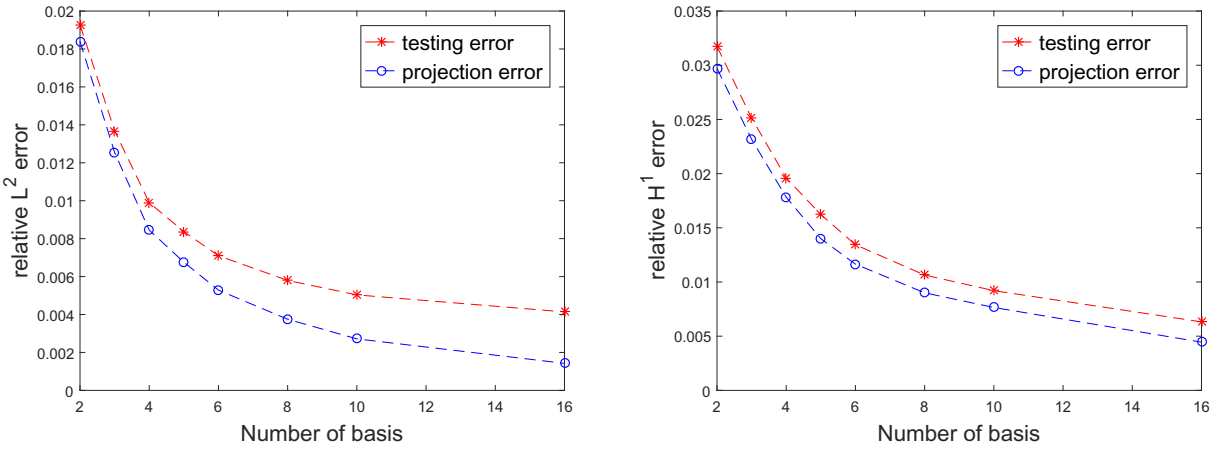


Figure 6: Relative L^2 and H^1 error with increasing number of basis for the local problem of Sec.4.2.

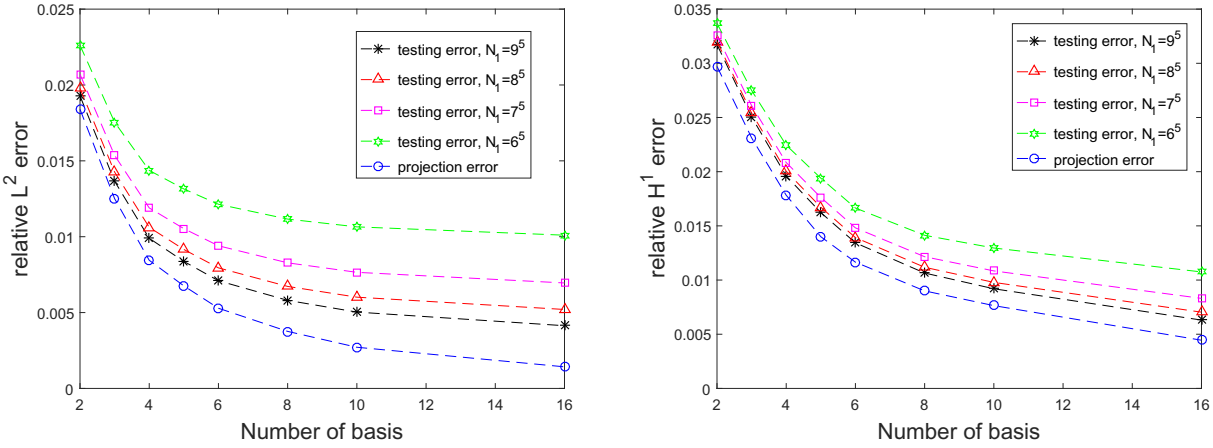


Figure 7: Relative L^2 and H^1 error of the solution computed by the nonlinear map $\mathbf{F}$ based on different uniform grids for the local problem of Sec.4.2.

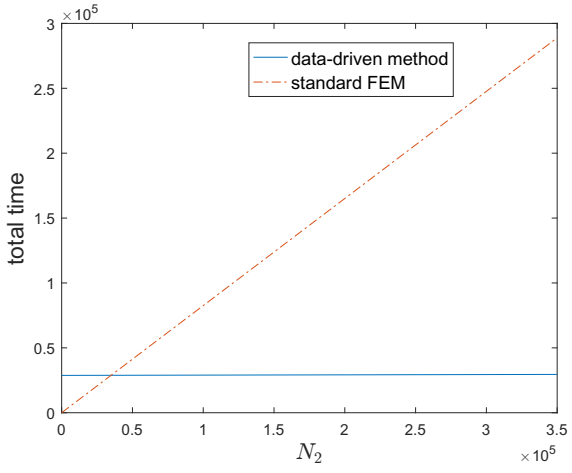
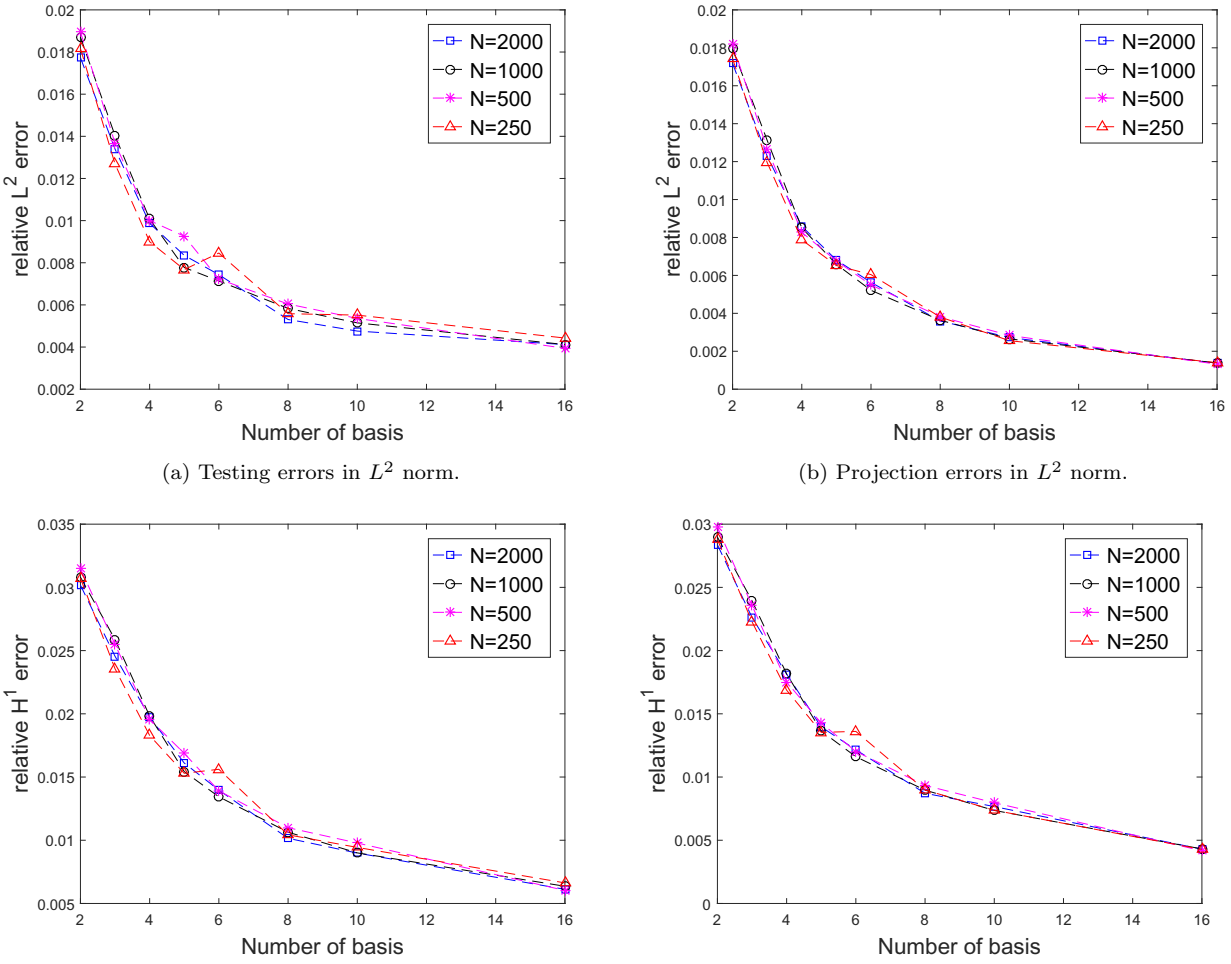


