SIAM J. SCI. COMPUT.
Vol. 41, No. 3, pp. A1984–A2011

A DOMAIN DECOMPOSITION MODEL REDUCTION METHOD FOR LINEAR CONVECTION-DIFFUSION EQUATIONS WITH RANDOM COEFFICIENTS*

LIN MU† AND GUANNAN ZHANG†

Abstract. We developed a domain decomposition model reduction method for linear steady-state convection-diffusion equations with random coefficients. Of particular interest to this effort are the diffusion equation with random diffusivity and the convection-dominated transport equation with random velocity. We investigated two types of random fields, i.e., colored noises and discrete white noises, both of which can lead to high-dimensional parametric dependence in practice. The motivation is to exploit domain decomposition to reduce the parametric dimension of local problems in subdomains, such that an entire parametric map can be approximated with a small number of expensive partial differential equation (PDE) simulations. The new method combines domain decomposition with model reduction and sparse polynomial approximation, so as to simultaneously handle the high dimensionality and irregular behavior of the PDEs under consideration. The advantages of our method lie in three aspects: (i) online-offline decomposition, i.e., the online cost is independent of the size of the triangle mesh; (ii) sparse approximation of operators involving non-affine high-dimensional random fields; (iii) an effective strategy to capture irregular behaviors, e.g., sharp transitions of the PDE solutions. Two numerical examples are provided to demonstrate the advantageous performance of our method.

Key words. domain decomposition, colored noise, discrete white noise, uncertainty quantification, high dimensionality, sharp transition

AMS subject classifications. 65D15, 65N35, 65N12, 65N15, 65C20, 65C30

DOI. 10.1137/18M1170601

1. Introduction.

This paper focuses on linear steady-state convection-diffusion equations with random coefficients. Of particular interest to this effort are two types of partial differential equations (PDEs), i.e., the diffusion equations with random diffusivities and the convection-dominated transport equations with random velocities. We investigated two types of random fields: colored noises and discrete white noises. The PDEs of interest are widely used to describe subsurface flows in porous media. There are two major challenges in solving such PDEs, i.e., high-dimensional parameterization and irregular behaviors, e.g., solutions with sharp transitions. The parametric dimension depends on the approximation of the random fields. A colored noise is usually approximated by its truncated Karhunen–Loève (KL) expansion [22], where the parametric dimension is the number of retained modes in the truncated expansion. A discrete white noise is usually defined by a piecewise constant function,

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 13, 2018; accepted for publication (in revised form) April 2, 2019; published electronically June 20, 2019.
http://www.siam.org/journals/sisc/41-3/M117060.html

Funding: This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contracts ERKJ259 and ERKJ320; by the U.S. National Science Foundation, Computational Mathematics program under award 1620027; and by the Laboratory Directed Research and Development program at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831 (mul1@ornl.gov, zhangg@ornl.gov).

where the parametric dimension is the number of pieces in the physical domain. In this effort, we only consider *finite*-dimensional discrete white noises, similar to the "inclusion problem" studied in [3].

In the literature, three common approaches for solving the PDEs of interest are Monte Carlo methods [26, 32], stochastic spectral methods [9, 10, 16, 38], and reduced-basis methods [4, 5, 6, 13, 30, 31]. The Monte Carlo methods, including their multilevel/multifidelity variants [15, 28], are insensitive to the parametric dimension but feature slow convergence. One particular class of spectral methods utilizes orthogonal polynomials to build sparse approximations of the parametric map, which have been very successful in exploiting the sparsity of the parametric dependence (see, e.g., [9]). However, when the dimension is very large and/or the solution map does not have the desired regularity, we do not have sufficient sparsity to build accurate approximations with affordable computational effort. The reduced-basis methods approximate the solution manifold by constructing reduced bases in the finite element (FE) space, via either proper orthogonal decomposition (POD) or greedy algorithms. In this case, the best convergence rate is described by the decay of the Kolmogorov n-width [8] with n the dimension of the reduced subspace. Similar to the spectral methods, high dimensionality and irregularities may lead to a slow decay of the Kolmogorov n-width, thus deteriorating the performance of the reduced-basis methods.

The domain decomposition (DD) methods were originally proposed to develop parallel solvers for deterministic PDEs (see, e.g., [37] for details), and the same idea for parallelization can be extended to the stochastic PDE setting [33, 35]. Recently, more research has been conducted on the DD methods for PDEs with colored noises, based on the fact that the KL eigenvalues decay faster as the ratio of the physical domain size and the correlation length becomes smaller [19, 29, 34]. In [7], Chen et al. utilized the DD approach to solve stochastic elliptic PDEs, where the local solution in each subdomain was approximated in a low-dimensional parametric space. In [18], Hou, Li, and Zhang combined the DD strategy with multiscale finite element methods (FEMs) to solve elliptic PDEs with a high-contrast random medium. In [11], Contreras et al. developed a new approach to capture the correlation structure of local random variables in different subdomains, without performing expensive global KL expansion; and such a strategy was then incorporated into the DD framework to solve stochastic elliptic PDEs [12]. In [36], Tipireddy, Stinis, and Tartakovsky combined basis adaptation and local Hilbert-space KL expansion to approximate local solutions in the subdomains. In [14, 20, 24], the authors incorporated the DD strategy into reduced-basis approaches for stochastic PDEs. However, to our best knowledge, there is still a lack of efficient numerical capability that can simultaneously handle high dimensionality and irregular behaviors of the stochastic PDEs under consideration; and developing such a capability is the objective of this paper.

In this effort, we created a new domain decomposition model reduction (DDMR) method, which exploits the low-dimensional structure of the local problems from various perspectives. The resulting algorithm can be divided into an offline procedure and an online procedure. The offline procedure consists of four main stages. The first is to divide the physical domain into a set of nonoverlapping subdomains, generate local random fields, and establish the correlation among the local fields. For a discrete white noise, we only need to align the domain partition with the partition of the noise, and there is no correlation among the local fields. For a colored noise, we decompose the covariance function in the global domain and subdomains, respectively, and generate both the global and the local KL expansions. The correlation is obtained by constructing a least-squares projection from the global random variables to the

local random variables in the subdomains. The second stage is to generate two sets of training data, i.e., a set of snapshots of the PDE solutions and another set of snapshots of the local stiffness matrices. The third stage is to generate a set of reduced bases for the PDE solution in the interior and on the interfaces of the subdomains, as well as define reduced local stiffness matrices by multiplying the reduced bases by the corresponding blocks of the local stiffness matrices. The fourth stage is to establish sparse approximations to the entries of the reduced local stiffness matrices in low-dimensional local parametric spaces, which finishes the offline procedure. In the online phase, when a new realization of the global random field is generated, we map the global random variables to the local random variables, evaluate the sparse approximations of the reduced local stiffness matrices, assemble the reduced global Schur complement matrix, solve the coefficients of the reduced bases on the interfaces, assemble the reduced local Schur complement matrices, and solve the coefficients of the reduced bases in the interior of the subdomains.

The advantages and contributions of our method lie in the following three aspects. First, our method has the online-offline decomposition feature; i.e., the online computational cost is *independent* of the triangle mesh size. This is achieved by utilizing the generated reduced bases. The bases on the interfaces are used to reduce the global Schur complement matrix, and the bases in the interior of the subdomains reduce the sizes of the linear systems recovering the local solutions. Second, our method can handle the PDEs of interest with nonaffine high-dimensional random coefficients. The challenge caused by the nonaffine coefficients is resolved by approximating the entries of the reduced stiffness matrices. Third, the DDMR method can avoid building sparse approximations to local PDE solutions. This feature is very important in solving the convection-dominated PDEs with random velocities, where the irregular behavior propagates to the parametric space. However, the entries of local stiffness matrices are not affected at all by such irregularity, so that we can still achieve the spectral convergence in approximating the entries of the reduced stiffness matrices.

The outline of this paper is as follows. In section 2, we set up the context of this work by introducing the PDEs and the random fields under consideration. In section 3, we briefly recall the deterministic DD method using the Schur complement. Our DDMR method is described in section 4. In section 5, we demonstrate our method by solving the diffusion equation with random diffusivities and the convection-dominated transport equation with random velocities. Finally, some concluding remarks are given in section 6.

2. Problem setting. Let $D \subset \mathbb{R}^d$, $d = 1, 2, 3$, be a bounded domain with Lipschitz continuous boundary, and let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space with Ω the sample space, $\mathcal{F} \subseteq 2^\Omega$ a σ-algebra, and $\mathbb{P}$ the associated probability measure. We considered the following stochastic boundary value problem: find a function $u : \overline{D} \times \Omega \to \mathbb{R}$, such that it holds $\mathbb{P}$-a.e. in Ω that

$$\begin{cases} -\nabla \cdot \big(a(x,\omega)\nabla u(x,\omega)\big) + \boldsymbol{b}(x,\omega) \cdot \nabla u(x,\omega) = f(x) & \text{in } D, \\ u(x,\omega) = w(x) & \text{on } \partial D, \end{cases} \tag{2.1}$$

where $a(x,\omega)$ and $\boldsymbol{b}(x,\omega) := (b_1(x,\omega), \ldots, b_d(x,\omega))$ are the random diffusivity and velocity, respectively, and $f(x)$ and $w(x)$ are a deterministic forcing term and a boundary condition, respectively. We denote by $W(D)$ a Banach space and assume the underlying random input data are properly chosen, such that the corresponding stochastic system (2.1) is well-posed and has a unique solution $u(x,\omega) \in L^2_{\mathbb{P}}(\Omega; W(D))$, where

the function space

$$L^2_{\mathbb{P}}(\Omega; W(D)) := \Big\{ u : \overline{D} \times \Omega \to \mathbb{R} \;\Big|\; u \text{ is strongly measurable and } \int_\Omega \|u\|^2_{W(D)}\, d\mathbb{P}(\omega) < +\infty \Big\}$$

consists of Banach-space valued functions that have finite second moments. Two examples posed in this setting are given below.

Example 1 (the diffusion equation with random diffusivity). Find a function $u : \overline{D} \times \Omega \to \mathbb{R}$, such that it holds $\mathbb{P}$-a.e. in Ω that

$$\begin{cases} -\nabla \cdot (a(x,\omega)\nabla u(x,\omega)) = f(x) & \text{in } D, \\ u(x,\omega) = 0 & \text{on } \partial D, \end{cases} \tag{2.2}$$

where the well-posedness is guaranteed in $L^2_{\mathbb{P}}(\Omega; W(D)) = L^2_{\mathbb{P}}(\Omega; H^1_0(D))$ with $f(x) \in L^2(D)$ and $a(x,\omega)$ uniformly elliptic, i.e., for $\mathbb{P}$-a.e. $\omega \in \Omega$,

$$a_{\min} \le \|a(x,\omega)\|_{L^\infty(D)} \text{ with } a_{\min} \in (0,\infty). \tag{2.3}$$

Example 2 (the convection-dominated transport with random velocity). Find a function $u : \overline{D} \times \Omega \to \mathbb{R}$, such that it holds $\mathbb{P}$-a.e. in Ω that

$$\begin{cases} -\varepsilon \Delta u(x,\omega) + \boldsymbol{b}(x,\omega) \cdot \nabla u(x,\omega) = f(x) & \text{in } D, \\ u(x,\omega) = w(x) & \text{on } \partial D, \end{cases} \tag{2.4}$$

where $\varepsilon > 0$ and the boundary condition $w(x)$ is defined by

$$w(x) := \begin{cases} 1, & x \in \mathscr{D}, \\ 0, & x \in \partial D \backslash \mathscr{D}, \end{cases}$$

where $\mathscr{D}$ is a subset of the boundary ∂D. When ε is very small, the solution has a sharp transition layer whose location is determined by the velocity field.

The two examples exhibit different aspects of the parametric map. In Example 1, the random diffusion operator leads to a very smooth solution in both the physical domain D and the parametric domain Ω; in Example 2, the random velocity field $\boldsymbol{b}(x,\omega)$ results in sharp transitions of the solution u in the domain D, and such irregular behavior will propagate to the parametric domain Ω.

2.1. The random fields of interest. We first define a generic stochastic process on $(\Omega, \mathcal{F}, \mathbb{P})$, denoted by

$$\eta(x,\omega) : D \times \Omega \to \mathbb{R}. \tag{2.5}$$

For a fixed $x \in D$, $\eta(x,\cdot)$ is a real-value square integrable random variable, i.e.,

$$\eta(x,\cdot) \in L^2(\Omega, \mathcal{F}, \mathbb{P}) := \left\{ X : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega |X(\omega)|^2 d\mathbb{P}(\omega) < \infty \right\}, \tag{2.6}$$

where $L^2(\Omega, \mathcal{F}, \mathbb{P})$ is equipped with the inner product $\langle X, Y \rangle_{\mathbb{P}} := \mathbb{E}[XY]$ and the norm $\|X\|_{\mathbb{P}} = \langle X, Y \rangle^{1/2}_{\mathbb{P}}$. For notational simplicity, we assume that $\mathbb{E}[\eta(x,\cdot)] = 0$ for all $x \in D$. The covariance function, denoted by

$$\kappa(x,x') := \mathbb{E}[\eta(x,\omega)\eta(x',\omega)], \tag{2.7}$$

is symmetric and bounded. The random fields $a(x,\omega)$ and/or $\boldsymbol{b}(x,\omega)$ in (2.1) are (nonlinear) functions of $\eta(x,\omega)$. For example, the diffusion coefficient could be defined as $a(x,\omega) := \exp(\eta(x,\omega))$ to satisfy the assumption in (2.3); and the random velocity (for $d = 2$) could be defined as $b_1(x,\omega) := \cos(\eta(x,\omega))$ and $b_2(x,\omega) := \sin(\eta(x,\omega))$. Of particular interest to this effort are the colored noise discussed in section 2.1.1 and the discrete white noise discussed in section 2.1.2.

2.1.1. The colored noise. When the covariance $\kappa(x,x')$ is continuous in $D \times D$, we define a compact positive self-adjoint operator $K : L^2(D) \to L^2(D)$, i.e., $K[v](x) := \int_D \kappa(x,x')v(x')dx'$, such that $K[\cdot]$ has a complete set of eigenvectors $\{\eta_n(x), n \in \mathbb{N}^+\}$ in $L^2(D)$ and real eigenvalues $\{\lambda_n, n \in \mathbb{N}^+\}$. On the other hand, the continuity of $\kappa(x,x')$ implies that the random field $\eta(x,\omega)$ is a mean-square continuous stochastic process, i.e., $\lim_{\varepsilon\to 0}\mathbb{E}[(\eta(x+\varepsilon,\omega)-\eta(x,\omega))^2] = 0$. Thus, $\eta(x,\omega)$ can be represented in the space $span\{\xi_n(x), n \in \mathbb{N}^+\}$ as

$$\eta(x,\omega) = \sum_{n=1}^{\infty} \sqrt{\lambda_n}\,\xi_n(x)y_n(\omega), \tag{2.8}$$

where the random variables y_n are defined by

$$y_n(\omega) := \int_D \eta(x,\omega)\xi_n(x)dx \quad \text{for} \quad n = 1,2,\ldots,$$

satisfying $\mathbb{E}[y_n] = 0$, $\mathbb{E}[y_n y_m] = \delta_{nm}$, and $\mathbb{V}ar[y_n] = 1$. Note that, as long as the correlation is nonzero, the eigenvalues decrease with n. The decay rate depends on the covariance function. We can approximate $\eta(x,\omega)$ by truncating its KL expansion of the form

$$\eta_N(x,\omega) := \sum_{n=1}^{N} \sqrt{\lambda_n}\,\xi_n(x)y_n(\omega), \tag{2.9}$$

such that $\eta(x,\omega) \approx \eta_N(x,\omega)$ can be simulated by sampling the N-dimensional random vector $\boldsymbol{y} := (y_1,\ldots,y_N)^\top$. The representation in (2.9) can be viewed as an approximate parameterization of the original random field $\eta(x,\omega)$. For notational convenience, we write the truncated KL expansion as a function of $\boldsymbol{y}$, i.e., $\eta_N(x,\boldsymbol{y})$. Figure 3 shows three snapshots of the random field in (5.2) with the covariance function in (5.1) and $L = 0.25$.