Figure 8: CPU time for the local problem of Sec.4.2.

of samples N in constructing the data-driven basis. Although the numerical error decreases when the sampling number N is increased in general, the difference is very mild.



(a) Testing errors in L^2 norm.

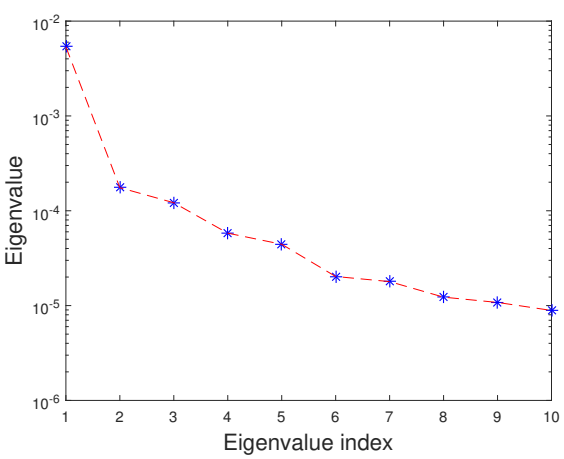
(b) Projection errors in L^2 norm.

(c) Testing errors in H^1 norm.

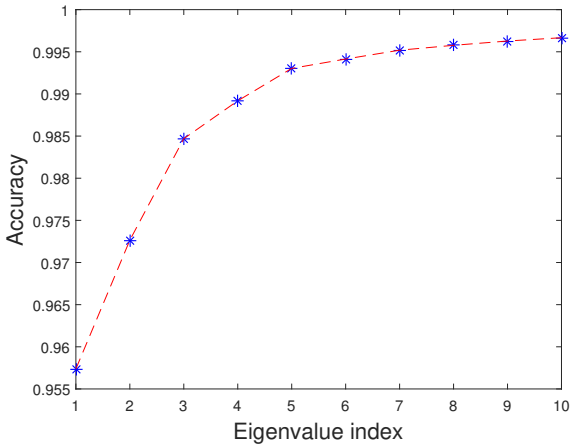
(d) Projection errors in H^1 norm.

Figure 9: The relative testing/projection errors in L^2 and H^1 norms with different number of samples (i.e. N) for the local problem of Sec.4.2.

Next, we test our method on the whole computation domain for (41) with coefficient (42). We choose $N_2 = 10N_1$. Figure 10 shows the decay property of eigenvalues. Similarly, we show magnitudes of the leading eigenvalues in Figure 10a and the ratio of the accumulated sum of the eigenvalues over the total sum in Figure 10b. We observe similar behaviors as before. Since we approximate the solution in the whole computational domain, we take the Galerkin approach described in Section 3.2 using the data-driven basis. In Figure 11, we show the mean relative error between our numerical solution and the reference solution in L^2 norm and H^1 norm, respectively. In practice, when the number of basis is 15, it takes about 0.084 second to compute a new solution by our method, whereas the standard FEM method costs about 0.82 second for one solution.

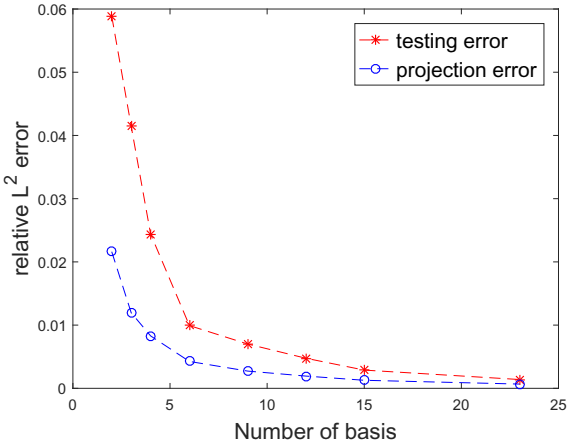


(a) Decay of the eigenvalues.

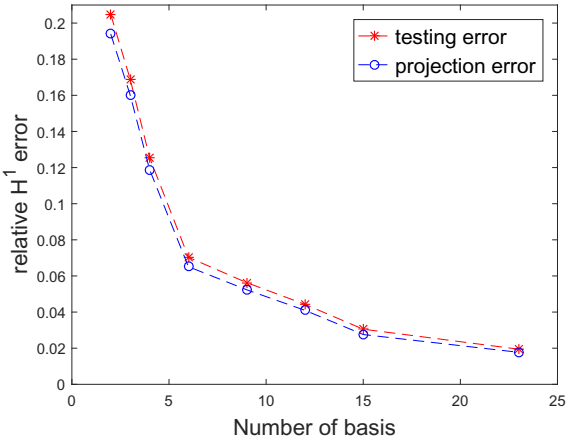


(b) $1-\sqrt{\sum_{j=n+1}^{N}\lambda_j/\sum_{j=1}^{N}\lambda_j}$, $n=1,2,....$

Figure 10: The decay properties of the eigenvalues for the global problem of Sec.4.2.



(a) Relative error in L^2 norm.



(b) Relative error in H^1 norm.

Figure 11: The relative errors with increasing number of basis for the global problem of Sec.4.2.

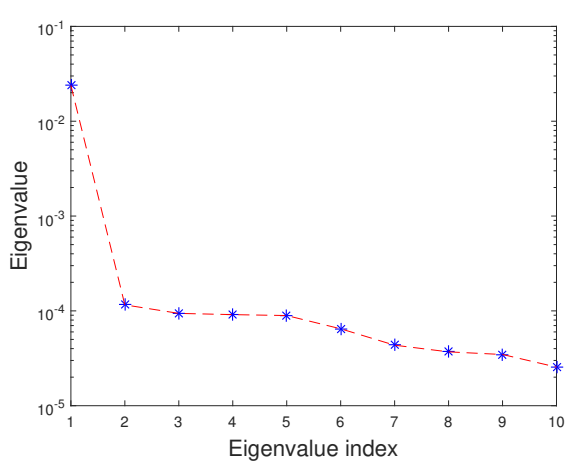
4.3. An example with an exponential type coefficient

We now solve the problem (41) with an exponential type coefficient. The coefficient is parameterized by eight random variables, which has the following form

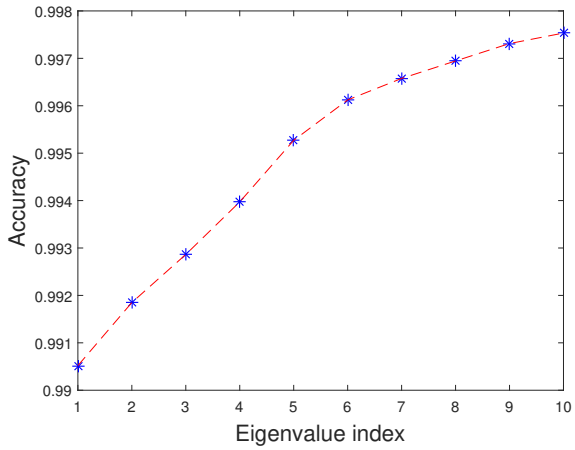
$$a(x,y,\omega) = \exp\Big(\sum_{i=1}^{8} \sin(\frac{2\pi(9-i)x}{9\epsilon_i})\cos(\frac{2\pi i y}{9\epsilon_i})\xi_i(\omega)\Big), \tag{43}$$

where the multiscale parameters $[\epsilon_1, \epsilon_2, \cdots, \epsilon_8] = [\frac{1}{43}, \frac{1}{41}, \frac{1}{47}, \frac{1}{29}, \frac{1}{37}, \frac{1}{31}, \frac{1}{53}, \frac{1}{35}]$ and $\xi_i(\omega)$, $i = 1, ..., 8$ are i.i.d. uniform random variables in $[-\frac{1}{2}, \frac{1}{2}]$. Hence the contrast ratio is $\kappa_a \approx 3.0 \times 10^3$ in the coefficient (43). The source function is $f(x,y) = \cos(2\pi x)\sin(2\pi y) \cdot I_{D_2}(x,y)$, where I_{D_2} is an indicator function defined on $D_2 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$. In the local problem, the subdomain of interest is $D_1 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{11}{16}, \frac{15}{16}]$.

In Figure 12, we show the decay property of eigenvalues. Specifically, in Figure 12a we show the magnitude of leading eigenvalues and in Figure 12b we show the ratio of the accumulated sum of the eigenvalues over the total sum. These results imply that the solution space has a low-dimensional structure, which can be approximated by the data-driven basis functions.



(a) Decay of eigenvalues.

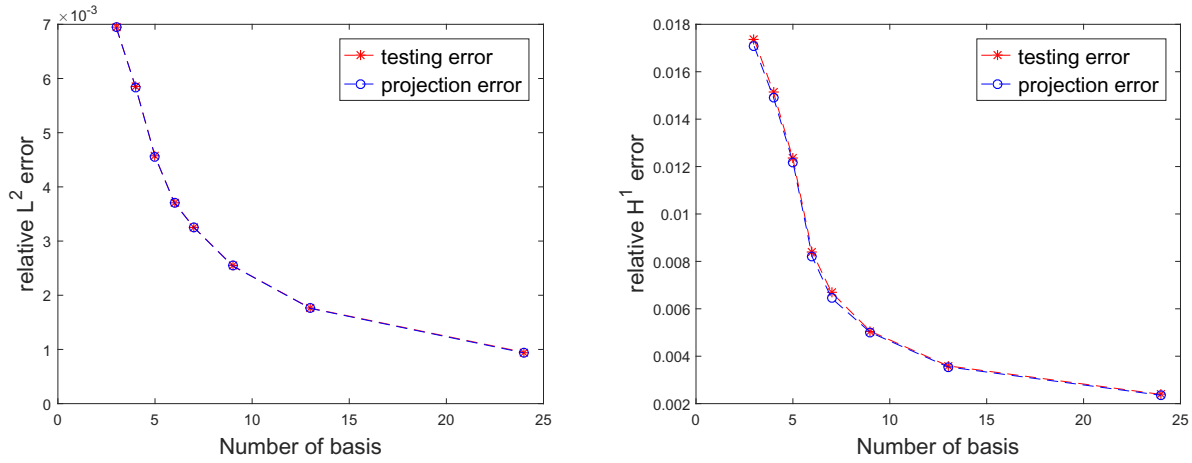


(b) $1 - \sqrt{\sum_{j=n+1}^{N} \lambda_j / \sum_{j=1}^{N} \lambda_j}$, $n = 1, 2, ...$

Figure 12: The decay properties of the eigenvalues in the problem of Sec.4.3.

Since the coefficient $a(x,y,\omega)$ is parameterized by eight random variables, it is expensive to construct the map $\mathbf{F} : \boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$ using the interpolation method with uniform grids. Instead, we use a sparse grid polynomial interpolation approach to approximate the map $\mathbf{F}$. Specifically, we use Legendre polynomials with total order less than or equal 4 (i.e. sparse grid of level 5) to approximate the map, where the total number of nodes is $N_1 = 2177$; see [11].

Figure 13a and 13b show the relative errors of the testing error and projection error in L^2 norm and H^1 norm, respectively, where $N_2 = 10N_1$. The sparse grid polynomial interpolation approach gives a comparable error as the best approximation error. We observe similar convergence results in solving the global problem (41) with the coefficient (43) (not shown here). Therefore, we can use the sparse grid method to construct maps for problems of a moderate number of random variables.



(a) Relative error in L^2 norm.

(b) Relative error in H^1 norm.

Figure 13: The relative errors with increasing number of basis in the problem of Sec.4.3.

We also study the approximation property of the nonlinear map $\mathbf{F}$ based on sparse grids of different levels. Specifically, sparse grids of accuracy level 3, 4, and 5 respectively contain $N_1 = 129$, 609, and 2177 grid points. Figure 14 shows the mean relative L^2 and H^1 errors of the testing errors and the project error (which does not depend on the grid partition), where $N_2 = 21700$. One can see that the nonlinear map $\mathbf{F}$ based on sparse grids of accuracy level 4 is accurate enough.

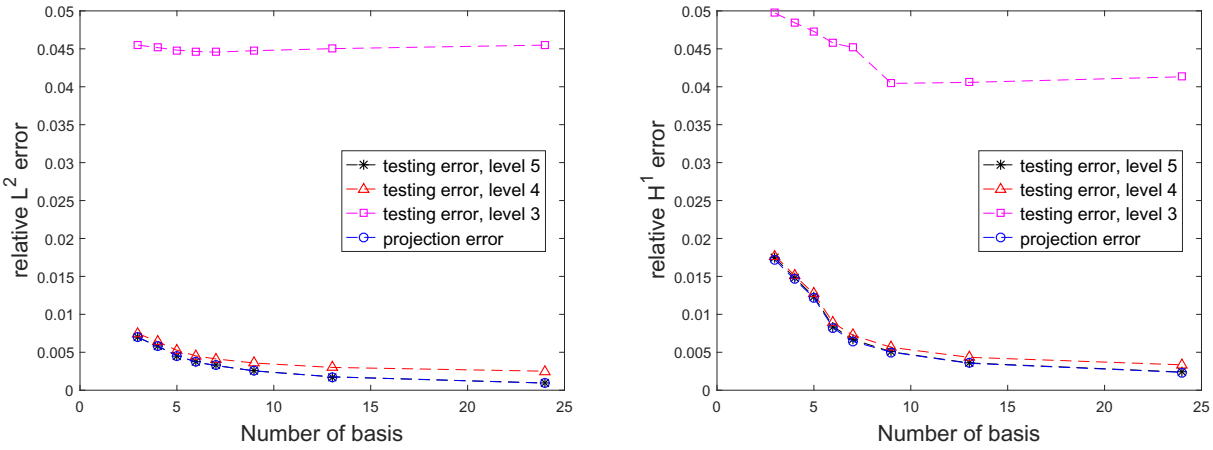


Figure 14: Relative L^2 and H^1 error of the solution computed by the nonlinear map $\mathbf{F}$ based on different sparse grids for the local problem of Sec.4.3.