Even though there exists a KL expansion for any mean-square continuous stochastic process, it is not easy to obtain the joint probability distribution of the random vector $\boldsymbol{y}$. In practice, Gaussian random fields are the most widely used models, which assume that $\eta(x,\omega)$ is a Gaussian random variable for any fixed $x \in D$. In this case, the noncorrelation property $\mathbb{E}[y_n y_m] = \delta_{nm}$ leads to the *independence* of $y_1,\ldots,y_N$, and thus $\eta_N(x,\boldsymbol{y})$ can be simulated by sampling the standard Gaussian distribution.

2.1.2. The discrete white noise. We assume the spatial domain D is the union of a set of nonoverlapping subdomains D_n^{WN} for $n = 1,\ldots,N$, i.e.,

$$D = \bigcup_{n=1}^{N} \overline{D_n^{\mathrm{WN}}} \quad \text{and} \quad D_n^{\mathrm{WN}} \cap D_m^{\mathrm{WN}} = \emptyset \quad \text{if} \ \ m \neq n. \tag{2.10}$$

Then, $\eta(x,\omega)$ is defined as a random variable in each subdomain D_n^{WN}, i.e.,

$$\eta_N(x,\omega) := \sum_{n=1}^{N} \mathbf{1}_{D_n^{\mathrm{WN}}}(x)y_n(\omega), \tag{2.11}$$

where $\mathbf{1}_{D_n^{\mathrm{WN}}}(x)$ denotes the indicator function of the subdomain D_n^{WN}. The random variables $y_1, \ldots, y_N$ could be either correlated or independent, bounded or unbounded. Once the joint probability distribution of $\boldsymbol{y}$ is defined, realizations of $\eta(x,\omega) = \eta_N(x,\omega)$ can be generated by Monte Carlo sampling. Figure 6 shows three snapshots of the discrete white noise with $N = 256$, $\mathbb{E}[y_n] = 0$, and $\mathbb{V}ar[y_n] = 1$.

3. The deterministic DD method. We briefly review the deterministic DD method for the PDE in (2.1) for a fixed parameter $\omega \in \Omega$, as well as set up necessary notations for our method in section 4. We define a conforming triangle mesh, denoted by $\mathcal{T}_h$, which satisfies regular geometric conditions. In particular, we utilize an FE space, denoted by $X_h \subset H^1(D)$, consisting of piecewise linear continuous basis functions on $\mathcal{T}_h$, and denote by X_h^0 the homogeneous counterpart of X_h.

For a fixed $\omega \in \Omega$, an FE scheme for the PDE in (2.1) is described as follows: seek a function $u_h \in X_h$ satisfying $u_h|_{\partial D} = w(x)$ and

$$\mathcal{A}(u_h, \nu; \omega) = (f, \nu) \ \ \forall \nu \in X_h^0, \tag{3.1}$$

where $\mathcal{A}(u_h, \nu; \omega)$ is a parameterized bilinear form. Then, the solution u_h can be represented in X_h in the form of

$$u_h(x,\omega) = \sum_{j=1}^{J} U_j(\omega)\, \psi_j(x),$$

where $\{\psi_j(x)\}_{j=1}^J$ is a basis of X_h with J degrees of freedom, and the vector $\mathbf{U}(\omega) := (U_1(\omega), \ldots, U_J(\omega))^\top$ consists of all the nodal values. It should be noted that different PDEs may need different definitions of the bilinear form in (3.1). For example, for the diffusion problem in Example 1, the bilinear form is simply $\mathcal{A}(\mu, \nu; \omega) := (a(x,\omega)\nabla\mu, \nabla\nu)$; for the convection-dominated transport problem in Example 2, we might need to use the streamline-upwind Petrov-Galerkin (SUPG) method, i.e.,

$$\mathcal{A}(\mu, \nu; \omega) := (\varepsilon\nabla\mu, \nabla\nu) + (\boldsymbol{b}\cdot\nabla\mu, \nu) + \sum_{\tau\in\mathcal{T}_h} \delta_\tau(-\varepsilon\Delta\mu + \boldsymbol{b}\cdot\nabla\mu - f, \boldsymbol{b}\cdot\nabla\nu)_\tau,$$

to stabilize the FE scheme, where δ_τ is a nonnegative stabilization parameter, and $(\cdot,\cdot)_\tau$ is the inner product within the triangle τ.

Now we introduce the domain decomposition. We decompose the physical domain D into S nonoverlapping subdomains, denoted by D_s^H, $s = 1, 2, \ldots, S$, such that

$$D = \bigcup_{s=1}^{N} \overline{D_s^H} \quad \text{and} \quad D_s^H \cap D_t^H = \emptyset \quad \text{if} \ \ s \neq t,$$

and we denote the collection of all the edges and interfaces by

$$\mathcal{E}^H := \bigcup_{s=1}^{S} \partial D_s^H \backslash \partial D.$$

One example of such a decomposition can be found in Figure 1(a), where D is a two-dimensional square domain. Nevertheless, our method can be used for domains with more complicated geometries, as long as the decomposition is embedded in the triangle mesh. Based on the embeddedness, we restrict $\mathcal{T}_h$ and X_h in $\overline{D_s^H}$, and define

$$\mathcal{T}_{h,s} := \mathcal{T}_h \cap \overline{D_s^H} \quad \text{and} \quad X_{h,s} := X_h|_{\overline{D_s^H}},$$

where J_s is the degrees of freedom of $X_{h,s}$.

Within each subdomain, we can write out a local weak formulation $\mathcal{A}(u_h, \nu; \omega)_s = (f, \nu)_s$ for all $\nu \in X_{h,s}$, which immediately leads to a local algebraic equation

$$\mathbb{A}_s \mathbf{U}_s = \mathbf{f}_s, \tag{3.2}$$

where $\mathbb{A}_s$ is the local stiffness matrix, $\mathbf{f}_s$ is the local right-hand-side vector, and $\mathbf{U}_s := (U_{s,1}, \ldots, U_{s,J_s})^\top$ is the vector of local nodal values in $\overline{D_s^H}$. The system in (3.2) is singular due to the lack of a boundary condition. Thus, we divide the components in $\mathbf{U}_s$ into two groups, i.e.,

$$\mathbf{U}_s := \begin{pmatrix} \mathbf{U}_s^0 \\ \mathbf{U}_s^{\mathrm{b}} \end{pmatrix},$$

where $\mathbf{U}_s^0$ and $\mathbf{U}_s^{\mathrm{b}}$ are the nodal values in the interior and on the boundary of D_s^H, respectively.[1] Then, we can recast the algebraic equation (3.2) in the following form:

$$\begin{pmatrix} \mathbb{A}_s^{00} & \mathbb{A}_s^{0\mathrm{b}} \\ \mathbb{A}_s^{\mathrm{b}0} & \mathbb{A}_s^{\mathrm{bb}} \end{pmatrix} \begin{pmatrix} \mathbf{U}_s^0 \\ \mathbf{U}_s^{\mathrm{b}} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_s^0 \\ \mathbf{f}_s^{\mathrm{b}} \end{pmatrix}, \tag{3.3}$$

where $\mathbf{f}_s^0$ and $\mathbf{f}_s^{\mathrm{b}}$ are the right-hand sides corresponding to $\mathbf{U}_s^0$ and $\mathbf{U}_s^{\mathrm{b}}$, respectively.

The system in (3.3) can be further manipulated to eliminate the interior unknowns $\mathbf{U}_s^0$ by representing them using $\mathbf{U}_s^{\mathrm{b}}$, i.e.,

$$\mathbf{U}_s^0 = (\mathbb{A}_s^{00})^{-1} \left(\mathbf{f}_s^0 - \mathbb{A}_s^{0\mathrm{b}} \mathbf{U}_s^{\mathrm{b}} \right), \tag{3.4}$$

as well as to define the local Schur complement

$$\mathbb{B}_s := \mathbb{A}_s^{\mathrm{bb}} - \mathbb{A}_s^{\mathrm{b}0} \left(\mathbb{A}_s^{00} \right)^{-1} \mathbb{A}_s^{0\mathrm{b}} \quad \text{and} \quad \mathbf{g}_s := \mathbf{f}_s^{\mathrm{b}} - \mathbb{A}_s^{\mathrm{b}0} \left(\mathbb{A}_s^{00} \right)^{-1} \mathbf{f}_s^0. \tag{3.5}$$

Substituting the relation (3.4) into the linear system (3.2), we can assemble a global system to solve the unknowns on the interfaces. Specifically, we need to define a manipulation matrix $\mathbb{T}_s$ for each subdomain, and assemble

$$\mathbb{B} := \sum_{s=1}^{S} \mathbb{T}_s^\top \mathbb{B}_s \mathbb{T}_s, \quad \mathbf{g} := \sum_{s=1}^{S} \mathbb{T}_s^\top \mathbf{g}_s, \quad \mathbf{U}^{\mathrm{b}} := \sum_{s=1}^{S} \mathbb{T}_s^\top \mathbf{U}_s^{\mathrm{b}}, \tag{3.6}$$

where the matrix $\mathbb{T}_s$ is used to put the entries of $\mathbb{B}_s$, $\mathbf{g}_s$, and $\mathbf{U}_s^{\mathrm{b}}$ to the correct locations in $\mathbb{B}$, $\mathbf{g}$, and $\mathbf{U}^{\mathrm{b}}$, respectively. Note that the size of the square matrix $\mathbb{B}$ is smaller than $\sum_{s=1}^{S} \dim(\mathbf{U}_s^{\mathrm{b}})$ due to shared interfaces between subdomains. After this, $\mathbf{U}^{\mathrm{b}}$ can be obtained by solving the condensed system

$$\mathbb{B}\mathbf{U}^{\mathrm{b}} = \mathbf{g}, \tag{3.7}$$

and $\mathbf{U}_s^0$ can be recovered by substituting $\mathbf{U}_s^{\mathrm{b}}$ into (3.4).

Our goal is to apply the DD method in the stochastic setting. It is easy to see that the main cost of assembling the condensed matrix $\mathbb{B}_s$ in (3.5) lies in the inversion of $\mathbb{A}_s^{00}$, especially on a very fine triangle mesh. We either need to compute the real inverse of $\mathbb{A}_s^{00}$ or solve the linear system $\mathbb{A}_s^{00}\mathbf{v} = \mathbf{w}$ with J_s different right-hand sides (see [37]

[1] Note that the superscript ^{0} of a vector or a matrix indicates that the entries associate with the nodal values in the interior of a subdomain. Analogously, the superscript $^{\mathrm{b}}$ indicates the association with nodal values on interfaces.

for details), both of which are very time consuming. In the stochastic setting, entries of $\mathbb{B}_s$ are functions of the random parameters, so that the inefficient computation of $\mathbb{B}_s$ for a large number of parameter samples is the bottleneck of applying the DD strategy to the parametric PDEs. Thus, how to efficiently approximate $\mathbb{B}_s$ in the parameter space is the focus of the next section.

4. The DDMR method. We describe the details of the proposed DDMR method in this section. The decomposition of random fields is discussed in section 4.1; the offline and the online procedures are discussed in sections 4.2 and 4.3, respectively.

4.1. Decomposition of the random fields. We intended to decompose the random fields of interest into the following form:

$$\eta(x,\omega) \approx \eta_{\boldsymbol{N}}(x,\omega) := \sum_{s=1}^{S} \eta^{\mathrm{loc}}_{s,N_s}(x,\omega)\mathbf{1}_{D_s^H}(x), \tag{4.1}$$

where $\boldsymbol{N} := (N_1,\dots,N_S)$ is the vector of dimensions of local random fields, $\mathbf{1}_{D_s^H}(x)$ is the indicator of the subdomain D_s^H, and $\eta^{\mathrm{loc}}_{s,N_s}(x,\omega)$ is the local random field with the support D_s^H. Both types of random fields introduced in section 2.1 can be decomposed and approximated by the form in (4.1). We discuss the colored noise case in section 4.1.1 and the discrete white noise case in section 4.1.2.

4.1.1. Local KL expansion for the colored noise. For any mean-square continuous random field η defined in D, we can restrict it to each subdomain $D_s^H \subset D$ and define a local KL expansion using the same covariance function κ, i.e.,

$$\eta^{\mathrm{loc}}_s(x,\omega) := \sum_{n=1}^{\infty} \sqrt{\lambda_{s,n}}\,\xi_{s,n}(x)y_{s,n}(\omega) \quad \text{for } s = 1,\dots,S, \tag{4.2}$$

such that $\eta(x,\omega) = \eta^{\mathrm{loc}}_s(x,\omega)$ for $x \in D_s^H$. Similarly, we can truncate the local KL expansion of η^{loc}_s and define the following approximation:

$$\eta^{\mathrm{loc}}_{s,N_s}(x,\omega) := \sum_{n=1}^{N_s} \sqrt{\lambda_{s,n}}\,\xi_{s,n}(x)\,y_{s,n}(\omega) \quad \forall x \in D_s^H \tag{4.3}$$

for $s = 1,\dots,S$. Analogously, we can also write $\eta^{\mathrm{loc}}_{s,N_s}(x,\boldsymbol{y}_s)$ as a function of the local random variables $\boldsymbol{y}_s := (y_{s,1},\dots,y_{s,N_s})^\top$. It is easy to see that the restriction of each realization of (2.8) in D_s^H corresponds to a unique realization of the local representation η^{loc}_s in (4.2), but it is not true for $\eta_N(x,\boldsymbol{y})$ and $\eta^{\mathrm{loc}}_{s,N_s}(x,\boldsymbol{y}_s)$ due to the truncation. For each sample of $\boldsymbol{y}$ in (2.9), we would like to find a sample of $\boldsymbol{y}_s$ to minimize the error $\eta_N(x,\boldsymbol{y}) - \eta^{\mathrm{loc}}_{s,N_s}(x,\boldsymbol{y}_s)$ in the L^2 sense, i.e., to solve the optimization problem

$$\boldsymbol{y}_s = \operatorname*{arg\,min}_{\boldsymbol{v}\in\mathbb{R}^{N_s}} \left\|\eta_N(\cdot,\boldsymbol{y}) - \eta^{\mathrm{loc}}_{s,N_s}(\cdot,\boldsymbol{v})\right\|^2_{L^2(D_s^H)}. \tag{4.4}$$

An immediate question about the problem (4.4) is how big the minimized L^2 error is. In fact, for any fixed $\boldsymbol{y}$, there exists an $\omega^* \in \Omega$ such that $\eta(x,\omega^*) = \eta^{\mathrm{loc}}_s(x,\omega^*)$ for any $x \in D_s^H$. We denoted by $\boldsymbol{y}^*_s(\omega^*)$ the image of ω^* and substituted $\boldsymbol{y}^*_s(\omega^*)$ into

(4.3). Then, the error minimized by $\boldsymbol{y}_s$ in (4.4) can be estimated by

$$\begin{aligned}
&\mathbb{E}\left[\left\|\eta_N(\cdot,\boldsymbol{y})-\eta^{\mathrm{loc}}_{s,N_s}(\cdot,\boldsymbol{y}_s)\right\|^2_{L^2(D_s^H)}\right]\\
&\le \mathbb{E}\left[\left\|\eta_N(\cdot,\boldsymbol{y})-\eta^{\mathrm{loc}}_{s,N_s}(\cdot,\boldsymbol{y}^*_s)\right\|^2_{L^2(D_s^H)}\right]\\
&\le \mathbb{E}\left[\left\|\eta_N(\cdot,\boldsymbol{y}(\omega^*))-\eta(\cdot,\omega^*)\right\|^2_{L^2(D_s^H)}\right]+\mathbb{E}\left[\left\|\eta^{\mathrm{loc}}_s(\cdot,\omega^*)-\eta^{\mathrm{loc}}_{s,N_s}(\cdot,\boldsymbol{y}^*_s(\omega^*))\right\|^2_{L^2(D_s^H)}\right]\\
&\le \mathbb{E}\left[\sum_{n=N+1}^{\infty}\lambda_n\, y_n^2(\omega^*)\right]+\mathbb{E}\left[\sum_{n=N_s+1}^{\infty}\lambda_{s,n}\, y_{s,n}^2(\omega^*)\right]\\
&= \sum_{n=N+1}^{\infty}\lambda_n+\sum_{n=N_s+1}^{\infty}\lambda_{s,n},
\end{aligned}$$

which implies that $\eta^{\mathrm{loc}}_{s,N_s}(x,\boldsymbol{y}_s)$ can provide a good approximation to $\eta_N(x,\boldsymbol{y})$ for sufficiently large N and N_s.

In practice, the first step to solve (4.4) is to obtain the eigenvalues and eigenvectors of the covariance function in D and D_s^H, respectively. It can be done analytically for certain types of covariance functions, e.g., exponential and Gaussian, or numerically using Galerkin projection (see section 2.1 in [23]) and efficient solvers for eigenvalue problems, e.g., ARPACK.[2] After that, the problem in (4.4) can be implemented using the discrete least-squares (DLS) method. Specifically, we drew a set of T uniformly distributed points that are not vertices of $\mathcal{T}_h$, denoted by $\{x_{s,i}\}_{i=1}^T$, in the subdomain D_s^H and formulated the following DLS problem:

$$\boldsymbol{y}_s=\arg\min_{\boldsymbol{v}\in\mathbb{R}^{N_s}}\sum_{i=1}^{T}\left|\eta_N(x_{s,i},\boldsymbol{y})-\eta^{\mathrm{loc}}_{s,N_s}(x_{s,i},\boldsymbol{v})\right|^2; \tag{4.5}$$

the optimal choice of $\boldsymbol{y}_s$ was obtained by solving the normal system

$$(\boldsymbol{\Xi}_s^\top\boldsymbol{\Xi}_s)\boldsymbol{y}_s=\boldsymbol{\Xi}_s^\top\boldsymbol{\eta}_N, \tag{4.6}$$

where

$$\begin{aligned}
&[\boldsymbol{\Xi}_s]_{ij}:=\sqrt{\lambda_{s,j}}\,\xi_{s,j}(x_{s,i}),\ \ i=1,\dots,T \text{ and } j=1,\dots,N_s,\\
&\boldsymbol{\eta}_N:=(\eta_N(x_{s,i},\boldsymbol{y}),\dots,\eta_N(x_{s,T},\boldsymbol{y}))^\top
\end{aligned}$$

are a $T\times N_s$ matrix and a T-dimensional vector, respectively. Note that the number of samples T needs to be bigger than N_s to guarantee the stability of the DLS method. After solving the DLS problems in all subdomains, we constructed an approximate global KL expansion by substituting all the local expansions in (4.3) into (4.1).

Remark 1. Since the colored noise is discretized on the triangle mesh $\mathcal{T}_h$ as a piecewise constant function, we do not need to force continuity of the local KL expansion obtained by solving (4.6), as long as the domain decomposition is embedded in the triangle mesh $\mathcal{T}_h$.

Remark 2. Since this effort does not focus on improving the KL expansion generation algorithms, the strategy used here is feasible but surely not optimal. It becomes inefficient when the triangle mesh $\mathcal{T}_h$ becomes very dense. In that case, the cost of decomposing the covariance function, i.e., solving a Fredholm integral equation of the

[2] http://www.caam.rice.edu/software/ARPACK/

second kind, is very time consuming. In fact, a DD-based parallel KL expansion generator was developed in [11], which can be directly applied in our setting to improve the efficiency of this step.

4.1.2. Decomposition of the discrete white noise. A straightforward way to decompose the discrete white noise in (2.10) is to align both decompositions, i.e., letting $S = N$ and $D_n^{\mathrm{WN}} = D_s^H$ for $s = n = 1, \ldots, S$. In this case, each subdomain only involves one random parameter, which can avoid the curse of dimensionality. This is the strategy we used for the numerical experiments because we only consider the case that N in (2.11) is finite in this work.

4.2. The offline procedure. The purpose of the offline procedure is to construct all the components that are needed in the reduced model, as well as finish all the expensive computation. Details about our offline procedure are given in sections 4.2.1–4.2.3, and a short summary is given in section 4.2.4.

4.2.1. Training data generation. We needed two sets of training data, one for the solution and the other for the local stiffness matrices. We take the colored noise case as an example to describe how to generate the training data. The described procedure can be directly applied to the case of having the discrete white noise.

To generate data of u_h, we set the dimension N in (2.9) sufficiently large, such that the error $\eta - \eta_N$ can be neglected. According to the definition of η_N in (2.9), we sampled the multivariate Gaussian distribution $\mathcal{N}(0, \mathbb{I})$ to generate K_u realizations of η_N, denoted by $\{\eta_N(x, \boldsymbol{y}(\omega_k)),\ k = 1, \ldots, K_u\}$, each of which was stored as a piecewise constant function on the mesh $\mathcal{T}_h$. Then, we substituted $\{\eta_N(x, \boldsymbol{y}(\omega_k)),\ k = 1, \ldots, K_u\}$ into the weak formulation in (3.1), and computed the realizations of all the unknowns, denoted by

$$\{\mathbf{U}_s(\omega_k), s = 1, \ldots, S, k = 1, \ldots, K_u\}. \tag{4.7}$$

Note that the data in (4.7) was obtained based on realizations of the global expansion η_N and the FE formulation in (3.1), such that $\mathbf{U}_s(\omega_k)$ does not contain the projection error caused by (4.5). This procedure was used to generate not only the training data, but also the validation data to test the performance of our method in the numerical examples in section 5.

Next, we discuss how to generate training data for the local stiffness matrices. To do this, we instead sampled from the local random variables $\boldsymbol{y}_s$. It is known that $\boldsymbol{y}_s$ in (4.3) also follow the multivariate standard normal distribution $\mathcal{N}(0, \mathbb{I})$. However, due to the truncation of both the global and the local KL expansions, the distribution of $\boldsymbol{y}_s$ obtained by (4.5) is perturbed from $\mathcal{N}(0, \mathbb{I})$. To avoid unexpected errors, we used an overkilling strategy, i.e., drawing K_y samples of $\boldsymbol{y}_s$ *uniformly* in a bounded rectangle domain $\Gamma_s \subset \mathbb{R}^{N_s}$. The domain Γ_s was set large enough such that the probability of having a sample $\boldsymbol{y}_s \sim \mathcal{N}(0, \mathbb{I})$ fall outside Γ_s is smaller than a tolerance. Due to the low dimensionality of $\boldsymbol{y}_s$, the number of samples of $\boldsymbol{y}_s$ that fell in the low probability region is relatively small. An alternative strategy is to use the results on polynomial approximations in irregular domains in [1]. After $\{\boldsymbol{y}_s(\omega_k),\ k = 1, \ldots, K_y\}$ was generated, we substituted them into (4.3) to obtain

$$\left\{\eta_{s,N_s}^{\mathrm{loc}}(x, \boldsymbol{y}_s(\omega_k)),\ k = 1, \ldots, K_y\right\}. \tag{4.8}$$

Substituting such a set into the stiffness matrix in (3.3), we assembled an approximate

local stiffness matrix, denoted by

$$\widetilde{\mathbb{A}}_s(\boldsymbol{y}_s(\omega_k)) := \begin{pmatrix} \widetilde{\mathbb{A}}_s^{00}(\boldsymbol{y}_s(\omega_k)) & \widetilde{\mathbb{A}}_s^{0\mathrm{b}}(\boldsymbol{y}_s(\omega_k)) \\ \widetilde{\mathbb{A}}_s^{\mathrm{b}0}(\boldsymbol{y}_s(\omega_k)) & \widetilde{\mathbb{A}}_s^{\mathrm{bb}}(\boldsymbol{y}_s(\omega_k)) \end{pmatrix} \approx \mathbb{A}_s(\boldsymbol{y}(\omega_k)). \tag{4.9}$$

Note that the error between $\widetilde{\mathbb{A}}_s(\boldsymbol{y}_s(\omega_k))$ and $\mathbb{A}_s(\boldsymbol{y}(\omega_k))$ comes from the projection in (4.5) from the truncated global expansion η_N onto the truncated local expansion $\eta_{s,N_s}^{\mathrm{loc}}$. Nevertheless, $\mathbb{A}_s(\boldsymbol{y}(\omega_k)) = \widetilde{\mathbb{A}}_s(\boldsymbol{y}_s(\omega_k))$, $k = 1, \ldots, K_y$, for the discrete white noise introduced in section 2.1.2 because there is no global-local projection in the domain decomposition.

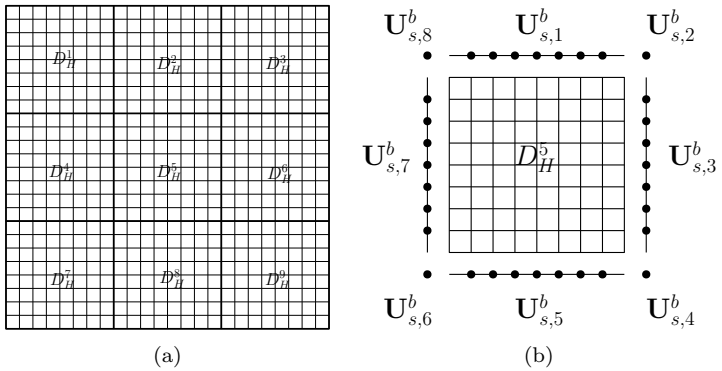
4.2.2. Constructing reduced global and local linear systems. To reduce the size of the equation in (3.7), we defined a set of snapshots for each interface of each subdomain,

$$\mathbb{V}_{s,j}^{\mathrm{b}} := [\mathbf{U}_{s,j}^{\mathrm{b}}(\omega_1), \ldots, \mathbf{U}_{s,j}^{\mathrm{b}}(\omega_K)] \quad \text{for} \quad j = 1, \ldots, E_s, \quad s = 1, \ldots, S, \tag{4.10}$$

where E_s denotes the number of *nonoverlapping* groups of the nodal values on $\partial D_s^H \backslash \partial D$ and $\mathbf{U}_{s,j}^{\mathrm{b}}$ denotes the vector of the nodal values of the jth group. Taking Figure 1(b) as an example, we can divide the interface nodal values of D_5^H into eight nonoverlapping groups, i.e., $E_s = 8$, where $\mathbf{U}_{s,1}^{\mathrm{b}}, \mathbf{U}_{s,3}^{\mathrm{b}}, \mathbf{U}_{s,5}^{\mathrm{b}}, \mathbf{U}_{s,7}^{\mathrm{b}}$ represent the unknowns on the edges and $\mathbf{U}_{s,2}^{\mathrm{b}}, \mathbf{U}_{s,4}^{\mathrm{b}}, \mathbf{U}_{s,6}^{\mathrm{b}}, \mathbf{U}_{s,8}^{\mathrm{b}}$ represent the unknowns at vertices. Then, we apply singular value decomposition (SVD) to $\mathbb{V}_{s,j}^{\mathrm{b}}$, so as to generate a reduced basis

$$\widehat{\mathbb{V}}_{s,j}^{\mathrm{b}} := [\mathbf{V}_{s,j,1}^{\mathrm{b}}, \ldots, \mathbf{V}_{s,j,M_{s,j}}^{\mathrm{b}}], \tag{4.11}$$

where $M_{s,j} < K_u$ is the number of retained left singular vectors. Note that it is possible that $\mathbb{V}_{s,j}^{\mathrm{b}} = \mathbb{V}_{s',j'}^{\mathrm{b}}$ when D_s^H and $D_{s'}^H$ share the interface. In this case, we only need to apply SVD to the interface once. In addition, since the groups representing the vertices of a subdomain, e.g., $\mathbf{U}_{s,2}^{\mathrm{b}}, \mathbf{U}_{s,4}^{\mathrm{b}}, \mathbf{U}_{s,6}^{\mathrm{b}}, \mathbf{U}_{s,8}^{\mathrm{b}}$ in Figure 1(b), only involve one degree of freedom, we do not perform SVD for those groups.



(a) (b)

FIG. 1. (a) *Illustration of coarse blocks* D_s^H $(s = 1, \ldots, 8)$. (b) *Illustration of how to divide the interface nodal values into different groups. We divide the interface nodal values of* D_5^H *into eight nonoverlapping groups, i.e.,* $E_s = 8$, *where* $\mathbf{U}_{s,1}^{\mathrm{b}}, \mathbf{U}_{s,3}^{\mathrm{b}}, \mathbf{U}_{s,5}^{\mathrm{b}}, \mathbf{U}_{s,7}^{\mathrm{b}}$ *represent the unknowns on the edges and* $\mathbf{U}_{s,2}^{\mathrm{b}}, \mathbf{U}_{s,4}^{\mathrm{b}}, \mathbf{U}_{s,6}^{\mathrm{b}}, \mathbf{U}_{s,8}^{\mathrm{b}}$ *represent the unknowns at vertices.*

After the reduced bases were generated for all the groups, we defined a matrix

$$\widehat{\mathbb{V}}_s^{\mathrm{b}} := \begin{pmatrix} \widehat{\mathbb{V}}_{s,1}^{\mathrm{b}} & & & \\ & \widehat{\mathbb{V}}_{s,2}^{\mathrm{b}} & & \\ & & \ddots & \\ & & & \widehat{\mathbb{V}}_{s,E_s}^{\mathrm{b}} \end{pmatrix}, \tag{4.12}$$

such that the unknowns $\mathbf{U}_s^{\mathrm{b}}$ can be approximated by projecting it onto $\widehat{\mathbb{V}}_s^{\mathrm{b}}$, i.e.,

$$\widehat{\mathbf{U}}_s^{\mathrm{b}} = \widehat{\mathbb{V}}_s^{\mathrm{b}}\, \widehat{\mathbf{C}}_s^{\mathrm{b}}, \tag{4.13}$$

where $\widehat{\mathbf{C}}_s^{\mathrm{b}}$ is the coefficient vector of size $\sum_{j=1}^{E_s} M_{s,j}$.