4.4. An example with discontinuous coefficients

We solve the problem (41) with a discontinuous coefficient, which is an interface problem. The coefficient is parameterized by twelve random variables and has the following form

$$a(x,y,\omega) = \exp\Big(\sum_{i=1}^{6} \sin(2\pi \frac{x\sin(\frac{i\pi}{6}) + y\cos(\frac{i\pi}{6})}{\epsilon_i})\xi_i(\omega)\Big) \cdot I_{D\setminus D_3}(x,y) + \exp\Big(\sum_{i=1}^{6} \sin(2\pi \frac{x\sin(\frac{(i+0.5)\pi}{6}) + y\cos(\frac{(i+0.5)\pi}{6})}{\epsilon_{i+6}})\xi_{i+6}(\omega)\Big) \cdot I_{D_3}(x,y), \tag{44}$$

where $\epsilon_i = \frac{1+i}{100}$ for $i = 1, \cdots, 6$, $\epsilon_i = \frac{i+13}{100}$ for $i = 7, \cdots, 12$, $\xi_i(\omega)$, $i = 1, \cdots, 12$ are i.i.d. uniform random variables in $[-\frac{2}{3}, \frac{2}{3}]$, and I_{D_3}, $I_{D\setminus D_3}$ are indicator functions. The contrast ratio in the coefficient (44) is $\kappa_a \approx 3 \times 10^3$. The subdomain D_3 consists of three small rectangles whose edges are parallel to the edges of domain D with width $10h$ and height 0.8. And the lower left vertices are located at $(0.3, 0.1), (0.5, 0.1), (0.7, 0.1)$ respectively. One can use the coefficient (44) to model channels in permeability field in the reservoir simulation. In Figure 15 we show two realizations of the coefficient (44).

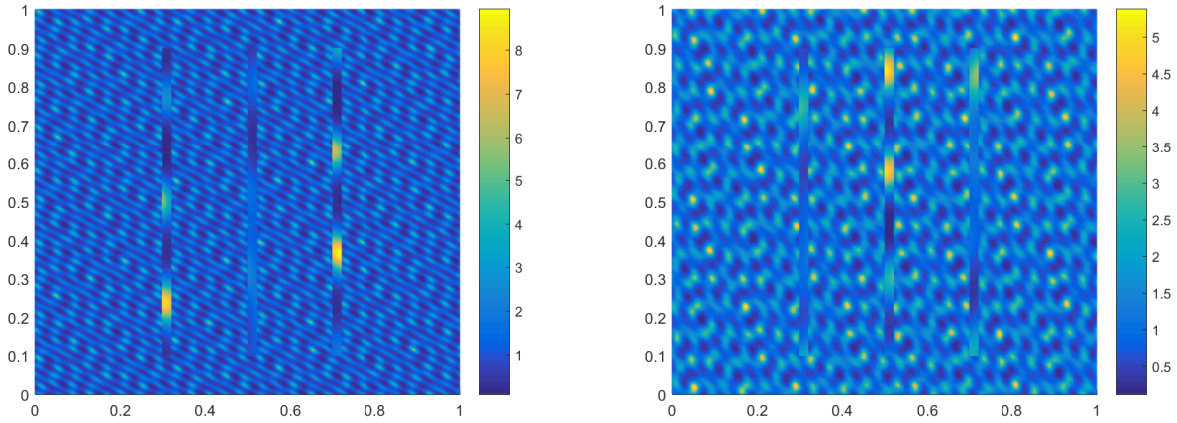
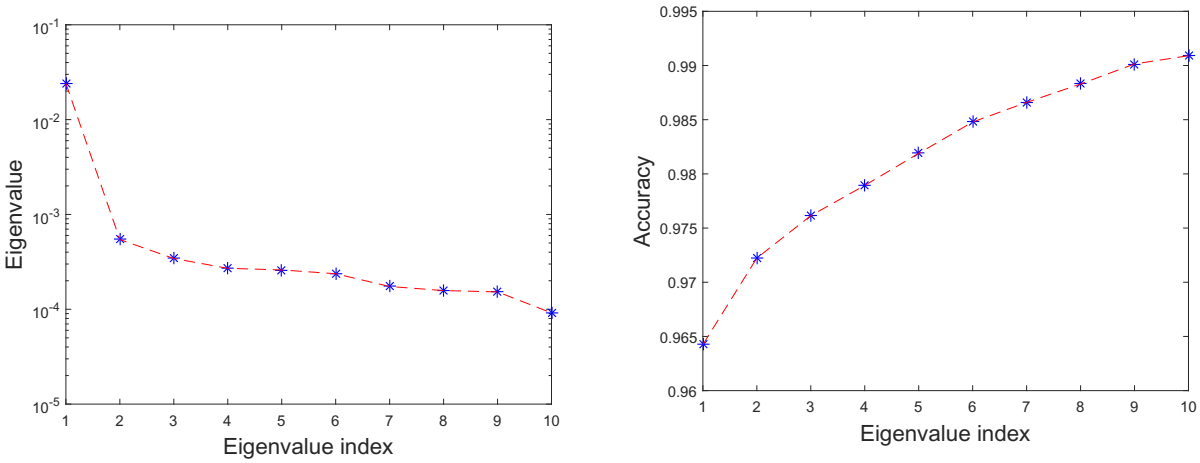


Figure 15: Two realizations of the coefficient (44) in the interface problem.

We now solve the local problem of (41) with the coefficient (44), where the domain of interest is $D_1 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{11}{16}, \frac{15}{16}]$. The source function is $f(x,y) = \cos(2\pi x)\sin(2\pi y) \cdot I_{D_2}(x,y)$, where $D_2 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$. In Figure 16a and Figure 16b we show the magnitude of dominant eigenvalues and approximate accuracy. These results show that only a few data-driven basis functions are enough to approximate all solution samples well.

Since the coefficient (44) is parameterized by twelve random variables, constructing the map $\mathbf{F} : \boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$ using the sparse grid polynomial interpolation becomes very expensive too. Here we use the least square method combined with the k-d tree algorithm for searching nearest neighbors to approximate the map $\mathbf{F}$.

In our method, we first generate $N_1 = 5000$ data pairs $\{(\boldsymbol{\xi}^n(\omega), \mathbf{c}^n(\omega)\}_{n=1}^{N_1}$ that will be used as training data. Then, we use $N_2 = 10N_1$ samples for testing in the online stage. For each new testing data point $\boldsymbol{\xi}(\omega) = [\xi_1(\omega), \cdots, \xi_r(\omega)]^T$ (here $r = 12$), we run the k-d tree algorithm to find its n nearest neighbors in the training data set and apply the least square method to compute the corresponding mapped value $\mathbf{c}(\omega) = [c_1(\omega), \ldots, c_K(\omega)]^T$. The complexity of constructing a k-d tree for N_1 data points is $O(N_1 \log N_1)$. Given the k-d tree, for each testing point the complexity of finding its n nearest neighbors is $O(n \log N_1)$ [40].



(a) Decay of eigenvalues.

(b) $1-\sqrt{\sum_{j=n+1}^{N}\lambda_j/\sum_{j=1}^{N}\lambda_j}$, $n=1,2,....$

Figure 16: The decay properties of the eigenvalues in the problem of Sec.4.4.