Similarly, we defined a set of snapshots for the interior nodal values of each subdomain, i.e., $\mathbb{V}_s^0 = [\mathbf{U}_s^0(\omega_1), \ldots, \mathbf{U}_s^0(\omega_{K_u})]$ for $s = 1, \ldots, S$, and performed SVD to obtain a reduced basis for the interior unknowns, i.e.,

$$\widehat{\mathbb{V}}_s^0 := [\mathbf{V}_{s,1}^0, \ldots, \mathbf{V}_{s,M_s}^0], \tag{4.14}$$

where $M_s < K_u$ is the dimension of the reduced basis. Consequently, we defined an approximation to the interior unknowns $\mathbf{U}_s^0$ by projecting it onto $\widehat{\mathbb{V}}_s^0$, i.e.,

$$\widehat{\mathbf{U}}_s^0 = \widehat{\mathbb{V}}_s^0\, \widehat{\mathbf{C}}_s^0, \tag{4.15}$$

where $\mathbf{C}_s^{\mathrm{b}}$ is the coefficient vector of size M_s.

Remark 3. Since the reduced bases were constructed locally, the behavior of the PDE solution restricted on an interface or in a subdomain is more regular than the global solution; e.g., the total variation is smaller. As such, random sampling was shown to be effective in constructing accurate bases for the PDEs under consideration (see Figures 5(b), 7(b), 9(b), and 11(a) for the demonstration).

Now we describe how to assemble the reduced versions of (3.4) and (3.7). For $s = 1, \ldots, S$, we defined a reduced system of (3.3) of the form

$$\begin{aligned} &\begin{pmatrix} \widehat{\mathbb{V}}_s^0 & \\ & \widehat{\mathbb{V}}_s^{\mathrm{b}} \end{pmatrix}^{\top} \begin{pmatrix} \widetilde{\mathbb{A}}_s^{00} & \widetilde{\mathbb{A}}_s^{0\mathrm{b}} \\ \widetilde{\mathbb{A}}_s^{\mathrm{b}0} & \widetilde{\mathbb{A}}_s^{\mathrm{bb}} \end{pmatrix} \begin{pmatrix} \widehat{\mathbb{V}}_s^0 & \\ & \widehat{\mathbb{V}}_s^{\mathrm{b}} \end{pmatrix} = \begin{pmatrix} (\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{00}\, \widehat{\mathbb{V}}_s^0 & (\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{0\mathrm{b}}\, \widehat{\mathbb{V}}_s^{\mathrm{b}} \\ (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \widetilde{\mathbb{A}}_s^{\mathrm{b}0}\, \widehat{\mathbb{V}}_s^0 & (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \widetilde{\mathbb{A}}_s^{\mathrm{bb}}\, \widehat{\mathbb{V}}_s^{\mathrm{b}} \end{pmatrix}, \\ &\begin{pmatrix} \widehat{\mathbb{V}}_s^0 & \\ & \widehat{\mathbb{V}}_s^{\mathrm{b}} \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{f}_s^0 \\ \mathbf{f}_s^{\mathrm{b}} \end{pmatrix} = \begin{pmatrix} (\widehat{\mathbb{V}}_s^0)^{\top} \mathbf{f}_s^0 \\ (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \mathbf{f}_s^{\mathrm{b}} \end{pmatrix}. \end{aligned} \tag{4.16}$$

The above system was further manipulated to eliminate the interior unknowns $\widehat{\mathbf{C}}_s^0$ by representing them using $\widehat{\mathbf{C}}_s^{\mathrm{b}}$, i.e.,

$$\widehat{\mathbf{C}}_s^0 = \left((\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{00}\, \widehat{\mathbb{V}}_s^0 \right)^{-1} \left((\widehat{\mathbb{V}}_s^0)^{\top} \mathbf{f}_s^0 - (\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{0\mathrm{b}}\, \widehat{\mathbb{V}}_s^{\mathrm{b}}\, \widehat{\mathbf{C}}_s^{\mathrm{b}} \right). \tag{4.17}$$

The reduced forms of the matrix and vector in (3.5) were then defined by

$$\widehat{\mathbb{B}}_s := (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \widetilde{\mathbb{A}}_s^{\mathrm{bb}}\, \widehat{\mathbb{V}}_s^{\mathrm{b}} - (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \widetilde{\mathbb{A}}_s^{\mathrm{b}0}\, \widehat{\mathbb{V}}_s^0 \left((\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{00}\, \widehat{\mathbb{V}}_s^0 \right)^{-1} (\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{0\mathrm{b}} \widehat{\mathbb{V}}_s^{\mathrm{b}} \tag{4.18}$$

and

$$\widehat{\mathbf{g}}_s := (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \mathbf{f}_s^{\mathrm{b}} - (\widehat{\mathbb{V}}_s^{\mathrm{b}})^{\top} \widetilde{\mathbb{A}}_s^{\mathrm{b}0}\, \widehat{\mathbb{V}}_s^0 \left((\widehat{\mathbb{V}}_s^0)^{\top} \widetilde{\mathbb{A}}_s^{00}\, \widehat{\mathbb{V}}_s^0 \right)^{-1} (\widehat{\mathbb{V}}_s^0)^{\top} \mathbf{f}_s^0. \tag{4.19}$$

Then, we canceled out $\widehat{\mathbf{C}}_s^0$ and assembled a global system to solve $\widehat{\mathbf{C}}_s^{\mathrm{b}}$. Specifically, we defined another manipulation matrix $\widehat{\mathbb{T}}_s$ for each subdomain, and assembled

$$\widehat{\mathbb{B}} := \sum_{s=1}^{S} \widehat{\mathbb{T}}_s^\top \widehat{\mathbb{B}}_s \, \widehat{\mathbb{T}}_s, \quad \widehat{\mathbf{g}} := \sum_{s=1}^{S} \widehat{\mathbb{T}}_s^\top \widehat{\mathbf{g}}_s, \quad \widehat{\mathbf{C}}^{\mathrm{b}} := \sum_{s=1}^{S} \widehat{\mathbb{T}}_s^\top \widehat{\mathbf{C}}_s^{\mathrm{b}}, \tag{4.20}$$

where the matrix $\widehat{\mathbb{T}}_s$ was used to put the entries of $\widehat{\mathbb{B}}_s$, $\widehat{\mathbf{g}}_s$, and $\widehat{\mathbf{C}}_s^{\mathrm{b}}$ to the correct locations in $\widehat{\mathbb{B}}$, $\widehat{\mathbf{g}}$, and $\widehat{\mathbf{C}}^{\mathrm{b}}$, respectively. Note that the size of the reduced square matrix $\widehat{\mathbb{B}}$ is smaller than $\sum_{s=1}^{S} \dim(\widehat{\mathbf{C}}_s^{\mathrm{b}})$ due to shared interfaces between subdomains. After this, $\widehat{\mathbf{C}}^{\mathrm{b}}$ can be obtained by solving

$$\widehat{\mathbb{B}} \, \widehat{\mathbf{C}}^{\mathrm{b}} = \widehat{\mathbf{g}}, \tag{4.21}$$

and $\widehat{\mathbf{C}}_s^0$ can be recovered by substituting $\widehat{\mathbf{C}}_s^{\mathrm{b}}$ into (4.17).

Since each matrix of snapshots $\mathbb{V}_s^0$ or $\mathbb{V}_{s,j}^{\mathrm{b}}$ only covers a small portion of nodal values, its singular values decay faster than the case of applying SVD to the matrix of snapshots of all nodal values of the global system. Similar to the local KL expansion, the decay rate of the singular values of $\mathbb{V}_s^0$ or $\mathbb{V}_{s,j}^{\mathrm{b}}$ also depends on the size of the subdomains. A large number of subdomains will lead to a faster decay, so that a smaller $M_{s,j}$ would be sufficient to achieve a prescribed accuracy. Nevertheless, it is difficult to design the optimal domain decomposition strategy to obtain a system in (4.21) with minimal size.

4.2.3. Sparse approximation of the local stiffness matrices. So far we managed to reduce the sizes of global and local systems by generating reduced bases in the subdomains and on the interfaces. The remaining task is to reduce the cost of assembling the stiffness matrices $\widehat{\mathbb{B}}_s(\omega)$ and the right-hand side $\widehat{\mathbf{g}}_s(w)$ for a large number of samples of ω. To do this, we constructed sparse polynomial approximations to the entries of the following matrices:

$$\begin{aligned} \widehat{\mathbb{A}}_s^{00}(\boldsymbol{y}_s) &:= (\widehat{\mathbb{V}}_s^0)^\top \widetilde{\mathbb{A}}_s^{00}(\boldsymbol{y}_s) \, \widehat{\mathbb{V}}_s^0, \qquad \widehat{\mathbb{A}}_s^{0\mathrm{b}}(\boldsymbol{y}_s) := (\widehat{\mathbb{V}}_s^0)^\top \widetilde{\mathbb{A}}_s^{0\mathrm{b}}(\boldsymbol{y}_s) \, \widehat{\mathbb{V}}_s^{\mathrm{b}}, \\ \widehat{\mathbb{A}}_s^{\mathrm{b}0}(\boldsymbol{y}_s) &:= (\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \widetilde{\mathbb{A}}_s^{\mathrm{b}0}(\boldsymbol{y}_s) \, \widehat{\mathbb{V}}_s^0, \qquad \widehat{\mathbb{A}}_s^{\mathrm{bb}}(\boldsymbol{y}_s) := (\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \widetilde{\mathbb{A}}_s^{\mathrm{bb}}(\boldsymbol{y}_s) \, \widehat{\mathbb{V}}_s^{\mathrm{b}} \end{aligned} \tag{4.22}$$

for $s = 1, \ldots, S$, where $\widetilde{\mathbb{A}}_s^{00}, \widetilde{\mathbb{A}}_s^{0\mathrm{b}}, \widetilde{\mathbb{A}}_s^{\mathrm{b}0}, \widetilde{\mathbb{A}}_s^{\mathrm{bb}}$ are as defined in (4.9). It should be noted that the matrices in (4.22) only depend on the local random vector $\boldsymbol{y}_s$ of dimension N_s, so that we can exploit the dimension reduction benefit in the sparse approximation. Specifically, we defined a sparse polynomial approximation in the bounded domain $\Gamma_s \subset \mathbb{R}^{N_s}$ using a Legendre basis, i.e.,

$$\begin{aligned} \widehat{\mathbb{A}}_s^{00,\mathrm{LS}}(\boldsymbol{y}_s) &:= \sum_{m=1}^{M_{s,p}^{\mathrm{LS}}} \widehat{\mathbb{K}}_{s,m}^{00} \, L_m(\boldsymbol{y}_s), \qquad \widehat{\mathbb{A}}_s^{0\mathrm{b},\mathrm{LS}}(\boldsymbol{y}_s) := \sum_{m=1}^{M_{s,p}^{\mathrm{LS}}} \widehat{\mathbb{K}}_{s,m}^{0\mathrm{b}} \, L_m(\boldsymbol{y}_s), \\ \widehat{\mathbb{A}}_s^{\mathrm{b}0,\mathrm{LS}}(\boldsymbol{y}_s) &:= \sum_{m=1}^{M_{s,p}^{\mathrm{LS}}} \widehat{\mathbb{K}}_{s,m}^{\mathrm{b}0} \, L_m(\boldsymbol{y}_s), \qquad \widehat{\mathbb{A}}_s^{\mathrm{bb},\mathrm{LS}}(\boldsymbol{y}_s) := \sum_{m=1}^{M_{s,p}^{\mathrm{LS}}} \widehat{\mathbb{K}}_{s,m}^{\mathrm{bb}} \, L_m(\boldsymbol{y}_s), \end{aligned} \tag{4.23}$$

where $L_m(\boldsymbol{y}_s)$ for $m = 1, \ldots, M_{s,p}^{\mathrm{LS}}$ are Legendre polynomials expanding the space $\mathcal{P}_{M_{s,p}^{\mathrm{LS}}}(\Gamma_s)$ with the subscript p showing the maximum polynomial order. The coefficient matrices in (4.23) were computed using the training data generated in (4.9). Note that we computed the upper triangle of the coefficient matrices in (4.23) and then copied the upper triangle to the lower triangle to obtain symmetry.

In the colored noise case, since the dimension N_s of the local random vector $\boldsymbol{y}_s$ is much smaller than the global dimension N, we only needed a small cardinality $M_{s,p}^{\mathrm{LS}}$ to achieve a prescribed accuracy. Moreover, the local dimension N_s can be further reduced by chopping the domain D into more subdomains. In this work, we used *anisotropic* total degree polynomial spaces [25] to define $\mathcal{P}_{M_{s,p}^{\mathrm{LS}}}(\Gamma_s)$, where the anisotropy was determined by the singular value decays of the local KL expansions. More advanced methods could be used to further exploit the sparsity, even though the simple anisotropic space is sufficient to illustrate the superior performance of our method. In the discrete white noise case, we set $D_n^{\mathrm{WN}} = D_s^H$, i.e., aligning the interfaces with the partition of the noise, so that the local dimension is $N_s = 1$.

A major difference between our method and the existing works, e.g., [12], is that we can approximate the local and global Schur complement matrices $\mathbb{B}_s$ and $\mathbb{B}$ without approximating a large number of local problems. Due to the inefficiency of inverting $\mathbb{A}_s^{00}$ and the lack of boundary condition in (3.3), the existing strategies to construct $\mathbb{B}_s$ decompose the local problem in (3.3) into a set of $\sum_{j=1}^{E_j} M_{s,j}$ subproblems and approximate the parametric dependence for all the subproblems. Nevertheless, in our method, we can reduce the sizes of $\widehat{\mathbb{A}}_s^{00}, \widehat{\mathbb{A}}_s^{0\mathrm{b}}, \widehat{\mathbb{A}}_s^{\mathrm{b}0}, \widehat{\mathbb{A}}_s^{\mathrm{bb}}$ to $M_s \times M_s$, $M_s \times M_{s,j}$, $M_{s,j} \times M_s$, $M_{s,j} \times M_{s,j}$ and the dimension of the parametric dependence to N_s. This makes direct approximation of all the entries of $\mathbb{B}_s$ computationally efficient, as well as makes it straightforward to handle nonaffine random coefficients without using empirical interpolation.