Since the n nearest neighbors (training data) are close to the testing data point $\boldsymbol{\xi}(\omega)$, for each training data $\left(\boldsymbol{\xi}^m(\omega), \mathbf{c}^m(\omega)\right)$, $m=1,....n$, we compute the first-order Taylor expansion of each component $c_j^m(\omega)$ at $\boldsymbol{\xi}(\omega)$ as

$$c_j^m(\omega) \approx c_j(\omega) + \sum_{i=1}^{r=12}(\xi_i^m - \xi_i)\frac{\partial c_j}{\partial \xi_i}(\omega), \quad j=1,2,\cdots,K, \tag{45}$$

where ξ_i^m, $i=1,...,r$, $c_j^m(\omega)$, $j=1,...,K$ are given training data, $c_j(\omega)$ and $\frac{\partial c_j}{\partial \xi_i}(\omega)$, $j=1,...,K$ are unknowns associated with the testing data point $\boldsymbol{\xi}(\omega)$. In the k-d tree algorithm, we choose $n=20$, which is slightly greater than $r+1=13$. By solving (45) using the least square method, we get the mapped value $\mathbf{c}(\omega)=[c_1(\omega),\ldots,c_K(\omega)]^T$. Finally, we use the formula (22) to get the numerical solution of Eq.(41) with the coefficient (44).

Because of the discontinuity and high-dimensional random variables in the coefficient (44), the problem (41) is more challenging. The nearest neighbors based least square method provides an efficient way to construct maps and achieve relative errors less than 3% in both L^2 norm and H^1 norm; see Figure 17. Alternatively, one can use neural network method to construct maps for this type of challenging problems; see Section 4.5.

4.5. An example with high-dimensional random coefficient and source function

We solve the problem (41) with an exponential type coefficient and random source function, where the total number of random variables is twenty. Specifically, the coefficient is parameterized by eighteen i.i.d. random variables, i.e.

$$a(x,y,\omega) = \exp\Big(\sum_{i=1}^{18}\sin(2\pi\frac{x\sin(\frac{i\pi}{18})+y\cos(\frac{i\pi}{18})}{\epsilon_i})\xi_i(\omega)\Big), \tag{46}$$

where $\epsilon_i=\frac{1}{2i+9}$, $i=1,2,\cdots,18$ and $\xi_i(\omega)$, $i=1,...,18$ are i.i.d. uniform random variables in $[-\frac{1}{5},\frac{1}{5}]$. The source function is a Gaussian density function $f(x,y)=\frac{1}{2\pi\sigma^2}\exp(-\frac{(x-\theta_1)^2+(y-\theta_2)^2}{2\sigma^2})$

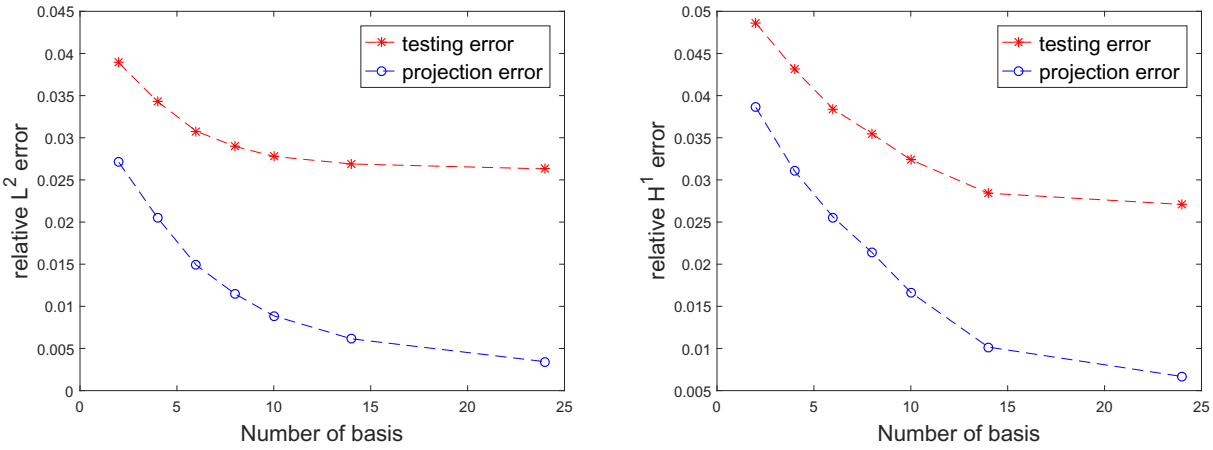
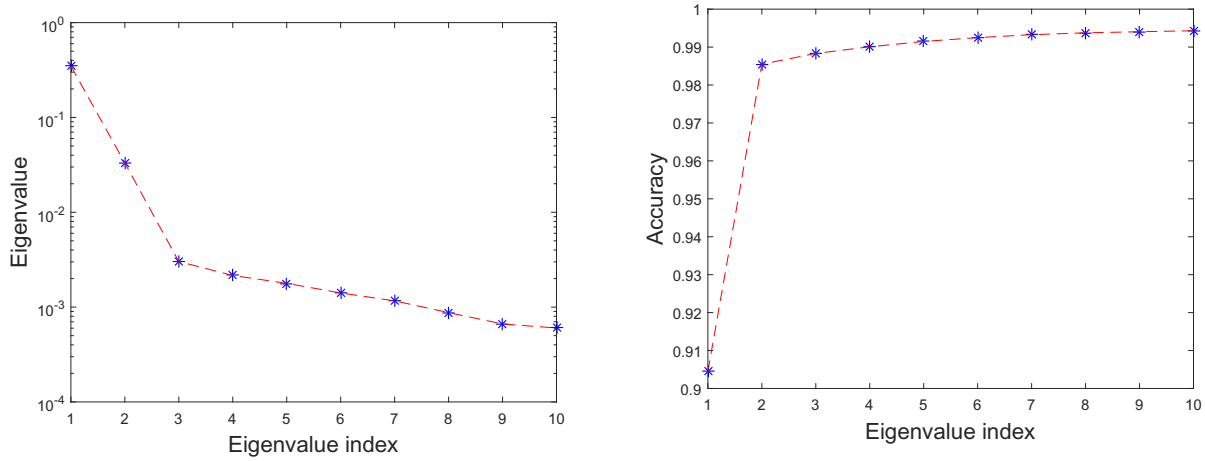


Figure 17: The relative errors with increasing number of basis in the local problem of Sec.4.4 .

with a random center (θ_1, θ_2) that is a random point uniformly distributed in the subdomain $D_2 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$ and $\sigma = 0.01$. When σ is small, the Gaussian density function $f(x, y)$ can be used to approximate the Dirac-δ function, such as modeling wells in reservoir simulations.

We first solve the local problem of (41) with $N = 2000$ samples of the coefficient (46), where the subdomain of interest is $D_1 = [\frac{1}{4}, \frac{3}{4}] \times [\frac{11}{16}, \frac{15}{16}]$. In Figures 18a and 18b, we show the magnitude of leading eigenvalues and the ratio of the accumulated sum of the eigenvalue over the total sum, respectively. We observe similar exponential decay properties of eigenvalues even if the source function contains randomness. These results show that we can still build a set of data-driven basis functions to solve problem (41) with coefficient (46).



(a) Decay of eigenvalues.

(b) $1 - \sqrt{\sum_{j=n+1}^{N} \lambda_j / \sum_{j=1}^{N} \lambda_j}$, $n = 1, 2, ...$

Figure 18: The decay properties of the eigenvalues in the problem of Sec.4.5.