On the other hand, the avoidance of building sparse approximations to local PDE solutions is particularly beneficial in solving the convection-dominated problem (see section 5.3). The sharp transition, caused by the discontinuous boundary condition, moves with the random parameters, which results in sharp transitions of the parametric map $\boldsymbol{y}(\omega) \to u_h(x, \omega)$. It is well known that approximating irregular functions is very challenging, especially in high-dimensional spaces (see, e.g., [21, 39]). However, the local stiffness matrices $\widehat{\mathbb{A}}_s^{00}, \widehat{\mathbb{A}}_s^{0\mathrm{b}}, \widehat{\mathbb{A}}_s^{\mathrm{b}0}, \widehat{\mathbb{A}}_s^{\mathrm{bb}}$ are not affected by such irregularity, so that the sparse approximation in (4.23) still features spectral convergence.

Remark 4. Since we assumed η is a Gaussian random field, it seems natural to use Hermite polynomials in (4.23). However, the least-squares projection in (4.5) causes an unknown perturbation of the local random variable $\boldsymbol{y}_s$, such that the Hermite polynomials are not orthogonal with respect to the perturbed kernel. To avoid some unexpected errors in the approximation in (4.23), we used an overkilling strategy, i.e., approximating the entries of the local stiffness matrices in a hypercube that is sufficiently large to cover the high probability region of $\boldsymbol{y}_s$. This is certainly not optimal because the accuracy of the Legendre approximation in the tail region, e.g., the corners of the hypercube, is wasted. Nevertheless, due to the dramatically reduced dimension and analytic regularity, constructing the Legendre approximation is still very efficient (see Figures 5(d), 7(d), 9(d), and 11(c) for demonstrations).

4.2.4. Summary of the offline procedure. The offline procedure discussed can be summarized in Algorithm 1, in the case of having the colored noise. The algorithm for handling the discrete white noise can be obtained by a slight modification.

Algorithm 1: *The offline procedure for the colored noise case*

Input: $D, f, w, a, \boldsymbol{b}$ in (2.1), triangulation $\mathcal{T}_h$, covariance $\kappa(x, x')$ in (2.7);
Output: $\{\lambda_n, \xi_n\}_{n=1}^{N}$, $\{\lambda_{s,n}, \xi_{s,n}\}_{n=1,s=1}^{N_s,S}$, $\{(\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \mathbf{f}_s^{\mathrm{b}}, (\widehat{\mathbb{V}}_s^{0})^\top \mathbf{f}_s^{0}\}_{s=1}^{S}$, $\{\widehat{\mathbb{V}}_s^{0}\}_{s=1}^{S}$, $\{\widehat{\mathbb{V}}_s^{\mathrm{b}}\}_{s=1}^{S}$, $\{\widehat{\mathbb{K}}_{s,m}^{00}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{0\mathrm{b}}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{\mathrm{b}0}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{\mathrm{bb}}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$;

1: Compute eigenvalues λ_n and eigenvectors ξ_n for $n = 1, \ldots, N$ for η_N in (2.9);
2: Decompose the domain D into D_s^H for $s = 1, \ldots, S$;
3: Compute eigenvalues $\lambda_{s,n}$ and eigenvectors $\xi_{s,n}$ for $\eta_{s,N_s}^{\mathrm{loc}}$ in (4.3);
4: Generate training data $\{\mathbf{U}_s(\omega_k),\ s = 1, \ldots, S,\ k = 1, \ldots, K_u\}$ in (4.7);
5: Use SVD to construct $\widehat{\mathbb{V}}_{s,j}^{\mathrm{b}}$ in (4.11) for $j = 1, \ldots, E_s,\ s = 1, \ldots, S$;
6: Assemble $\widehat{\mathbb{V}}_s^{\mathrm{b}}$ in (4.12) for $s = 1, \ldots, S$;
7: Use SVD to construct $\widehat{\mathbb{V}}_s^{0}$ in (4.14) for $s = 1, \ldots, S$;
8: Compute and store $(\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \mathbf{f}_s^{\mathrm{b}}$ and $(\widehat{\mathbb{V}}_s^{0})^\top \mathbf{f}_s^{0}$ for $s = 1, \ldots, S$;
9: Generate training data $\{\widetilde{\mathbb{A}}_s(\boldsymbol{y}_s(\omega_k)),\ s = 1, \ldots, S,\ k = 1, \ldots, K_y\}$ in (4.9);
10: Construct reduced data $\widehat{\mathbb{A}}_s^{00}(\boldsymbol{y}_s), \widehat{\mathbb{A}}_s^{0\mathrm{b}}(\boldsymbol{y}_s), \widehat{\mathbb{A}}_s^{\mathrm{b}0}(\boldsymbol{y}_s), \widehat{\mathbb{A}}_s^{\mathrm{bb}}(\boldsymbol{y}_s)$ using (4.22);
11: Solve the coefficient matrices $\widehat{\mathbb{K}}_{s,m}^{00}, \widehat{\mathbb{K}}_{s,m}^{0\mathrm{b}}, \widehat{\mathbb{K}}_{s,m}^{\mathrm{b}0}, \widehat{\mathbb{K}}_{s,m}^{\mathrm{bb}}$ in (4.23);

In terms of the number of operations, the dominant cost lies in the generation of $\{\mathbf{U}_s(\omega_k),\ s = 1, \ldots, S,\ k = 1, \ldots, K_u\}$ and $\{\widetilde{\mathbb{A}}_s(\boldsymbol{y}_s(\omega_k)),\ s = 1, \ldots, S,\ k = 1, \ldots, K_y\}$. Both numbers K_u and K_y could be reduced by dividing D into more subdomains. The number K_u is an important factor on the quality of the solution manifold coverage by the training data. As the size of each subdomain becomes smaller, the set $\{\mathbf{U}_s(\omega_k),\ k = 1, \ldots, K_u\}$ only covers a smaller submanifold in a lower-dimensional space, and thus a smaller value of K_u can provide sufficient coverage to achieve a prescribed accuracy. The number K_y is related to the cardinality $M_{s,p}^{\mathrm{LS}}$ of the polynomial space. Thus, it is easy to see that a smaller subdomain will lead to a better anisotropy of the local KL expansion, such that $M_{s,p}^{\mathrm{LS}}$ can be further reduced for a given polynomial order p. We would like to emphasize again that the convergence of our method does not require the number of subdomains to go to infinity.

In terms of the storage requirement, the sizes of the matrices for storing bases functions, i.e., $\{\xi_n\}_{n=1}^{N}$, $\{\xi_{s,n}\}_{n=1,s=1}^{N_s,S}$, $\{\widehat{\mathbb{V}}_s^{0}\}_{s=1}^{S}$, and $\{\widehat{\mathbb{V}}_s^{\mathrm{b}}\}_{s=1}^{S}$, depend on the triangle mesh size h, which is inevitable. A major improvement of this effort is that the sizes of the sparse approximation coefficients, i.e., $\widehat{\mathbb{K}}_{s,m}^{00}$, $\widehat{\mathbb{K}}_{s,m}^{0\mathrm{b}}$, $\widehat{\mathbb{K}}_{s,m}^{\mathrm{b}0}$, $\widehat{\mathbb{K}}_{s,m}^{\mathrm{bb}}$, are of the sizes $M_s \times M_s$, $M_s \times M_{s,j}$, $M_{s,j} \times M_s$, $M_{s,j} \times M_{s,j}$, respectively, which are, again, independent of triangle mesh size h. Thus, the space required to store those coefficients is on the order of $\mathcal{O}(SM_{s,p}^{\mathrm{LS}}(M_s + M_{s,j})^2)$ when using the same $M_s, M_{s,j}$, and $M_{s,p}^{\mathrm{LS}}$ for all subdomains.

4.3. The online procedure. The online procedure is to use the output of the offline procedure to recover the solution at a set of new samples. We summarize the online procedure in Algorithm 2 for the case of having the colored noise. The algorithm for handling the discrete white noise can be obtained by a slight modification.

Algorithm 2: *The online procedure for the colored noise case*

Input: $\{\lambda_n, \xi_n\}_{n=1}^{N}$, $\{\lambda_{s,n}, \xi_{s,n}\}_{n=1,s=1}^{N_s,S}$, $\{(\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \mathbf{f}_s^{\mathrm{b}}, (\widehat{\mathbb{V}}_s^{0})^\top \mathbf{f}_s^{0}\}_{s=1}^{S}$, $\{\widehat{\mathbb{V}}_s^{0}\}_{s=1}^{S}$, $\{\widehat{\mathbb{V}}_s^{\mathrm{b}}\}_{s=1}^{S}$, $\{\widehat{\mathbb{K}}_{s,m}^{00}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{0\mathrm{b}}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{\mathrm{b}0}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$, $\{\widehat{\mathbb{K}}_{s,m}^{\mathrm{bb}}\}_{s=1,m=1}^{S,M_{s,p}^{\mathrm{LS}}}$;

Output: $\widehat{\mathbf{C}}^{\mathrm{b,LS}}$, $\{\widehat{\mathbf{C}}_s^{0,\mathrm{LS}}\}_{s=1}^{S}$;

1: Generate a sample of $\boldsymbol{y} \in \mathbb{R}^N$;
2: Project $\boldsymbol{y}$ to $\boldsymbol{y}_s$ for $s = 1, \ldots, S$ using (4.6);
3: Substitute $\boldsymbol{y}_s$ into (4.23) to evaluate $\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}$, $\widehat{\mathbb{A}}_s^{0\mathrm{b},\mathrm{LS}}$, $\widehat{\mathbb{A}}_s^{\mathrm{b}0,\mathrm{LS}}$, $\widehat{\mathbb{A}}_s^{\mathrm{bb},\mathrm{LS}}$;
4: Compute $\big(\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}\big)^{-1}$ for $s = 1, \ldots, S$;
5: Construct $\widehat{\mathbb{B}}_s^{\mathrm{LS}}(\boldsymbol{y}_s) := \widehat{\mathbb{A}}_s^{\mathrm{bb},\mathrm{LS}} - \widehat{\mathbb{A}}_s^{\mathrm{b}0,\mathrm{LS}}\big(\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}\big)^{-1}\widehat{\mathbb{A}}_s^{0\mathrm{b},\mathrm{LS}}$ for $s = 1, \ldots, S$;
6: Construct $\widehat{\mathbf{g}}_s(\boldsymbol{y}_s) := (\widehat{\mathbb{V}}_s^{\mathrm{b}})^\top \mathbf{f}_s^{\mathrm{b}} - \widehat{\mathbb{A}}_s^{\mathrm{b}0}\big(\widehat{\mathbb{A}}_s^{00}\big)^{-1}(\widehat{\mathbb{V}}_s^{0})^\top \mathbf{f}_s^{0}$;
7: Assemble $\widehat{\mathbb{B}}^{\mathrm{LS}} := \sum_{s=1}^{S}(\widehat{\mathbb{T}}_s)^\top \widehat{\mathbb{B}}_s^{\mathrm{LS}} \widehat{\mathbb{T}}_s$, $\widehat{\mathbf{g}}^{\mathrm{LS}} := \sum_{s=1}^{S}(\widehat{\mathbb{T}}_s)^\top \widehat{\mathbf{g}}_s^{\mathrm{LS}}$;
8: Solve the global system $\widehat{\mathbb{B}}^{\mathrm{LS}}\,\widehat{\mathbf{C}}^{\mathrm{b,LS}} = \widehat{\mathbf{g}}^{\mathrm{LS}}$;
9: Assign $\widehat{\mathbf{C}}^{\mathrm{b,LS}}$ to $\widehat{\mathbf{C}}_s^{\mathrm{b,LS}}$ for $s = 1, \ldots, S$;
10: Recover the local unknowns $\widehat{\mathbf{C}}_s^{0,\mathrm{LS}} = \big(\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}\big)^{-1}\left[(\widehat{\mathbb{V}}_s^{0})^\top \mathbf{f}_s^{0} - \widehat{\mathbb{A}}_s^{0\mathrm{b},\mathrm{LS}}\,\widehat{\mathbf{C}}_s^{\mathrm{b,LS}}\right]$;

The key feature of the online procedure is that the cost of solving $\widehat{\mathbf{C}}^{\mathrm{b,LS}}$, $\{\widehat{\mathbf{C}}_s^{0,\mathrm{LS}}\}_{s=1}^{S}$ for each sample $\boldsymbol{y} \in \mathbb{R}^N$ is independent of the triangle mesh size h. First, we can see that the cost of mapping $\boldsymbol{y}$ to $\boldsymbol{y}_s$ in each subdomain involves solving a linear system of size $T \times T$ in (4.6), where $T > N_s$ only needs to be big enough to guarantee numerical stability. Second, since the sizes of $\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}$, $\widehat{\mathbb{A}}_s^{0\mathrm{b},\mathrm{LS}}$, $\widehat{\mathbb{A}}_s^{\mathrm{b}0,\mathrm{LS}}$, and $\widehat{\mathbb{A}}_s^{\mathrm{bb},\mathrm{LS}}$ are $M_s \times M_s$, $M_s \times M_{s,j}$, $M_{s,j} \times M_s$, $M_{s,j} \times M_{s,j}$, respectively, evaluation of those matrices involves a total of $S(M_s + M_{s,j})^2$ vector-vector multiplications, where the vectors are of size $M_{s,p}^{\mathrm{LS}}$. A major cost lies in the inversion of $\widehat{\mathbb{A}}_s^{00,\mathrm{LS}}$, which requires $\mathcal{O}(SM_s^3)$ operations. In addition, since the matrix $\widehat{\mathbb{B}}_s^{\mathrm{LS}}$ is of size $\sum_{j=1}^{E_s} M_{s,j} \times \sum_{j=1}^{E_s} M_{s,j}$, the size of the global matrix $\widehat{\mathbb{B}}^{\mathrm{LS}}$ is smaller than $\sum_{s=1}^{S}\sum_{j=1}^{E_s} M_{s,j} \times \sum_{s=1}^{S}\sum_{j=1}^{E_s} M_{s,j}$ due to shared nodal values, so that the cost of solving $\widehat{\mathbb{B}}^{\mathrm{LS}}\,\widehat{\mathbf{C}}^{\mathrm{b,LS}} = \widehat{\mathbf{g}}^{\mathrm{LS}}$ is also independent of the triangle mesh size h.

4.4. Discussion on approximation errors. This effort focuses on the development of a new DD method for the PDEs with random inputs, and rigorous error analysis will be conducted in future work. Nevertheless, it is not difficult to identify the main error sources from Algorithms 1 and 2. Basically, there are four main error sources, i.e., (i) FE discretization, (ii) discretization of random fields, (iii) reduced-basis representation, and (iv) sparse approximation of the local stiffness matrices. The first error can be estimated by following the standard FE analysis. The second error only applies to the colored noise cases, where the error comes from the truncations of the global and local KL expansions, as well as the least-squares projection from the global to the local random variables. According to the theories on KL expansion (see, e.g., [34]), this error can be controlled by increasing the dimension of the KL expansions, i.e., increasing N and N_s in (2.9) and (4.3), respectively. The error from the third source is related to the Kolmogorov n-width (i.e., the optimal error) for a submanifold of the solution, i.e., the manifold of the solution on an interface or in a subdomain. This error is not easy to analyze due to its correlation with the domain partition. The smaller each subdomain, the faster the n-width decays for a local manifold. Nevertheless, such faster n-width decay does not necessarily lead to smaller size of the global system matrix $\widehat{\mathbb{B}}$ in (4.21), as more subdomains have to be

handled. Thus, an important question to be answered is how to partition the domain in an efficient way to minimize the size of the reduced global system in (4.21). At last, the error of approximating the reduced stiffness matrices depends on the regularity of the entries of those matrices with respect to the local parameters. In fact, since the bilinear form in (3.1) is a linear or quadratic function of the random coefficients, the entries of the reduced stiffness matrices share the same regularity as the coefficients. For the PDEs of interest, the coefficients have analytic regularity, so that the sparse approximations to $\widetilde{\mathbb{A}}_s^{00}, \widetilde{\mathbb{A}}_s^{0\mathrm{b}}, \widetilde{\mathbb{A}}_s^{\mathrm{b}0}, \widetilde{\mathbb{A}}_s^{\mathrm{bb}}$ are expected to have spectral accuracy. In addition, perturbation theory is needed to analyze how such a matrix approximation error propagates to PDE solutions.

5. Numerical examples. To test the performance of our method, we carried out numerical experiments based on the two PDEs given in Examples 1 and 2, where the physical domain D is set to a two-dimensional box $D := [0,1] \times [0,1]$. Our algorithms are implemented in `Matlab 2016a` and simulated on a workstation with `Intel(R) Xeon(R) CPU E5-2699 v4`. For each example, we tested two random fields. One is the truncated colored noise $\eta_N(x,\omega)$ in (2.9) with the Gaussian covariance

$$\kappa(x,x') := \exp\left(-\frac{\|x-x'\|_2^2}{L^2}\right), \tag{5.1}$$

where L is the correlation length, and $\boldsymbol{y} := (y_1,\ldots,y_N)^\top$ are assumed to follow the N-dimensional standard Gaussian distribution $\mathcal{N}(0,\mathbb{I})$. The other one is the discrete white noise $\eta_N^{\mathrm{WN}}(x,\omega)$ defined in (2.11), where the random variables $y_1,\ldots,y_N$ are assumed to follow the N-dimensional Gaussian distribution $\mathcal{N}(0,\sigma^2\mathbb{I})$ with σ being the standard deviation.

5.1. Tests on the local KL expansion. In the colored noise case, we tested the error between the global and local KL expansions caused by the truncations and the least-squares projection, discussed in section 4.1.1. To do this, we defined a 1024×1024 Cartesian mesh in D and discretized both the global and the local random fields on the mesh $\mathcal{T}_h$ as piecewise constant functions. For simplicity, we assumed the subdomains are of the same size and shape, such that all the local KL expansions feature the same eigenvalue profile. The decay of the eigenvalues $\sqrt{\lambda_n}$ is shown in Figure 2(a). As expected, the smaller the subdomain, the faster the eigenvalues decay, illustrating the motivation of using domain decomposition. In Figure 2(b), the error $\|\eta_N - \eta_{\boldsymbol{N}}\|_{L^2(D)}$ was computed for the four cases considered in Figure 2(a), where the global KL expansions were truncated at the 200th term, and the local expansions were truncated at $N_s = 1,2,3,4,5,6$ for $s = 1,\ldots,S$. It shows that the error $\|\eta_N - \eta_{\boldsymbol{N}}\|_{L^2(D)}$ is dominated by the largest neglected eigenvalue of the local KL expansion.

5.2. The diffusion equation with random diffusivity. We considered the two-dimensional elliptic PDE given in Example 1, where $D = [0,1]^2$ and $f(x) = 100$. A piecewise linear FE basis was used to discretize the PDE in D.

5.2.1. The colored noise case. The diffusivity $a(x,\omega)$ in (2.2) is defined by

$$a(x,\omega) := \exp\left(\frac{1}{5}\eta(x,\omega)\right), \tag{5.2}$$

with $\eta(x,\omega)$ defined in (2.8) based on the covariance function in (5.1) with the correlation length being $L = 1$ and $L = 0.25$. Figure 3 shows three snapshots of the

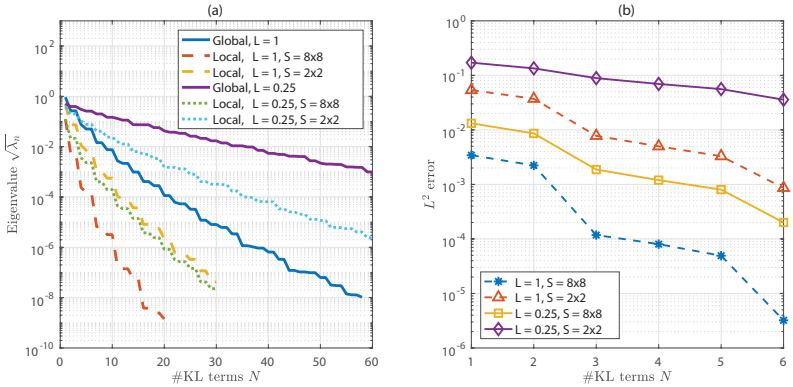


FIG. 2. (a) *The decay of eigenvalues* $\sqrt{\lambda_n}$ *of the global KL expansion* η_N *in* (2.9) *and the local KL expansion* $\eta^{\text{loc}}_{s,N_s}$ *in* (4.3) *for correlation lengths* $L = 1$ *and* $L = 0.25$*;* (b) *the* L^2 *error between the global and local truncated KL expansions, where the global KL expansion is truncated at the* 200*th term.*

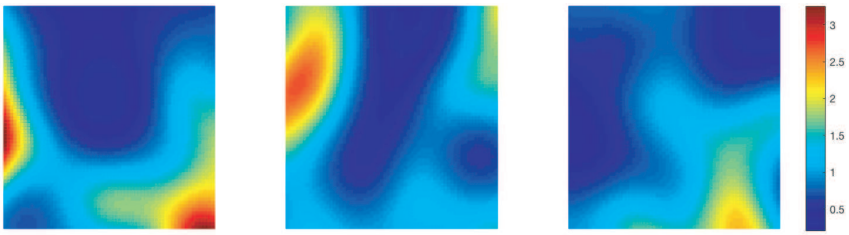


FIG. 3. *Illustration of three snapshots of the random field in* (5.2) *with* $L = 0.25$.

random field in (5.2) with $L = 0.25$. To compute the total approximation error, we defined the reference solution as the numerical solution obtained by solving the PDE in (2.2) on a triangle mesh with $h = 1/2^{12}$, i.e., a total of 4096×4096 grid points, using the truncated global KL expansion η_N with $N = 200$. Figure 2(a) shows that $N = 200$ is sufficient to neglect the global KL truncation error. The random variables $y_1, \ldots, y_N$ follow the N-dimensional standard Gaussian distribution. For the local KL expansion, the bounded domains Γ_s, introduced in section 4.2.1, were set to $\Gamma_s = [-5, 5]^{N_s}$ for $s = 1, \ldots, S$, such that the probability of having a sample $\boldsymbol{y}_s$ fall outside Γ_s is about 5×10^{-7}. In the domain Γ_s, we used sparse Legendre polynomials to approximate the reduced local stiffness matrices.

To illustrate the effectiveness of SVD, we drew 1000 random samples of $\boldsymbol{y} \in \mathbb{R}^N$, generated snapshots by solving the expensive FE problems, and performed SVDs. The singular value[3] decay curves along an interface and in the interior of a subdomain are plotted in Figure 4. We observed that the more subdomains, the faster the singular values decay. As such, we can keep a small number of singular vectors along the interfaces to reduce the size of the global stiffness matrix $\mathbb{B}$ in (3.7), as well as keep a small number of interior singular vectors in each subdomain to reduce the size of the local stiffness matrices $\mathbb{A}_s^{00}$ in (3.4). Figure 4 demonstrates that obtaining fast singular value decay is another advantage of using domain decomposition.

Now we show the accuracy of our DDMR approach by examining the error decay

[3]The plotted singular values are normalized by the largest singular value in each case.

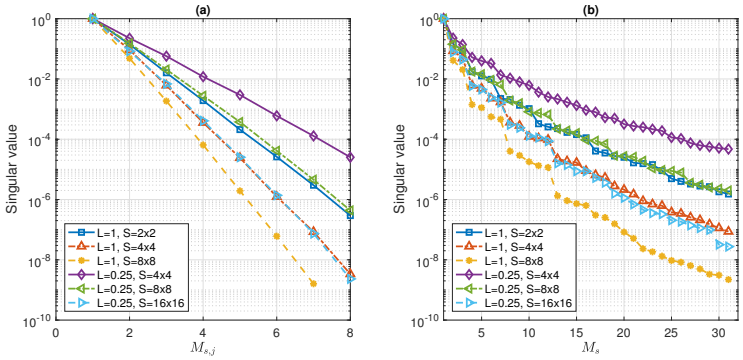


FIG. 4. (a) *The decay of singular values of the solution on one interface;* (b) *the decay of singular values of the solution in the interior of one subdomain* D_s^H.

of the reduced model with respect to four quantities, i.e.,

- N_s: the dimension of the truncated local KL expansion in (4.3);
- $M_{s,j}$: the dimension of $\widehat{\mathbb{V}}_{s,j}^{\mathrm{b}}$ in (4.11) along the jth interface of D_s^H;
- M_s: the dimension of $\widehat{\mathbb{V}}_s^0$ in (4.14) in subdomain D_s^H;
- $M_{s,p}^{\mathrm{LS}}$: the cardinality of the polynomial space $\mathcal{P}_{M_{s,p}^{\mathrm{LS}}}(\Gamma_s)$ of (4.23).

Each of the above four quantities could be different for each subdomain or interface. In this work, we used the same number over all subdomains for each quantity. This strategy is not optimal but sufficient to demonstrate the performance of our method. We run simulations in four different scenarios, i.e.,

(i) $L = 1$ with $S = 8 \times 8$ subdomains;
(ii) $L = 1$ with $S = 16 \times 16$ subdomains;
(iii) $L = 0.25$ with $S = 8 \times 8$ subdomains;
(iv) $L = 0.25$ with $S = 16 \times 16$ subdomains.

The results are shown in Figure 5. In each scenario, we first computed the errors by setting $N_s = 6$, $M_{s,j} = 6$, $M_s = 19$, and $p = 9$ (leading to $M_{s,p}^{\mathrm{LS}} = 332$). These errors are shown as the last markers of the error decay curves plotted in Figure 5. Then, in each subfigure of Figure 5, we decreased one of the four quantities while leaving the other three unchanged. In each scenario, we generated 1000 random samples of $\boldsymbol{y}$ and executed 1000 time-consuming FE solvers (using the 4096×4096 mesh) to generate data of the solution defined in (4.7). On the other hand, we generated another 1000 realizations of $\boldsymbol{y}_s$ for $s = 1, \ldots, S$ to generate the data of the local stiffness matrices defined in (4.9). The error was computed in the relative L^2 norm using another 1000 random samples of $\boldsymbol{y} \sim \mathcal{N}(0, \mathbb{I})$. The error decays, as expected. We would like to point out that the smallest error comes from the case of having a bigger correlation and more subdomains. Thus, for a small correlation length, we can introduce more subdomains to reduce the total error.

Next, we discuss the efficiency of our approach to achieve a prescribed accuracy. We used the case with $L = 0.25$ and three triangle meshes of sizes 1024×1024, 2048×2048, and 4096×4096, respectively. For each of the three meshes, we constructed our reduced model by setting $N_s = 6$, $M_{s,j} = 6$, $M_s = 19$, and $p = 9$ (i.e., $M_{s,p}^{\mathrm{LS}} = 332$) in our algorithm.[4] The complexity of our approach is divided into the offline cost, i.e., the cost of Algorithm 1, and the online cost, i.e., the cost of Algorithm 2. Both the

[4]The model reduction error is balanced with the FE error on the finest mesh of size 4096×4096.

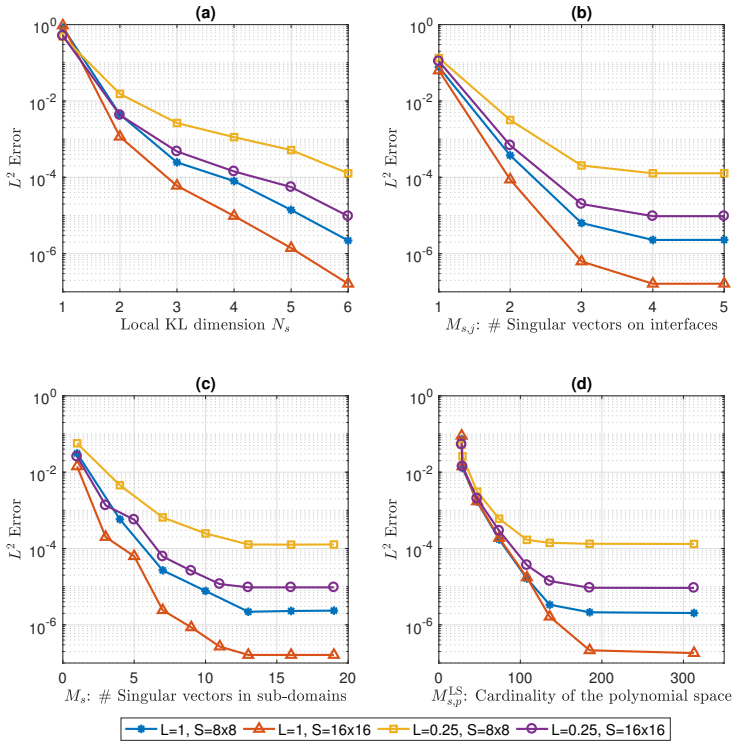


FIG. 5. *Illustration of the error decays with respect to N_s, $M_{s,j}$, M_s, and $M_{s,p}^{\mathrm{LS}}$.* (a) *Error decay with respect to N_s while setting $M_{s,j} = 6$, $M_s = 19$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (b) *error decay with respect to $M_{s,j}$ while setting $N_s = 6$, $M_s = 19$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (c) *error decay with respect to M_s while setting $N_s = 6$, $M_{s,j} = 6$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (d) *error decay with respect to $M_{s,p}^{\mathrm{LS}}$ while setting $N_s = 6$, $M_{s,j} = 6$, and $M_s = 19$.*

online and the offline costs are measured in a relative way by

$$\text{Cost} = \frac{\text{The offline (online) CPU time}}{\text{The CPU time of one expensive FE simulation}}, \tag{5.3}$$

which, in other words, is the number of expensive FE simulations. To be more harsh to our method, the CPU time of one expensive FE simulation is only measured by the time of solving the final linear system using the "\" solver in `Matlab 2016a`, regardless of assembly cost and other operations that may not be optimally implemented.