Notice that both the coefficient and source contain randomness here. We put the random variables $\boldsymbol{\xi}(\omega)$ in the coefficient and the random variables $\boldsymbol{\theta}(\omega)$ in the source together when we construct the map $\mathbf{F}$. Moreover, the dimension of randomness, 18+2=20, is too large even

for sparse grids. Here we construct the map $\mathbf{F} : (\boldsymbol{\xi}(\omega), \boldsymbol{\theta}(\omega)) \mapsto \mathbf{c}(\omega)$ using the neural network as depicted in Figure 19. The neural network has 4 hidden layers and each layer has 50 units. Naturally, the number of the input units is 20 and the number of the output units is K. The layer between input units and first layer of hidden units is an affine transform. So is the layer between output units and last layer of hidden units. Each two layers of hidden units are connected by an affine transform, a tanh (hyperbolic tangent) activation and a residual connection, i.e. $\mathbf{h}_{l+1} = \tanh(\mathbf{A}_l \mathbf{h}_l + \mathbf{b}_l) + \mathbf{h}_l$, $l = 1, 2, 3$, where $\mathbf{h}_l$ is l-th layer of hidden units, $\mathbf{A}_l$ is a 50-by-50 matrix and $\mathbf{b}_l$ is a 50-by-1 vector. Under the same setting of neural network, if the rectified linear unit (ReLU), which is piecewise linear, is used as the activation function, we observe a much bigger error. Therefore, we choose the hyperbolic tangent activation function and implement the residual neural network (ResNet) here [25].

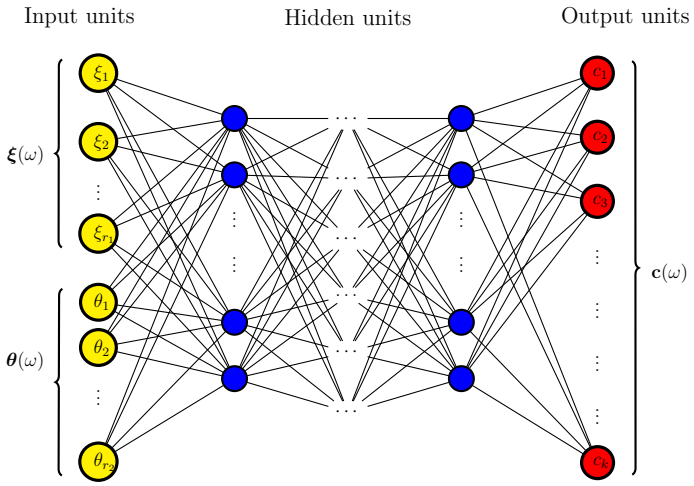


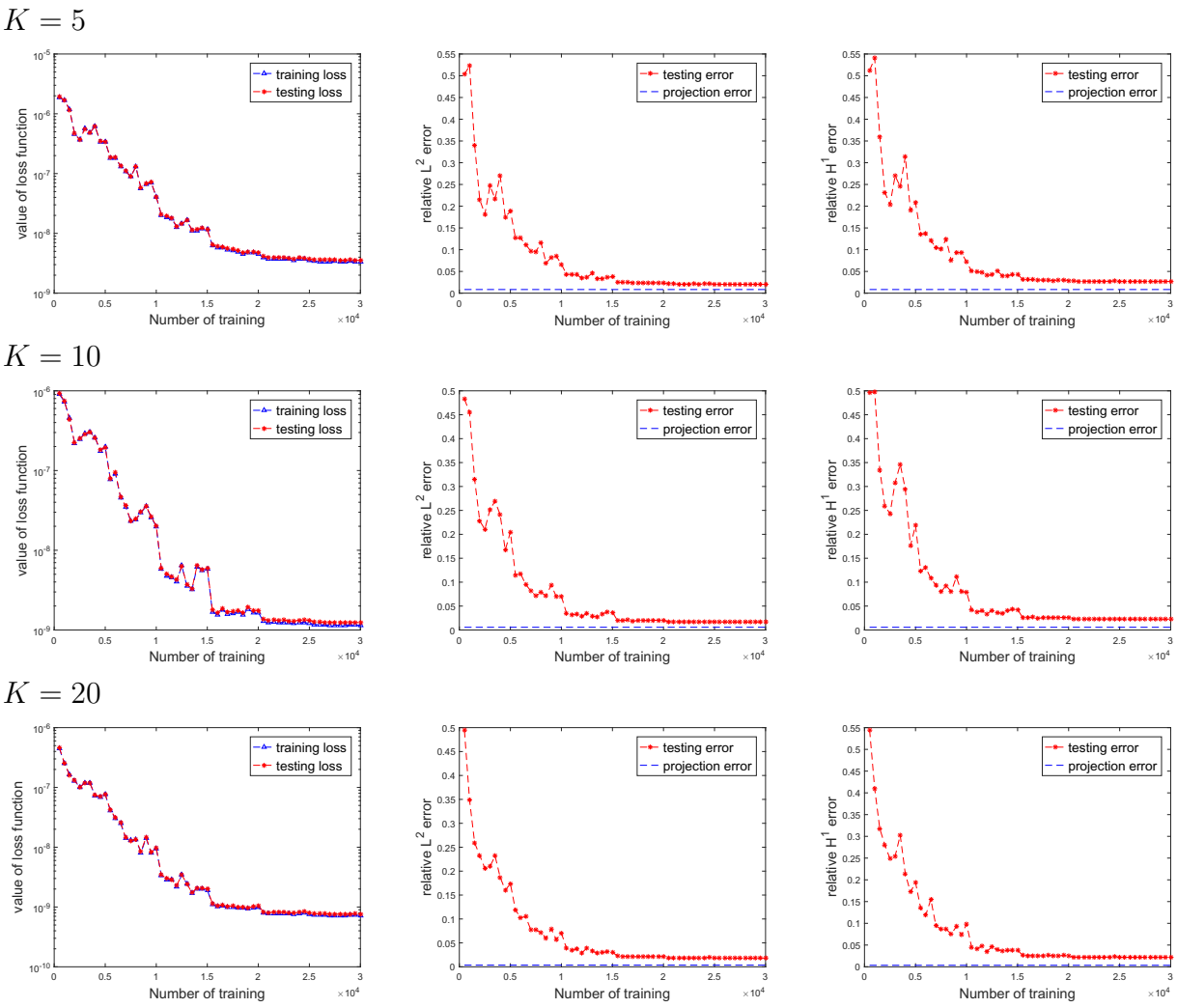
Figure 19: Structure of neural network, where $r_1 = 18$ and $r_2 = 2$.

We use $N_1 = 5000$ samples for network training in the offline stage and $N_2 = 10N_1$ samples for testing in the online stage. The sample data pairs for training are $\{(\boldsymbol{\xi}^n(\omega), \boldsymbol{\theta}^n(\omega)), \mathbf{c}^n(\omega)\}_{n=1}^{N_1}$, where $\boldsymbol{\xi}^n(\omega) \in [-\frac{1}{5}, \frac{1}{5}]^{18}$, $\boldsymbol{\theta}^n(\omega) \in [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{16}, \frac{5}{16}]$, and $\mathbf{c}^n(\omega) \in R^K$. We define the loss function of network training as

$$loss\big(\{\mathbf{c}^n\}, \{\hat{\mathbf{c}}^n\}\big) = \frac{1}{N_1} \sum_{n=1}^{N_1} \frac{1}{K} |\mathbf{c}^n - \hat{\mathbf{c}}^n|^2, \tag{47}$$

where $\mathbf{c}^n$ are the training data and $\hat{\mathbf{c}}^n$ are the output of the neural network (see Figure 19).

Figure 20a shows the value of loss function during training procedure. Figure 20b shows the corresponding mean relative error of the testing samples in L^2 norm. Eventually the relative error of the neural network reaches about 1.5%. Figure 20c shows the corresponding mean relative error of the testing samples in H^1 norm. We remark that many existing methods become extremely expensive or infeasible when the problem is parameterized by high-dimensional random variables. Our data-driven basis method based on neural network still provides a satisfactory result.



(a) Loss. (b) Relative L^2 error. (c) Relative H^1 error.