The results are shown in Table 1 for the case of having $S = 16 \times 16$ subdomains. In the offline procedure, the data generation is still the dominant part, and our dimension reduction strategy successfully helps achieve $\mathcal{O}(10^{-6})$ error with only 1000 expensive FE simulations. For the SVD algorithm, since we only needed a few of the largest singular values and singular vectors, we did not need to run full SVD. Instead, we use the Lanczos bidiagonalization methods [2] to reduce the cost of running SVDs. On the other hand, the wall time of the online cost is $0.42 \times 4\,\text{sec} = 1.68\,\text{sec}$, $0.07 \times 25\,\text{sec} = 1.75\,\text{sec}$, $0.004 \times 381\,\text{sec} = 1.52\,\text{sec}$ for the cases of having 1024×1024, 2048×2048, and 4096×4096 meshes, respectively. This verifies that the online cost is independent of the triangle mesh size h, so that the finer the original mesh, the more savings our

TABLE 1

Computational cost for solving Example 1 *with the diffusivity being the colored noise ($L = 0.25$). The domain D is decomposed into 16×16 subdomains. The unit FE time only takes into account the CPU time of solving the final linear system; the offline and online costs are measured by the number of expensive FE simulations.*

FE cost	# FE nodal values	2^{20}	2^{22}	2^{24}
	Unit FE time	4 sec	25 sec	381 sec
Offline cost ($\frac{\text{Wall time}}{\text{Unit FE time}}$)	KL expansion	2.19	2.16	0.53
	FE solves for $\mathbf{U}_s(w_k)$, $k = 1, \ldots, 1000$	1000	1000	1000
	SVD on the interfaces for $\widehat{\mathbb{V}}^{\mathrm{b}}_{s,j}$	22.70	4.71	0.33
	SVD in the subdomains for $\widehat{\mathbb{V}}^{0}_{s}$	34.13	29.00	12.38
	Assembling $\widetilde{\mathbb{A}}_s(\omega_k)$, $k = 1, \ldots, 1000$	256.25	301.33	389.78
	Computing $\widehat{\mathbb{K}}^{00}_s, \widehat{\mathbb{K}}^{0\mathrm{b}}_s, \widehat{\mathbb{K}}^{\mathrm{b}0}_s, \widehat{\mathbb{K}}^{\mathrm{bb}}_s$	6.89	1.45	0.08
	Total	1315.27	1338.65	1403.1
Online cost ($\frac{\text{Wall time}}{\text{Unit FE time}}$)	Solving $\widehat{\mathbf{C}}^{\mathrm{b,LS}}$	0.09	0.02	0.001
	Solving $\widehat{\mathbf{C}}^{0,\mathrm{LS}}_s$ for $s = 1, \ldots, S$	0.33	0.05	0.003
	Total	0.42	0.07	0.004

method can provide.

5.2.2. The discrete white noise case. Now we show the discrete white noise case by replacing the random field η in (5.2) with the one in (2.11), which is assumed to be a uniformly partitioned piecewise constant random field. The random parameters $\boldsymbol{y}$ follow the multidimensional Gaussian distributions, denoted by $\mathcal{N}(0, \sigma^2\mathbb{I})$, where σ is the standard deviation and $\mathbb{I}$ is the N-dimensional identity matrix. Figure 6 shows three snapshots of the discrete white noise with $N = 16 \times 16$ and $\sigma = 1$.

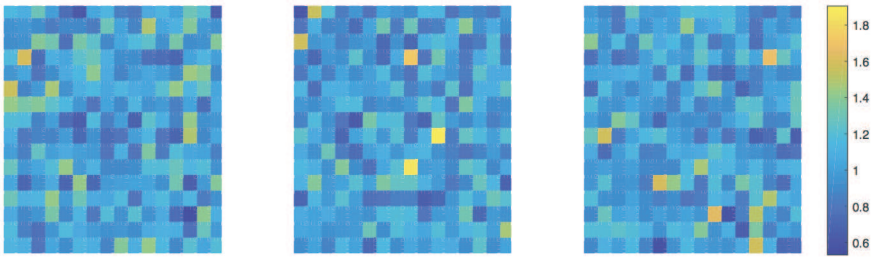


FIG. 6. *Illustration of three snapshots of the discrete white noise with $N = 16 \times 16$ and $\sigma = 1$.*

We run simulations in four different cases, i.e.,

(i) $\sigma = 0.1$ with $N = S = 8 \times 8$ subdomains;
(ii) $\sigma = 0.1$ with $N = S = 16 \times 16$ subdomains;
(iii) $\sigma = 1$ with $N = S = 8 \times 8$ subdomains;
(iv) $\sigma = 1$ with $N = S = 16 \times 16$ subdomains.

For each case, we assume $D^{\mathrm{WN}}_n = D^H_s$, i.e., align the domain decomposition with the partition of the random field. As such, the approximation of the local stiffness matrices

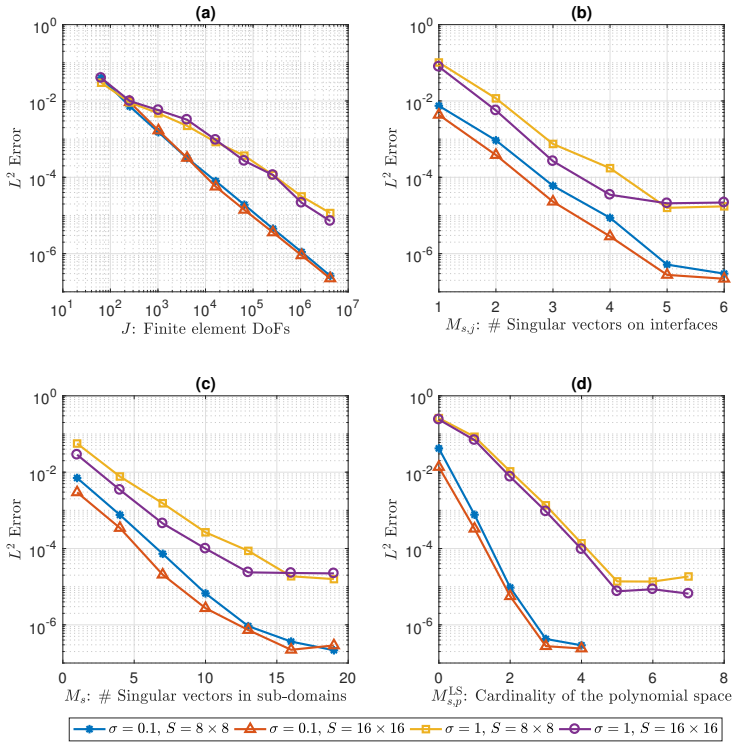


FIG. 7. *Illustration of the error decays with respect to J, $M_{s,j}$, M_s, and $M_{s,p}^{\mathrm{LS}}$.* (a) *Error decay with respect to J while setting $M_{s,j} = 6$, $M_s = 19$, and $M_{s,p}^{\mathrm{LS}} = 10$;* (b) *error decay with respect to $M_{s,j}$ while setting $J = 2^{24}$, $M_s = 19$, and $M_{s,p}^{\mathrm{LS}} = 10$;* (c) *error decay with respect to M_s while setting $J = 2^{24}$, $M_{s,j} = 6$, and $M_{s,p}^{\mathrm{LS}} = 10$;* (d) *error decay with respect to $M_{s,p}^{\mathrm{LS}}$ while setting $J = 2^{24}$, $M_{s,j} = 6$, and $M_s = 19$.*

becomes a one-dimensional approximation problem. We still used 1000 realizations to conduct SVD, but we only use 100 realizations for the DLS approximation of the local stiffness matrices. The local parameter domain Γ_s in section 4.2.1 is set to $[-5\sigma, 5\sigma]$. The reference solution was obtained by solving the PDE on a very fine mesh with $h = 1/2^{12}$, i.e., 4096×4096 unknowns. All the other settings were the same as in section 5.2.1.

The results are shown in Figure 7, where we used the same strategy as in Figure 5 to generate the error decay curves. Since there is no KL expansion setting, we plotted the error with respect to J in (3.1). The error was computed in the relative L^2 norm using another 1000 random samples of $\boldsymbol{y} \sim \mathcal{N}(0, \sigma^2 \mathbb{I})$. As expected, the error of case (ii) is the smallest due to the smaller variance. The flat toes in Figures 7(b)–(d) are due to the dominancy of the error caused by the triangle mesh size. As shown in Figure 7(d), an important advantage of our method is that it can reduce local stiffness matrix approximation to a set of one-dimensional problems that completely overcomes the curse of dimensionality.

Next, we show the efficiency of our approach using the 256-dimensional random field, i.e., $N = S = 16 \times 16$ in (2.11), with $\sigma = 0.1$. Three triangle meshes of sizes 1024×1024, 2048×2048, and 4096×4096 were used. For each mesh, we constructed

TABLE 2

Computational cost for solving Example 1 *with diffusivity being the* 256*-dimensional discrete white noise. The standard deviation of each random variable is* $\sigma = 0.1$*. The unit FE time only takes into account the CPU time of solving the final linear system; the offline and online costs are measured by the number of expensive FE simulations.*

FE cost	# FE nodal values	2^{20}	2^{22}	2^{24}
	Unit FE time	4 sec	27 sec	394 sec
Offline cost ($\frac{\text{Wall time}}{\text{Unit FE time}}$)	FE solves for $\mathbf{U}_s(w_k)$, $k = 1, \ldots, 1000$	1000	1000	1000
	SVD on the interfaces for $\widehat{\mathbb{V}}_{s,j}^{\mathrm{b}}$	23.82	4.17	0.32
	SVD in the subdomains for $\widehat{\mathbb{V}}_s^0$	35.42	29.51	12.18
	Assembling $\widetilde{\mathbb{A}}_s(\omega_k)$, $k = 1, \ldots, 100$	28.90	34.83	49.67
	Computing $\widehat{\mathbb{K}}_s^{00}, \widehat{\mathbb{K}}_s^{0\mathrm{b}}, \widehat{\mathbb{K}}_s^{\mathrm{b}0}, \widehat{\mathbb{K}}_s^{\mathrm{bb}}$	0.005	0.007	0.001
	Total	1088.15	1068.52	1062.17
Online cost ($\frac{\text{Wall time}}{\text{Unit FE time}}$)	Solving $\widehat{\mathbf{C}}^{\mathrm{b,LS}}$	0.111	0.018	0.001
	Solving $\widehat{\mathbf{C}}_s^{0,\mathrm{LS}}$ for $s = 1, \ldots, S$	0.423	0.071	0.002
	Total	0.534	0.089	0.003

a reduced model by setting $M_{s,j} = 6$, $M_s = 19$, and $p = 9$ (i.e., $M_{s,p}^{\mathrm{LS}} = 10$) in our algorithm. The complexity of our approach is divided into the offline cost, i.e., the cost of Algorithm 1, and the online cost, i.e., the cost of Algorithm 2. Both the online and the offline costs are measured in a relative way by the formula given in (5.3). The results are shown in Table 2. As expected, we managed to make the online cost independent of the original triangle mesh size. In the offline procedure, the cost of assembling $\widetilde{A}_s$ is much smaller than the case of having colored noise because of the one-dimensional parametric dependence of the local stiffness matrices.

5.3. The convection-dominated transport with random velocity. We considered the two-dimensional PDE given in Example 2, where $f(x) = 0$, $D = [0,1]^2$, and the set $\mathscr{D}$ is a subset of ∂D defined by $\{x_1 = 0,\ x_2 \in [0, 0.5]\} \cup \{x_1 \in [0,1],\ x_2 = 0\}$. The SUPG scheme is used to discretize the PDE in the physical domain D.

5.3.1. The colored noise case. The random velocity field $\boldsymbol{b}(x,\omega)$ is defined by

$$\boldsymbol{b}(x,\omega) := \begin{pmatrix} \cos\left(\frac{1}{5}\eta(x,\omega)\right) \\ \sin\left(\frac{1}{5}\eta(x,\omega)\right) \end{pmatrix}, \tag{5.4}$$

where $\eta(x,\omega)$ is defined in (2.8) based on the covariance function given in (5.1) with correlation length $L = 0.25$. The reference solution is obtained by solving the PDE in (2.4) on a mesh with $h = 1/2^{12}$ using the truncated global KL expansion η_N with $N = 200$. The random variables $y_1, \ldots, y_N$ follow the N-dimensional standard Gaussian distribution $\mathcal{N}(0, \mathbb{I})$. Three snapshots of the velocity field and the corresponding solution field are given in Figure 8.

We defined a total of $S = 16 \times 16$ subdomains. The bounded domain Γ_s introduced in section 4.2.1 is set to $\Gamma_s = [-5,5]^{N_s}$ for $s = 1, \ldots, S$, as in Example 1. Similar to

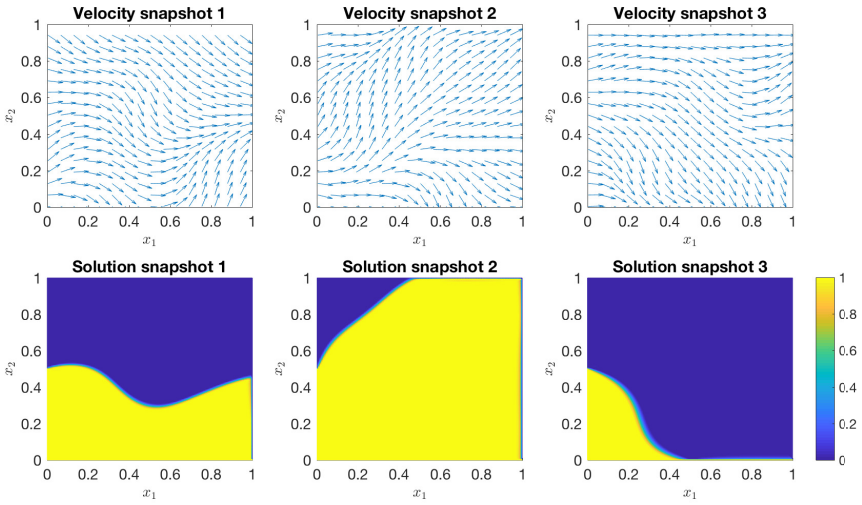


FIG. 8. *Three snapshots of the colored noise velocity field defined in* (5.4) *and the corresponding solution field of the PDE in* (2.4).

Figure 5, we show the accuracy of our approach by examining the error decays of the reduced model with respect to the four quantities, i.e., N_s, $M_{s,j}$, M_s, and $M_{s,p}^{\mathrm{LS}}$. We ran simulations with three diffusion coefficients, i.e., $\varepsilon = 10^0, 10^{-2}, 10^{-4}$.

The results are shown in Figure 9. We first computed the errors by setting $N_s = 6$, $M_{s,j} = 6$, $M_s = 36$, and $p = 9$ (leading to $M_{s,p}^{\mathrm{LS}} = 332$). These errors are shown as the last markers of the error decay curves plotted in Figure 8. Then we used the same strategy as in Figure 5 to generate Figure 8. We observe that the errors become bigger as we decreased the value of ε, i.e., increasing the sharpness of the transition area. Moreover, when $\varepsilon = 10^0$, the solution has a smooth transition, in which case the error decays fast with respect to $M_{s,j}$ and M_s. In comparison, when $\varepsilon = 10^{-4}$, the errors from SVDs are dominant, as both the errors caused by local KL expansion and the sparse approximation hit flat toes. In addition, we also observed that the decay rate of the error with respect to $M_{s,p}^{\mathrm{LS}}$ remained the same as we decreased the value of ε because of the fact that the smoothness of the local stiffness matrices was not changed with ε.

5.3.2. The discrete white noise case. Now we demonstrate the discrete white noise case by replacing the random field η in (5.4) with the one in (2.11), which is a uniformly partitioned piecewise constant random field. The random parameters in $\boldsymbol{y}$ follow $\mathcal{N}(0, \sigma^2\mathbb{I})$, as in section 5.2.2. In the case of $N = 2 \times 2$, $\sigma = 0.5$, three snapshots of the velocity field and the corresponding solution field are given in Figure 10 for illustration.

To test the accuracy, we run simulations in four different cases, i.e.,

(i) $\sigma = 0.5$ with $\varepsilon = 10^{-4}$;
(ii) $\sigma = 0.5$ with $\varepsilon = 10^{-2}$;
(iii) $\sigma = 0.1$ with $\varepsilon = 10^{-4}$;
(iv) $\sigma = 0.1$ with $\varepsilon = 10^{-2}$.

For each case, we assume $D_n^{\mathrm{WN}} = D_s^H$, as in Example 1. Again, the approximation of the reduced local stiffness matrices becomes a one-dimensional approximation prob-

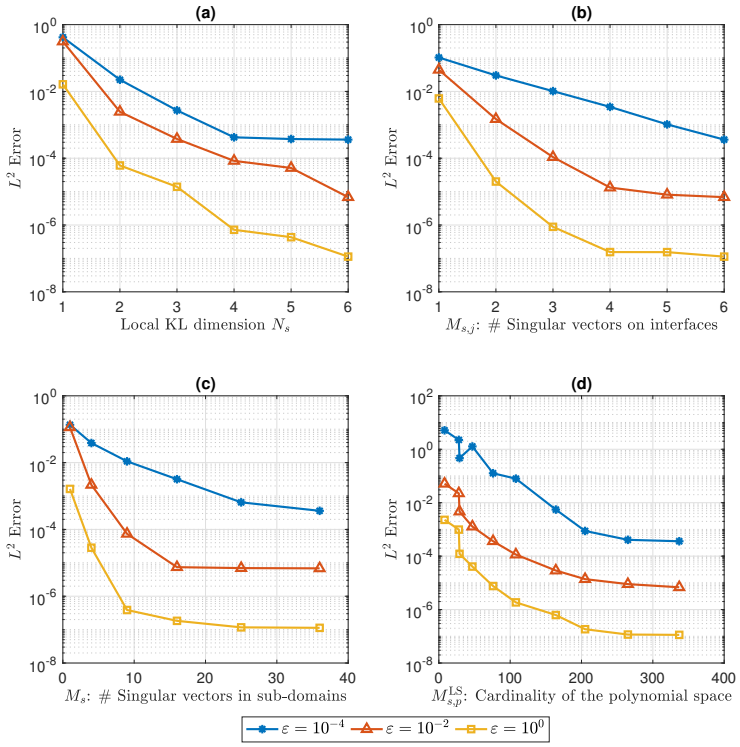


FIG. 9. *Illustration of the error decays with respect to N_s, $M_{s,j}$, M_s, and $M_{s,p}^{\mathrm{LS}}$.* (a) *Error decay with respect to N_s while setting $M_{s,j} = 6$, $M_s = 36$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (b) *error decay with respect to $M_{s,j}$ while setting $N_s = 6$, $M_s = 36$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (c) *error decay with respect to M_s while setting $N_s = 6$, $M_{s,j} = 6$, and $M_{s,p}^{\mathrm{LS}} = 332$;* (d) *error decay with respect to $M_{s,p}^{\mathrm{LS}}$ while setting $N_s = 6$, $M_{s,j} = 6$, and $M_s = 36$.*

lem. We used 1000 realizations to conduct SVD, but we only used 100 realizations for the DLS approximation of the local stiffness matrices. The local parameter domain Γ_s in section 4.2.1 is set to $[-5, 5]$. All the other settings are the same as in section 5.2.2. The results are shown in Figure 11. The error was computed in the relative L^2 norm using another 1000 random samples of $\boldsymbol{y} \sim \mathcal{N}(0, \sigma^2 \mathbb{I})$. As expected, the error of case (iv) is the smallest due to the smaller σ and the bigger ε. As ε decreases, the SVD errors become more and more dominant due to the slow decay of singular values around the sharp transition layer.

Remark 5. For the computational cost of our method, we can conduct similar discussions and draw the same conclusions as in Example 1 by generating tables analogous to Tables 1 and 2. Thus, we omit the cost analysis in this example and only refer to the discussions on complexity in section 5.2.

6. Concluding remarks. We developed a new model reduction method for stochastic convection-diffusion equations by integrating domain decomposition, a local reduced-basis method, and sparse approximation of operators. Our method can overcome the curse of high dimensionality and achieve online-offline decomposition,

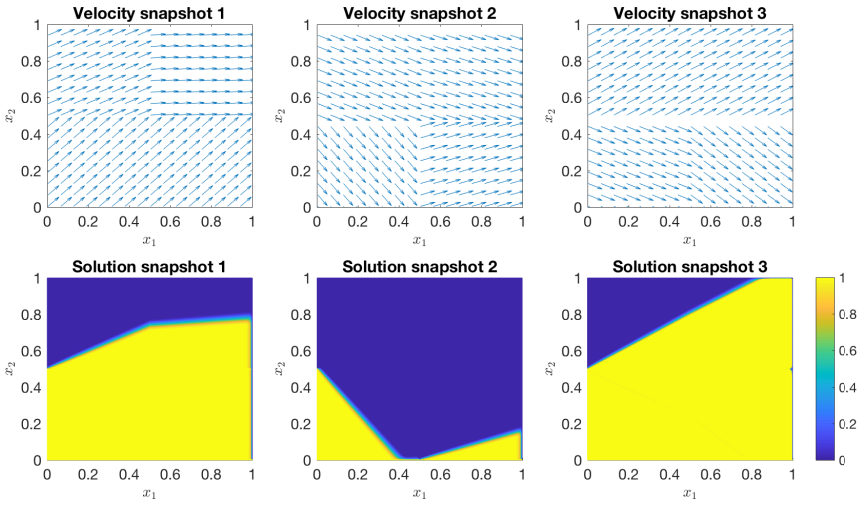


FIG. 10. *Three snapshots of the discrete white noise velocity field defined in* (5.4) *and the corresponding solution field of the PDE in* (2.4).

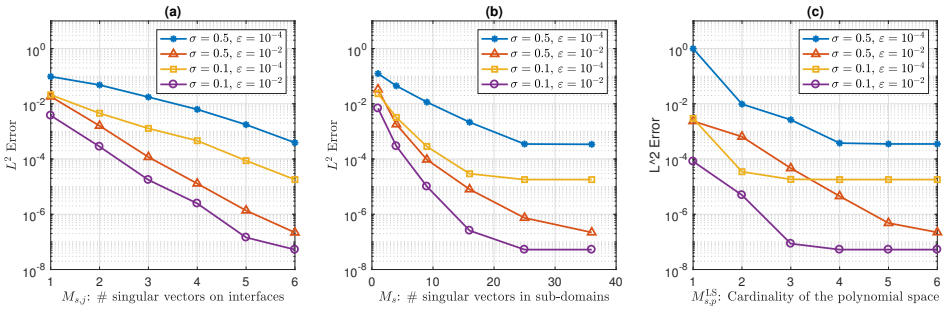


FIG. 11. *Illustration of the error decays with respect to* $M_{s,j}$, M_s, *and* $M_{s,p}^{\mathrm{LS}}$. (a) *Error decay with respect to* $M_{s,j}$ *while setting* $M_s = 36$ *and* $M_{s,p}^{\mathrm{LS}} = 7$; (b) *error decay with respect to* M_s *while setting* $M_{s,j} = 6$ *and* $M_{s,p}^{\mathrm{LS}} = 7$; (c) *error decay with respect to* $M_{s,p}^{\mathrm{LS}}$ *while setting* $M_{s,j} = 6$ *and* $M_s = 36$.

as well as handle the convection-dominated problem with irregular behavior. Even though our strategy shows very promising performance, it could be further improved in several aspects. The first direction would be incorporating greedy algorithms [13, 17] to generate the local reduced bases, which requires a posteriori error estimates and a systematic way to coordinate the greedy search in different subdomains. Another direction is to extend our method to a nonintrusive algorithm (see, e.g., [27]), which will make it much easier to couple with large-scale simulation code.

REFERENCES

[1] B. Adcock and D. Huybrechs, *Approximating Smooth, Multivariate Functions on Irregular Domains*, preprint, https://arxiv.org/abs/1802.00602, 2018.

[2] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput., 27 (2005), pp. 19–42, https://doi.org/10.1137/04060593X.

[3] J. Beck, F. Nobile, L. Tamellini, and R. Tempone, *Convergence of quasi-optimal stochastic Galerkin methods for a class of PDES with random coefficients*, Comput. Math. Appl., 67 (2014), pp. 732–751.

[4] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715.

[5] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM J. Math. Anal., 43 (2011), pp. 1457–1472, https://doi.org/10.1137/100795772.

[6] P. Chen, A. Quarteroni, and G. Rozza, *Comparison between reduced basis and stochastic collocation methods for elliptic problems*, J. Sci. Comput., 59 (2013), pp. 187–216.

[7] Y. Chen, J. Jakeman, C. Gittelson, and D. Xiu, *Local polynomial chaos expansion for linear differential equations with high dimensional random inputs*, SIAM J. Sci. Comput., 37 (2015), pp. A79–A102, https://doi.org/10.1137/140970100.

[8] A. Cohen and R. DeVore, *Kolmogorov widths under holomorphic mappings*, IMA J. Numer. Anal., 36 (2016), pp. 1–12.

[9] A. Cohen, R. DeVore, and C. Schwab, *Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs*, Found. Comput. Math., 10 (2010), pp. 615–646.

[10] A. Cohen, R. DeVore, and C. Schwab, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs*, Anal. Appl. (Singap.), 9 (2011), pp. 11–47.

[11] A. A. Contreras, P. Mycek, O. P. Le Maître, F. Rizzi, B. Debusschere, and O. M. Knio, *Parallel domain decomposition strategies for stochastic elliptic equations. Part A: Local Karhunen–Loève representations*, SIAM J. Sci. Comput., 40 (2018), pp. C520–C546, https://doi.org/10.1137/17M1132185.

[12] A. A. Contreras, P. Mycek, O. P. Le Maître, F. Rizzi, B. Debusschere, and O. M. Knio, *Parallel domain decomposition strategies for stochastic elliptic equations. Part B: Accelerated Monte Carlo sampling with local PC expansions*, SIAM J. Sci. Comput., 40 (2018), pp. C547–C580, https://doi.org/10.1137/17M1132197.

[13] R. DeVore, G. Petrova, and P. Wojtaszczyk, *Greedy algorithms for reduced bases in Banach spaces*, Constr. Approx., 37 (2013), pp. 455–466.

[14] J. L. Eftang and A. T. Patera, *Port reduction in parametrized component static condensation: Approximation and a posteriori error estimation*, Internat. J. Numer. Methods Engrg., 96 (2013), pp. 269–302.

[15] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328.

[16] M. D. Gunzburger, C. G. Webster, and G. Zhang, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numer., 23 (2014), pp. 521–650.

[17] J. S. Hesthaven, B. Stamm, and S. Zhang, *Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 259–283.

[18] T. Y. Hou, Q. Li, and P. Zhang, *Exploring the locally low dimensional structure in solving random elliptic PDEs*, Multiscale Model. Simul., 15 (2017), pp. 661–695, https://doi.org/10.1137/16M1077611.

[19] S. P. Huang, S. T. Quek, and K. K. Phoon, *Convergence study of the truncated Karhunen–Loeve expansion for simulation of stochastic processes*, Internat. J. Numer. Methods Engrg., 52 (2001), pp. 1029–1043.

[20] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, *A static condensation reduced basis element method: Approximation and a posteriori error estimation*, ESAIM Math. Model. Numer. Anal., 47 (2012), pp. 213–251.

[21] J. D. Jakeman, R. Archibald, and D. Xiu, *Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids*, J. Comput. Phys., 230 (2011), pp. 3977–3997.

[22] M. Kac and A. J. F. Siegert, *An explicit representation of a stationary Gaussian process*, Ann. Math. Statistics, 18 (1947), pp. 438–442.

[23] O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer, Dordrecht, The Netherlands, 2010.

[24] Q. Liao and K. Willcox, *A domain decomposition approach for uncertainty analysis*, SIAM J. Sci. Comput., 37 (2015), pp. A103–A133, https://doi.org/10.1137/140980508.

[25] F. Nobile, R. Tempone, and C. G. Webster, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442, https://doi.org/10.1137/070680540.

[26] B. Peherstorfer, M. Gunzburger, and K. Willcox, *Convergence analysis of multifidelity Monte Carlo estimation*, Numer. Math., 45 (2018), pp. 1–25.

[27] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction*, Comput. Methods Appl. Mech. Engrg., 306 (2016),

pp. 196–215.
[28] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*, SIAM Rev., 60 (2018), pp. 550–591, https://doi.org/10.1137/16M1082469.
[29] S. Pranesh and D. Ghosh, *Addressing the curse of dimensionality in SSFEM using the dependence of eigenvalues in KL expansion on domain size*, Comput. Methods Appl. Mech. Engrg., 311 (2016), pp. 457–475.
[30] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Unitext 92, Springer, Cham, 2016.
[31] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275.
[32] A. Rukavishnikova, S. M. Ermakov, and G. S. Fishman, *Monte Carlo*, Springer-Verlag, New York, 1996.
[33] A. Sarkar, N. Benabbou, and R. Ghanem, *Domain decomposition of stochastic PDEs: Theoretical formulations*, Internat. J. Numer. Methods Engrg., 77 (2009), pp. 689–701.
[34] C. Schwab and R. A. Todor, *Karhunen–Loève approximation of random fields by generalized fast multipole methods*, J. Comput. Phys., 217 (2006), pp. 100–122.
[35] W. Subber and A. Sarkar, *Domain decomposition method of stochastic PDEs: A two-level scalable preconditioner*, J. Phys. Conf. Ser., 341 (2012), 012033.
[36] R. Tipireddy, P. Stinis, and A. M. Tartakovsky, *Basis adaptation and domain decomposition for steady-state partial differential equations with random coefficients*, J. Comput. Phys., 351 (2017), pp. 203–215.
[37] A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer, Berlin, Heidelberg, 2005.
[38] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139, https://doi.org/10.1137/040615201.
[39] G. Zhang, C. G. Webster, M. Gunzburger, and J. Burkardt, *Hyperspherical sparse approximation techniques for high-dimensional discontinuity detection*, SIAM Rev., 58 (2016), pp. 517–551, https://doi.org/10.1137/16M1071699.