Figure 20: First column: the value of loss function during training procedure. Second column and third column: the mean relative errors of the testing set during training procedure in L^2 and H^1 norm respectively.

4.6. An example with unknown random coefficient and source function

Finally, we present an example where the models of the random coefficient and source are unknown. Only a set of sample solutions are provided as well as a few censors can be placed at certain locations for solution measurements. This kind of scenario appears often in practice. We take the least square fitting method as described in Section 3.3. Our numerical experiment is still based on (41), which is used to generate solution samples (instead of experiments or measurements in real practice). But once the data are generated, we do not assume any knowledge of the coefficient or the source when computing a new solution.

To be specific, the coefficient takes the form

$$a(x,y,\omega) = \exp\Big(\sum_{i=1}^{24} \sin(2\pi \frac{x\sin(\frac{i\pi}{24}) + y\cos(\frac{i\pi}{24})}{\epsilon_i})\xi_i(\omega)\Big), \tag{48}$$

where $\epsilon_i = \frac{1+i}{100}$, $i = 1, 2, \cdots, 24$ and $\xi_i(\omega)$, $i = 1, ..., 24$ are i.i.d. uniform random variables in $[-\frac{1}{6}, \frac{1}{6}]$. The source function is a random function $f(x,y) = \sin(\pi(\theta_1 x + 2\theta_2))\cos(\pi(\theta_3 y + 2\theta_4)) \cdot I_{D_2}(x,y)$ with i.i.d. uniform random variables $\theta_1, \theta_2, \theta_3, \theta_4$ in $[0,2]$. We first generate $N = 2000$ solutions samples (using standard FEM) $u(x_j, \omega_i), i = 1, \ldots, N, j = 1, \ldots, J$, where x_j are the points where solution samples are measured. Then, a set of K data-driven basis $\phi_k(x_j), j = 1, \ldots, J, k = 1, \ldots, K$ are extracted from the solution samples as before.

Next we determine M good sensing locations from the data-driven basis so that the least square problem (26) is not ill-conditioned. We follow the method proposed in [32]. Define $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K] \in R^{J\times K}$, where $\boldsymbol{\phi}_k = [\phi_k(x_1), \ldots, \phi_k(x_J)]^T$. If $M = K$, QR factorization with column pivoting is performed on Φ^T. If $M > K$, QR factorization with pivoting is performed on $\Phi\Phi^T$. The first M pivoting indices provide the measurement locations. Once a new solution is measured at these M selected locations, the least square problem (26) is solved to determine the coefficients $c_1, c_2, \ldots, c_K$ and the new solution is approximated by $u(x_j, \omega) = \sum_{k=1}^{K} c_k \phi_k(x_j)$.

In Figure 21 and Figure 22, we show the results of the local problem and global problem, respectively. In these numerical results, we compared the error between the reconstructed solutions and the reference solution. We find the our proposed method works well for problem (41) with a non-parametric coefficient or source as well.

5. Conclusion

In this paper, we propose a data-driven approach to solve multiscale elliptic PDEs with random coefficients or random sources. This type of multiscale problem has many applications, such as heterogeneous porous media flow problems in water aquifer and oil reservoir simulations. Motivated by the existence of approximate low dimensional structures in the solution space of the multiscale problems, we construct a set of problem-specific data-driven basis functions directly from samples solutions or experimental data. Once the data-driven basis is available, depending on different problem setups, we design several ways to compute a new solution efficiently.

Error analysis based on the sampling error of the coefficients and the projection error of the data-driven basis is presented to provide some guidance on the implementation of our method.

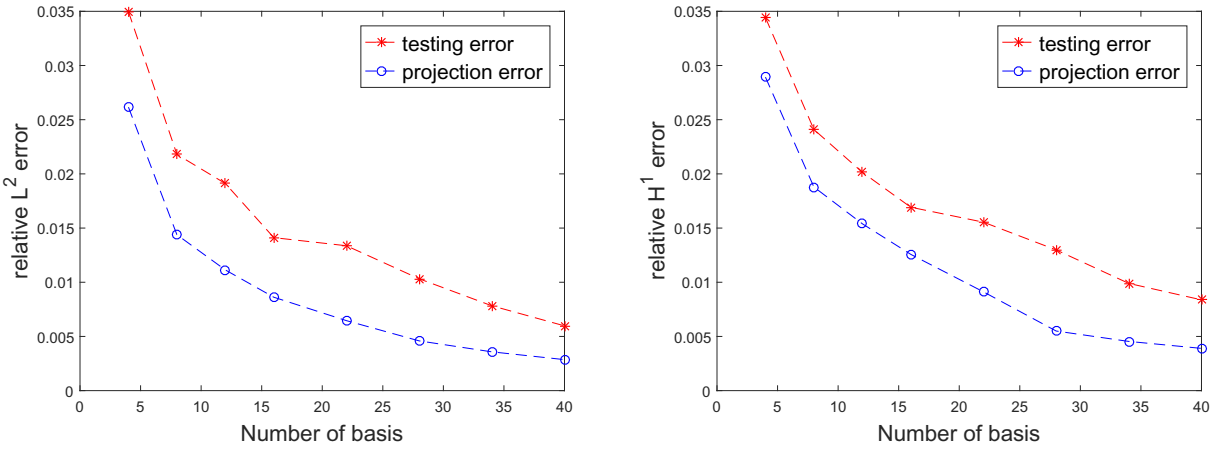


Figure 21: The relative errors with increasing number of basis in the local problem of Sec.4.6 .

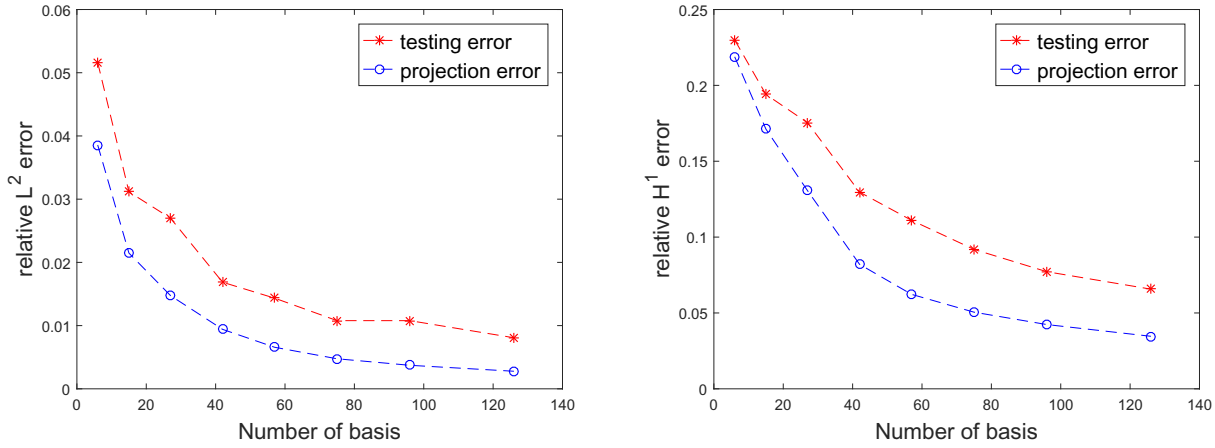


Figure 22: The relative errors with increasing number of basis in the global problem of Sec.4.6.

Numerical examples show that the proposed method is very efficient in solving multiscale elliptic PDEs with random input, especially when the random input is relative high dimensional. Therefore, these data-driven basis functions indeed provide a nearly optimal approximation to the low dimensional structures in the solution space.

Acknowledgements

The research of S. Li is partially supported by the Doris Chen Postgraduate Scholarship. The research of Z. Zhang is supported by the Hong Kong RGC General Research Funds (Projects 27300616, 17300817, and 17300318), National Natural Science Foundation of China (Project 11601457), Seed Funding Programme for Basic Research (HKU), and Basic Research Programme (JCYJ20180307151603959) of The Science, Technology and Innovation Commission of Shenzhen Municipality. The research of H. Zhao is partially supported by NSF grant DMS-1622490 and DMS-1821010. This research is made possible by a donation to the Big Data Project Fund, HKU, from Dr Patrick Poon whose generosity is gratefully acknowledged.

References

[1] A. Abdulle, A. Barth, and C. Schwab, *Multilevel Monte Carlo methods for stochastic elliptic multiscale PDEs*, Multiscale Modeling & Simulation, 11 (2013), pp. 1033–1070.

[2] M. Arnst and R. Ghanem, *Probabilistic equivalence and stochastic model reduction in multiscale analysis*, Comput. methods Appl. Mech. Engrg, 197(43) (2008), pp. 3584–3592.

[3] B. V. Asokan and N. Zabaras, *A stochastic variational multiscale method for diffusion in heterogeneous random media*, Journal of Computational Physics, 218 (2006), pp. 654–676.

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

[5] I. Babuska, R. Tempone, and G. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.

[6] M. Bachmayr, A. Cohen, R. DeVore, and G. Migliorati, *Sparse polynomial approximation of parametric elliptic PDEs. Part II: lognormal coefficients*, ESAIM: Mathematical Modelling and Numerical Analysis, 51 (2017), pp. 341–363.

[7] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339(9) (2004), pp. 667–672.

[8] M. Bebendorf and W. Hackbusch, *Existence of H-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ coefficients*, Numerische Mathematik, 95 (2003), pp. 1–28.

[9] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Review, 57 (2015), pp. 483–531.

[10] G. Berkooz, P. Holmes, and J. L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual review of fluid mechanics, 25(1) (1993), pp. 539–575.

[11] H. J. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica, 13 (2004), pp. 147–269.

[12] M. Cheng, T. Y. Hou, M. Yan, and Z. Zhang, *A data-driven stochastic method for elliptic PDEs with random coefficients*, SIAM J. UQ, 1 (2013), pp. 452–493.

[13] M. Cheng, T. Y. Hou, and Z. Zhang, *A dynamically bi-orthogonal method for stochastic partial differential equations I: derivation and algorithms*, J. Comput. Phys., 242 (2013), pp. 843–868.

[14] ——, *A dynamically bi-orthogonal method for stochastic partial differential equations II: adaptivity and generalizations*, J. Comput. Phys., 242 (2013), pp. 753–776.

[15] E. Chung, Y. Efendiev, W. Leung, and Z. Zhang, *Cluster-based generalized multiscale finite element method for elliptic PDEs with random coefficients*, Journal of Computational Physics, 371 (2018), pp. 606–617.

[16] A. Cohen and R. DeVore, *Approximation of high-dimensional parametric PDEs*, Acta Numerica, 24 (2015), pp. 1–159.

[17] G. Dolzmann and S. Müller, *Estimates for green's matrices of elliptic systems byl p theory*, Manuscripta mathematica, 88 (1995), pp. 261–273.

[18] Y. Efendiev, C. Kronsbein, and F. Legoll, *Multilevel Monte Carlo approaches for numerical homogenization*, Multiscale Modeling & Simulation, 13 (2015), pp. 1107–1135.

[19] B. Engquist and H. Zhao, *Approximate separability of the Green's function of the Helmholtz equation in the high frequency limit*, Communications on Pure and Applied Mathematics, 71 (2018), pp. 2220–2274.

[20] R. Ghanem and P. Spanos, *Stochastic finite elements: a spectral approach.*, Springer-Verlag, New York, 1991.

[21] I. Graham, F. Kuo, D. Nuyens, R. Scheichl, and I. Sloan, *Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications*, Journal of Computational Physics, 230 (2011), pp. 3668–3694.

[22] I. G. Graham, F. Y. Kuo, J. A. Nichols, R. Scheichl, C. Schwab, and I. H. Sloan, *Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients*, Numerische Mathematik, 131(2) (2015), pp. 329–368.

[23] M. Grüter and K. Widman, *The green function for uniformly elliptic equations*, Manuscripta Mathematica, 37 (1982), pp. 303–342.

[24] N. Halko, P. Martinsson, and J. Tropp, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review, 53 (2011), pp. 217–288.

[25] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[26] V. Hoang and C. Schwab, *N-term wiener chaos approximation rates for elliptic PDEs with lognormal Gaussian random inputs*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 797–826.

[27] T. Hou and P. Liu, *A heterogeneous stochastic FEM framework for elliptic PDEs*, Journal of Computational Physics, 281 (2015), pp. 942–969.

[28] T. Hou, P. Liu, and Z. Zhang, *A localized data-driven stochastic method for elliptic PDEs with random coefficients*, Bull. Inst. Math. Acad. Sin. (N.S.), 1 (2016), pp. 179–216.

[29] T. Hou, D. Ma, and Z. Zhang, *A model reduction method for multiscale elliptic PDEs with random coefficients using an optimization approach*, Multiscale Modeling & Simulation, 17 (2019), pp. 826–853.

[30] T. Y. Hou, W. Luo, B. Rozovskii, and H. M. Zhou, *Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics*, J. Comput. Phys., 216 (2006), pp. 687–706.

[31] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidid, O. Runborg, and C. Theodoropoulos, *Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis*, Communications in Mathematical Sciences, 1(4) (2003), pp. 715–762.

[32] K. Manohar, B. Brunton, J. Kutz, and S. Brunton, *Data-driven sparse sensor placement for reconstruction*, arXiv:1701.07569, (2017).

[33] H. G. Matthies and A. Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Method Appl. Mech. Eng., 194 (2005), pp. 1295–1331.

[34] H. N. Najm, *Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics*, Annual Review of Fluid Mechanics, 41 (2009), pp. 35–52.

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

[36] H. Owhadi and L. Zhang, *Metric-based upscaling*, Communications on Pure and Applied Mathematics, 60 (2007), pp. 675–723.

[37] T. Sapsis and P. Lermusiaux, *Dynamically orthogonal field equations for continuous stochastic dynamical systems*, Physica D: Nonlinear Phenomena, 238 (2009), pp. 2347–2360.

[38] L. Sirovich, *Turbulence and the dynamics of coherent structures. I. Coherent structures*, Quarterly of applied mathematics, 45(3) (1987), pp. 561–571.

[39] A. Stuart, *Inverse problems: a Bayesian perspective*, Acta numerica, 19 (2010), pp. 451–559.

[40] I. Wald and V. Havran, *On building fast kd-trees for ray tracing, and on doing that in O (N log N)*, in 2006 IEEE Symposium on Interactive Ray Tracing, IEEE, 2006, pp. 61–69.

[41] J. Wan and N. Zabaras, *A probabilistic graphical model approach to stochastic multiscale partial differential equations*, Journal of Computational Physics, 250 (2013), pp. 477–510.

[42] X. L. Wan and G. Karniadakis, *Multi-element generalized polynomial chaos for arbitrary probability measures*, SIAM J. Sci. Comp., 28 (2006), pp. 901–928.

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

[44] D. Xiu and G. Karniadakis, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, J. Comput. Phys., 187 (2003), pp. 137–167.

[45] Z. Zhang, M. Ci, and T. Y. Hou, *A multiscale data-driven stochastic method for elliptic PDEs with random coefficients*, SIAM Multiscale Model. Simul., 13 (2015), pp. 173–204.